3D motion estimation and assessment in fluorescence microscopy volume sequences

Sandeep Manandhar

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE RENNES 1

THESE DE DOCTORAT DE

L'UNIVERSITE DE RENNES 1
COMUE UNIVERSITE BRETAGNE LOIRE

ECOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : Signal, Image, Vision

Par
Sandeep MANANDHAR

3D Motion Estimation and Assessment in Fluorescence Microscopy Volume Sequences

Thèse présentée et soutenue à Rennes, le 28/11/2019
Unité de recherche : Centre Inria Rennes – Bretagne Atlantique

Rapporteurs avant soutenance :

Fabrice HEITZ	Professeur, Université de Strasbourg
Michael LIEBLING	Senior researcher, Institut de Recherche IDIAP, Suisse

Composition du Jury :

Président :	Christine GUILLEMOT	DR Inria, Centre Inria Rennes
Examinateurs :	Olivier DAMERON	Maître de Conférences, Université Rennes 1
	Cédric DEMONCEAUX	Professeur, Université de Bourgogne
	Christophe ZIMMER	Chercheur, Institut Pasteur, Paris
	Fabrice HEITZ	Professeur, Université de Strasbourg
	Michael LIEBLING	Senior researcher, Institut de Recherche IDIAP, Suisse
Dir. de thèse :	Patrick BOUTHEMY	DR Inria, Centre Inria Rennes
Co-dir. de thèse :	Charles KERVRANN	DR Inria, Centre Inria Rennes

Table des matières

Table des matières iii

1 Introduction 1
- 1.1 Light-Sheet Microscopy . . . 2
  - 1.1.1 Fluorescence and Fluorophores . . . 2
  - 1.1.2 Light-Sheet Fluorecence Microscopy . . . 4
  - 1.1.3 Axially Swept Light-Sheet Microscopy (ASLM) . . . 4
  - 1.1.4 Diagonally Scanned Light-Sheet Microscopy (DiaSLM) . . . 5
- 1.2 Biological Context . . . 6
  - 1.2.1 MV3 Melanoma (MV3) Cell . . . 6
  - 1.2.2 Extra Cellular Matrix (ECM) and Collagen . . . 6
  - 1.2.3 Locomotion of MV3 cell . . . 7
- 1.3 Towards Motion Estimation . . . 8
- 1.4 Research Aims . . . 11
- 1.5 Structure of the Manuscript . . . 12

2 State-of-the-Art 13
- 2.1 Variational Methods . . . 13
  - 2.1.1 Minimization . . . 15
    - 2.1.1.1 Euler-Lagrange equations : . . . 15
    - 2.1.1.2 Splitting methods : . . . 16
  - 2.1.2 Data term . . . 17
  - 2.1.3 Regularization . . . 21
    - 2.1.3.1 Image-Driven Approach . . . 21
    - 2.1.3.2 Flow-Driven Approach . . . 21
    - 2.1.3.3 Joint Image- and Flow-Driven approach . . . 22
    - 2.1.3.4 Non-Local Regularization . . . 22
- 2.2 Matching . . . 23
  - 2.2.1 Sparse and Semi-dense Matching . . . 23
  - 2.2.2 Densification of Sparse Matches . . . 24
  - 2.2.3 Combination of Matching and Variational Method . . . 24
- 2.3 Convolution Neural Network-based Optical Flow . . . 25
- 2.4 In the Context of Bio-Imaging . . . 28
  - 2.4.1 3D Motion Estimation in Bio-Imaging . . . 28
    - 2.4.1.1 Amat's Method . . . 29
- 2.5 Remarks . . . 31

3 Visualization and Assessment **33**
3.1 Data Description . . . 33
3.1.1 MV3 cell . . . 33
3.1.2 Collagen network . . . 34
3.2 3D Flow Field Visualization . . . 35
3.2.1 Preliminaries . . . 36
3.2.2 Requirements for good visualization . . . 36
3.2.3 Middlebury Style Color Coding . . . 38
3.2.4 3D Flow Field Visualization : No Free Lunch . . . 40
3.2.5 Mapping 3D motion field to HSV (3DHSV) . . . 41
3.2.6 Mapping 3D motion field from three orthogonal planes (3PHS) . . . 42
3.2.7 Mapping 3D motion field to Maximum Intensity Points . . . 45
3.2.7.1 Remarks . . . 47
3.3 Accuracy estimation : . . . 49
3.3.1 Computation of 3D Structure Tensor . . . 51
3.3.1.1 Structural Angular Error . . . 52
3.3.1.2 Choice of Smoothing Window σ . . . 52
3.3.1.3 Eigen Decomposition . . . 53
3.3.2 Conclusion . . . 53

4 Matching **55**
4.1 A short overview of the PatchMatch algorithm . . . 55
4.2 3D PatchMatch . . . 56
4.2.1 Scanning and sequential analysis . . . 56
4.2.2 Super-pixel Representation . . . 57
4.2.2.1 Super-pixel matching per slice . . . 58
4.2.2.2 Sparse-to-Dense Interpolation . . . 59
4.2.2.3 Experimental results . . . 61
4.2.2.4 Remarks . . . 63
4.2.3 Voxel Representation . . . 63
4.2.3.1 Coarse-to-Fine 3D PatchMatch . . . 65
4.2.3.2 Similarity Cost . . . 65
4.2.3.3 Residual Matching Cost . . . 67
4.3 Triangular Mesh Representation . . . 69
4.3.1 Heat Kernel Signature . . . 69
4.3.2 Discrete Mesh Laplacian . . . 70
4.3.3 Heat Kernel Signature of MV3 cell surface . . . 70
4.3.3.1 Matching Blebs in 3D+t Mesh Sequence . . . 74
4.3.3.2 Experimental Results . . . 75
4.4 Conclusion . . . 76

5 Variational Methods **81**
5.1 Introduction . . . 81
5.2 Census Constancy Assumption . . . 81
5.2.1 Census Transform in the continuous setting . . . 82
5.2.2 Euler Lagrange Equations . . . 84
5.2.2.1 Numerical Solver . . . 86

5.2.3 CS approximation factor . . . 87
5.3 Pyramidal approach/Coarse-to-Fine strategy . . . 87
5.4 Experiments . . . 89
5.4.1 Pseudo-ground truth generation . . . 89
5.4.1.1 Dataset 1 . . . 90
5.4.1.2 Dataset2 . . . 91
5.4.2 Parameterization . . . 94
5.4.3 Assessment of ASAE behaviour . . . 95
5.4.4 Results in real sequences . . . 99
5.4.5 Blebbing MV3 cell . . . 99
5.4.6 Collagen network in motion . . . 104
5.4.6.1 Foreground masking of flow fields . . . 110
5.5 Robust Regularization . . . 111
5.6 Time Complexity . . . 112
5.7 Conclusion . . . 113

6 Combination of the Matching and the Variational methods 123
6.1 Introduction . . . 123
6.1.1 Combining 3D PatchMatch with our variational method . . . 125
6.1.2 Experimental Results . . . 125
6.2 Combining 3D PatchMatch and the novel TV-based method . . . 127
6.2.1 Sub-problem #1 . . . 129
6.2.2 Sub-problem #2 . . . 130
6.2.3 Preliminary Results . . . 131
6.3 Conclusion . . . 133

7 Conclusion 139
7.1 Research Contributions . . . 139
7.2 Perspectives . . . 141
7.2.1 Short-term perspectives . . . 141
7.2.1.1 Parallelization of our method . . . 141
7.2.1.2 Sparse variation for regularization . . . 141
7.2.1.3 Visualization tools . . . 141
7.2.2 Long-term perspectives . . . 142
7.2.2.1 Triangular mesh-based representation . . . 142
7.2.2.2 Towards Deep-learning optical flow methods . . . 143
7.2.2.3 A 3D computer vision library for fluorescence volumetric data 145

8 Appendix **147**
8.1 SAE analysis of MV3 C 147
8.2 List of publications 148
8.2.1 Conference with abstract 148
8.2.2 Conference papers 148
8.2.3 Journal paper 149

References **151**

Résumé en français

Contexte

La microscopie optique 3D à fluorescence permet notamment d'enregistrer la dynamique cellulaire à une bonne résolution en espace et temps. L'analyse semi-automatique, voire automatique, des séquences volumiques (3D+t) ainsi acquises devient cruciale pour étudier et comprendre les mécanismes sous-jacents de la dynamique cellulaire. Cela pourra également contribuer aux progrès de la médecine moderne.

La première étape vers l'analyse du contenu dynamique des séquences volumiques est l'estimation du mouvement des éléments imagés, et plus précisément la mesure d'un champ de vitesses (ou flot optique) 3D.

Les intérêts de l'obtention de cartes de mouvement 3D en bio-imagerie sont au moins de trois ordres:

- *Aide aux microscopistes et focalisation de l'acquisition*.

Ce n'est pas une tâche facile d'amener une cellule à se comporter d'une certaine manière pour faciliter l'étude de certains aspects de son fonctionnement. Cela se fait généralement en modifiant l'environnement biochimique externe en fonction de connaissances préalables. Une telle observation est un processus long et exigeant de patience. Habituellement, un réseau de cellules ayant subi un traitement similaire est observé sous microscope, de sorte qu'elles peuvent être étudiées de manière parallèle. De plus, il est pratique d'utiliser deux microscopes, l'un à champ de vision large mais de basse résolution, l'autre à champ de vision restreint mais de haute résolution. Le premier peut permettre de trouver une région intéressante dans la cellule ou dans le tissu, pour que le second puisse permettre de se concentrer sur cette région particulière. Il est difficile de trouver automatiquement ces régions intéressantes. L'estimation de mouvement peut aider à résoudre ce problème en exhibant des régions dynamiques saillantes.

- *Evaluation d'autres quantités biophysiques*.

Des quantités telles que la force de traction ou le taux de croissance sont des mesures vitales pour comprendre ce qui est normal ou anormal dans le comportement d'une cellule. Les quantifier permet de répondre de manière quantitative à des questions telles que le comportement de la cellule dans un

environnement modifié ou l'effet d'un médicament sur celle-ci. La première étape vers une telle quantification est l'estimation du mouvement 3D.

- *Contribution à d'autres types de calcul dans l'image*.

L'estimation du mouvement 3D peut être utilisée comme un bloc fonctionnel dans un pipeline de segmentation, de recalage, de suivi, voire dans d'autres objectifs plus élaborés d'analyse automatique de l'image.

Objectifs

Si l'estimation du mouvement 2D (ou calcul du flot optique) dans des séquences d'images usuelles 2D, étudiée depuis plusieurs décennies, est aujourd'hui forte d'un corpus important de méthodes performantes, l'estimation du flot optique 3D dans des séquences 3D, et en particulier en imagerie microscopique, n'a pas fait l'objet de telles recherches approfondies. Outre la dimension ajoutée, les verrous résident dans la difficulté de formuler des hypothèses sur l'évolution de l'intensité entre deux images, l'absence d'outils d'évaluation appropriés, la nature anisotrope des volumes de données, le coût calcul des méthodes d'estimation de flot 3D. Néanmoins, les microscopes les plus récents sont capables de produire des séquences 3D à relativement haute résolution selon les quatre axes, et avec moins de bruit d'acquisition (voir Fig. 1). Nos recherches portent précisément sur la conception d'une solution à ces problèmes, qui soit la plus générale possible tout en assurant des niveaux de performance en précision et complexité satisfaisants.

Pour atteindre ces objectifs, nous avons conçu : (1) un algorithme d'appariement 3D PatchMatch, (2) une méthode variationnelle de flot optique 3D, les deux s'appuyant sur la signature Census, discrète dans le premier cas, continue dans le second, (3) des schémas de visualisation du flot 3D clairs et analysables, (4) une nouvelle métrique d'erreur pour le flot optique calculé lorsqu'aucune vérité-terrain n'est disponible.

Mise en correspondance 3D

PatchMatch (Barnes et al. 2009) est un algorithme de recherche du voisin le plus proche entre deux images 2D, populaire pour son exécution rapide. Il procède par balayage itératif de chaque point d'une image où de bonnes correspondances sont propagées depuis les points voisins, avec de plus une recherche aléatoire dans une région délimitée. Le coût d'appariement est défini par une mesure de similarité reposant sur l'intensité des points dans un patch. Cependant, sa transposition directe à des images 3D conduit à des temps de calcul élevés.

Recours aux super-pixels. Une première solution à ce problème est de sous-échantillonner l'ensemble des voxels du volume, de sorte que le temps de balayage complet de l'algorithme soit réduit. Cependant, le volume lui-même a une distribution éparse de points significatifs, et contient également des régions à faible rapport signal sur bruit. Dans ce cas, nous devons échantillonner judicieusement les voxels. L'un des moyens consiste à segmenter le volume en super-voxels et à ne scanner que leurs centroïdes afin de trouver leurs correspondants dans l'autre volume. C'est le principe de notre première méthode 3D PatchMatch. Puisque, pour un mouvement cohérent, les points voisins sont supposés se déplacer de manière similaire, nous pouvons considérer que le mouvement du centroïde vaut pour le mouvement des points appartenant à ce super-voxel. Cependant, les volumes étant très anisotropes, les centroïdes peuvent être disposés de manière très irrégulière, et le balayage nécessaire à la mise en œuvre de la propagation des correspondances devient alors difficile. Aussi, nous traitons le volume couche par couche, et du coup nous extrayons des super-pixels dans chaque couche du volume. En utilisant l'algorithme de super-pixels SLIC (Achanta et al. 2012), nous générons des super-pixels dans chaque tranche du volume qui sont ainsi de taille et de distribution relativement régulières. Nous pouvons ainsi utiliser l'ordre habituel de balayage d'une image 2D. Pour densifier les correspondances, nous avons développé une interpolation pondérée où les poids sont calculés à partir des directions des vecteurs de déplacement et des différences d'intensité entre les voxels voisins. Pour tester la méthode, nous avons synthétisé des séquences avec un mélange de mouvements de translation, de rotation et d'homothétie. Nous avons comparé la performance de notre méthode à celle d'Amat et al (2013) et nous avons constaté que notre méthode est plus performante. Cependant, si cette solution est bien adaptée au canal d'acquisition de microscopie de fluorescence correspondant à la cellule, pour le canal correspondant au collagène, qui entoure la cellule et contient principalement des structures filamentaires minces, la segmentation n'est plus opérante. Un balayage plus dense devient nécessaire, mais alors avec une incidence forte sur le coût calcul.

Recours à la multi-résolution. Pour concilier balayage dense et coût calcul raisonnable, nous avons adopté une approche multi-résolution pour une seconde version de notre méthode 3D PatchMatch. Nous calculons tout d'abord les correspondances entre une paire de volumes à résolution grossière, où l'initialisation se fait de façon aléatoire, et la recherche aléatoire est faite dans une région plus large du volume. Les correspondances obtenues à partir du niveau grossier sont utilisées comme initialisation à des niveaux plus fins successifs tout en réduisant à travers les échelles la région de recherche aléatoire à un plus petit volume. Cela réduit considérablement le temps de calcul. La méthode est de fait applicable à la fois aux canaux de la cellule et du collagène.

Nous avons testé et comparé trois mesures de similarité différentes, à savoir : la somme des différences quadratiques (SSD), la signature Census (CS) et la corrélation croisée normalisée tronquée (TZNCC). Sur la base d'une évaluation qualitative et en tenant compte de l'efficacité computationnelle, nous avons trouvé que CS était la mesure la plus performante pour les séquences de volumes de microscopie à fluorescence.

Nous avons d'autre part exploré la représentation par une variété 2D des images volumiques 3D du canal d'acquisition correspondant à la cellule. En utilisant l'algorithme du « marching cube » (Lorensen et Cline 1987), nous créons un maillage triangulaire 3D des données. Cela nous permet d'extraire des caractéristiques géométriques plutôt que des caractéristiques photométriques. Nous avons calculé la signature du « heat kernel » (Sun et al. 2009) à partir du maillage triangulaire, et ainsi nous pouvons détecter les « blebs » ou boutons, qui se rapportent à la région fortement courbée de la surface de la cellule. Nous segmentons la région du bouton et établissons la correspondance entre deux instants. Notre méthode permet de suivre les boutons, même si le volume et la surface spécifique des boutons évoluent avec le temps.

Mesure du flot 3D par une approche variationnelle

Avec la conclusion tirée de la comparaison des mesures de similarité pour 3D PatchMatch, nous avons également opté pour l'utilisation de CS dans le terme d'attache aux données de la méthode variationnelle de calcul du flot 3D que nous avons développée. Proposée à l'origine par Horn et Schunck (1981), l'approche variationnelle classique, et ses variantes ultérieures, utilisent l'hypothèse de conservation de l'intensité par le mouvement entre deux images. Cependant, cette hypothèse devient assez vite caduque en présence de bruit important (et pas nécessairement gaussien), et de fluctuation de l'intensité au cours du temps, comme dans les images de microscopie de fluorescence.

Nous formulons tout d'abord notre fonction d'énergie globale pour l'estimation du flot optique 3D en utilisant la pénalisation quadratique à la fois sur l'hypothèse de conservation de la CS pour le terme d'attache aux données et sur le lissage au premier ordre du flot pour le terme de régularisation. Nous avons comparé expérimentalement notre méthode à deux autres méthodes : une extension 3D de la méthode de Horn et Schunck (1981) que nous avons programmée, la méthode proposée par Amat et al. La méthode d'Amat et al utilise une pénalisation robuste de la conservation de l'intensité et un modèle markovien construit sur une carte de super-voxels. Nous avons montré

sur des exemples synthétiques et sur des exemples réels que notre méthode les surpasse toutes les deux. Avec ces expériences, nous démontrons que l'hypothèse de conservation de la CS est bien mieux adaptée que celle de conservation de l'intensité pour des images de microscopie de fluorescence.

De plus, nous fournissons une mise en œuvre de notre méthode avec une régularisation robuste du flot 3D. Les résultats de cette dernière version montrent une meilleure prise en compte des discontinuités du champ des vitesses 3D près des bords des structures dynamiques présentes dans le volume. Ces différentes mises en œuvre sont d'autre part fondées sur une approche multi-résolution pour appréhender correctement les grands déplacements.

Nous avons également conçu un pipeline pour combiner 3D PatchMatch et la méthode variationnelle, où les grands déplacements sont calculés par 3D PatchMatch et les petits déplacements résiduels par notre méthode variationnelle. Dans ce pipeline, 3D PatchMatch utilise la CS discrète et la méthode variationnelle la CS continue. Nous avons aussi proposé une implémentation parallélisable suivant Zach et al (2007), où nous utilisons notre extension de la CS continue en 3D dans l'attache aux données et la variation totale pour le terme de régularisation.

Visualisation des flots 3D

L'évaluation et l'étude comparative de flots optiques 3D sont difficiles à réaliser, surtout lorsqu'aucune vérité-terrain n'est disponible. La visualisation est l'un des moyens qualitatifs de comparer différents champs de vitesses. Un premier moyen couramment utilisé dans le cas 2D, consiste en des tracés de vecteurs sous forme de flèches, qui indiquent la direction et l'amplitude du mouvement en chaque point, ce qui fournit un moyen immédiat et intuitif d'appréhension de la nature physique du flot. Toutefois, pour éviter des chevauchements qui nuisent à la compréhension du champ ainsi tracé, seul un sous-ensemble de points sous-échantillonnés sur une grille régulière contribuent au tracé. Alors, les discontinuités éventuellement présentes dans le flot sont difficiles à cerner précisément. Ces constats sont encore plus marqués pour le cas 3D. Le tracé d'un flot 3D à l'aide de flèches est vite encombré, difficile à manier et encore moins intelligible.
Une alternative largement utilisée aujourd'hui pour la visualisation de champs de vitesses 2D est le recours à un code couleur. Il permet une visualisation dense du champ (au point près) et ainsi un affichage précis des discontinuités du flot, au détriment toutefois d'une appréhension très aisée de la nature physique du flot tracé. Il rend également plus facile les comparaisons entre champs calculés. Abramoff et al (2000) ont présenté une façon de représenter les flots 2D en utilisant la teinte et la

saturation des couleurs. La direction du mouvement est codée par la teinte et l'amplitude par la saturation (plus l'amplitude est grande, plus la saturation est élevée). L'extension de cette technique en 3D n'est cependant pas si simple. Premièrement, il n'y a pas assez de couleurs perceptibles à l'œil humain pour coder des vecteurs 3D de toutes orientations et amplitudes, même avec l'utilisation de l'information supplémentaire que représente la valeur dans l'espace des couleurs HSV (Hue – Saturation – Value). Deuxièmement, la visualisation des flots 3D en bio-imagerie doit être faite intelligemment, de façon à ce que l'outil affiche plus le flot sur le contenu pertinent que sur le fond de l'image, et que le rendu soit le plus intuitif possible pour le biologiste.

A cette fin, nous proposons trois schémas de codage couleur pour les flots 3D calculés dans des séquences de volumes de fluorescence :

- *Schéma 3DHSV*.

Dans ce schéma, nous utilisons l'axe de valeur pour représenter la part du mouvement entre les couches, à savoir le long de l'axe vertical du volume. Si le vecteur est dirigé vers le haut (axe des *Z* positifs), la valeur est plus claire. Inversement, s'il est dirigé vers le bas (selon l'axe des *Z* négatifs) la valeur est plus sombre. Les deux dimensions de l'espace des couleurs, teinte et saturation, continuent à encoder direction et amplitude respectivement, mais uniquement pour la part 2D du vecteur dans la couche du volume concernée. Ce schéma conduit à une visualisation couche par couche (ou tranche par tranche) du flot 3D au sein de l'image volumique, ce qui est généralement fait pour les images elles-mêmes. Cela révèle également les discontinuités dans le flot au sein de la tranche.

- *Schéma 3PHS*.

Dans ce schéma, nous partitionnons les données selon un plan. Seuls les vecteurs émanant de ce plan sont pris en compte. Ensuite, ces vecteurs sont projetés sur ce plan de section. Le flot 2D projeté qui en résulte est ensuite codé comme usuellement dans l'espace teinte et saturation. Nous choisissons trois plans orthogonaux (et tout d'abord un point de référence dans le volume) pour partitionner et projeter le flot 3D, afin d'obtenir la meilleure visualisation possible du mouvement sous-jacent. Il est primordial de choisir les plans qui fournissent le plus de contenus pertinents. Avec ce schéma, les discontinuités dans le flot deviennent aussi plus évidentes.

- *MIP-flow*.

Ce schéma s'inspire de la projection d'intensité maximale (PMI ou MPI en anglais) des données. L'intensité maximale le long d'un axe est calculée et projetée sur le plan orthogonal. Étant donné que les points les plus lumineux du volume se rapportent aux points marqués (ou labellisés) par fluorescence, la projection ne contient ainsi que des régions labellisées, donc d'intérêt pour le biologiste. En utilisant les vecteurs de mouvement provenant uniquement de ces points 3D, nous les

projetons aussi sur le plan orthogonal. Le flot 2D résultant peut être alors visualisé en utilisant classiquement les espaces teinte et saturation. Nous pouvons en outre superposer le flot 2D codé en couleur sur le MIP des données d'image pour produire une visualisation encore mieux localisée par rapport au contenu du volume. Nous pensons que ce schéma convient bien aux biologistes et/ou aux biophysiciens, qui ne s'intéressent qu'à ce qui est labellisé dans les données.

Evaluation quantitative sans vérité-terrain

La façon habituelle de procéder pour l'évaluation quantitative d'un champ de vecteurs de vitesses 2D (ou flot 2D) consiste à calculer l'écart entre les extrémités des vecteurs calculés et de ceux fournis par la vérité-terrain (EPE pour « End-Point Error » en anglais) ou l'erreur angulaire moyenne (AAE pour « Average Angle Error ») entre ces vecteurs. Ces deux mesures d'erreur exigent donc qu'une vérité-terrain soit disponible. Dans le contexte de la vision par ordinateur et de l'estimation de flots 2D, la vérité-terrain sur le mouvement est associée à la génération de séquences d'images de synthèse réalistes, comme dans le benchmark MPI-Sintel, où par construction l'information sur la scène 3D et son mouvement 3D est définie et le flot 2D vrai induit peut en être calculé. Des alternatives existent également comme pour la base KITTI, où les vidéos acquises sont réelles, mais une instrumentation supplémentaire permet d'obtenir l'information sur la scène 3D et la configuration de mouvement est particulière. Cependant, cela n'est pas possible pour les images de fluorescence 3D, car (1) nous ne comprenons pas suffisamment la dynamique aux niveaux cellulaire et sous-cellulaire pour pouvoir correctement les simuler, (2) la création d'un ensemble de données large et varié est un plus grand défi en soi.

Pour contourner ce problème, nous proposons une nouvelle mesure d'erreur appelée erreur angulaire structurelle (SAE pour « Structural Angular Error » en anglais). L'erreur dans le flot 3D, estimé entre un volume source et un volume cible, repose sur l'écart en chaque point entre le tenseur de structure de l'image source originale et celui de la source reconstruite par recalage inverse à partir du flot 3D calculé par la méthode d'estimation du flot. Plus précisément, seule l'orientation principale des tenseurs de structure locaux est utilisée, donnée par le vecteur propre correspondant. La différence angulaire entre les deux vecteurs propres forme l'erreur du flot calculé en chaque point. Nous produisons la carte des erreurs SAE pour une analyse locale détaillée du flot calculé, et la moyenne globale de l'erreur (ASAE) pour comparer la performance des méthodes testées. Nous avons aussi comparé le comportement des mesures ASAE et AEPE (moyenne globale des EPE) en présence de vérité-terrain, pour valider le bien-fondé de la mesure ASAE. Pour ce faire, nous calculons un flot 3D entre différentes paires de volumes à l'aide de la version 3D de Horn et

Schunck (1981), qui peut ensuite servir comme vérité-terrain. A partir de cet ensemble de champs de vitesses 3D, nous générons de nouvelles paires de volumes source et cible, sur lesquelles nous pouvons à la fois comparer notre méthode avec celle développée par Amat et al, et valider le comportement de la mesure ASAE au vu de celui de l'AEPE. Nous avons ainsi constaté que le comportement de l'ASAE était très semblable à celui de l'EPE, ce qui confirme que l'ASAE peut être utilisé pour une évaluation quantitative lorsqu'il n'y a pas de vérité-terrain disponible. De plus, la mesure ASAE peut être utilisée pour définir la bonne paramétrisation de la méthode de calcul du flot 3D.

Synthèse des contributions

Nous avons proposé deux approches de mise en correspondance 3D de type PatchMatch : super-pixel et multi-résolution. A partir des expériences menées avec notre méthode 3D PatchMatch, nous avons conclu que la signature Census est le choix optimal (en termes de performance et d'efficacité de calcul) dans sa version discrète d'origine pour 3D PatchMatch et dans une version continue pour concevoir le terme d'attache aux données d'une méthode variationnelle 3D. Cette dernière a surpassé la méthode de Horn-and-Schunck 3D et la méthode de Amat et al. Pour une évaluation visuelle et qualitative, qui représente un vrai enjeu en bio-imagerie 3D, nous avons proposé trois schémas de visualisation. D'autre part pour une évaluation quantitative en absence de vérité-terrain, nous avons défini une nouvelle mesure d'erreur SAE s'appuyant sur le tenseur de structure. Le schéma de visualisation n'est pas seulement utile pour le développement de méthodes de flot optique, mais aussi pour les biologistes et les biophysiciens. Notre mesure SAE a montré un comportement similaire à la mesure standard EPE en présence de vérité-terrain. La mesure SAE peut être utilisée pour produire une carte de confiance de n'importe quel flot 3D calculé sur des séquences volumiques réelles. De cette façon, nous avons réalisé l'estimation et l'évaluation de flots 3D pour des séquences de volumes de microscopie à fluorescence.

CHAPITRE 1

INTRODUCTION

The orderly course of the macroscopic organic life is an articulation of the microscopic biological world. The mechanics of how these two worlds interact is still a mystery to us. Microscopy has brought some light to the intricacies of many of these interplays. Robert Hooke (1635 -1703) was the first to discover a cell in 1665, when he had observed a cork under his microscope. Antoni van Leeuwenhoek (1632-1723) discovered microorganisms like protozoa and bacteria. Unlike Hooke's observation of a dead plant cell, Leeuwenhoek's witnessed these micro specimen to be motile which led to the conclusion that they were alive. This uncovered a whole new world of unicellular organisms co-existing with multicellular beings. Robert Koch (1843-1910) discovered bacteria causing diseases like tuberculosis, cholera, and anthrax using his microscope. He developed a drug as an attempt to cure tuberculosis. Though not successful, it is still used today for tuberculosis diagnosis. He also studied malaria infected inhabitants of New Guinea to discover Plasmodium parasite in their blood sample. He then observed how immune they were to the parasite, whereas outsiders were prone to the disease. He could not devise any medicine against the disease with that study but it shed light to an important phenomenon called acquired immunity.

More discoveries were made with microscopes which were related to the cell anatomy, their functions and morphologies (Mazzarello 1999). We have come to know that any perturbation of the underlying cellular or sub-cellular functions leads to abnormal behaviour of the cell. Deciphering such processes, and understanding what aggravates them or bolsters them, certainly helps the modern medicine. In order to do so, it is necessary to have proper imaging device that can record the dynamics of these processes, and computational tools that can measure the

bio-physical quantities from these records.

Earlier microscopes were limited to making 2D observation. Meticulous drawings were drafted for the record. It was in 1986 when a digital microscope with a digital camera was made by Hirox Co. LTD., Tokyo, Japan. Large amount of spatio-temporal (2D+time) information could then be collected from the microscope. 3D bio-imaging with a microscope, on the other hand, is not an easy task. In 1981, Gerd Binnig and Heinrich Rohrer invented the scanning tunneling microscope based on the wave nature of the electron to make a 3D scan. The source of information was electron rather than the photon. Guckenberger *et al.* (1988) describes its application to biological studies, where a conducting medium is used to support a biological specimen. An auxiliary light microscope was used to identify interesting area to scan. A current standard approach to 3D scanning of specimen is light-sheet fluorescence microscopy (LSFM). While the main work in this thesis is developing computation tools for 3D motion estimation, we believe it is necessary to introduce the microscopy techniques that are common in 3D acquisition of biological data. Keeping the review of microscopy limited to the scope of our work, now we discuss LSFM.

1.1 Light-Sheet Microscopy

Light-sheet microscopy was first introduced in 1902 as the ultra microscopy by Siedentopf and Zsigmondy (1902), where diffraction limit was realized by passing sunlight through an adjustable slit aperture. Based on this, Voie's group in 1993 proposed an imaging device called orthogonal plane fluorescence optical sectioning (OPFOS). Figure 1.1 depicts a simplified scheme of the microscope. Mainly, it consists of a pair of cylindrical lenses. A laser beam is passed through the lens on one end to produce a thin light sheet. Orthogonal to the light sheet, an objective lens is placed such that the thinnest part of the beam (waist) is at its focus. The specimen is placed at the waist of the light sheet. The emitted signal from the section of the specimen lying in the waist is captured through the objective lens. This is called optical sectioning. The specimen, when moved via a movable stage in z-axis, usually by a high precision stepper motor, enables the microscope to image sections in the third dimension. The stack of such images forms a 3D volume of the specimen. Later in 2004, Huisken *et al.* (2004) applied this principle with fluorescence microscopy system calling it selective plane illumination microscopy (SPIM). They generated images of Medaka and Drosophilia during embryogenesis using SPIM.

1.1.1 Fluorescence and Fluorophores

Fluorescence is a phenomenon where a substance emits visible or invisible radiation when a radiation with much higher frequency is incident on it. Fluorophores are such chemical

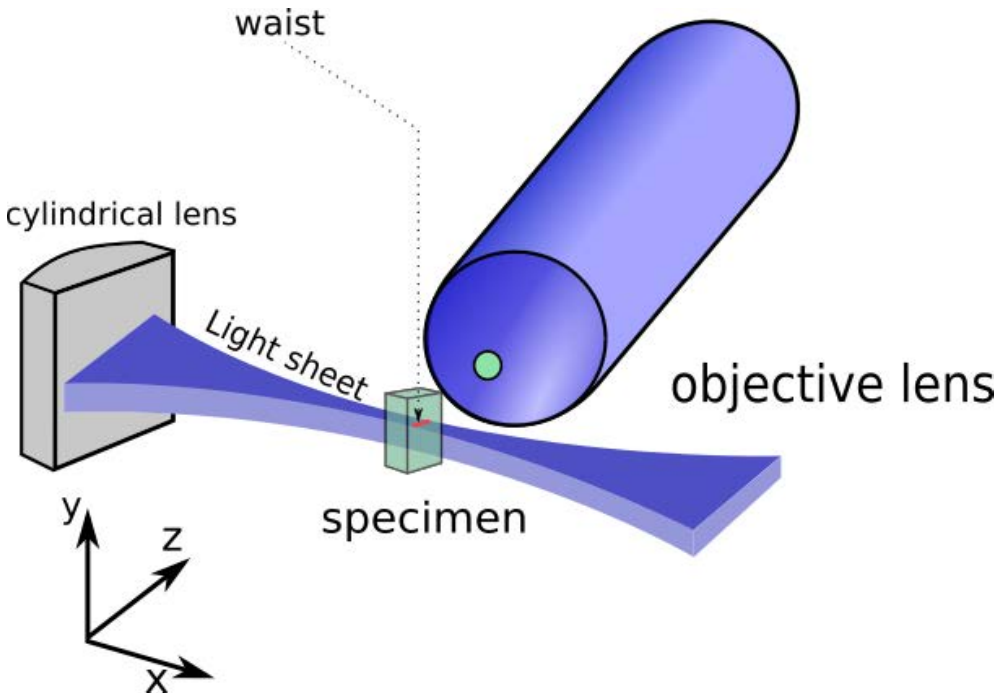


FIGURE 1.1 – A simplified view of a light-sheet microscope. Cylindrical lens in gray (the one at the other end of the light sheet is not shown in the figure). Objective lens is shown in green. The beam waist is highlighted in red.

compounds which exhibit this phenomenon. They are naturally stable, but when they are excited with an energy wave, their electrons absorb photons to reach higher energy state. In order to become stable again, they re-emit photons, but with lower energy as some energy is spent in the vibration. The low energy wave, now with higher wavelength can be captured to observe the fluorophores. Fluorophores have their own excitation and emission wavelengths which are exploited to distinguish different targets within an object.

A naturally occurring fluorophore was described as photogenic tissue in a jellyfish (Hydromedusae) by Davenport *et al.* (1955). It emitted green colored light. However, it was Shimomura *et al.* (1962), who identified the green fluorescent component as a protein (GFP), and also provided a way to isolate the protein from the jellyfish. Chalfie *et al.* (1994) demonstrated that GFP expression can be used to monitor the gene expression and protein localization in living organisms. Ever since, many variants of GFP have been engineered, and used in labelling of other proteins in a biological specimen. GFP can be encoded to the gene of another specimen (Dhandayuthapani *et al.* 1995). If expressed by the cell of the specimen, it glows green upon application of the excitation wave, which can be identified and measured with LSFM. Its expression has paved a way to understand many biological functions.

1.1.2 Light-Sheet Fluorecence Microscopy

Over the years, many variations of the LSFM have been developed to capture fluorescence efficiently. Most of them try to solve common issues like, light scattering and absorption in the out-of-focus sections of large specimen, time resolution, photobleaching and phototoxicity. Continuous cycle of excitation and emission of energy waves degrades the fluorophore's structure. Over a long duration, the fluorophores can no longer emit radiation eventhough the excitation energy is high enough. This is known as the photobleaching. On the other hand, phototoxicity is induced when the cycle of excitation and emission of energy waves, as well as the electronically sensitive intracellular constituents eventually degrade the cell's health. To minimize this effect, excitation beam is lowered in intensity and restricted to the focal plane. For instance, MuVi-SPIM (Krzic *et al.* 2012) dealt with light scattering in large specimen using two objective lens instead of one to obtain two images at a time. The pair of images was processed to reduce distortion induced by scattering. Additionally, they used a fast camera sensor for higher acquisition speed. Thus, the effect of photobleaching in acquisition became less adverse.

Typically, a Gaussian beam is employed to generate the light sheet. However, upon increasing the field of view (FOV), the beam waist also thickens (Dean *et al.* 2016). Consequently, the axial resolution worsens. Application of lattice beams (Betzig 2005), Bessel beams (Planchon *et al.* 2011), and Airy beams (Vettenburg *et al.* 2014) were proposed as the remedies to the aforementioned problem. A review on variants of LSFM can be found in (Krishnendu *et al.* 2018). However, we briefly describe the microscopes that were employed to acquire the images we have processed during the thesis. Our work was carried out in collaboration with Danuser Lab, UT Southwestern, Texas within the Inria associated-tem program, which motivates the microscopy acquisition and the related biological context.

1.1.3 Axially Swept Light-Sheet Microscopy (ASLM)

Dean *et al.* (2015) acquired images with a large FOV, while using Gaussian beam to generate the light sheet. For this, they restrict the acquisition to only a narrow strip of in-focus region of the optical section (see Fig. 1.2). The strip of active pixels in the sCMOS sensor is translated across the sensor at a specified velocity. In tandem, the axial position of the beam is modulated with an remote focusing method. Finally, a slice is accumulated by superposition of many small and overlapping images. Lateral resolution (XY-plane) of 380 ± 20 nm and a Z-resolution of 390 ± 6 nm are reported. They go on to demonstrate the 3D biological imaging ability of the microscope by acquiring a large volume of $162 \times 162 \times 100 \mu m^3$ containing multiple MV3 melanoma cells. The authors also acquire the collagen environment where the cells are suspended. The resolution of spatio-temporal data was high enough to perform 3D particle

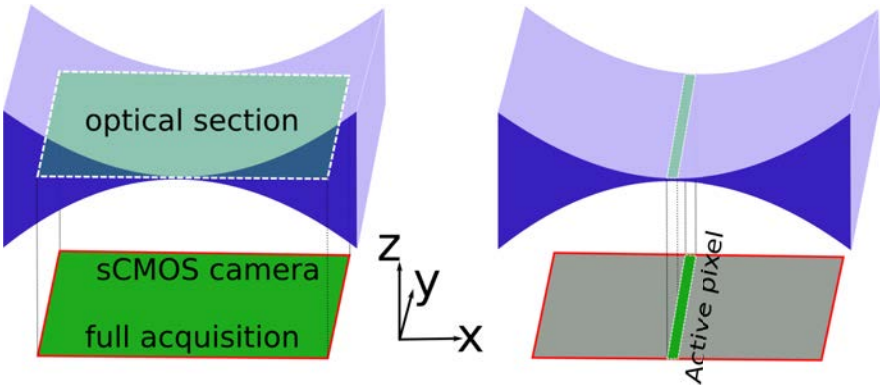


FIGURE 1.2 – Left : LSFM scheme where full acquisition of the optical section is made by a camera. Right : ASLM scheme where only a narrow strip of optical section is read by the camera.

tracking (Jaqaman *et al.* 2008).

1.1.4 Diagonally Scanned Light-Sheet Microscopy (DiaSLM)

DiaSLM (Dean *et al.* 2016) mainly differs from ASLM in the sample orientation while scanning. In this type of microscope, a sample is mounted 45° to the optical axis of light-sheet and camera optics as seen in Fig. 1.3. This allows optical sectioning of even the tallest part of a cell. The cell specimen is placed in a glass cover slip. To avoid the need of longer light-sheet, the cover-slip is swept diagonally while scanning. Volumetric imaging of cells up to 3.5Hz rate is reported. The raw data do not exhibit out-of-focus blur, and hence requires no extra post processing of deconvolution. The data is in Euclidean frame and thus, requires no signal-altering transformations.

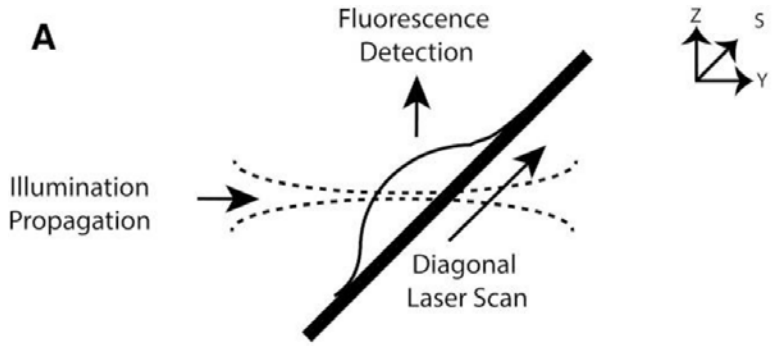


FIGURE 1.3 – Scheme of DiaSLM. Picture courtesy of (Dean *et al.* 2016).

1.2 Biological Context

Before we delve into the objectives of the thesis, it is necessary to provide some context to the type of scenarios and related biological problems we are dealing with. For this reason, we discuss the specimen and their dynamics which have been acquired by the microscopes discussed in the previous subsections, and used in the experimental evaluation of our method. We are not driving towards answering biological questions but devising tools that could be helpful to the biologists and the biophysicists studying cell dynamics.

1.2.1 MV3 Melanoma (MV3) Cell

Skin cancer is the result of malignant melanoma (MV3) cell. It is tenacious in survival, and has baffled the biomedical community by its resistance to apoptosis and drugs. Few therapeutic strategies have been approved such as single or combined sessions of chemotherapy, electro-chemotherapy, administration of drugs, immunotherapy and others, which can be reviewed in (Domingues *et al.* 2018). However, these treatments face challenges such as : patient's health level to undergo the treatment(s), planning a targeted therapy in advance, no guarantee of affirmative response by the patient's immune system.

Once the melanoma cells become motile, they move along external matrix of tissues and/or organs. This state is called metastasis of the MV3 cell. It is believed that the metastasized MV3 cells are invulnerable to body's immune system (Zbytek *et al.* 2008). They migrate via circulatory or lymphatic system ; they extravasate and colonize distant secondary tissues. This makes them hard to contain and treat. Numerous studies have been made *in vivo* and *in vitro* to understand this migratory behaviour, which could possibly lead to a cure. Cancer cells displace themselves *in vivo* in an extracellular matrix (ECM) with a migratory track, often created by a leader cell (Gaggioli *et al.* 2007), or sometimes by other anatomical structures (Friedl and Alexander 2011). Experiments to study migration of cancer cells have been conducted *in vitro*, where ECMs are engineered for a better control of migration space, substrate stiffness and other constraints (Pathak and Kumar 2012).

1.2.2 Extra Cellular Matrix (ECM) and Collagen

ECM is a non-cellular environment where cells are confined to, and are ubiquitous within all tissues and organs. Beside providing geometrical integrity, aesthetics and tenacity to the tissues and the organs, they play crucial role in biomechanical regulation of cells, like migration as discussed in previous subsection. Though heterogeneous in composition and topology, the ECM mainly constitutes of water, proteins and polysaccharides. Its macromolecular composition is described in (Frantz *et al.* 2010) as Proteoglycans (PGs) and Fibrous proteins.

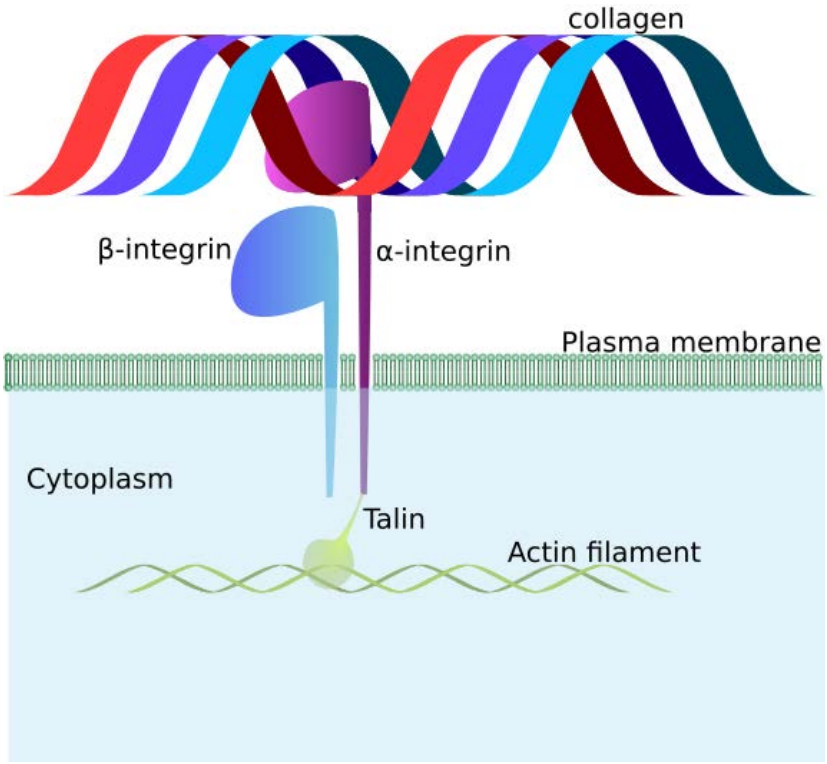


FIGURE 1.4 – A simplified view of how a cell might adhere to collagen via integrin.

PGs are long polysaccharide chains that function to provide buffering and reinforcement to the ECM composition. Their hydrophilic nature forms hydrogel which protects ECM from external stresses.

Collagen, which is one of the most abundant fibrous protein, is what has gathered interests of biologists and biophysicists. Collagen is a trimer that provides the main structure to the ECM. Beside strengthening the matrix, it regulates cell adhesion, migration and tissue growth (Plant *et al.* 2009). It is roughly 50 nm in diameter, and evidently possesses enormous tensile strength. Figure 1.4 shows a scenario in which a cell (not necessarily a cancer cell) attaches itself to a nearby collagen fiber. The actin-filament present in the cytoskeleton near the cell membrane is attached to a protein called talin, which then is chained to another protein called integrin, that protrudes beyond the cell membrane to hook itself to the collagen. This is one of the ways, how the cell adheres to ECM. The signaling between these proteins is still an open topic for research (Khalili and Ahmad 2015).

1.2.3 Locomotion of MV3 cell

The network of collagen is porous in nature. This allows cells to pass through. Studies suggest (refered within (Colin *et al.* 2016)) that when the pores of ECM are not large enough,

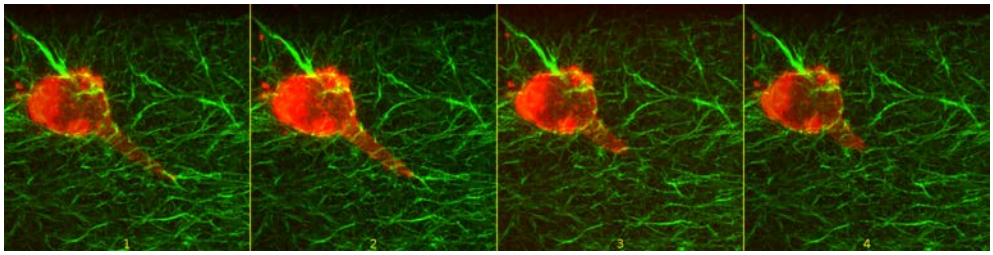


FIGURE 1.5 – Maximum intensity projection (MIP) in xy-plane : MV3 cell (in red) migrating in a collagen (in green) environment. The filopodium attaches to the collagen fiber, builds tension, unlatches itself (in 3^{rd} frame) to slingshot itself to the top-left corner of the frame. The image was acquired with ASLM (Dean *et al.* 2015). Courtesy of Danuser Lab, UT Southwestern, Texas, USA.

normally a cell ceases to move. This assures the protection of rigid cell nucleus. Enzymes called Matrixins or MMPs are capable of degrading and remodeling fibers of ECM to create migratory tracks. A leader cancer cell is able to take advantage of these tracks. The followers take the paved path. Additionally, it can also invade in an MMP-independent manner, where it takes amoeboidal form (Wyckoff *et al.* 2006). This form allows it to protrude its cell membrane to produce any of the locomotory structures : blebs (spherical protrusion), lamellipodia (broad and thin projection), filopodia (long and slender protrusion). It grows protrusion(s) in the leading edge, which can adhere to collagen scaffoldings, and generate enough tension inside the cell by contraction to slide itself in the trailing edge (Friedl and Alexander 2011). This sequence is depicted in Fig. 1.5. The maximum intensity projections in XY plane are shown in the figure. It is a convenient way to view 3D volumetric data, where only the points with maximum intensity along the normal of the projection plane viewed.

Blebbing is another interesting phenomenon, which happens at the surface of a MV3 cell in absence of ECM. Blebs are sphere-like protrusions that are devoid of actin filament. Normally, for a non-cancerous cell, it occurs towards the end of its life. The bleb protrusion expands with tearing of cell membrane to enroll cytosol and lipids into it. The expansion lasts around 30 seconds. Gradually, it rejects the cytosol into the cell while actin is recruited in its cortex. The retraction lasts for around 2 minutes (Charras 2008). MV3 cell utilizes this phenomenon to drive itself in a non-adhesive environment. Figure 1.6 depicts the blebbing of an MV3 cell placed on a cover-slip.

1.3 Towards Motion Estimation

Having stated the typical scenarios of cell dynamics, now we motivate the 3D motion estimation in biological structures. Broadly speaking, the benefits of the motion estimation in

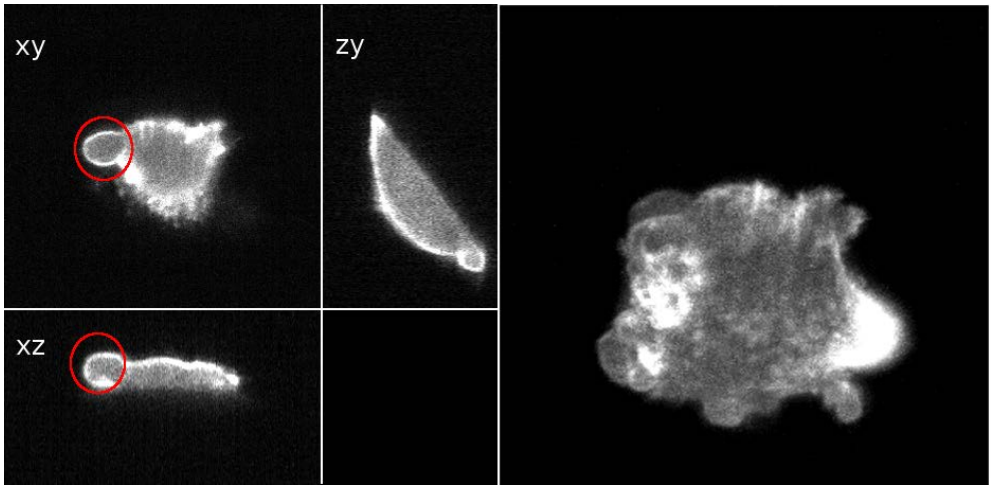


FIGURE 1.6 – MV3 cell in a cover-slip. Left : Orthogonal projection of the volume. A bleb is circled in red. The angular cover-slip used in DiaSLM (Dean *et al.* 2016) can be infered from the zy projection. Right : MIP in xy-plane. Courtesy of Dansur Lab, UT Southwestern, Texas, USA.

bio-imaging are threefold :

- **Quantification of other bio-physical quantities** Quantities like force, traction strength or rate of growth are vital measurements to understand what is normal and abnormal behaviour of a cell. With this, questions like how does the cell behave in an altered environment and how a drug affects it, can be answered in a quantitative way. One of the first steps towards such quantification is estimating motion from sequences of light microscopy volumes.
- **Facilitation of microscopists for acquisition** It is not an easy task to get a cell to behave in a certain way. It is usually done by manipulating the protein signalling or external bio-chemical environment based upon our prior knowledge. Such observation is a long and patience demanding process. Usually, an array of cells that have all undergone a similar treatment are observed under a microscope, so that they can be observed in a parallel manner. Furthermore, it is a practice to use two microscopes, one with low FOV at high resolution and the other with high FOV at lower resolution. The latter one finds an interesting region in the cell so that the former can focus on that particular region. It is difficult to automatically define such interesting regions. Motion estimation can help solving this issue by finding saliently dynamic regions, and help us to define what is interesting and what is not.
- **The aid to other computational process** Motion estimation can be used as a functional block in the pipeline of segmentation, registration, tracking and other higher-level image processing.

.

With such motivations, 2D and 3D optical flow methods for biological images have been researched over the past decades. Horn and Schunck (1981) developed a variational model for motion estimation in a pair of generic 2D images which assumed brightness constancy of objects along their trajectory. Baron (2004) experimented with this model in the context of 3D MRI cardiac images. Delpiano *et al.* (2012) tested the variants of the model in fluorescence point confocal 2D microscopy images. Based on graph of super-voxels, Amat *et al.* (2013) devised a 3D variational optical flow method, which is primarily targeted to cells that have relative translational motion, e.g., the scenario of Drosophila embryo development. The method was quantitatively tested with synthetic data exhibiting fluorescent nuclei with linear motion types. Boquet-Pujadas *et al.* (2017a) presented another 3D variational motion estimation in an amoeboid cell based on data assimilation. Some bio-physical measurements were obtained beforehand with other methods to model the cell. However, the model becomes obsolete when testing a different specimen, for e.g., collagen network.

Tracking particles in fluorescence images is also of great interest. However, the motion estimation is not as dense as in optical flow. Jaqaman *et al.* (2008) proposed a tracking algorithm to link particles between successive frames by the means of combinatorial optimization. Vallmitjana *et al.* (2017), to track mitochondrial movement in neurons, detected and segmented dynamic structures from 2D time-lapse fluorescence microscopy images. Then, a probabilistic tracking was performed to cluster the centroids of the segments. Particle image velocimetry (PIV) (Raffel *et al.* 1998) is another method to compute displacements in images. For localization of a particle, it relies on cross-correlation of intensities between a pair of patches from source and target images. However, the cross-correlation operation is computationally expensive. Usually, short spatial displacements are computed by limiting the search region to a smaller area. The method has been employed to estimate 2D motion of migrating cells in (Petitjean *et al.* 2010), spreading of breast cancer in (Weiger *et al.* 2013), bead tracking in (Style *et al.* 2014), to mention a few. Dhruv *et al.* (2016) compared the performance of the PIV method with an optical flow method (Lucas and Kanade 1981). They report the optical flow method to be 26 times faster than the PIV method while producing similar results in 2D. The faster performance of optical flow method makes it more attractive than the PIV method in order to estimate motion in 3D.

We adopt variational approach to 3D optical flow computation in fluorescence image sequences. The approach comprises of a data-term and a regularization term. The data-term mainly drives the method to a solution based on some assumption such as the brightness constancy of voxels. The regularization, on the other hand diffuses the motion vectors to the region where the assumption is not valid. Brightness constancy is a bold assumption for the

fluorescence volumetric images. The image has regions with different signal to noise ratio. The void region is characterized by gaussian noise, while the fluorescence labelled regions have poisson-gaussian noise. On top of that, the photobleaching leads to variation in the intensity levels. Therefore, it is necessary to devise a data-term which can overcome such issues.

Additionally, a strategy has to be devised to deal with large-scale motion as the variational methods are valid for smaller displacements. The usual coarse-to-fine approach can be employed. However, this approach is flawed by its inability to capture fast moving smaller objects, and the requirement to determine the number of coarse levels beforehand, which is not obvious. Efficient matching of correspondences (Barnes *et al.* 2009), (Weinzaepfel *et al.* 2013) can be employed to deal with large displacements. Thus obtained flow field can then be refined with a variational optical flow method (Brox and Malik 2011). A detailed literature review about the motion estimation can be found in the following chapter.

With additional dimension to motion estimation problem, the complexity rises not only in the computation, but also in the validation of the method. First of all, reliable ground truth motion fields for fluorescence method are not available. The synthesis of 3D motion field that reflects the underlying biological dynamics is another challenge in itself. It is so because (1) mimicking such dynamics computationally requires a deeper understanding of the specimen (2) efficient tools to manipulate volumetric data in 3D space are unavailable. Therefore, we motivate the development of no-reference based assessment. Visualization is another way to qualitatively evaluate the computed flow fields. However, 3D flow field visualization is not straightforward as its 2D counterpart. Especially, in case of 3D+t fluorescence images, where the data themselves demand smart ways of visualization, the associated 3D flow field certainly needs appropriate visualization in order to be able to make a qualitative evaluation.

1.4 Research Aims

Our work is driven by the knowledge that the currently available 3D motion estimation tools for fluorescence images are limited to certain specimen and/or motion types, and they do not provide proper assessment tools. Therefore, the aim of this thesis is to mitigate the limitations by developing 3D motion estimation and assessment tools, which can allow to study the dynamics of generic biological objects.

The following objectives are necessary for the achievement of this aim :

- To test various data-terms that could make motion estimation in fluorescence volumetric images more reliable.
- To test the methods with data with diverse shape and motion types : We are aware that the biological objects are diverse in morphology and dynamics. However, we believe

that the experiments with the cell and the collagen are complex and diverse (in both structure and motion types) enough to direct us towards formulation of a generic motion estimation tool.

- To explore the ways to color code 3D vector fields for qualitative evaluation : Color coded flow field is widely accepted visualization of 2D flow fields. However, its 3D extension is not straight forward. Additionally, the volume is mostly void and interesting dynamics happen somewhere in smaller regions. Thus, visualization demands smarter way to describe the flow fields in those regions.
- To devise a confidence measure for estimated flow fields in absence of ground truth : Assessment of 3D motion estimation methods is difficult to perform as no ground truth is available on real sequences. However, quantification of error is necessary not only for comparison of methods, but also for the parameter tuning.

1.5 Structure of the Manuscript

We now provide a brief walk through of the chapters to come.

In Chapter 2, we will review the literature of the 2D and 3D optical flow methods. We go through the work that we consider milestones in the research of 2D optical flow field. Variational methods as well as learning based methods are reviewed. We also review the current state-of-the-art method for 3D motion estimation in fluorescence image sequences.

Chapter 3 describes the data that we have used to demonstrate our work in this manuscript. It also describes our proposed visualization techniques and assessment measure that will be used in chapter 5 for evaluation of optical flow methods.

Chapter 4 describes various matching strategies we have developed. We present super-pixel based 3D PatchMatch as well as coarse-to-fine 3D PatchMatch. In pursuit of appropriate data-term, we test three different similarity measures for matching voxels in real sequences with our 3D PatchMatch. We also explore the triangular mesh based representation of 3D volumetric data. We use heat kernel signature to detect, segment and track blebs in a MV3 cell.

Chapter 5 describes our proposed 3D variational method that uses Census signature based data term. We will evaluate the performance of our method against the classical 3D Horn-and-Schunck and the state-of-the-art Amat's method. Further, we will evaluate the performance of our assessment measure with respect to the standard optical flow error, namely End-point-Error.

Chapter 6 describes efficient ways to combine the results of 3D PatchMatch and 3D variational method to deal with small and large motion amplitudes.

Chapter 7 provides our concluding remarks on our work, and perspectives and prospects of 3D motion estimation for fluorescence volumetric data.

Chapitre 2

State-of-the-Art on Optical Flow

In computer vision, there has been a plethora of work for motion estimation in 2D image sequences. We restrict ourselves to the study of methods that have been milestones of the research work in optical flow problem. We start by studying the 2D methods and extending them in the 3D setting. We consider an image I in domain Ω, at time step t, which has a point $\boldsymbol{x}$ that moves with velocity $\boldsymbol{w}(\boldsymbol{x})$, with $\boldsymbol{w}$ being the flow field between the image sequence. Even though we review 2D methods, we keep the dimension of the domain generic as those methods are tractable to higher dimension.

2.1 Variational Methods

The pioneering work of Horn and Schunck (1981) led to the development of various variational methods for 2D optical flow in natural images (Sun *et al.* 2010; Fortun *et al.* 2015). The methods in this family requires an energy functional to be minimized which generally consists of data fidelity term/s and regularization term/s :

$$E(\boldsymbol{w}) = \int_{\Omega} D(I, \boldsymbol{w}) + \alpha \mathcal{R}(\boldsymbol{w}) d\boldsymbol{x}, \qquad \boldsymbol{x} \in \Omega \tag{2.1}$$

The data term D in Eq. 2.1 drives the minimization towards the solution, while the regularizer $\mathcal{R}$ imposes certain constraints to keep the problem away from being ill-posed. Horn and Schunck (1981) defined the data term as the quadratic norm of the displaced intensity difference at each point between an image pair. This implies that a point along its trajectory should remain similar in its appearance. This assumption is named as brightness constancy assumption (BCA). Similarly, they proposed quadratic norm of the velocity gradient $\nabla \boldsymbol{w}$ as the regularizer, assuming

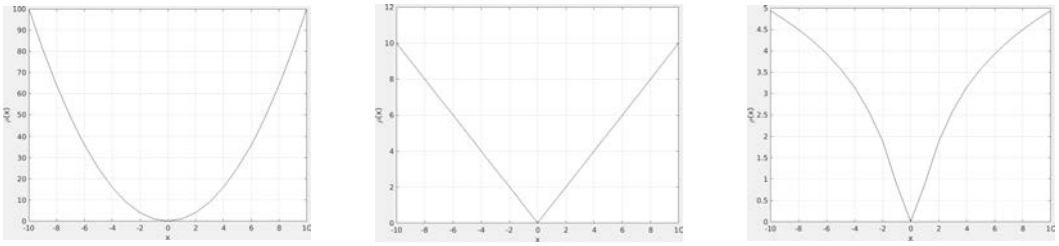


FIGURE 2.1 – From left to right : L_2, Charbonnier, and Lorentzian norms

a smooth velocity field over the image domain Ω. This helps in filling the region with flow field where the BCA becomes invalid. This constraint is known as flow smoothness constraint. α is an empirical scalar value to balance the two terms. Equation 2.2 presents the Horn-and-Schunck energy formulation :

$$E(\boldsymbol{w}) = \int_{\Omega} |I(\boldsymbol{x} + \boldsymbol{w}, t + 1) - I(\boldsymbol{x}, t)|^2 + \alpha |\nabla \boldsymbol{w}|^2 d\boldsymbol{x}, \tag{2.2}$$

The linearization of the data term calls for Taylor series expansion which is generally limited to first order. This is written as :

$$I(\boldsymbol{x} + \boldsymbol{w}, t + 1) - I(\boldsymbol{x}, t) = I_x u + I_y v + I_t, \tag{2.3}$$

where I_x, I_y are the spatial derivatives of I, and I_t is the temporal derivative.

Two strong limitations of the method is that it is valid only for small displacements, and it does not preserve motion discontinuities. To address these problems, a more accurate model was proposed by Heitz and Bouthemy (1993). They integrated edge-based motion constraint to handle occlusions and discontinuities, while dealing with the large displacements using the multi-resolution setup proposed by Anandan (1989) in context of dense image pair matching. The basic idea of this setup is to reduce large displacements into smaller ones by reducing the image pair resolution. This allows fast computation of global motion in coarsest level, which is upscaled and warped in next finer resolution. The motion field is then iteratively refined in subsequent finer stages by a series of energy minimization. This scheme is also referred to as coarse-to-fine framework.

Mémin and Pérez (2002) and Brox *et al.* (2004) derived this scheme as a single minimization problem by integrating warping into mathematical derivation. The latter work, in addition to BCA, uses the gradient constancy assumption (GCA) (Haralick and Lee (1983); Uras *et al.* (1988)), which is more robust to small variations in image intensity. The robustness to such variations was also addressed by Black and Anandan (1996), by proposing a non-convex norm called Lorentzian norm for both data and regularization terms. The penalizers reduces the influence of outliers (points where constraints are not valid) in the estimation. Brox *et al.* (2004) made Charbonnier penalty popular in practice for computing optical flow. Table 2.1 presents

TABLE 2.1 – Penalizers in variational energy

Norms	Equations
Quadratic (L_2)	$\rho(x) = x^2$
Charbonnier (L_1^{approx})	$\rho(x) = (x^2 + \epsilon^2)^a$
Lorentzian	$\rho(x) = log\left(1 + \frac{x^2}{2\epsilon}\right)$

FIGURE 2.2 – 2D optical flow for a pair of images displayed using Middlebury style color coding. Left : ground-truth, Middle : Horn-and-Schunck method, Right : Brox's method.

the formulation of these norms with a fixed scalar value ϵ wherever mentioned. Figure 2.1 plots their behaviour. A generalized Charbonnier function is used to approximate L_1-norm which is convex when $a \geq 0.5$. Sun *et al.* (2010) showed that it is more robust than quadratic/L_2-norm or other non-convex norms. This penalization in regularization has another desirable effect which is the estimation of flow fields with discontinuities. Figure 2.2 shows the color coded flow field computed by Horn-and-Schunck method, which is smooth even across motion boundaries, and Brox's method (Brox *et al.* 2004), which preserves discontinuity at the motion boundaries.

This completes the basic skeleton of variational optical flow which consists of the data term and the regularization term. The choice of these components and the minimization of to the energy function vary greatly across the literature. Now, we briefly describe the common ways to finding the solution.

2.1.1 Minimization

2.1.1.1 Euler-Lagrange equations :

To minimize the optical flow energy function, the associated Euler-Lagrange (E.L.) equations have to be found and then solved. Let F be a function encapsulating all the terms in Eq. 2.1.

$$E(\boldsymbol{w}) = \int_{\Omega} F(x, y, u, v, u_x, u_y, v_x, v_y)dxdy. \tag{2.4}$$

From calculus of variation, the minimizer of the function is located where its first order derivatives vanish. We write the associate E.L. equations as :

$$F_u - \frac{\partial}{\partial x} F_{u_x} - \frac{\partial}{\partial y} F_{u_y} \approx 0, \tag{2.5}$$

$$F_v - \frac{\partial}{\partial x} F_{v_x} - \frac{\partial}{\partial y} F_{v_y} \approx 0, \tag{2.6}$$

with reflecting Neumann boundary conditions $\boldsymbol{n}^T \nabla u = 0, \boldsymbol{n}^T \nabla v = 0, \boldsymbol{n}^T \nabla w = 0$, where $\boldsymbol{n}$ is normal at boundary points. For solution, Horn and Schunck (1981) discretized the associated partial derivatives using finite element methods. The system of PDEs can then be rewritten in matricial form of $Ax = b$, and solved for iteratively using Jacobi or Gauss-Seidel methods (Hamming (1973)). Successive over-relaxation (David M. Young (1971)) is another technique which has faster convergence rate than the previous two methods. However, neither of them are parallelization friendly methods. A faster and parallelization friendly method is reported in Mémin and Pérez (1997). Gwosdek *et al.* (2012) implemented the method of Brox *et al.* (2004) for efficient GPU computation, based on fast explicit diffusion (Grewenig *et al.* (2010)).

2.1.1.2 Splitting methods :

Another variant of minimization of optical flow energy is based on splitting methods. It separates the data and regularization terms, and minimize the subproblems turn-wise. Proposed by Zach *et al.* (2007), it is a real-time and massively parallelizable method. The proposed energy function reads :

$$E_\theta(\boldsymbol{w}) = \int_\Omega |I(\boldsymbol{x} + \tilde{\boldsymbol{w}}, t+1) - I(\boldsymbol{x}, t)| d\boldsymbol{x} + \frac{1}{2\theta} \int_\Omega \|\boldsymbol{w}(\boldsymbol{x}) - \tilde{\boldsymbol{w}}(\boldsymbol{x})\|^2 d\boldsymbol{x} + \alpha \int_\Omega |\nabla \boldsymbol{w}(\boldsymbol{x})| d\boldsymbol{x}, \tag{2.7}$$

where $\tilde{\boldsymbol{w}}$ is an auxiliary flow field that is supposed to be close to the original solution, θ is the coupling term. The auxiliary flow field is generally found by up-sampling the flow field computed in coarser level. The authors use true L_1 norm instead of the approximation proposed by Brox *et al.* (2004). The consideration of total variation in the regularization and L_1 penalization in the data term led to its name, TV-L1 optical flow. The minimization alternates between (i) fixing $\tilde{\boldsymbol{w}}$ and solving for $\boldsymbol{w}$, which has the form of :

$$\min_{\boldsymbol{w}} \int_\Omega \left(\frac{1}{2\theta} \|\boldsymbol{w}(\boldsymbol{x}) - \tilde{\boldsymbol{w}}(\boldsymbol{x})\|^2 + |\nabla \boldsymbol{w}(\boldsymbol{x})| \right) d\boldsymbol{x}, \tag{2.8}$$

and (ii) fixing $\boldsymbol{w}$ and solving for $\tilde{\boldsymbol{w}}$, which has form of :

$$\min_{\tilde{\boldsymbol{w}}} \int_\Omega \left(\frac{1}{2\theta} \|\boldsymbol{w}(\boldsymbol{x}) - \tilde{\boldsymbol{w}}(\boldsymbol{x})\|^2 + |I(\boldsymbol{x} + \tilde{\boldsymbol{w}}, t+1) - I(\boldsymbol{x}, t)| \right) d\boldsymbol{x}. \tag{2.9}$$

Equation 2.8 takes the total variation based denoising model of Rudin *et al.* (1992) which is solved using the algorithm described in (Chambolle 2004). The second half of the problem is solved by point-wise thresholding. Both of the sub problems are massively parallelizable and even appear in OpenCV (Bradski 2000), a popular 2D image processing library. Wedel *et al.* (2009) further improved the results of the method by adding a median filtering stage between iterations. Recently, Chen *et al.* (2018) presented Split-Bregman algorithm (Goldstein and Osher 2009) to compute 2D optical flow that also involves L_1-norm optimization. Fast convergence and better stability has been reported. Various TV regularization schemes with combination of L_1 and L_2 norms for the data term can be found in Burger *et al.* (2016).

2.1.2 Data term

In our previous discussion, BCA was used to describe the basic skeleton of variational methods for 2D optical flow estimation. We have also mention GCA that can be used as a complementary data term. Let ρ be a penalizer for any assumption used as its argument.

Here are few data terms that have be studied and worth mentioning.

- Brightness Constancy Assumption (BCA) : This is the most widely used assumption in image-based optical flow computation. The data term E_{data} can be written as :

$$E_{data}(\boldsymbol{w}) = \int_{\Omega} \rho(I(\boldsymbol{x}+\boldsymbol{w}, t+1) - I(\boldsymbol{x}, t))d\boldsymbol{x} \tag{2.10}$$

  As mentioned before, preliminary work linearized this assumption using first order Taylor expansion, which made a further assumption of small displacement. However, Nagel and Enkelmann (1986), Alvarez *et al.* (2000), and Mémin *et al.* (1996) refrained from linearizing BCA as it is problematic for large displacements. Keeping up with this trend, Brox *et al.* (2004) used arbitrary downsampling factor for building image pyramid which was not as small as in Horn and Schunck (1981).

  With unlinearized data-term, we rewrite the energy function using BCA penalized by robust norm as :

$$E(\boldsymbol{w}) = \int_{\Omega} \psi_D((I(\boldsymbol{x}+\boldsymbol{w}, t+1) - I(\boldsymbol{x}, t))^2) + \alpha\psi_S(|\nabla \boldsymbol{w}|^2)d\boldsymbol{x}, \tag{2.11}$$

  where $\psi_{D/S}(x^2) = \sqrt{x^2+\epsilon^2}$, a form of Charbonnier norm, and called the robust norm defined in Table 2.1. Let $\psi'_{D/S}$ be its first-order derivative. The associated E.L. equation for the u-component (and similarly v-components) of the flow field without linearization of the data term reads :

$$\psi'_D.(I(\boldsymbol{x}+\boldsymbol{w}, t+1) - I(\boldsymbol{x}, t)).I_x(\boldsymbol{x}+\boldsymbol{w}, t+1) - \alpha div(\psi'_S.\nabla u) = 0, \tag{2.12}$$

where,

$$\psi'_D = \psi'((I(\boldsymbol{x}+\boldsymbol{w}, t+1) - I(\boldsymbol{x}, t))^2),$$
$$\psi'_S = \psi'(|\nabla \boldsymbol{w}|^2). \tag{2.13}$$

As we can see, the Taylor expansion has not been introduced yet. Mémin *et al.* (1996) and Brox *et al.* (2004) suggested iterative linearization of $I(\boldsymbol{x}+\boldsymbol{w}, t+1)$ by Taylor expansion. This leads to :

$$I(\boldsymbol{x}+\boldsymbol{w}^{k+1}, t+1) \approx I(\boldsymbol{x}+\boldsymbol{w}^{k}, t+1) + \nabla I(\boldsymbol{x}+\boldsymbol{w}^{k}, t+1)^T d\boldsymbol{w}^k,$$
$$I_x(\boldsymbol{x}+\boldsymbol{w}^{k+1}, t+1) \approx I_x(\boldsymbol{x}+\boldsymbol{w}^{k}, t+1) + \nabla I_x(\boldsymbol{x}+\boldsymbol{w}^{k}, t+1)^T d\boldsymbol{w}^k, \tag{2.14}$$

where k is the iteration number and $d\boldsymbol{w}^k = \boldsymbol{w}^{k+1} - \boldsymbol{w}^k$. Now $d\boldsymbol{w}^k$ being the unknown, the optimization scheme involves implicit increments in the flow vector and thus theoretically justifies warping. The remaining nonlinearities due to $\psi'_{D/S}$ is handled by additional fixed point iteration.

- Gradient Constancy Assumption (GCA) : This assumption originates from the work of Uras *et al.* (1988) and later adopted by Tistarelli (1996), Brox *et al.* (2004), Papenberg *et al.* (2006) and many others. The data term E_{data} can be written as :

$$E_{data}(\boldsymbol{w}) = \int_{\Omega} \rho(\nabla I(\boldsymbol{x}+\boldsymbol{w}, t+1) - \nabla I(\boldsymbol{x}, t)) d\boldsymbol{x} \tag{2.15}$$

  The calculus to minimize BCA applies in case of GCA as well, for both linear and non-linear cases. GCA is invariant to additive illumination changes and holds true for translational and divergent motion. For rotational invariance, the usage of norm of the gradient was proposed by Papenberg *et al.* (2006). It should be noted that it performs poorly in low textured and noisy regions. Therefore, it is generally used as a complementary term to BCA as in Brox *et al.* (2004). Recently, Qianglong *et al.* (2017) used GCA alone to compute optical flow in context of particle image velocimetry (PIV).

- Higher-order Constancy Assumptions : Higher order derivatives beyond GCA as data terms are possible. Papenberg *et al.* (2006) reports Hessian ($\mathcal{H}_2$), Laplacian and their respective norms as constancy assumptions penalized by L_2 norm. Like GCA, these assumptions hold true for translational and divergent motions, but ineffective to rotation, and sensitive to noise. We list down these assumptions in Table 2.2.

Most of these assumptions are point-wise in nature, and do not hold true in every region of the image. A more local optimization may lead to an easier use of the constancy assumptions. Lucas and Kanade (1981)'s original proposition came at around the same time as Horn and Schunck (1981)'s. They also start off with the linearized constraint defined in Eq. 2.3. However,

TABLE 2.2 – Higher-order assumptions reported by Papenberg *et al.* (2006)

Constancy Assumptions	Data term
Hessian	$\|\mathcal{H}_2(I(\boldsymbol{x}+\boldsymbol{w},t+1)) - \mathcal{H}_2(I(\boldsymbol{x},t))\|_2^2$
Laplacian	$\|\Delta(I(\boldsymbol{x}+\boldsymbol{w},t+1)) - \Delta(I(\boldsymbol{x},t))\|_2^2$
Norm of gradient	$(\|\nabla(I(\boldsymbol{x}+\boldsymbol{w},t+1))\| - \|\nabla(I(\boldsymbol{x},t))\|)^2$
Norm of Hessian	$(\|\mathcal{H}_2(I(\boldsymbol{x}+\boldsymbol{w},t+1))\| - \|\mathcal{H}_2(I(\boldsymbol{x},t))\|)^2$
Determinant of Hessian	$(\det\mathcal{H}_2(I(\boldsymbol{x}+\boldsymbol{w},t+1)) - \det\mathcal{H}_2(I(\boldsymbol{x},t)))^2$

instead of global regularization, they suggest the displacement vector is constant within some neighbourhood of size σ. Then, it becomes possible to determine the components of the vector from weighted least square fit by minimizing :

$$E_{LK}(\boldsymbol{w}) = G_\sigma \star (\boldsymbol{w}^T \nabla I + I_t), \tag{2.16}$$

where $\star$ denotes the convolution operation with G_σ, a Gaussian averaging kernel with a bandwidth σ that defines the influence of the local neighbourhood.

Bruhn *et al.* (2005) combined the local approach of Lucas and Kanade (1981) with the global approach of Horn and Schunck (1981). They reformulate the linearized data term using structure tensor defined by spatio-temporal derivatives. The use of averaging kernel to compute the structure tensor accommodates the local information.

Steinbrucker *et al.* (2009a) showed that linearization is not absolutely necessary while using the framework of Zach *et al.* (2007). This encourages various similarity measures originating from the literature of image registration (Barbara and Jan 2003) and stereo-vision (Delon and Rougé 2007) to be used as the data term such as :

— Sum of Absolute/Squared Difference (SAD/SSD) : This requires a patch around the point in image to be considered. Steinbrucker *et al.* (2009a) used L_1 norm i.e. sum of absolute difference of image intensities between two patches. L_2 norm would lead to sum of squared difference. The data term favours matching of points in image with similar local structure, and consequently becomes more discriminative, but with the implicit assumption of local translation.

— Zero-Normalized Cross Correlation (ZNCC) : It is another patch-based similarity measure that first centers the mean intensity of the patch to 0 and normalizes the patch intensities by local standard deviation. This allows it to be very robust to Gaussian type noise (Drulea and Nedevschi 2013). The related data term is given as :

$$\rho_{ZNCC}(\boldsymbol{x},\boldsymbol{w}) = \sum_{\boldsymbol{P}\in\Omega}\left(1 - \frac{1}{|\mathcal{N}_{\boldsymbol{x}}|}\sum_{p\in\mathcal{N}_{\boldsymbol{x}}}\frac{(I(p+\boldsymbol{w}(p),t+1) - \mu(\boldsymbol{x}+\boldsymbol{w}(\boldsymbol{x}),t+1).(I(p,t) - \mu(\boldsymbol{x},t))}{\sigma(\boldsymbol{x}+\boldsymbol{w}(\boldsymbol{x}),t+1)\sigma(\boldsymbol{x},t)}\right), \tag{2.17}$$

where μ and σ are the mean and the standard deviation computed in patch $\boldsymbol{P}$ around $\boldsymbol{x}$ defined by the neighbourhood $\mathcal{N}_{\boldsymbol{x}}$. Manuel *et al.* (2010) proposed truncated version of the measure by disregarding the negative correlations as :

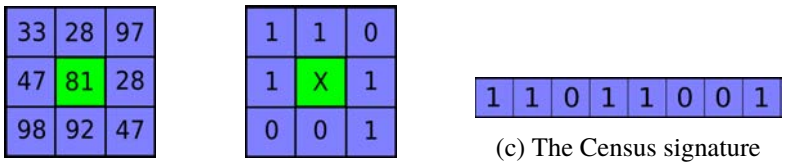
$$\delta(\boldsymbol{x}, \boldsymbol{w}) = min\{1, \rho_{ZNCC}(\boldsymbol{x}, \boldsymbol{w})\} \tag{2.18}$$

They tackled the non-linearity by a direct second-order Taylor expansion on the point-wise data term. One major drawback of this measure is the demanding computation time.

— Census Signature (CS) : Zabih and Woodfill (1994) proposed Census transform in the context of correspondence problem. It is a patch-based feature vector of binary bits arising from directional derivatives. Let H be a function that assigns a bit in a binary string such that :

$$H(r) = \begin{cases} 0, & \text{if } r < 0 \\ 1, & \text{otherwise,} \end{cases}$$

where $r = I(\boldsymbol{x}) - I(\boldsymbol{p})$ and $\boldsymbol{p} \in \mathcal{N}_{\boldsymbol{x}}$ (neighbourhood of $\boldsymbol{x}$). Figure 2.3 illustrates the computation of Census signature of a 2D patch.



(a) A patch around $\boldsymbol{x}$(green)

(b) $H(r)$ in the patch

(c) The Census signature

FIGURE 2.3 – Computing Census signature in a 2D patch.

The Hamming distance between two CS gives the similarity measure. It is given by the summation of bits obtained after binary "OR" operation between the two signatures. Ranftl *et al.* (2012) demonstrated the robustness of CS in challenging lightning conditions in the context of stereo matching. They proposed a scale-invariant descriptor by sampling from stencil of different radii. CS for optical flow was first proposed by Stein (2004). However, it is difficult to linearize, and appears in the original discrete form in the work of Müller *et al.* (2011), Demetz *et al.* (2013), Vogel *et al.* (2013), Hermann and Werner (2014). While these works integrate CS into variational energy, Mohamed and Mertsching (2012) used it to coarsely match points while refining it in variational stage that used the data term given by Bruhn *et al.* (2005). It was Hafner *et al.* (2013) that approximated Census signature in continuous form and successfully linearized it for optical flow estimation.They go on to show that their formulation of CS is a generalized gradient constancy assumption.

2.1.3 Regularization

We reiterate that regularization is necessary to keep the energy formulation from being ill-posed problem. The so-called aperture problem, where one or more components of the flow vector can take any value, is tackled by regularization constraints. Horn and Schunck (1981) imposed smoothness constraint by quadratic penalization on gradient of the flow field. However, it oversmooths the flow field. To prevent such behaviour, various techniques were proposed by Nagel and Enkelmann (1986), Heitz and Bouthemy (1993), Cohen (1993), Schnörr and Sprengel (1994), Black and Anandan (1996), Alvarez *et al.* (1999), and many others to prevent smoothing across flow boundaries. Borrowing from the taxonomy of Weickert *et al.* (2006), we provide a brief overview of different regularization models. Again, let ρ be the penalty function for the regularization term.

2.1.3.1 Image-Driven Approach

The basis for this approach is that flow discontinuities are subset of image discontinuities. As a consequence, we penalize the flow gradient by a weighting function derived from image gradients. Two different ways have been explored. Alvarez *et al.* (1999) simply uses the norm of the image gradient while disregarding the direction of the edges. This is behaves in isotropic way. It is given by :

$$\mathcal{R}(\boldsymbol{w}) = \int_{\Omega} g(|\nabla I|^2)\rho(|\nabla \boldsymbol{w}|^2), \tag{2.19}$$

where g is a positive valued weighting function which assigns pixels not belonging to edges, higher values resulting in smoothing in only those points.

Nagel and Enkelmann (1986) used a tensor-based approach to take direction of the edges into account. A 2D structure tensor $\boldsymbol{S}$ is given as $\boldsymbol{S} := G_\sigma \star \left[\nabla I \nabla I^T\right] := \sum_{i=1}^{2} \lambda_i \boldsymbol{s_i} \boldsymbol{s_i}^T$, where λ_i and $\boldsymbol{s}_i$ are the i^{th} order eigenvalue and eigenvector of $\boldsymbol{S}$. The regularization term for 2D flow field as formulated by Zimmer *et al.* (2011) writes as :

$$\mathcal{R}(\boldsymbol{w}) = \int_{\Omega} \frac{\kappa^2}{tr\boldsymbol{S} + 2\kappa^2}(|\nabla_{\boldsymbol{s}_1} \boldsymbol{w}|^2) + \frac{tr\boldsymbol{S} + \kappa^2}{tr\boldsymbol{S} + 2\kappa^2}(|\nabla_{\boldsymbol{s}_2} \boldsymbol{w}|^2) d\boldsymbol{x}, \tag{2.20}$$

κ is a regularization parameter and ∇_i denotes directional derivative along $\boldsymbol{s}_i$. When $\kappa \to 0$, the smoothing is done along $\boldsymbol{s}_2$ direction, i.e., along the edges only. This gives rise to anisotropic behaviour.

2.1.3.2 Flow-Driven Approach

The obvious drawback of image-driven approach is that they may over-segment the flow field by trying to match flow boundaries in image boundaries. It is natural to use flow gradients

to determine discontinuity in the flow field itself. Schnorr (1994) proposed the robust norm $\psi(.)$ as $\rho(.)$. It was made popular by the seminal work of Brox *et al.* (2004). The behaviour is isotropic. Anisotropic flow-driven regularization is formulated in Weickert and Schnörr (2001) and given as :

$$\mathcal{R}(\boldsymbol{w}) \int_{\Omega} tr(\psi(\nabla \boldsymbol{w} \nabla \boldsymbol{w}^T) d\boldsymbol{x}, \tag{2.21}$$

where $\nabla \boldsymbol{w} \nabla \boldsymbol{w}^T$ produces a 2×2 symmetric positive semi-definite matrix. Hence, there exists two orthogonal eigenvectors which points to the normal and the tangential directions of the flow. The behaviour is similar to anisotropic image-driven approach, but it takes flow direction into account. The drawback of the approach is that flow boundaries are poorly localised.

2.1.3.3 Joint Image- and Flow-Driven approach

To find the remedy to over-segmentation problem in image-driven approach and poor-localization in flow-driven approach, Sun *et al.* (2008) combined both approaches. Direction of smoothing was derived from image structures while the degree of smoothness was obtained from flow gradient. Zimmer *et al.* (2011) improved the joint regularization by reducing smoothing across edges and adding rotation invariance.

2.1.3.4 Non-Local Regularization

Discretization by finite element methods leads to regularization only upto neighbours at unit distance in the grid considered. Non-local approach to regularization proposed by Manuel *et al.* (2010) considers additional regularization that accommodates neighbours at larger distance. The key idea is that pixels belonging to a surface move similarly. They proposed to weight the norm of the difference between flow vectors of the neighbours using the spatial and color information. It is given as :

$$\mathcal{R}(\boldsymbol{w}) = \int_{\Omega} W(\boldsymbol{x}, \tilde{\boldsymbol{x}})(\rho(|\boldsymbol{w}(\boldsymbol{x}) - \boldsymbol{w}(\tilde{\boldsymbol{x}})|) d\boldsymbol{x}, \qquad \tilde{\boldsymbol{x}} \in \mathcal{N}_{\boldsymbol{x}}, \tag{2.22}$$

where $\tilde{\boldsymbol{x}}$ lies in the neighbourhood $\mathcal{N}_{\boldsymbol{x}}$ of $\boldsymbol{x}$, and weight $W(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ is given as :

$$W(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \exp\left(-\frac{|\boldsymbol{x} - \tilde{\boldsymbol{x}}|^2}{2\sigma_d^2} - \frac{|I(\boldsymbol{x}) - I(\tilde{\boldsymbol{x}})|^2}{2\sigma_I^2}\right), \tag{2.23}$$

where σ_d, σ_I are the normalization factors and ρ is the Huber norm (Huber 1964). Sun *et al.* (2010) described the regularization as weighted median filtering. They used neighbourhood of size 15×15 with Charbonnier norm as the penalizer. While being very accurate, the method performs slowly for a moderately sized neighbourhood. Zhang *et al.* (2014) have implemented a version of 2D weighted median filtering, which they claim to be 100+ times faster.

2.2 Matching

Matching points between images in early stage in optical flow problem to deal with large displacements is motivated by the inherent flaws of the coarse-to-fine scheme (Brox and Malik 2011; Revaud *et al.* 2015). In this scheme, the flow field from coarser level is upscaled, and used in warping the target frame to get a new source frame for flow field estimation in next fine scale. Usually bilinear/bicubic (in 2D) and trilinear (in 3D) interpolation is used for warping and upscaling. Because of the down-sampling, smaller objects are lost at coarser levels. Thus, the motion of smaller objects cannot be recovered. Further, the number of levels required to turn the largest displacement into smaller one in the coarsest level is arbitrary. The coarser the image, the more loss of finer details in the flow field. Over the past decade, significant works have been carried out, drawing methods from stereo matching and image registration, to compute large displacements. These matching methods have better flexibility as (i) they do not require differential optimization and thus, linearization, and (ii) more powerful data terms referred to as matching cost between image features are possible. Though, the computed initial flow field contains only integer displacements, may not exhibit global regularity, and is usually sparse or at least semi-dense, the aforementioned flexibility plus the stable behaviour makes them appealing.

2.2.1 Sparse and Semi-dense Matching

The earliest work of Torr (1997) tried to get around the aforementioned flaws by establishing correspondence using normalized cross correlation. Wills *et al.* (2006) computed sparse correspondences between image pair using edge features, and thin-plate spline model to densify them. Liu *et al.* (2008) used SIFT descriptor (Lowe 1999) for robust matching. They formulate the matching problem as a discrete optimization problem. Brox and Malik (2011) penalized the disparity between flow and HOG (Histogram of Gradient)-based matches in variational energy. Barnes *et al.* (2009) developed PatchMatch algorithm that can quickly estimate approximate nearest neighbourhood field.

This benefited the optical flow methods of Lu *et al.* (2013), Bao *et al.* (2014), Hu *et al.* (2016). Lu *et al.* (2013) used superpixels computed using SLIC (Achanta *et al.* 2012) as the basic matching block that respected image boundaries. Hu *et al.* (2016) and Li *et al.* (2018) used a coarse-to-fine approach to PatchMatch algorithm that boosted the performance speed of the method. They match only the subsampled points from a fixed image grid. SIFT descriptor was used. The method produced semi-dense matches after filtering the outliers. Weinzaepfel *et al.* (2013) proposed Deep-matching : a method based on a deep predefined convolutional architecture but without learning weights. It correlates patches at multiple image scales. Each

patch is broken down into a hierarchy of sub-patches, locally matched up to a certain range of motion, and propagated up the hierarchy via maxpooling and aggregation. The downside of the method is that the matches could be sparse in some scenario, which would not provide enough information for subsequent densification. Further, the method is computationally heavy for a moderately sized image pair.

2.2.2 Densification of Sparse Matches

Revaud *et al.* (2015) used Deep matching to compute semi-dense matching and used interpolation based on geodesic-like weights to densify the matches. They apply the Delaunay triangulation around matched points, and compute geodesic-like weights based on spatial and color information for neighbouring pixels in the triangulation. Finally, these weights are used for interpolation using Nadaraya-Watson estimator (Wasserman (2010)). The interpolation is named "EpicFlow interpolation", and has been used as an essential step in the works of Hu *et al.* (2016), Li *et al.* (2018), Bailer *et al.* (2017), Chen *et al.* (2018) and many others. Hu *et al.* (2017) presented a robust interpolation of matches computed with superpixels. Zweig and Wolf (2016) proposed a data-driven sparse-to-dense interpolation based of convolutional network. The method is inspired by filling-in process in the visual cortex. Additionally, variational refinement is added at the end of this pipeline. Revaud *et al.* (2015) used energy similar to Brox *et al.* (2004) as the final refinement, which no longer required coarse-to-fine scheme. Another interesting work wais presented by Fortun *et al.* (2015) which aggregates multiple motion candidates estimated by PatchMatch. The best candidates are selected while imposing a global smoothness without the need of coarse-to-fine variational step.

2.2.3 Combination of Matching and Variational Method

It is also possible to compute a dense flow field by combining the matching with single level variational method. Brox and Malik (2011) included the additional term in their energy function which penalized the discrepancy between the flow vectors and the displacement vectors for the matched points. The optimization problem is given as :

$$E(\boldsymbol{w}) = E_{data}(\boldsymbol{w}) + \alpha E_{smooth}(\boldsymbol{w}) + \beta E_{match}(\boldsymbol{w}, \tilde{\boldsymbol{w}}) + E_{desc}(\tilde{\boldsymbol{w}}), \tag{2.24}$$

where $E_{data}, E_{smooth}, E_{match}, E_{desc}$ are the energy related to data, smoothness, matching displacements, and descriptors, respectively. $\tilde{\boldsymbol{w}}$ is the displacement field obtained from the

sparse matching stage. The last two terms are given as :

$$E_{match}(\boldsymbol{w}, \tilde{\boldsymbol{w}}) = \int_{\Omega} \delta(\boldsymbol{x}) C(\boldsymbol{x}) \rho(\boldsymbol{w}(\boldsymbol{x}) - \tilde{\boldsymbol{w}}(\boldsymbol{x})) d\boldsymbol{x} \tag{2.25}$$

$$E_{desc}(\tilde{\boldsymbol{w}}) = \int_{\Omega} \delta(\boldsymbol{x}) |\boldsymbol{f}_2(\boldsymbol{x} + \tilde{\boldsymbol{w}}(\boldsymbol{x})) - \boldsymbol{f}_1(\boldsymbol{x})|^2 d\boldsymbol{x}, \tag{2.26}$$

where $\boldsymbol{f}_1, \boldsymbol{f}_2$ are the feature vectors in source and target frames, respectively. The function C provides the confidence of matching. The function δ defines the existence of matches on a fixed spatial grid. The minimization is done by solving the usual E.L. equations. The main drawback of the method is that the balance between E_{data} and E_{match} is determined by the confidence measure. Thus, the quality of final flow field is highly dependent on the quality of matches.

Following works of Chambolle (2004) and Zach *et al.* (2007), Steinbrucker *et al.* (2009b) proposed to solve Eq. 2.7 from Section 2.1.1.2 in primal-dual manner, while using matches as an auxiliary flow field. Even though computationally cumbersome, they proposed to perform a complete search for matches in the image. Consequently, no coarse-to-fine scheme is required. Similarly, Chen *et al.* (2018) used Bregman iterations (Goldstein and Osher (2009)) with matches computed by the method of Weinzaepfel *et al.* (2013). They claim their method to be better and faster in public 2D optical flow benchmarking dataset (Baker *et al.* 2011a; Butler *et al.* 2012) than Steinbrucker *et al.* (2009b)'s.

2.3 Convolution Neural Network-based Optical Flow

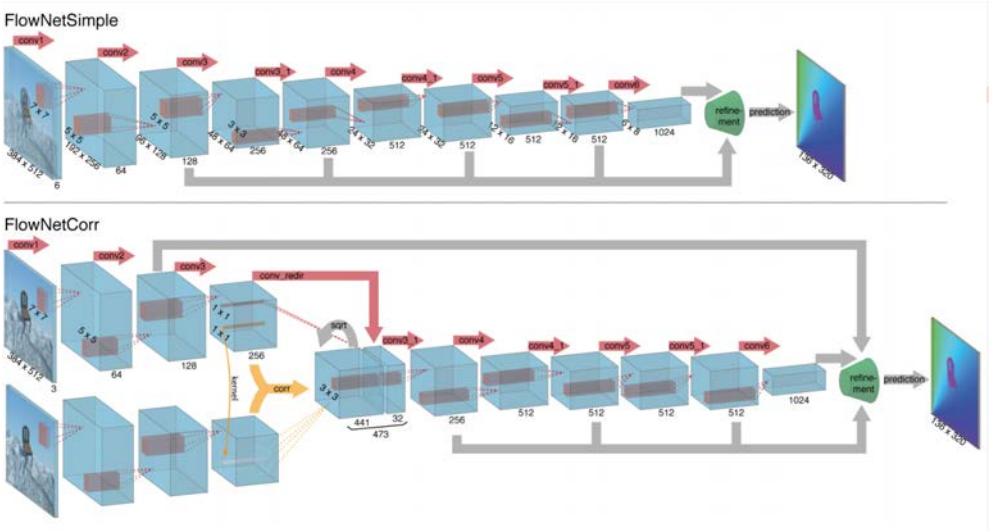


FIGURE 2.4 – FlowNetSimple (top) and FlowNetCorr (bottom), Philipp *et al.* (2015)

This is the newest avenue of 2D optical flow computation, which takes a data driven approach. This type of methods, recently demonstrate the best performance in public optical

flow dataset (Baker *et al.* (2011b), Butler *et al.* (2012)). Compared to variational methods, they are much more flexible in terms of features to exploit as the data term. They can model complex transformation between images while making few assumptions on the input data. They are also faster than the variational counterparts.

Philipp *et al.* (2015) were among the first to propose an end-to-end convolutional neural network (CNN) to learn optical flow in supervised manner. Besides the architecture of the CNN layers, two major components for supervised learning is the loss function of the training and a large training set. They used the Euclidean distance between the estimated and the ground truth 2D velocity components at different scales. For training, they used the synthetic dataset containing chairs exhibiting simple motion of varying scales. Data augmentation (Krizhevsky *et al.* (2012)) that included geometric transformations was included in the training to avoid over-fitting. They proposed two different architectures : (i) FlowNetSimple (ii) FlowNetCorr as seen in fig. 2.4. The first architecture stacks the source and target frame together and learns the optical flow as the weights of convolutional layer that will minimized the distance between learned flow field and the ground truth. The second architecture uses a correlation layer to match the CNN features learned by passing the source and target through identical CNN layers. Because of the maxpooling stage, the correlation layer can localize the best match in a small search window. For refinement, they perform a series of up-convolution and finally predict a flow field which is 4 times smaller in resolution than the input. Ranjan and Black (2016), to deal with long range motion, proposed a CNN network called SpyNet that acts in coarse-to-fine manner. It estimates small displacements using image pairs in coarser levels and upsamples the flow field to finer levels. SpyNet is claimed to be 96% smaller than Flow-net. Ilg *et al.* (2017) presented cascaded network of the previous two architectures leading to the so-called FlowNet 2.0 architecture. They implemented incremental flow update by warping the target image in intermediate stages. By scheduling the training difficulty, they increased the accuracy of estimation while also being able to compute small displacements. For this, their network is first trained on the dataset exhibiting simple 2D motion and no realistic lighting. Eventually, they fine tune the network using the dataset produced by 3D motion in realistic lighting condition. Sun *et al.* (2017) presented a compact CNN architecture that is 17 times smaller and more accurate than Flow-net 2.0, which is called PWC-net. They take inspiration from coarse-to-fine optical flow computation model. They compute a pyramid of CNN features, which is learned during the training. Instead of the intensity image, they warp the CNN features in each pyramid levels. Cost volume layer with small search range is computed between the source and target CNN features. For training loss, they adopted the same multiscale loss as in Philipp *et al.* (2015). The results produced by Flow-net 2.0 seems to preserve flow discontinuity better than that by PWC-net. Hui *et al.* (2018) proposed an architecture called LiteFlow-net, which they

claim is 30 times smaller and 1.36 times faster than FlowNet 2.0. It improves the pyramidal feature extraction layers and adds a flow regularization layer to reject outliers.

CNN approach has been adapted to image matching as well. Zagoruyko and Komodakis (2015) and Jure and Yann (2014) proposed architectures to learn CNN features in image patches for stereo-vision matching. Bailer *et al.* (2017) presented a supervised approach to optical flow computation which is not end-to-end but inspired by CNN feature matching. They implemented siamese network that was put forward by Bromley *et al.* (1993), which takes triplet of patches as input during the training. Among them, two are the matched pair. The network is trained to reduce the L_2 distance between the matched pair while increasing the distance between unmatched pair. Such a loss function, is termed the hinge embedding loss. They train their network to obtain CNN features for each patch in source and target images, and match the features in the paradigm of PatchMatch. They used EpicFlow-interpolation for further refinement. Zhao *et al.* (2017) implemented multi-scale CNN feature learning and matching with additional network to model the cross-scale relationship of the matches.

All these supervised methods excel in performance because of the availability of ground truth and possibility of data augmentation. Towards unsupervised learning of optical flow field, Aria and Ioannis (2016) proposed an end-to-end fully convolutional network. They used the Charbonnier penalty on linearized BCA as the training loss. Similarly, Ren *et al.* (2017) exploited the data term used by Brox *et al.* (2004) as their training loss. Simon *et al.* (2017) found Census loss to perform better than BCA-based loss for the training. Their work suggests that a better loss function modeling could lead to more accurate estimation.

Though, the CNN-based approach outperforms the classical optical flow methods in public benchmark tests, certainly, there are drawbacks to the CNN-based approach to optical flow estimation. The first and foremost problem arises when preparing a suitable and large training data set. The training set usually consists of image pairs rendered using 3D computer graphics software such as Blender (Hess 2010). Such a software can simulate realistic object and camera motion with good textures and realistic lighting conditions. Blender also generates the 2D optical flow which is required to render image frames. The optical flow is used as the ground truth as described in Wulff *et al.* (2012). However, such tool cannot synthesize image sequences of different modalities or dimensionalty. Further, the aforementioned CNN-based methods show good performance in public benchmarks but generalization to complex scene and different image modalities are difficult to achieve. The large number of parameters, which is in order of millions, do not betray the inner workings of the methods. This makes it difficult to study the efficiency and sensitivity of the layers involved in the architecture.

2.4 In the Context of Bio-Imaging

So far, we have discussed about the development of 2D optical flow methods over past decades and its current trend. Though the methods have been usually tested on videos and images, they also find their use in the analysis of different modalities of bio-medical images (MRI, CT, ultrasound, etc.). In context of quantitative bioimaging, optical flow methods can be found in the pipeline of motion analysis (Behbahani *et al.* (2007)), image registration (Maintz and Viergever (1998)), tracking (Duan *et al.* (2005)), and even surface reconstruction (Nan *et al.* (1996)). Thirion (1998) presented a non-rigid matching algorithm called Demons that considered the object under motion as a deformable grid. Using attraction and diffusion models for the particles in the grid, they were able to compute the deformation transformation. Such a transformation can be used to obtain the flow field. However, large displacements cannot be addressed by this method. Delpiano *et al.* (2012) compared the performance of methods by Horn and Schunck (1981) and Bruhn *et al.* (2005) for the 2D motion analysis of confocal microscopy data. The latter method was deemed to perform better. Of course, the off-the-plane motion goes missing in 2D acquisition. This adversely affects the image gradients, and eventually the optical flow estimation.

2.4.1 3D Motion Estimation in Bio-Imaging

Towards 3D optical flow estimation, Baron (2004) implemented 3D version of Lucas and Kanade (1981) and Horn and Schunck (1981) for cardiac motion analysis in gated MRI data. A multi-grid implementation of Horn-and-Shunck in the same context was demonstrated by Mostafa *et al.* (2007). Pock *et al.* (2007), implemented the method of Zach *et al.* (2007) in 3D for registration of clinical CT lung data. Hubený *et al.* (2007) tested three 2D optical flow methods with robust penalizers on 2D synthetic dataset to find a proper regularization parameter. Then, they extended the methods to 3D while fixing the value of regularization obtained from 2D test cases. 3D results were presented qualitatively. Chan and Liebling (2015) took an interesting approach to reconstruct a 3D divergence-free cardiac blood flow in zebra-fish larva from multiple 2D optical flow field of 2D projections. They used radial basis functions instead of a constrained regularization to induce smoothing in the 3D flow field. Validation was limited to simulated fields. Amat *et al.* (2013), similar to the 2D approach of Gkamas and Nikou (2011), proposed an algorithm involving BCA of presegmented super-voxels, and smoothness constraint imposed by Markov Random Fields over the graph of the super-voxels. The method, which we refer to as Amat's method, computes a 3D translation vector per super-voxel by assuming that objects do not undergo complex motion. The method was tested on synthetic dataset exhibiting mostly translational motion, and performs accurately in real examples when

FIGURE 2.5 – The scheme of Amat's method

the objects in the image volume undergo such a simple motion. The resulting flow field is sparse and piecewise constant in the supervoxel segment, which was densified for the purpose of visualization by Kappe *et al.* (2016). We will briefly discuss Amat's method in the following subsection as we will use it as a baseline method.

Tektonidis and Rohr (2017) implemented a SSD-based local optical flow estimation for multiple consecutive frames to determine diffeomorphic transformations in log-domain. They employ weighted median filtering proposed by Sun *et al.* (2010) to preserve flow discontinuity. Boquet-Pujadas *et al.* (2017b) took a data assimilation approach to compute motion during amoeboid cell migration. Following the work of Marion *et al.* (2005), they model the intra-cellular material as a homogeneous, incompressible fluid while ignoring other organelles. The motion of the cell is then governed by Navier-Stokes equations for fluid which is imposed as additional constraints to Horn-and-Schunck's optical flow model. Besides the 3D flow field, the pressure and the forces were measured. The accuracy of the method depends on the modeling of the cell. Though the authors claim it to be generic enough, the model of the cell varies form specimen to specimen. Further, not all specimen can be modeled as a fluid, for e.g., Collagen.

2.4.1.1 Amat's Method

Here we briefly describe Amat's method as we consider it to be a baseline method. The method proceeds in four stages as depicted in fig. 2.5. The method requires a segmentation of the foreground objects, which are supposed to be labeled. The implementation provided by the authors does not perform the segmentation. The segmentation maps have to be provided as the inputs along with the images. They use 3D SLIC method provided by Achanta *et al.* (2012) for super-voxel generation. The use of super-voxel facilitates the use of local intensity information and simpler translation motion model like in Lucas and Kanade (1981), while exploiting irregular boundaries instead of rectangular/cuboid. Let $\Re$ be the set of suer-voxels and S be a super-voxel containing a voxel $\boldsymbol{x}$. The method calculated a single translation $\boldsymbol{w}_S$ for each S. The resulting data term in its discrete form is given as :

$$E_{data} = \sum_{S \in \Re} \sum_{\boldsymbol{x} \in S} \rho_D(I_t(\boldsymbol{x}) - I_{t+1}(\boldsymbol{x} + \boldsymbol{w}_S)). \tag{2.27}$$

Besides, helping in regularization of flow field in textureless regions, the super-voxel based data term reduces the dimensionality of the search makes the method run faster.

The next step is to generate a Markov random field between the super-voxels. Considering each super-voxels as a node, an edge is defined between two super-voxels if their centers of mass are below a distance threshold d_{max}. Finally, we can write the regularization term as :

$$E_{smooth} = \alpha \sum_{S \in \Re} \sum_{R \in \mathcal{N}_S} W_{R,S} \rho_C(\boldsymbol{w}(R) - \boldsymbol{w}(S)), \tag{2.28}$$

where, α is the usual balancing parameter, R is a supervoxel that belongs to the neighbourhood $\mathcal{N}_S$ of S and $W_{R,S}$ gives the weight as :

$$W_{R,S} = \exp\left\{ -0.5 \left(\frac{d_{R,S}}{d_{max}} \right)^2 \right\} \frac{vol(R) + vol(S)}{2max_{A \in \Re}(vol(A))}, \tag{2.29}$$

with $d_{R,S}$ being the distance between the centroids of R and S, and $vol(A)$ is the volume of any super-voxel A. The weights decrease the interaction between far-apart and/or smaller super-voxels. The weights along with the super-voxel segmentations imposes the smoothness constraint only on the objects possibly having common dynamics. However, the regularization depends on $\mathcal{N}_S$ which is difficult to determine empirically, given the large variation in cellular structures and dynamics.

The authors combine the Eqs. 2.27 and 2.28, and solve its minimizer using LBFGS method provided by Byrd *et al.* (1995). ρ_D and ρ_C are both defined as Huber cost function (Huber 1964). The experiments were done in the dataset of Drosophilia and zebrafish embryos (see fig. 2.6) which consisted of many cells suspended in a volume and apparently moving with small translation between consecutive time steps. These cells could be properly segmented using supervoxels and put in a MRF setting. Given the nature of the data, the method performs well in determining the translational motion of the cells.

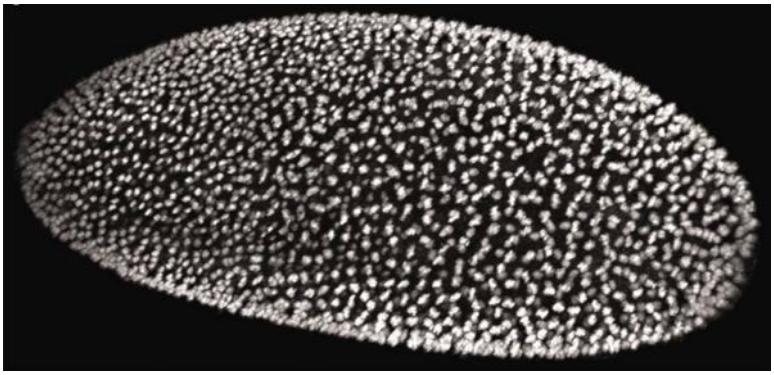


FIGURE 2.6 – 3D volume of Drosophila embryo. From Amat *et al.* (2013).

2.5 Remarks

Most of the aforementioned methods in Bio-Imaging make use of the classical BCA from Lucas and Kanade (1981) and Horn and Schunck (1981). While image registration methods have been benefited by using more elaborate data terms (Sotiras *et al.* (2013)), the optical flow methods in 3D bio-medical images still rely on the classical assumption. We also stress on the lack of availability of their implementation. Most of the methods are not available publicly. The demonstrations are made for specific image modality and specimen. Lack of public benchmarking dataset also adds difficulty to comment on the robustness of the methods. To this end, we are motivated to extend 2D optical methods to 3D, so that a comparative study can be carried out.

CHAPITRE 3

VISUALIZATION AND ASSESSMENT

3.1 Data Description

In this section, we present the data that we have used for describing our work in this thesis manuscript. We deal with the dynamics of two specimen : the MV3 cell and the collagen network.

3.1.1 MV3 cell

3D sequences of three specimen of MV3 cells have been acquired, of which two exhibit blebbing in a coverslip (no collagen environment). The last one demonstrates a large scale retraction of its filopodium to migrate in a collagen environment. We will refer to the blebbing MV3 sequences to as MV3 A and MV3 B, and the last one as MV3 C as depicted (only a single volume) in Figs. 3.1a, 3.1b and 3.2a, respectively.

The MV3 A and MV3 B were labelled with Tractin-GFP. It binds with actin filament (F-actin) cytoskeleton of the cell, which lies close to the cell surface. Thus, we are able to view the topology of the cell where F-actin is present. Volumetric images, each of size $50 \times 50 \times 30 \mu m^3 (300 \times 300 \times 83)$, were acquired with DiaSLM (Dean *et al.* 2016) at the rate of 2.86 Hz. 1000 time points were acquired. We make use of first few time points where we find motion of the bleb(s) to be easy to interpret.

MV3 C contains ezrin-labelled MV3 cell. Ezrin is a protein that links plasma membrane of the cell with the actin cytoskeleton. ASLM was employed to acquire the volumetric images at the rate of 0.04 Hz. Each volume is of size $80 \times 80 \times 40 \mu m^3 (512 \times 512 \times 101)$. The sequence

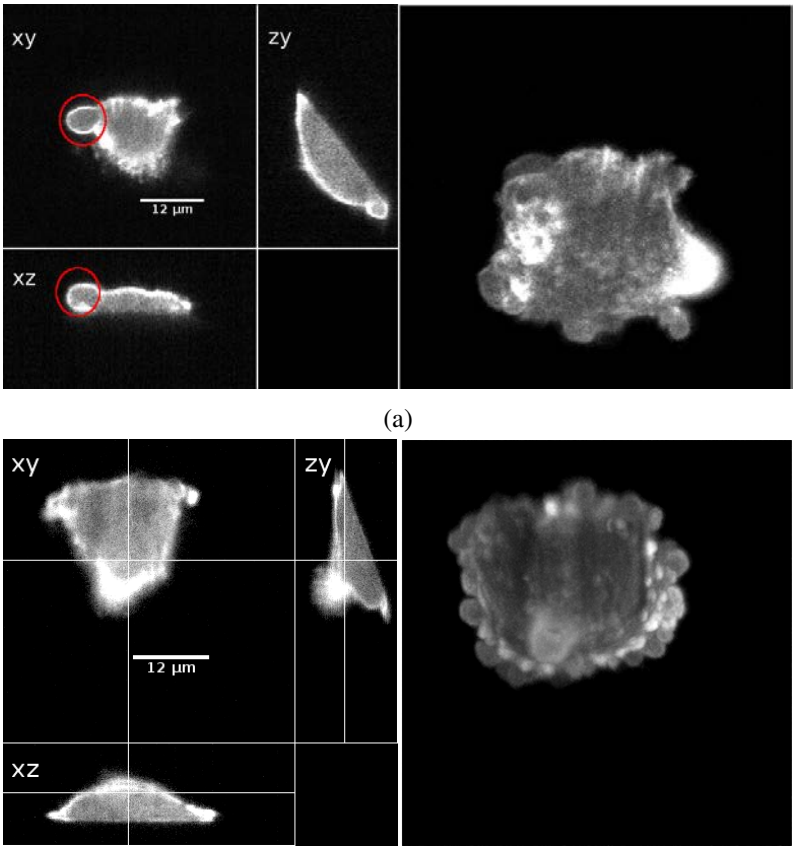
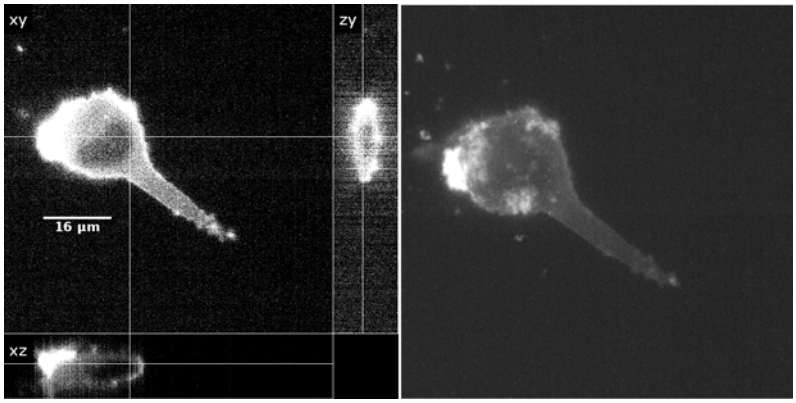


FIGURE 3.1 – Orthogonal slices (on left) and Maximum intensity projection (MIP, on right) in XY-plane of a volume in the sequence (a) MV3 A (b) MV3 B. A bleb in the cell surface of MV3 A is circled in red. Several small sphere-like protrusions in the cell surface of MV3 B are the blebs.

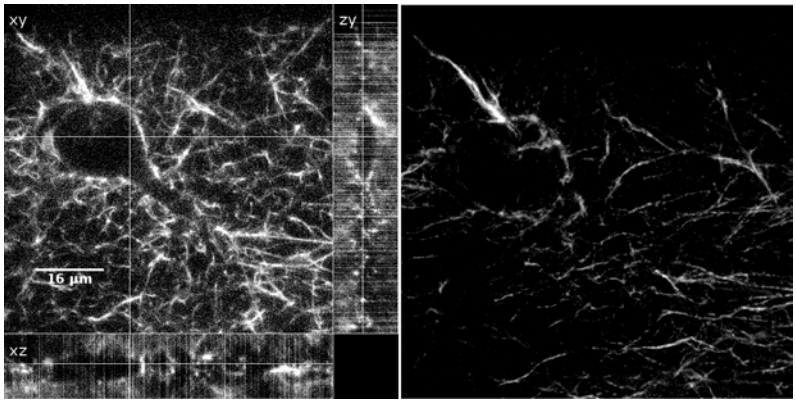
starts with a filopodium latched to a collagen network. The filopodia eventually unlatches, resulting in a motion in the direction of retraction that betrays a slingshot-like behaviour. We only make use of a subset of the sequence where interesting motion lies.

3.1.2 Collagen network

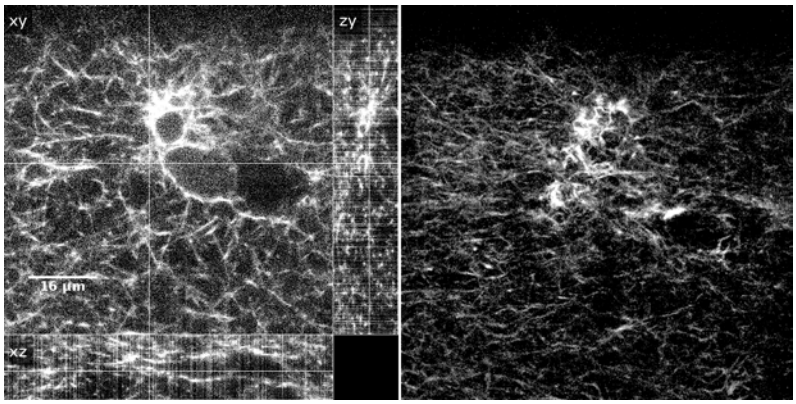
The 3D sequences of collagen network has been acquired in the context of MV3 cell migration. We perform experiments with two acquired volume sequences of collagen network. We name them Collagen 1 and Collagen 2, which are depicted in Figs. 3.2b and 3.2c, respectively. Collagen 1 embeds the MV3 cell of the sequence MV3 C. The embedded cell for Collagen 2 has not been described in this manuscript, as we do not find any interesting motion within. The collagen networks were labelled with DyLight 521. The acquisitions were made with ASLM at the rate of 0.04 Hz. Each volume is of size $80 \times 80 \times 40 \mu m^3 (512 \times 512 \times 101)$.



(a)



(b)



(c)

FIGURE 3.2 – Orthogonal slices (on left) and Maximum intensity projection (MIP, on right) in XY-plane of a volume in the sequence (a) MV3 C (b) Collagen 1 (c) Collagen 2.

3.2 3D Flow Field Visualization

Scientific visualization is a complementary yet essential component to the design, implementation and mathematics behind algorithms. It comes in as a common tool to not only assess the end results but in feedback loop during the development of the algorithm to check the

sensitivity of various parameters within the algorithm. It is a crucial part of research for the communities studying fluid dynamics, computer graphics, computer vision, climate modeling, to name a few. Of course, the components to visualize are not the same in all communities. Some might be interested in scalar fields like uncertainty, eigenvalues, image labels, temperature, humidity, etc. and some in vector fields like motion map, eigenvectors, vorticity, surface normals etc. Likewise, tensor fields. Time series data of these fields requires more sophisticated methods to visualize. Take the case of computational fluid dynamics (CFD) for example, the researchers in this field are less interested in instantaneous flow field or flow between two frames and prefer the statistics of flow field collected over large time interval. The reason being, the behaviour of the fluid in one time step cannot be exactly reproduced but can be characterized over large sample. So they use streak-lines, streamlines, path-lines and even time series map of scalar components of vectors. In our case, we mostly work with a pair of volumes. So we do not rely on the time series visualization methods.

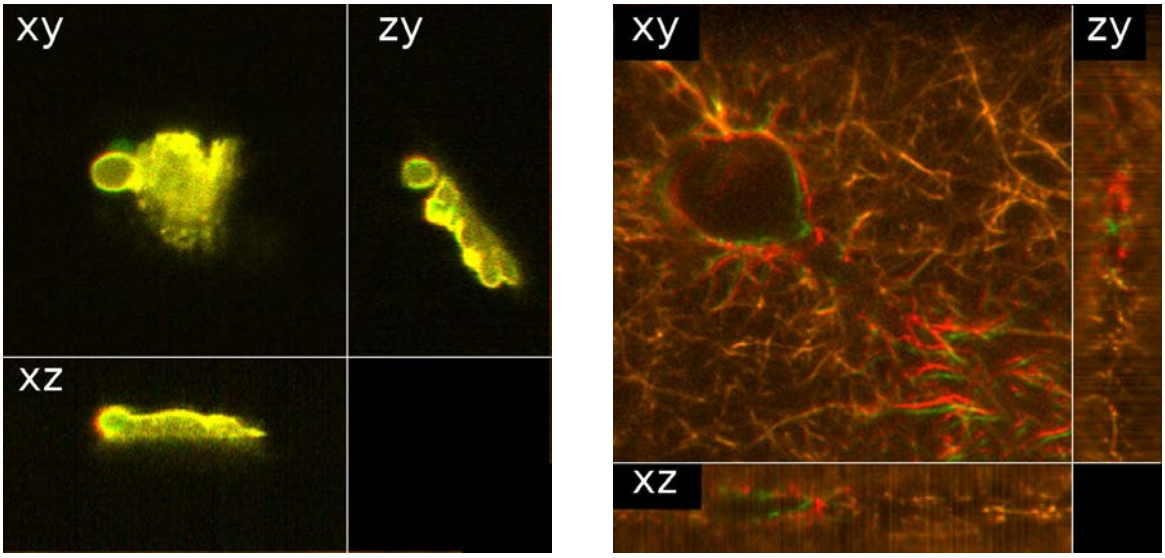
3.2.1 Preliminaries

Let I_1 and I_2 be the source and the target volumes of size $M \times N \times P$, respectively. U, V, W are the matrices of same size as I_1, which contain the scalar components u, v, w along X, Y, Z axes of the 3D flow field, respectively. The matrices and the image data are defined in the same coordinate system. In what follows, we illustrate the main ideas using the motion computed in the volume pairs of MV3 A and Collagen 1 shown in Fig. 3.3. The 3D motion is computed using Amat *et al.* (2013)'s optical flow method, which we will be referring to as Amat's method.

3.2.2 Requirements for good visualization

In the development of our thesis, we drive ourselves to design a proper visualization of 3D motion field for fluorescence microscopy volume sequence. For the sake of qualitative assessment, we have to rely on some form of visualization. Further, a good visualization method would be interesting to biologists and microscopists, who are concerned with the dynamic behaviour of specimen under study. The usual way to display motion field would be to use arrows/glyphs. But in 3D, this technique, while requiring interactive software package, in many aspects, falls short in providing enough information. Another approach derived from the community of fluid dynamics, would be to render three separate heat maps of scalar components of the 3D flow field. But it remains to the viewer to make sense of the direction and magnitude of the motion. So we set forth to design appropriate visualization methods while keeping following points in mind :

— **Clarity** : A common visualization of flow field is the usage of arrows to convey the



(a) a volume pair in MV3 A

(b) a volume pair in Collagen 1.

FIGURE 3.3 – Overlay of slices of I_1 (in red) and I_2 (in green) of the sequence MV3 B (left) and Collagen 1 (right)

direction and magnitude of motion vectors. This approach is effective when sparse view of the motion field is not an issue. Conversely, it lacks in clarity when motion field is dense as vectors overlap with each other. Often, sub-sampling is done to prevent overlapping. Further, smaller vectors become unintelligible when both large and smaller vectors are being displayed together. Similarly, when the motion magnitude of the vector field is relatively smaller than the scale of the moving object it self, the global view of the flow field might not be clear for the viewer. In that case, upscaling of the magnitude might be required. Discontinuities and smoothness become less obvious (see Figs. 3.4 and 3.9c).

— **Coherence** : The coherence of motion map and moving structures is of importance when local behaviour at various points in the scene is to be observed. For example, motion in a collagen network during cell migration. The usage of arrows (see Fig. 3.5) here fails to create a coherent visualization.

— **Compactness** : A vector field can be separated to scalar components. It is a common practice in the fluid dynamics community to visualize scalar orthogonal components of a flow field. In case of 3D, a slice from the volume is chosen and its scalar components are mapped into three different renderings as in Fig. 3.6. It remains up to the viewer to figure out the direction of the motion.

— **Commensurability** : A visualization should allow to make comparisons between flow fields. Different motion estimation algorithms may produce different motion fields

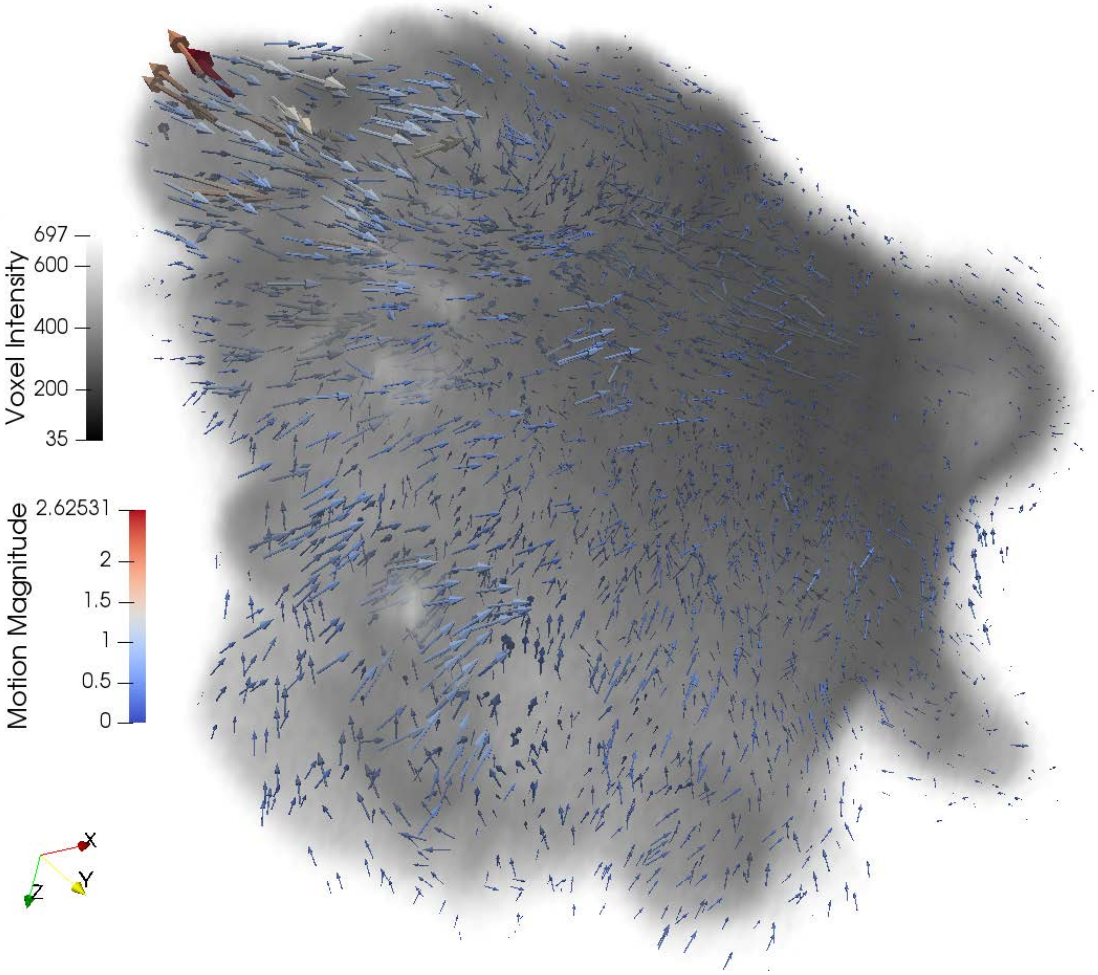


FIGURE 3.4 – 3D flow field computed for a volume pair in MV3 A. The motion magnitude is given by the color code on the right. The direction of the motion is shown by the arrows. The MV3 cell is depicted as the translucent volume from where the arrows are casted out. The length of arrows shows amplitude of the motion, but has been scaled by the factor of 10 for the sake of perception.

for the same sequence. The difference could be subtle or drastic. Also, for the same algorithm, the parameters involved may produce different flow fields upon tuning. Having a visualization method that allows comparison of different flow fields helps in the research and development of optical flow methods.

3.2.3 Middlebury Style Color Coding

We take inspiration from visualization of 2D optical flow field. The 2D flow field can be compactly visualized using the Middlebury style color coding (Baker *et al.* (2011b)). The color code allows the direction and magnitude of 2D vectors to be mapped to hue and saturation

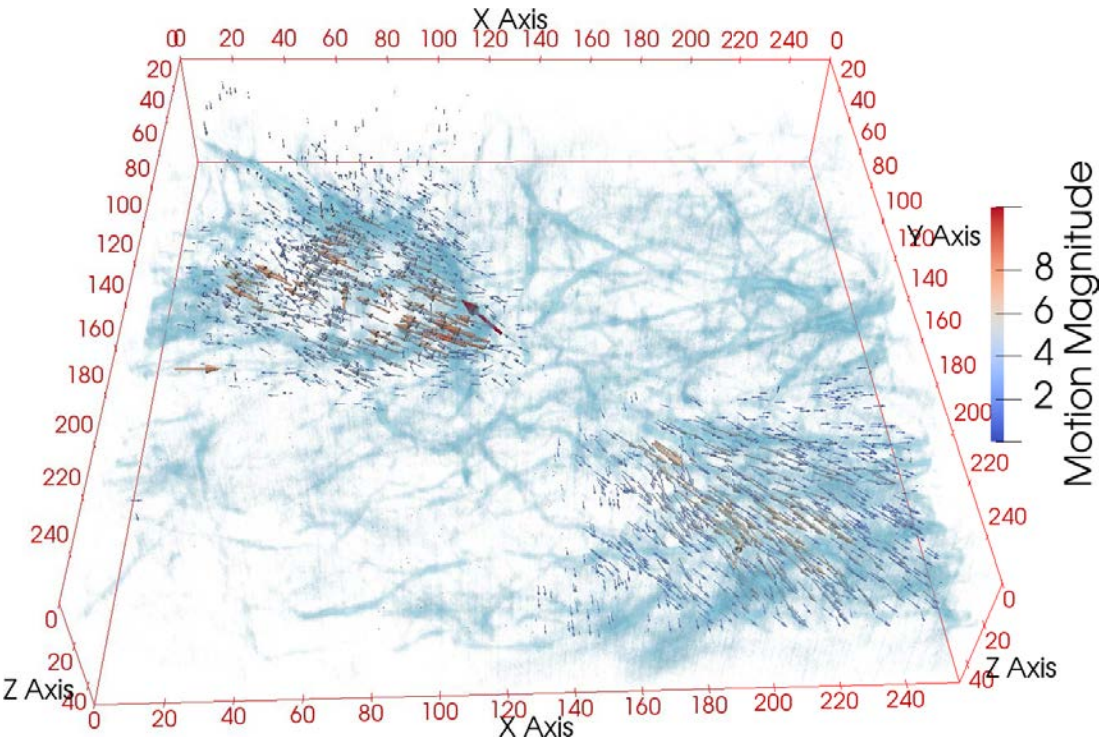


FIGURE 3.5 – 3D flow field computed for a volume pair in Collagen 1 depicted by 3D glyphs. The location for arrows are sampled from a uniform spatial distribution. Only 10000 3D vectors are being rendered.

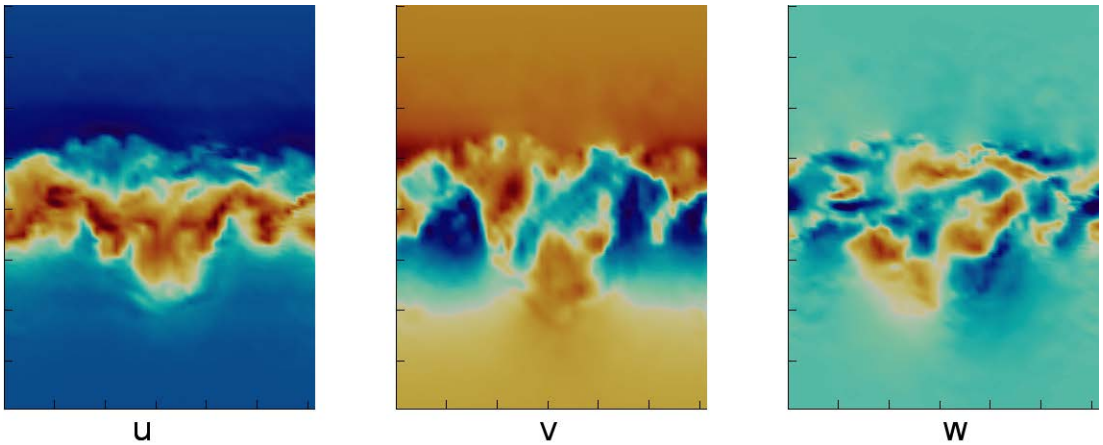


FIGURE 3.6 – Scalar components of a 3D flow field of a fluid (Chandramouli *et al.* 2018).

space. For a normalized system, hue and saturation take values between $[0°, 360°]$ and $[0, 1]$, respectively. Let u and v be two scalar components of a 2D vector in a plane. Then hue is obtained by mapping the arctangent of u and v between $[0°, 360°]$ and saturation as $\sqrt{u^2 + v^2}$, normalized by some value m. m can be the maximum motion magnitude of the plane or user selected value to adjust the saturation level of the visualization. Figure 3.8 shows the color mapping. As an

example, we show a 2D optical flow field computed between a pair of 2D images and visualized using this style in Fig. 3.9d. It clearly shows the smoothness and discontinuities of the flow field which are few key components when comparing different flow fields.

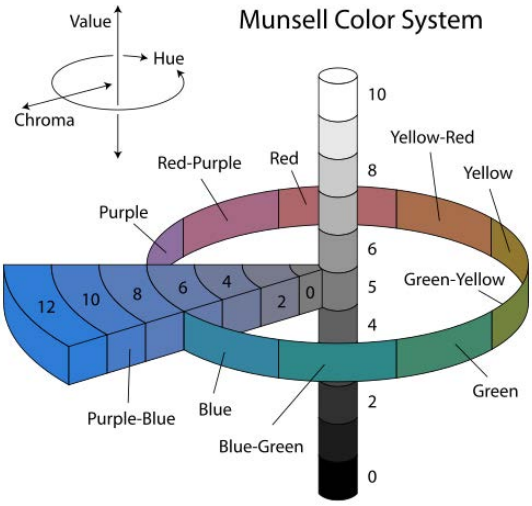


FIGURE 3.7 – Munsell color system which describes a color using Hue, Saturation (Chroma) and Value spaces. The image is provided by Jacob Rus licensed under CC BY-SA 3.0 license (url : https ://creativecommons.org/licenses/by-sa/3.0/deed.en).

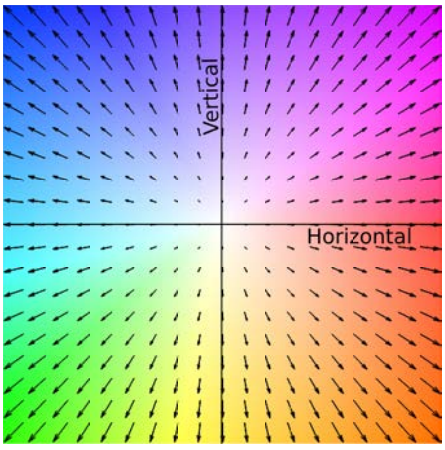


FIGURE 3.8 – Middlebury style color coding for 2D motion field

3.2.4 3D Flow Field Visualization : No Free Lunch

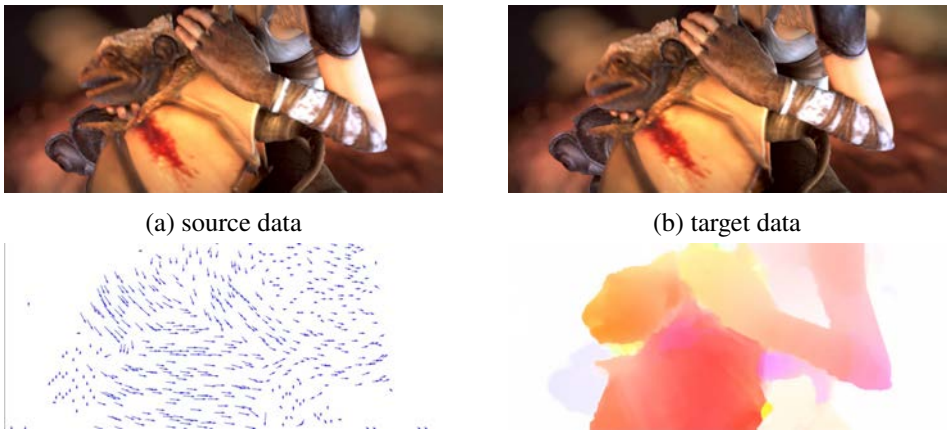
Color-coded flow map is now a common practice for viewing 2D optical flow fields (Baker *et al.* 2011a). Figure 3.8 depicts one of the popular color codes for 2D flow fields. Color mapping

of 2D vector field was described in the work of Abramoff *et al.* (2000), where they proposed to map the Hue subspace to the direction and the Saturation subspace to the magnitude of the vector, while keeping the Value of the map to 1. However, creating a 3D color map for 3D flow fields, especially for those pertaining to the dynamics of 3D fluorescence image sequence, it is not trivial for two reasons :

— Even when the Value subspace is used for the 3D vector representation, there are not enough colors discernible to human eyes to accommodate 3D vectors of all orientations and amplitudes. If we go back to the original HSV coding of colors (see Fig. 3.7), put forward by Albert H. Munsell in 1905, we can see that the color code of Abramoff *et al.* (2000) for 2D flow field is just a single slice of HSV color system. Apparently, the manipulation of value as the third dimension could help in direct extension of 3D color code. However, Munsell discovered that all possible colors defined by perceptual range of the three components does not form a regular 3D surface. This means, not all possible 3D vectors can be mapped to HSV space.

— Viewing 3D flow field for fluorescence data in 2D planes has to be done smartly as not all portions of the flow field are informative.

One of our contributions lies in resolving these issues by introducing three different techniques that makes color mapping of 3D flow fields possible.



(a) source data (b) target data

(c) arrow-based visualization (d) Middlebury color map

FIGURE 3.9 – 2D optical flow for a pair of images (from Revaud *et al.* (2015))

3.2.5 Mapping 3D motion field to HSV (3DHSV)

The first method takes slice-by-slice approach which we call 3DHSV mapping. We map the direction of the motion to Hue, the magnitude to Saturation and the off-the-plane motion to Value subspaces. We use only u and v components to map the direction to hue. The magnitude

is computed using all three components. The value-components are given by the w-component of the flow field such that, voxels in the slice with downward and upward trajectories in Z-axis are brighter and darker, respectively. The color code is depicted as HSV cylinder in Fig. 3.10.

Figure 3.11 shows overlay of slices of I_1 (in red) and I_2 (in green) of MV3 B. It can be noted that the bleb (the circular structure on the top-left side) moves towards the bottom right side of the image plane, however with small displacement. Figure 3.12 shows the 3DHSV map of the motion field of the sequence. The motion of the bleb in the mentioned direction is apparent. The patchiness of the flow field across the plane can be noticed. The stationary region is rendered with gray color. Off-the-plane motion is rendered with different values which appear as darker or brighter pixels. Some scaling in value space might be required to make the off-the-plane motion more discernible to the user.

FIGURE 3.10 – 3DHSV color code

3.2.6 Mapping 3D motion field from three orthogonal planes (3PHS)

Our second proposition is to simply slice the volume in a desired plane and project the 3D vector corresponding to the intersected voxel to the plane. Projection of a 3D vector $\boldsymbol{\omega}$ onto a 2D plane defined by the normal vector $\hat{n}$ such that $|\hat{n}| = 1$ is given as $\boldsymbol{Proj}_{\hat{n}}(\boldsymbol{\omega})$ in the Eq. 3.1. The resulting 2D vector field can then be mapped using the Middlebury color code 3.8. We propose to use three orthogonal planes, i.e., $XY_{\hat{n}=[0,0,1]^T}$, $XZ_{\hat{n}=[1,0,0]^T}$ and $ZY_{\hat{n}=[0,1,0]^T}$ planes, mainly because these three planes are the most commonly viewed ones.

$$\boldsymbol{Proj}_{\hat{n}}(\boldsymbol{\omega}) = \boldsymbol{\omega} - (\boldsymbol{\omega}.\hat{n})\hat{n} \tag{3.1}$$

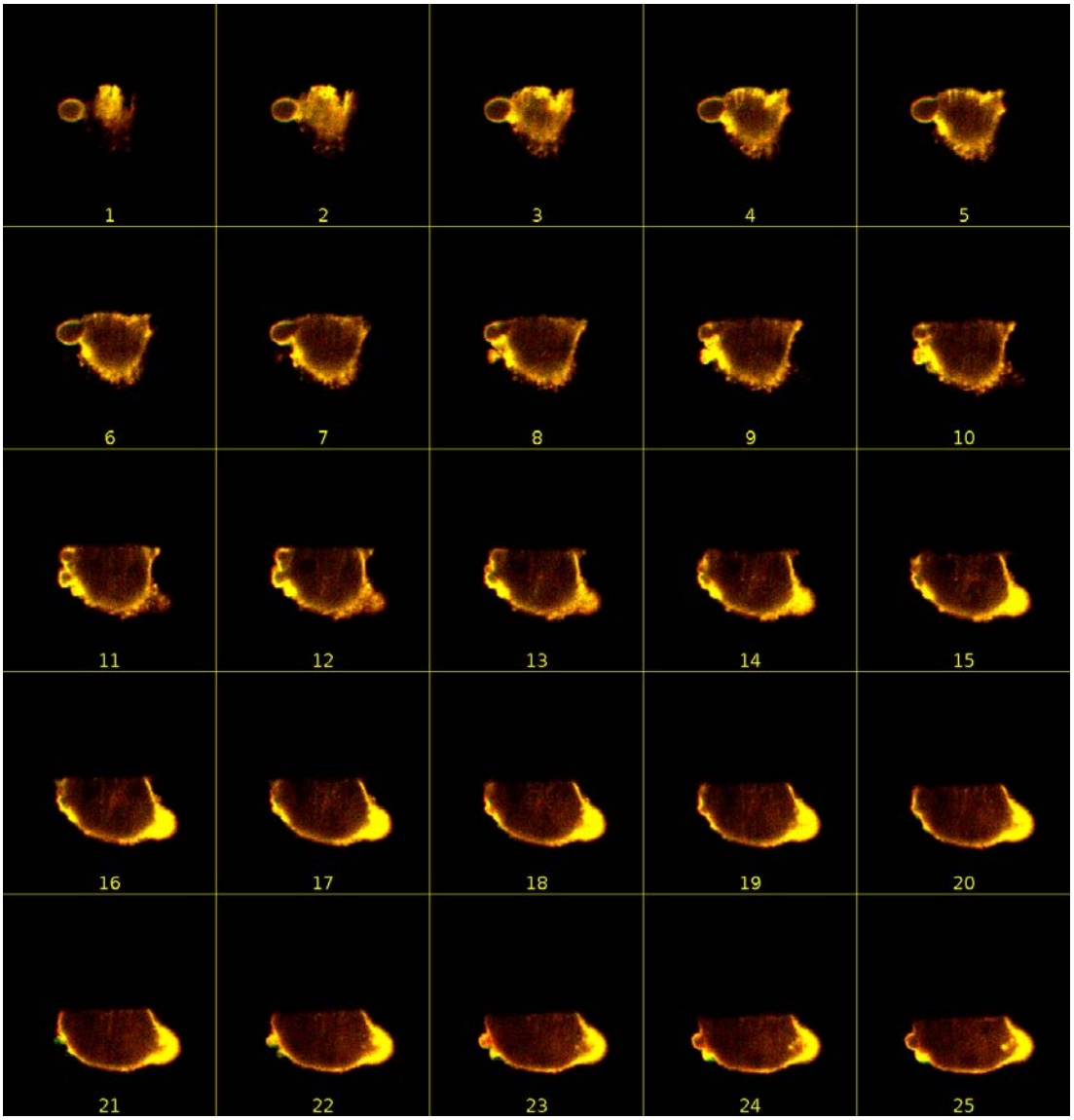


FIGURE 3.11 – Slice-by-slice overlay of I_1 (in red) and I_2 (in green) in MV3 A. The slice number is labelled in the respective subfigures.

For each voxel located at $\boldsymbol{x}$, we project the 3D vector $(u(\boldsymbol{x}), v(\boldsymbol{x}), w(\boldsymbol{x}))$ onto XY-plane using Eq. 3.1.

$$\boldsymbol{Proj}_{[0,0,1]^T}\left(\begin{bmatrix} u(\boldsymbol{x}) \\ v(\boldsymbol{x}) \\ w(\boldsymbol{x}) \end{bmatrix}\right) = \begin{bmatrix} u(\boldsymbol{x}) \\ v(\boldsymbol{x}) \\ w(\boldsymbol{x}) \end{bmatrix} - \begin{bmatrix} u(\boldsymbol{x}) & v(\boldsymbol{x}) & w(\boldsymbol{x}) \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{3.2}$$

$$\boldsymbol{Proj}_{[0,0,1]^T}\left(\begin{bmatrix} u(\boldsymbol{x}) \\ v(\boldsymbol{x}) \\ w(\boldsymbol{x}) \end{bmatrix}\right) = \begin{bmatrix} u(\boldsymbol{x}) \\ v(\boldsymbol{x}) \\ 0 \end{bmatrix} \tag{3.3}$$

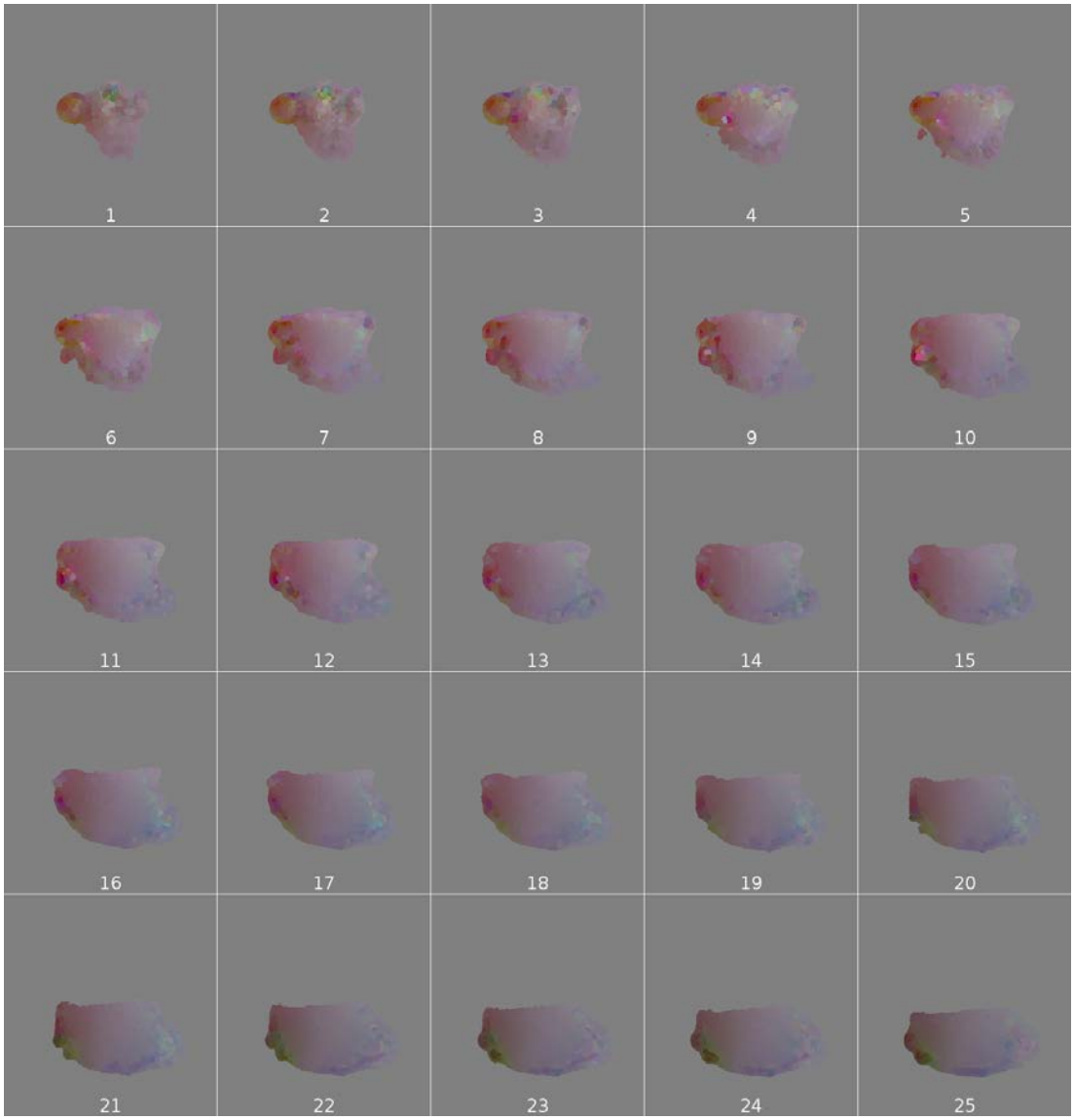


FIGURE 3.12 – 3DHSV map of the 3D flow field computed using Amat's method for the volume pair in MV3 A. The slice number is labelled in the respective cell. Refer to 3.10 for the color coding scheme.

Of course, the user has to select a point in the 3D volume where these three planes intersect. We then map the hue and the saturation components of the color code for each planes separately. The arc tangent of the horizontal and the vertical components provide the angle to be mapped to hue which ranges from $0 - 360°$. The saturation level is given by normalizing the magnitude of the 2D vector between $0 - 1$.

Figure 3.14 shows the 3PHS map of the flow fields computed using Amat's method for a pair of volumes in MV3 A. The effects of different values of the parameters of the method on the result are clearly visualized.

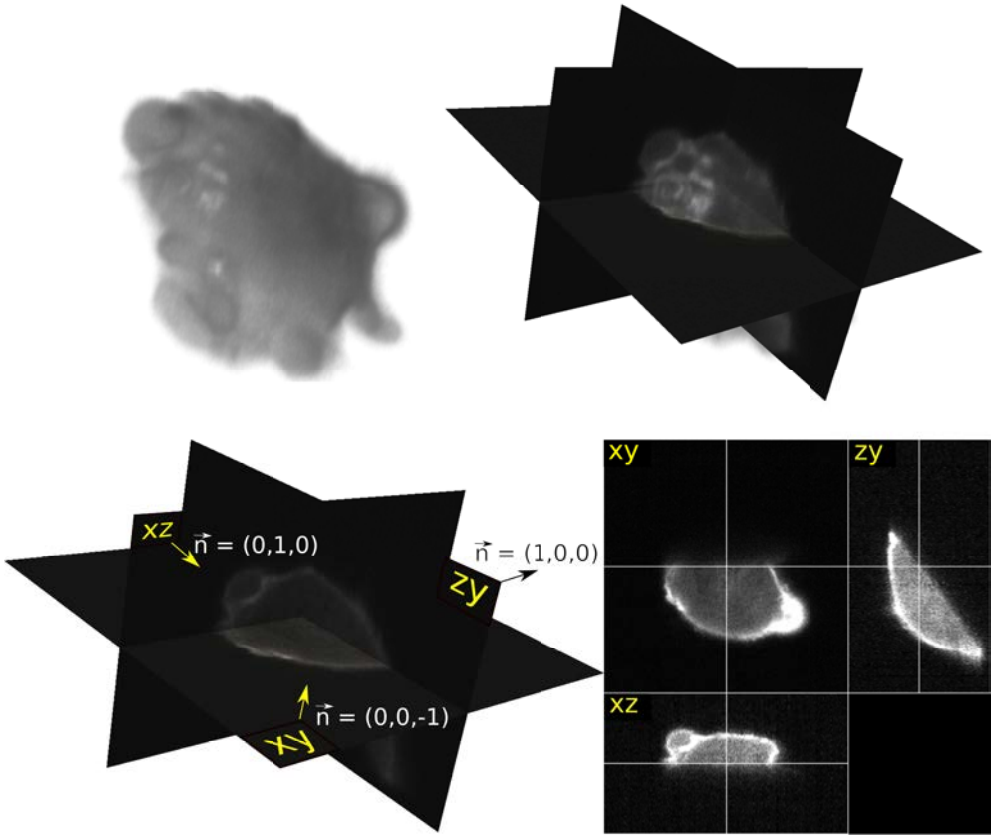


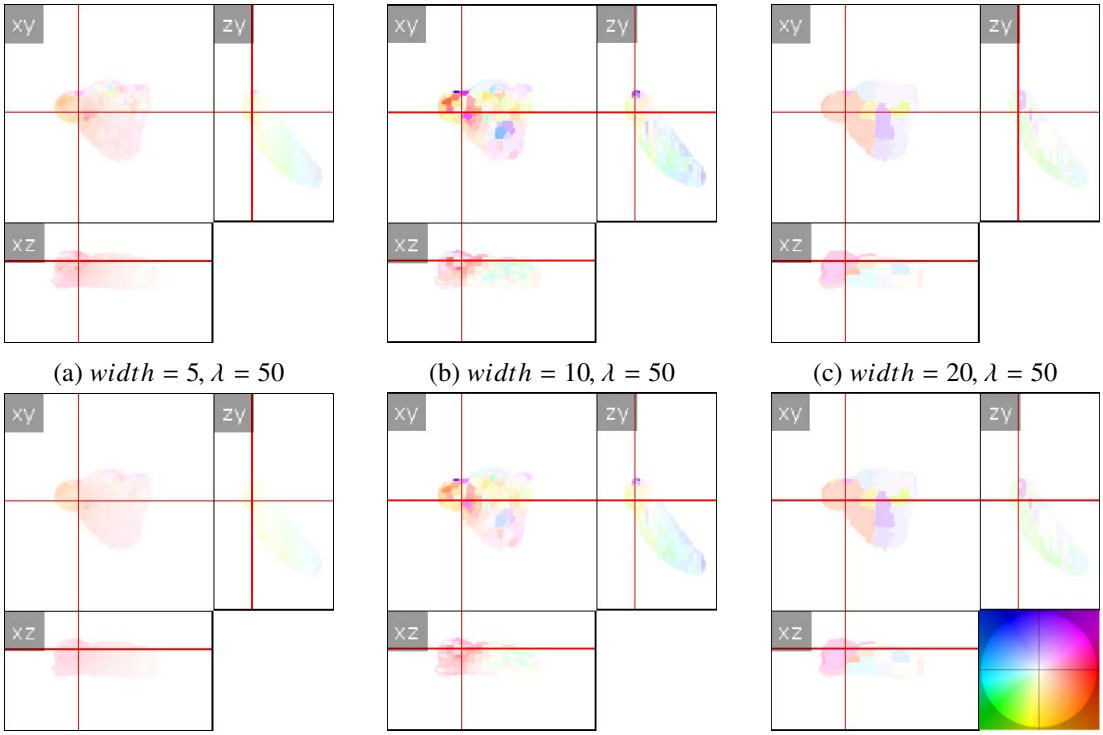
FIGURE 3.13 – Slicing the volumetric data in MV3 A using three orthogonal planes.

3.2.7 Mapping 3D motion field to Maximum Intensity Points

Maximum intensity projection (MIP) imaging Wallis *et al.* (1989) is one of the convenient ways to visualize 3D volumetric image data in a 2D plane where only the voxel with maximum intensity in the direction perpendicular to the plane is mapped to the plane (see Fig. 3.15). This technique enhances the points in the volumes where fluorophores are concentrated while removing the weakly stained or noisy points to form a clear view of the 3D scene in a 2D form. Structures which manifest themselves in more than one planes are difficult to discern in the slice-by-slice approach. MIP images reveals such structures in one single plane as seen in Fig. 3.16. Multiple MIPs might be required to form an understanding of a 3D scene.

Based on this, we now build a 3D flow field visualization technique, which we call MIP-Flow map. It leverages the representative capability of MIP images. For this, we will project the 3D flow field onto a 2D plane similar to 3PHS map. However, we will select only the voxels that have been accounted for in MIP image, i.e. voxels with maximum intensity along the normal of the projection plane.

For example, we consider XY-plane as the projection plane. To compute MIP, for each pair



(a) $width = 5, \lambda = 50$ (b) $width = 10, \lambda = 50$ (c) $width = 20, \lambda = 50$

(d) $width = 5, \lambda = 200$ (e) $width = 10, \lambda = 200$ (f) $width = 20, \lambda = 200$

FIGURE 3.14 – 3PHS rendering of 3D flow field for a volume pair in MV3 A computed by Amat's method. Width refers to the supervoxel width ; λ refers to the smoothness parameter. The block-like artifact becomes more apparent with larger width or lower values of λ. The color code is given on the bottom-right side of (f).

(x, y) such that $x \in X$ and $y \in Y$, along Z-axis, we obtain the voxels with maximum intensities. Equation 3.4 presents its formulation.

$$\tilde{I}(x, y) = \max_{z=1,\ldots,P} I(x, y, z), \tag{3.4}$$

where $\tilde{I}$ is the MIP image. Let $\mathcal{M}$ be the set of location of voxels with maximum intensity along their respective line of projection. For each voxel located at $\{\boldsymbol{x}_m : \boldsymbol{x}_m \in \mathcal{M}\}$, we project the 3D vector $(u(\boldsymbol{x}_m), v(\boldsymbol{x}_m), w(\boldsymbol{x}_m))$ onto the XY-plane using Eq. 3.1. The projected 2D vector is simply $(u(\boldsymbol{x}_m), v(\boldsymbol{x}_m))$ as calculated in Eq 3.5.

Figure 3.17 shows the MIP images in XY, XZ and ZY planes, and the respective MIP-flow maps of the flow field between a volume pair in MV3 A computed using Amat's method. The patchiness of the flow field is apparent. Motion in unlabelled region is significantly visible. Motion of the blebs which are mostly formed at the left hand side of the XY-plane can be appreciated as well. Figure 3.18 shows the 3PHS (left) and MIP-flow (mid) maps of flow field computed in Collagen 1. The latter tends to better summarize the general motion of the whole

scene. MIP-flow map can be further overlaid on top of the MIP of original source volume as depicted in 3.18 (right) to highlight the motion of structures.

$$
\begin{aligned}
\boldsymbol{Proj}_{[0,0,1]^T}\left(\begin{bmatrix} u(\boldsymbol{x}_m) \\ v(\boldsymbol{x}_m) \\ w(\boldsymbol{x}_m) \end{bmatrix}\right) &= \begin{bmatrix} u(\boldsymbol{x}_m) \\ v(\boldsymbol{x}_m) \\ w(\boldsymbol{x}_m) \end{bmatrix} - \begin{bmatrix} u(\boldsymbol{x}_m) & v(\boldsymbol{x}_m) & w(\boldsymbol{x}_m) \end{bmatrix} . \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
\boldsymbol{Proj}_{[0,0,1]^T}\left(\begin{bmatrix} u(\boldsymbol{x}_m) \\ v(\boldsymbol{x}_m) \\ w(\boldsymbol{x}_m) \end{bmatrix}\right) &= \begin{bmatrix} u(\boldsymbol{x}_m) \\ v(\boldsymbol{x}_m) \\ 0 \end{bmatrix}
\end{aligned} \tag{3.5}
$$

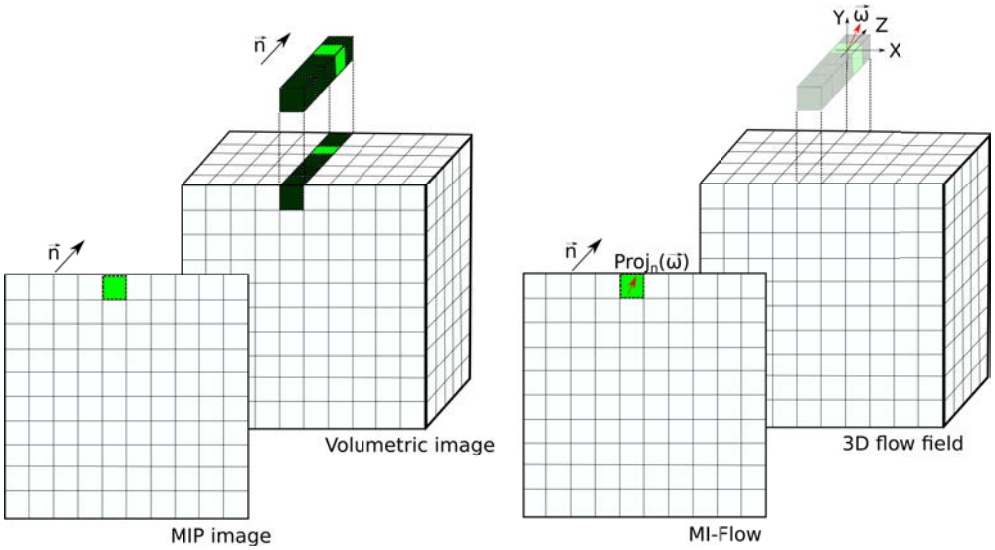
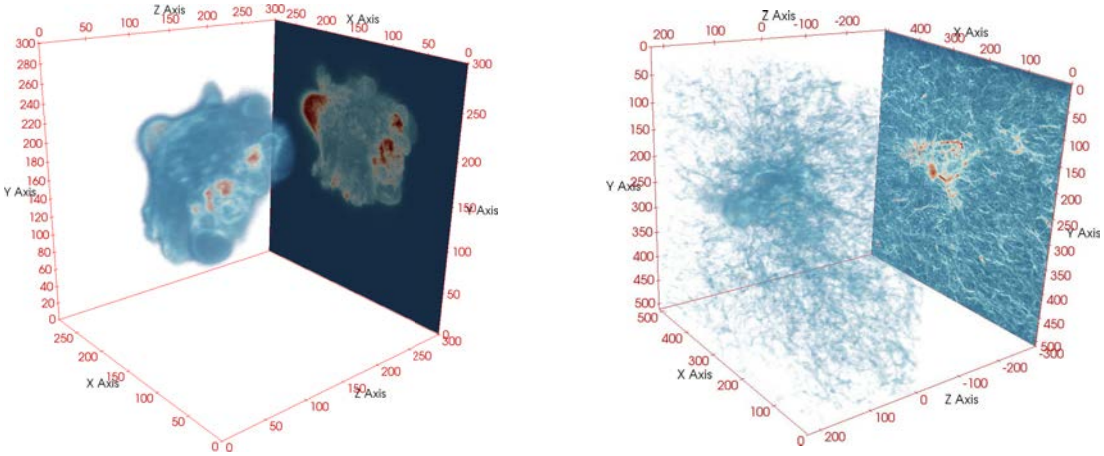


FIGURE 3.15 – The bright green voxel has the maximum intensity. Thus, it is used as the projected value in the MIP image plane.

3.2.7.1 Remarks

We have presented three different visualization techniques for 3D flow field. The methods are based on projection of 3D flow fields onto some 2D plane. Each of the methods have their advantage over the other. 3DHSV and MIP-flow map can be generated for the whole 3D volume in one pass. 3PHS requires a user input to select a voxel location in 3D space where the projected planes intersect. It becomes impractical to generate one map per point in 3D. However, 3PHS provides precise information about the 3D flow field. Unlike 3DHSV map, magnitude is precisely computed for each projection planes and unlike MIP-flow map, all the intersecting voxels are accounted for rendering. The smoothness and discontinuities of the flow field is more apparent. On the other hand, MIP-flow map produces the best summary of the



(a) MIP image of a volume in MV3 A projected on a XY plane.

(b) MIP image of a volume in Collagen 1 projected on a XY plane.

FIGURE 3.16 – MIP of different types of sample.

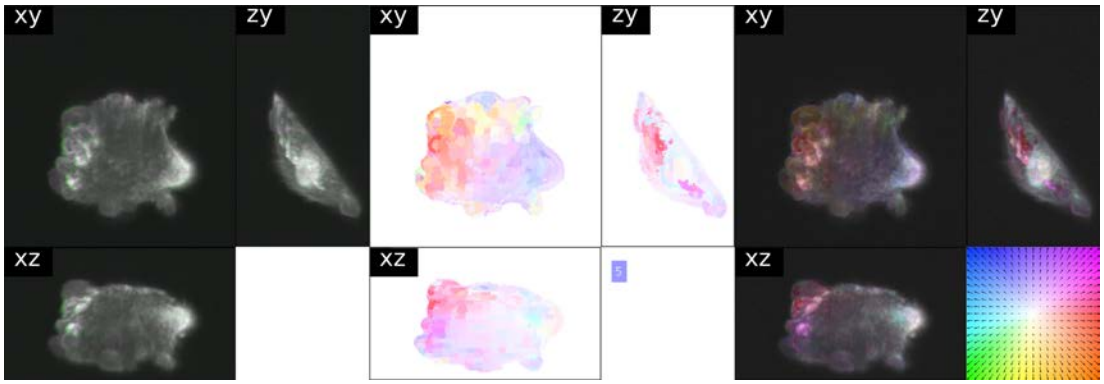


FIGURE 3.17 – From left to right : Overlay of MIPs of I_1 (in green) and I_2(in magenta) in MV3 A ; MIP-flow of the 3D flow field computed using Amat's method ; Overlay of MIP-flow on MIP of I_1. The color code for the flow is given in the subimage on the bottom-right side.

3D flow-field out of the three. Specially in the collagen channel, where thin structures mostly span out to different planes, MIP-flow is better suited to depict the general motion (see Figs. 3.18 and 3.18). This map additionally requires the source image. During the development of the thesis, 3PHS map was mostly used for the assessment. We now briefly summarize our subjective evaluation of the three methods based on the requirements that we proposed in 3.2.2.

Flow Map	Clarity	Coherence	Compactness	Commensurability
3DHSV	medium	high	high	low
3PHS	high	high	medium	high
MIP-flow	high	high	high	medium

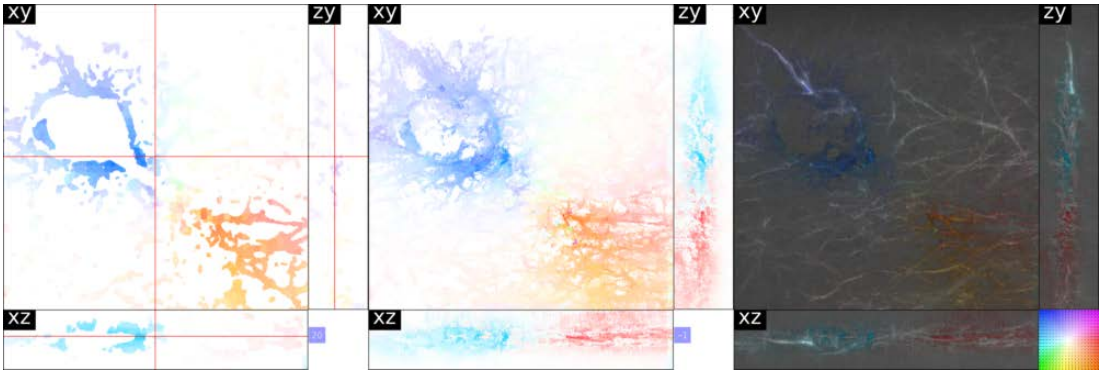


FIGURE 3.18 – From left to right : 3PHS map of a flow field computed using Amat's method for a volume pair in Collagen 1 ; MIP-flow of the 3D flow field ; Overlay of MIP-flow on MIP of I_1. The color code for the flow is given in the subimage on the bottom-right side.

3.3 Accuracy estimation :

Quantitative evaluation of optical flow field was emphasized by Barron *et al.* (1994). In their work, they compare various 2D optical flow algorithms quantitatively, using synthetic image sequences. To synthesize the sequence, they simulated camera and object motion. The ground truth flow field was mostly smooth and translation type. They also created diverging motion field by simulating inwards moving camera. Following the work of Fleet and Jepson (1990), they exploited angular error between estimated flow field and the ground truth flow field. For this, they report cosine angle between the estimated and the ground truth vector, and sine angle between between the normal component of estimated vector and the ground truth vector. The work motivated creation of more extensive benchmarking dataset like Baker *et al.* (2011b), Butler *et al.* (2012), Menze and Geiger (2015) to name a few popular ones. These publicly available dataset assess results based on error measures known as End-Point-Error (EPE) and Angular-Error (AE). EPE is simply the L_2 norm between estimated and the ground truth vector. AE is inspired by the work of Barron *et al.* (1994). It is computed by normalizing the vectors in 3D homogeneous coordinate system and taking the inverse cosine of their dot product. The first two dataset are synthetic ones generated by 3D artists and by the use of computer graphics software. The later one is based on real image acquisition, where motion is computed using 3D visual odometry and laser measurements.

In the context of 3D optical flow benchmarking, no such dataset are available publicly. Hubený *et al.* (2007) synthesized 2D microscopy image sequences of a migrating cell to assess the performance and tune the parameters of 2D variational optical flow methods. They proceed to use the parameters tuned in 2D case in 3D case as well of the assessment of optical flow methods. However, no quantitative evaluation was performed in 3D dataset. Likewise, Delpiano *et al.* (2012) performed evaluation of 2D local-global optical flow methods on synthetic image series

of model dendrites. It was Amat *et al.* (2013) who reported EPE error for their 3D optical flow method, evaluated on synthetic 3D image sequences. But the synthesis was done using simple translation motion of spheroid objects. It is evident that a live-cell image sequence exhibits a lot more complex motion than just a simple translation. Boquet-Pujadas *et al.* (2017a) took a data-assimilation approach that used a model of amoeba, meaning they already had physical measurements relating to the physics of the cellular body. This exempts the method from the need of quantitative assessment of the estimated flow field. However, the method would not be appropriate for specimen without a model. Towards the synthesis of images simulating cellular dynamics, recently Sorokin *et al.* (2018) generated image sequence of protruding and retracting filopodia in a cell. The work was conducted to generate few traceable points in the filopodia in order to have a benchmark for tracking algorithms. Unfortunately, the method, as of now, does not produce dense flow field which could otherwise have been helpful for optical flow benchmarking as well.

Zhu and Milanfar (2010) proposed an image-content metric for image restoration, when no reference image is available. This no-reference metric is based upon singular values of local image gradient matrix. They use the global average of the metric to assess the performance of denoising algorithms and automatically set the parameters. In a similar spirit, we propose a novel metric for assessing the performance of optical flow. Due to the lack of ground truth, we rely on the comparison between the optical flow based reconstruction of source volume and the original source volume for quantitative evaluation. Our metric is based on the eigenvectors of local structure tensors.

We reconstruct the approximation of I_1 by backward warping I_2 using the computed 3D flow field. Then, we compare the structures of the reconstructed volume I_r with the real source volume I_1. Comparing structure is better than the commonly used DFD or displaced frame difference (Brox and Malik (2011)) as it is robust to noise and independent of type of interpolation used for warping. DFD tends to be lower where intensity gradient is relatively low, even though the flow vector is erroneous, and it tends to be high where intensity gradient is strong (i.e., around a sharp edge) even though the flow vector is almost accurate. The structure tensor based comparison, however, does not manifest such disproportionate behaviour. Our idea is to rely on the underlying image structure rather than on the image intensity itself as in DFD. The former is both more constraining and differentiating for the flow. Under this premise, we propose a new error measure called Structural Angular Error (SAE). Let us emphasize that it is still a quantitative error measure, not a qualitative one, in the absence of ground truth.

We compute and compare the principal orientation of 3D structure in the volume using the eigenvectors of structure tensor (Ahmad *et al.* (2015)). The average of SAE over the foreground region is used to assess the accuracy of the motion estimation.

3.3.1 Computation of 3D Structure Tensor

Following the work on autocorrelation by Hannah (1974), Förstner and Gülch (1987) and Harris and Stephens (1988) popularized the use of local structures for feature extraction and matching. In the field of bio-imaging, the autocorrelation matrix or structure tensor (ST) has been used for various purposes. Fonck *et al.* (2009), in their study of morphological characterization of elastin fibers, used ST analysis to assess the orientation of the fibers. They made a plug-in for ImageJ called "OrientationJ" available for 2D ST analysis. Later, Rezakhaniha *et al.* (2012) used it to investigate the waviness and orientation of collagen fibers. Budde and Frank (2012) did 2D ST analysis to study crossing fibers in histological section of rat brian. Pop *et al.* (2013) used it to enhance the contrast of images in the cell membrane region. Zhang *et al.* (2016) demonstrated its robustness in 2D and 3D synthetic data in presence of noise, anisotropy, deformation, blur, and variation in resolution of image. Khan *et al.* (2015) used 3D ST to analyze and validate light microscopy and diffusion MRI data. More recently, Teh *et al.* (2017) registered ST volumes of synchrotron radiation imaging (SRI) to the diffusion tensor imaging (DTI) data to validate DTI measurements. Angles derived from the projection of eigenvectors of DTI and ST of SRI were used to assess the correspondence. This approach is similar to our work, for we want to validate two supposedly same volumes. First volume is the original source volume itself. The second volume is the reconstructed volume computed by backward warping of target volume using the computed optical flow field. This boils down to validation of registration result.

We do not use 2D angles of projected vectors like Teh *et al.* (2017). Instead, we take the cosine of angles between two corresponding 3D eigenvectors. This does not necessitate the choice of projection planes and keeps the validation simple.

Consider a 3×3 tensor $\boldsymbol{A}$ computed using 3D image gradient ∇I as follows,

$$\boldsymbol{A} = \nabla I \nabla I^T = \begin{bmatrix} I_x^2 & I_x I_y & I_x I_z \\ I_x I_y & I_y^2 & I_y I_z \\ I_x I_z & I_y I_z & I_z^2. \end{bmatrix} \tag{3.6}$$

We now compute the structure tensor as :

$$\boldsymbol{J} = G_\sigma \star A, \tag{3.7}$$

where G_σ is a smoothing kernel of bandwidth σ and "$\star$" is the convolution operation acting independently on each element of $\boldsymbol{A}$. Here, we use a 3D Gaussian kernel. The matrix $\boldsymbol{J}$ is a symmetric positive semidefinite matrix as a consequence of Eq. 3.6. Thus, we can state that :

— The eigen values of $\boldsymbol{J}$ are not negative such that : $\lambda_0 \geq \lambda_1 \geq \lambda_2 \geq 0$.

— The eigen vectors $\boldsymbol{e}_1$, $\boldsymbol{e}_2$ and $\boldsymbol{e}_3$ associated to λ_0, λ_1 and λ_2, respectively, are orthogonal to each other.

Since, $\boldsymbol{A}$ takes the form of covariance matrix of intensity gradients, few more comments can be made about the eigen-components of $\boldsymbol{J}$:

— The eigen vector $\boldsymbol{e}_0$ associated to the largest eigen valueλ_0 gives the direction of the biggest intensity change.

— The eigen vector $\boldsymbol{e}_2$ associated to the smallest eigen valueλ_2 gives the direction of the least intensity change.

The eigenvectors $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ are the normal and the tangential components of the orientation of surface in the image. The former component points to the direction of highest gradient change whereas the latter follows along the surface with least change. We propose to use either of the components to evaluate the reconstruction after optical flow computation.

3.3.1.1 Structural Angular Error

Let $\boldsymbol{e}_2^r(\boldsymbol{p})$ and $\boldsymbol{e}_2^1(\boldsymbol{p})$ be the smallest order eigenvectors of the structure tensors at $\boldsymbol{p}$ in I_r and I_1 respectively. We then compute the structural angular error (SAE) between $\boldsymbol{e}_2^r(\boldsymbol{p})$ and $\boldsymbol{e}_2^1(\boldsymbol{p})$ according to :

$$SAE(\boldsymbol{p}) = \arccos\left(\frac{\boldsymbol{e}_2^r(\boldsymbol{p})}{|\boldsymbol{e}_2^r(\boldsymbol{p})|} \cdot \frac{\boldsymbol{e}_2^1(\boldsymbol{p})}{|\boldsymbol{e}_2^1(\boldsymbol{p})|}\right). \tag{3.8}$$

The eigenvectors, depending upon the numerical solver, could have flipped direction since :

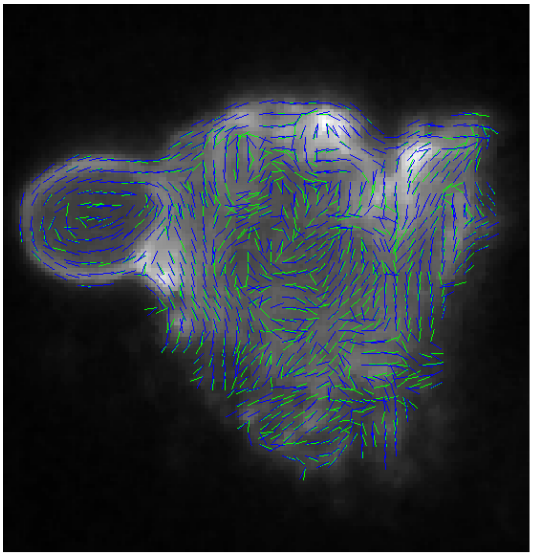
$$\boldsymbol{J}(-\boldsymbol{e}) = \lambda(-\boldsymbol{e}). \tag{3.9}$$

This leads to high angular error even though the surfaces have a good match. To handle such case, we compute the acute angle between the corresponding vectors. The normal components, i.e. $\boldsymbol{e}_0^r(\boldsymbol{p})$ and $\boldsymbol{e}_0^1(\boldsymbol{p})$, can also be used instead of the tangential ones. Lower the value of SAE, the better. We compute the average of SAE over the foreground region to quantify the error in estimated flow field. The method is useful when the complete flow field of the foreground region is available and thus, it is not appropriate for assessment of sparse matching. Heat map of SAE, shown in Fig. 3.19b, is useful for closer analysis of flow field error.

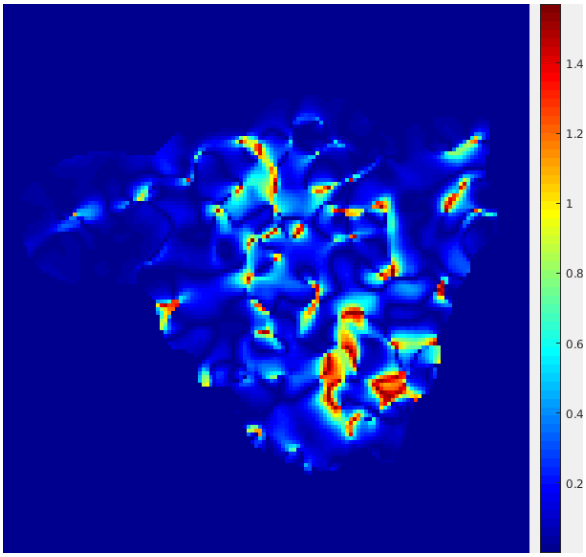
3.3.1.2 Choice of Smoothing Window σ

The only parameter in this method is the value for σ. It controls the size of smoothing neighborhood. σ should be large enough to include gradient information in all direction. If it is very large, the eigenvectors will be smoother, and will loose directional information of smaller objects. For isotropic case, we can use the Gaussian kernel for smoothing given as :

$$G_\sigma(x, y, z) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2 + y^2 + z^2}{2\sigma^2}\right) \tag{3.10}$$



(a) Tangential components (projected on XY plane) of a slice of I_1 (in green) and I_r (in blue) overlaid on the slice of the I_1 in MV3 A.



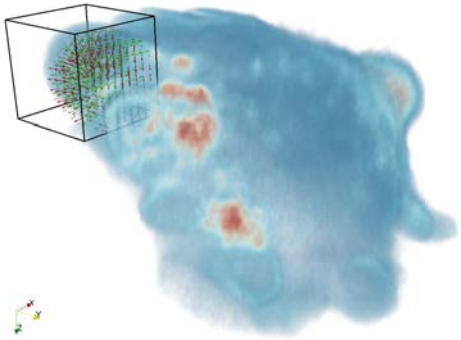
(b) Heat map of the computed SAE on a slice. The color code is given on the right side.

3.3.1.3 Eigen Decomposition

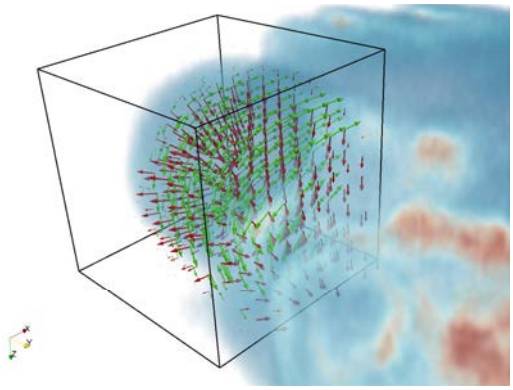
Iterative and non iterative methods for the eigen decomposition of 3×3 symmetric matrices are discussed in Eberly (2018). A closed form solution for the eigen-decomposition of 3D ST is discussed in Ginkel (2002). Time complexity of a 3×3 symmetric matrix is $O(3^3)$. For a volume of size $M \times N \times P$, the complexity reaches $M.N.P.O(3^3)$, which is time and memory consuming. We propose to subsample I_r and I_1 before performing ST analysis. This would require to reduce the size of σ as well. Another trick, is to use solver that can compute either the highest or the smallest eigen-components of $\boldsymbol{J}$. We have used MATLAB's "$eigs$" (Stewart (2001)) routine for this. Figures 3.20a and 3.20b depict the orthogonal (in red) and tangential (in green) components of the surface of the bleb region in the bounding box. The eigenvectors $\boldsymbol{e}_0^r$ and $\boldsymbol{e}_1^r$ order were used to obtain the orthogonal and tangential components, respectively. Figures 3.21a and 3.21b depict the tangential components of the bleb surface in I_1 (in green) and I_r (in blue), respectively. Figure 3.19a depicts these tangential components on a slice. Figure 3.19b depicts the heatmap of the SAE on the slice, which can be taken as the confidence measure of the estimated optical flow.

3.3.2 Conclusion

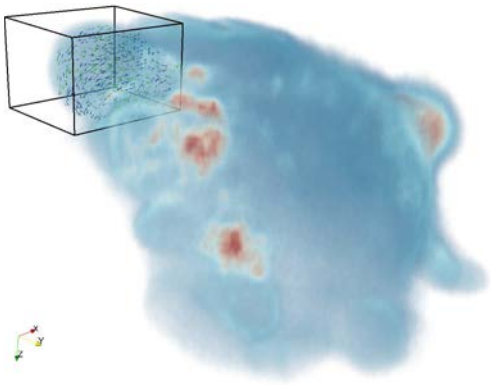
In this chapter, we presented three different 3D flow field visualization techniques and a way to assess the quality of computed optical flow field. The visualization method renders the magnitude and the direction using hue, saturation and value spaces for color. The visualizations



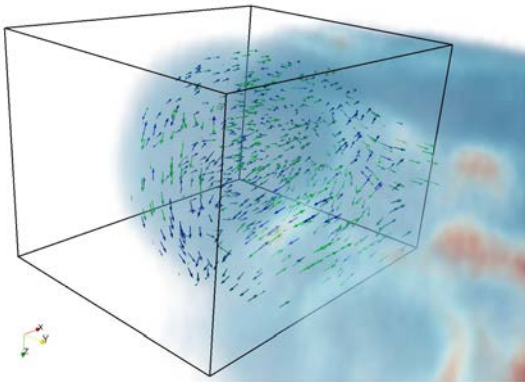
(a) Orthogonal (in red) and tangential (in green) components of the bleb region



(b) Zoomed version



(a) Tangential components of I_1 (in green) and I_r (in blue).



(b) Zoomed version. Some of the corresponding vectors are 180° out of phase.

produced are clear, coherent, compact and commensurable. They require neither the hassle of proper file formatting of the flow fields, nor the knowledge of special rendering software. We mostly rely on 3PHS and MIP-flow map to visualize the effects of different parameters in different optical flow computation models. We believe the visualizations are not only useful for the research and development of 3D motion estimation algorithms, but also to the concerned microscopists and biologists for better understanding of the dynamic scenes. We have presented a measure to assess the quality of the computed motion field in absence of the ground truth motion field. It is based on the computation of 3D eigenvectors of structure tensor of the 3D volume. The average of angle between the principal eigenvectors at each voxel location is computed as the final measure of estimation error. We will investigate its performance with respect to the standard EPE measure in chapter 5.

CHAPITRE 4

MATCHING

In this chapter, we describe the methods we have explored to match 3D data. Matching between 3D images can be achieved in sparse or semi-dense manner. Sparse matching is faster but has less number of correspondences. It requires additional elaborate densification scheme. Semi-dense matching has more correspondence, and thus has better chances of good refinement. Nonetheless, it is slow. The element to be matched could be first voxels or patches of voxels. When considering a single image plane extracted from the volume, one may consider a voxel as a pixel. This approach is a image-based representation. The straight-forward features for matching are the intensity values, gradients, textures, etc. Another paradigm of matching is to pursue the triangular mesh representation of data rather than the image data itself. In this case, the element to be matched could be a vertex or a patch of the surface. The features to be exploited could be curvature, area, spectral signatures, etc.

In the first part of the chapter, we describe the methods involving the image representation. We extend the well-known 2D PatchMatch method proposed by Barnes *et al.* (2009) to 3D. The method proves to be efficient and accurate in 2D image registration problems.

4.1 A short overview of the PatchMatch algorithm

We briefly recall the 2D PatchMatch method. The goal of PatchMatch is to find the best matching pair of patches $\boldsymbol{P}$ and $\boldsymbol{P}^c$ centered at $\boldsymbol{p}$ in I_1 and $\boldsymbol{p}^c$ in I_2, respectively, which results in a displacement vector $\boldsymbol{w}(\boldsymbol{p}) = \boldsymbol{p}^c - \boldsymbol{p}$. It achieves this in an iterative manner where displacements having best matching score are propagated from neighbours, and refined using random search

(Barnes *et al.* 2009). The idea is to avoid brute force matching (i.e. exhaustive search as in block matching). PatchMatch relies on natural coherence in the images to propagate good matches. As a result, the method performs faster than other matching algorithms (Muja and Lowe 2009), while keeping a low memory footprint. The 2D PatchMatch algorithm proceeds in three stages :1) initialization, 2) propagation, and 3) random search, of which the last two are iterative steps. Now, we describe the stages.

Initialization : In this stage, we initialize the 2D displacement field either with random vectors or vectors approximated with coarse matching. Originally, Barnes *et al.* (2009) used random initialization. Lu *et al.* (2013) used potentially good displacement vectors induced by super-pixel similarity in a close range of vicinity. Hu *et al.* (2016) used initialization from matches obtained in coarser levels.

Propagation : In this stage, for any iteration i, the displacement vectors having good matching scores are propagated. A raster scan is performed from top-left to bottom-right. In the scanning process, each current matching score for a given pixel is compared with those of its causal neighbours, i.e., neighbours just previously scanned which probably have refined matches. If a neighbour has the best matching score, its displacement vector is copied to the current pixel of interest. This implicitly imposes a smoothness in the resulting flow field. The matching score can determined by one of the similarity measures discussed in 4.2.3.2.

Random Search : In this stage, we try to get a better match by first randomly selecting pixels in I_2 around the current best estimate. The random draw is done in a squared region having side length of R. If the randomly selected pixel tends to have a better matching score, it becomes the new best corresponding voxel. A new random sample is drawn again from a region with side length of $R/2$ around the current best estimate. These iterations of random sampling continue by halving the side length of the square until the square region has unit area.

4.2 3D PatchMatch

Now, we describe our 3D PatchMatch extension and its application to 3D microscopy image volumes. We refer to $\boldsymbol{p}$ as a voxel, and $\boldsymbol{P}$ as a 3D patch of voxels. The displacement field $\boldsymbol{w}$ to be computed consists of 3D vectors.

4.2.1 Scanning and sequential analysis

Two choices can be made regarding the scanning : 1) sparsity and 2) density. The straightforward way would be to scan each voxel in raster scan order, slice-by-slice, and propagate the matches from its directly connected neighbours. This leads to dense matching, which after outlier removal turns out to be semi-dense. To speed up the matching process, Hu *et al.* (2016)

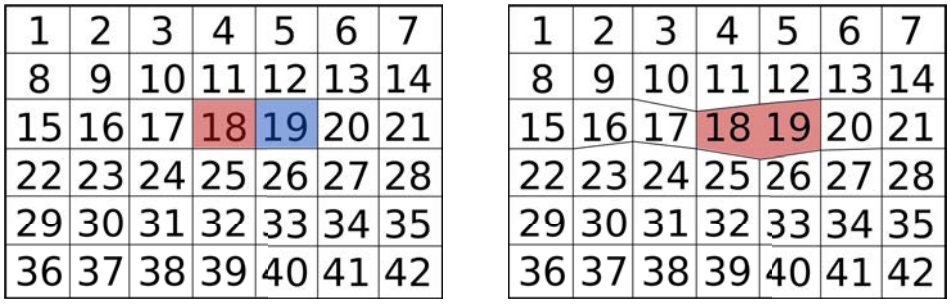


FIGURE 4.1 – From left to right : numbering the grid region, and merging of grid region 18 and 19.

proposed to use seed points on a regular grid. Giraud *et al.* (2019) used a patch consisting of pixels in a fixed radius around the center of a super-pixel. These schemes greatly reduces the scanning time. For sparse method, we take the super-pixel-based approach described in what follows.

4.2.2 Super-pixel Representation

In a scene of smoothly moving objects, the neighbouring points of a dynamic element move similar to the element itself. More often, the neighbouring points look similar to the element. Therefore, the motion of the element in this similar looking vicinity can be a good estimate of the neighbouring points. This motivates us to exploit super-pixel segmentation of an image slice, so that the motion of segments can be computed more efficiently. Here, we use the term pixel instead of voxels as we deal with a slice of the volumetric data.

A LM image volume consists of a stack of 2D slices where each slice depicts information corresponding to a given depth layer of the cell. Considering this volume structure, we compute the 3D displacement vectors only at the points of the stack grid, that is, at the pixels of every slice grid. In what follows, the term displacement vector is interchangeably used as flow vector.

Our framework starts with the computation of a set of correspondences in a pair of LM image volumes I_1 and I_2. To be efficient, matching is performed on a sparse set of points. Superpixels are good candidates for that purpose, since we can easily control their density, they are almost regularly distributed over the image while taking into account the intensity patterns. We adopt the SLIC method Achanta *et al.* (2012). The SLIC algorithm lets us control the size and number of super-pixels. Because it uses a regular gridded initialization, one can roughly assign a index-number to the super-pixel as in Fig. 4.1. This becomes useful while performing raster-scan even in an irregular distribution of super-pixels. It may be a case, when two or more grid cells merge during segmentation. Then such region will be scanned each time the merged grid cells are indexed in the raster-scan (see Fig. 4.1).

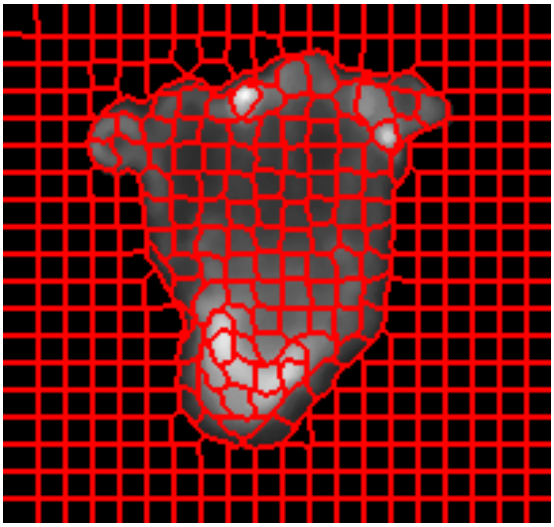


FIGURE 4.2 – SLIC segmentation of a slice.

On the contrary, super-voxels are difficult to manipulate when raster scan is to be performed. Because the segments are irregular in all three dimensions, the causal neighbours for propagation are difficult to assign. Anisotropy and number of slices can be problematic as well. Therefore, we opt for super-pixel based representation rather than the super-voxel representation.

We obtain K non overlapping super-pixels in the slices of the volume I_1. Super-pixels in a slice of real data is depicted in Fig. 4.2. Let $\mathcal{S}$ be the set of super-pixels and s_k be the k^{th} super-pixel. For each s_k, we define a set of 3D neighbouring super-pixels $\mathcal{N}_k = \{s_{k_i}, i = 1..n\}$ in I_1 which can lie in different slices. This neighborhood will be used in the propagation step of the 3D PatchMatch method. The set $\mathcal{N}_k$ comprises n super-pixels with minimum distance to s_k according to the distance measure :

$$d_{k,j} = \exp\left(-\frac{(\boldsymbol{c}_j - \boldsymbol{c}_k)^2}{\sigma_d} - \frac{(I_1(\boldsymbol{c}_j) - I_1(\boldsymbol{c}_k))^2}{\sigma_I}\right) \tag{4.1}$$

where $\boldsymbol{c}_k$ and $\boldsymbol{c}_i$ are the centres of super-pixels s_k and s_j, and σ_d and σ_I are respectively the spatial distance and intensity value normalizers.

4.2.2.1 Super-pixel matching per slice

Correspondences between points of I_1 and I_2 are defined by the displacement vectors in $\boldsymbol{w}$ such that $I_1(\boldsymbol{x})$ and $I_2(\boldsymbol{x} + \boldsymbol{w}(\boldsymbol{x}))$ form a matched pair, where $\boldsymbol{x}$ is the location of a point in I_1.. A smooth flow field suggests that similar voxels in a close vicinity move similarly. Then, we estimate a displacement vector for only the center c_k of super-pixel s_k, as representative of its vicinity.

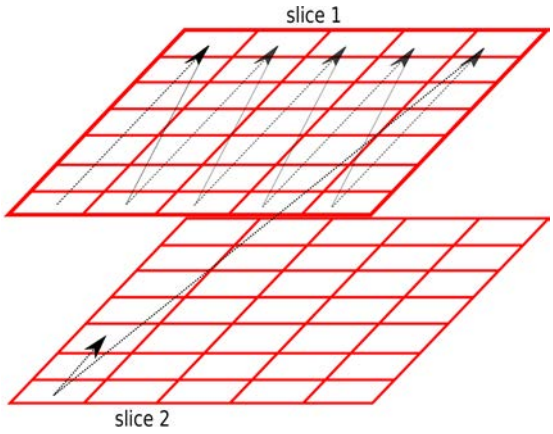


FIGURE 4.3 – Scanning order in the stack for 3D PatchMatch. The slices are regularly tessellated with cell size adapted to super-pixel mean size. Each cell encapsulates a super-pixel center.

For initialization, $\boldsymbol{w}(\boldsymbol{c}_k)$ is chosen as the 3D vector to the center $\boldsymbol{c}_{k'}$ of super-pixel $s_{k'}$ in I_2 selected as the most similar one according to the distance given in Eq. 4.1, where $I_2(\boldsymbol{c}_{k'})$ is substituted for $I_1(\boldsymbol{c}_j)$. In our experiment, it proved to be better than the random initialization. Then, we iterate propagation and random search. Scanning of the volume is performed in raster-scan order in each slice, visiting each s_k according to their index order. The slices are visited in a top-to-bottom way through the stack, as shown in Fig. 4.3.

We now briefly explain propagation in 3D setting. To update $\boldsymbol{w}(\boldsymbol{c}_k)$, we consider the current displacement vectors at the centers of the super-pixels in $\mathcal{N}_k$. This provides n possible correspondences for $\boldsymbol{c}_k$. We select the vectors, among the n labels plus the current label of $\boldsymbol{c}_k$, with a patch-based similarity measure. The patch is a local 3D patch of size $9 \times 9 \times 3$ around $\boldsymbol{c}_k$. For patch similarity cost, we adopt the SSD between two patches.

During the random search, a voxel is drawn at random from a cuboid region having side length of R. If the random selection tends to getting a better matching score, a new random sample is drawn again from a cuboid region with half the previous side length, i.e., $R/2$. The random sampling continues by halving the side length of the cuboid until the cuboid has unit volume.

At the very end of the alternate iterations, a sparse 3D flow field is obtained. A median filter is applied to the 3D vectors of the field to remove outliers that significantly disobey smoothness in the flow field (Sun *et al.* 2010). The filtered vectors are then used to interpolate displacements for the remaining voxels in I_1 as described in the next section.

4.2.2.2 Sparse-to-Dense Interpolation

To get a dense 3D flow field $\boldsymbol{w}$ between I_1 and I_2, the 3D displacement vectors for all the points in each super-pixel segment need to be computed. To address the issue, we locally

interpolate the sparse set of displacement vectors supplied by the first stage. Of course, we do not want to just duplicate the flow vector of c_k to all points of super-pixel segment s_k, the undergone motion being not necessarily locally translational.

To this end, we design an interpolation scheme inspired by EpicFlow interpolation introduced in Revaud *et al.* (2015). EpicFlow interpolation is an edge-aware interpolation technique that has been known to perform well in 2D optical flow benchmarks. It assumes that motion discontinuities are subset of intensity edges, and derives weights using an approximation of geodesic distance between points in Voronoi cells with intensity edges as cost map. In our case, we exploit the super-pixel segmentation instead of Voronoi tessellation. In addition, motion boundaries will be taken into account in the interpolation stage in a different manner.

Let us consider a point $\boldsymbol{x}$ in super-pixel s_k. The question is how to involve known vectors in the neighbourhood to interpolate the displacement or flow vector at $\boldsymbol{x}$. For this, we introduce a *flow-aware* weight that encodes the respective orientation of known vectors. We allow interpolation with vectors of neighbours only when the flow vector at c_k and those of the super-pixel centers of its neighbours in $\mathcal{N}_k^\star$ are converging. Here, $\mathcal{N}_k^\star$ is the subset of neighbouring super-pixels lying in the same slice as s_k. Given the anisotropy of the volume, we do not involve off-the-plane super-pixels. Also, we stress that we interpolate sparse and possibly irregularly distributed 3D vectors.

We build an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of super-pixel centers c_k's , and $\mathcal{E}$ is the set of edges between adjacent super-pixels in $\mathcal{S}$. Let $\boldsymbol{a}$ and $\boldsymbol{b}$ be any two vertices (centers of super-pixel s_a and s_b respectively). We define edge weights w_e using an approximation of geodesic distance between two vertices in $\mathcal{V}$ defined by :

$$D_g(\boldsymbol{a}, \boldsymbol{b}) = \sum_{p_i \in L_{\boldsymbol{a},\boldsymbol{b}}} q_{p_i}, \tag{4.2}$$

where p_i is a point in the line segment $L_{\boldsymbol{a},\boldsymbol{b}}$ between $\boldsymbol{a}$ and $\boldsymbol{b}$. The scalar q_{p_i} is the intensity edge response at p_i which can be derived from an intensity edge detector or simply from local intensity contrast. Noisy responses are filtered out.

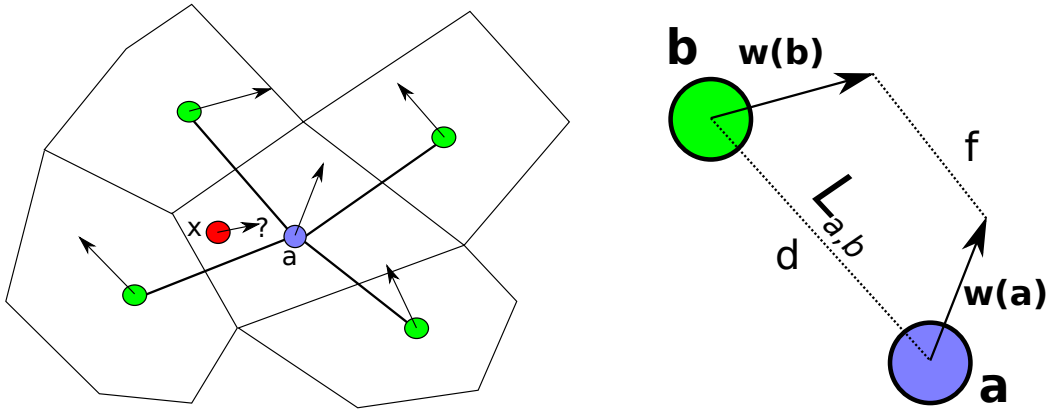
We aim to nullify the interaction between two flow vectors if they are moving away from each other. Assuming no strong local curl in $\boldsymbol{w}$, we encourage interpolation from $\boldsymbol{w}(\boldsymbol{a})$ and $\boldsymbol{w}(\boldsymbol{b})$ to get flow vector at $\boldsymbol{x}$ in super-pixel s_a, if the flow vectors $\boldsymbol{w}(\boldsymbol{a})$ and $\boldsymbol{w}(\boldsymbol{b})$ are converging as illustrated in Fig.4.4. Now, we define edge weights on $\mathcal{E}$ as :

$$w_e(\boldsymbol{a}, \boldsymbol{b}) = \begin{cases} \frac{1}{D_g(\boldsymbol{a},\boldsymbol{b})} & \text{if } \frac{d_{a,b}}{f} > 1 \\ 0 & \text{otherwise,} \end{cases} \tag{4.3}$$

where $d_{a,b}$ is the Euclidean distance between $\boldsymbol{a}$ and $\boldsymbol{b}$, and f the distance between points $\boldsymbol{a} + \boldsymbol{w}(\boldsymbol{a})$ and $\boldsymbol{b} + \boldsymbol{w}(\boldsymbol{b})$ as shown in Fig.4.4. Given the weights derived from the respective

TABLE 4.1 – Error evaluation on 3D synthetic tests.

Method	Ours		Amat's method	
Error type	AEPE	AAE	AEPE	AAE
Translation	0.19	0.26	0.17	0.02
Rotation+Translation	1.74	0.14	1.93	0.11
Rotation+Scale	4.30	0.67	7.83	1.17



(a) Local sparse flow field (b) Respective orientation.

FIGURE 4.4 – Interpolation of flow vector at $\boldsymbol{x}$ (in red) according to the central flow vector (in blue) and the neighbouring flow vectors (in green). The blue and green points are the centers of their respective super-pixels delineated by black lines.

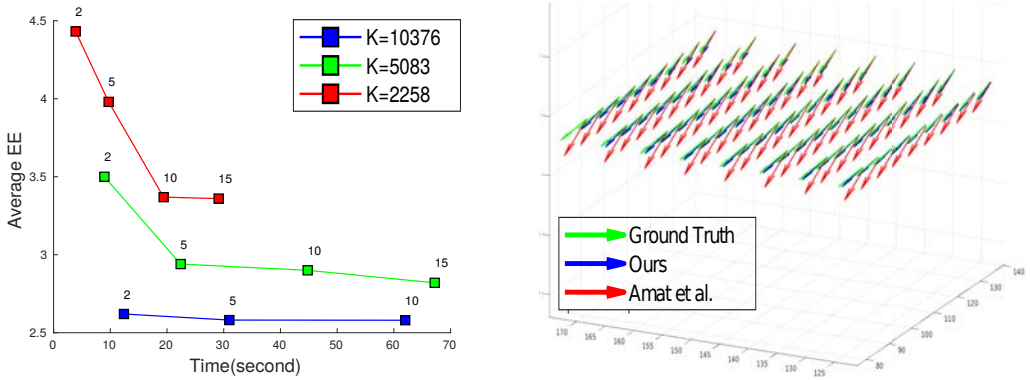
orientation of the flow vectors and the spatial location of the super-pixels, we interpolate for the flow vector at $\boldsymbol{x}$ as follows :

$$\boldsymbol{w}(\boldsymbol{x}) = \frac{w_{a,x}\boldsymbol{w}(\boldsymbol{a}) + \sum_{\boldsymbol{i}\in\mathcal{N}_a^\star} w_e(\boldsymbol{a},\boldsymbol{i})\boldsymbol{w}(\boldsymbol{i})}{w_{a,x} + \sum_{\boldsymbol{i}\in\mathcal{N}_a^\star} w_e(\boldsymbol{a},\boldsymbol{i})}, \forall \boldsymbol{x} \in s_a, \tag{4.4}$$

where $w_{a,x} = 1/D_g(\boldsymbol{a},\boldsymbol{x})$. Finally, the interpolation of the 3D flow is performed slice by slice.

4.2.2.3 Experimental results

First, we objectively evaluated the method performance on computer-generated sequences with ground-truth, obtained by applying known transformations on real volumes issued from cell tracking challenge (Maska *et al.* 2014). The image volume is of size $275 \times 231 \times 35$ voxels. One hundred uniformly random transformations were applied for each class of transformation, from 3 to 10 units for x- and y-translation, from 2 to 3 units for z-translation, rotation range around the z-axis was $[-10°, 10°]$, and scaling range $[0.5, 3]$ in x and y directions, and $[0.5, 1.5]$ in z direction. We compared the 3D flow field obtained by our method against Amat's method. Table 4.1 contains average end-point error (AEE) and average angular error (AAE). Our method handles motion including rotation or scaling better than Amat's method. Unlike our method,



(a) AEE vs time in 3D.

(b) 3D flow field in a slice.

FIGURE 4.5 – Left : Accuracy converges to a value after few iterations (number of iterations is mentioned above each plotted points). Right : 3D flow vectors estimated in a slice with ground-truth in green, ours in blue, Amat's method in red (ground-truth : rotation of 10° around z-axis, 3D translation of [10,-20,3]).

Amat's method computes a vector for a supervoxel, and assigns the same vector for the involved voxels which is not appropriate to handle complex motion. Figure 4.5b contains a portion of flow fields in a slice in the case of rotation and translation. The flow field computed by our method closely follows the ground-truth flow field.

We have studied the tradeoff between runtime and accuracy with a varying number of super-pixels. The method is implemented in C++ (not optimized) and run on a computer with 2.8 GHz Intel i7 processor and 16 GB of RAM. The parameters are set as follows : the size of initial random search region is $[40 \times 40 \times 7]$, $\sigma_I = 0.3$ and $\sigma_d = 100$. Plots in Fig.4.5a show a gain in accuracy with smaller size of super-pixels (larger the K, smaller the super-pixel size). The convergence is reached after 5-10 iterations (best viewed in electronic form for zooming or in color printed form).

We also applied our method on two real sequences acquired using microscopes described in Dean *et al.* (2015, 2016). The first sequence is MV3 B described in Chapter 3 (Fig.4.6a-b). Figure 4.7 plots the 3D displacement vectors in one of the protrusions in the cell. The vectors tend to stretch the cell membrane. The computation time to estimate the 3D flow field between two volumes was 163 seconds with 19, 120 and 24 seconds for super-pixel generation, 3D patch matching, and flow interpolation, respectively

The second sequence is a MV3 cell undergoing large scale protrusion depicted in Fig.4.6c-d. The main body of the cell in middle section squeezes as protrusions abruptly appear. A cross-section view of its 3D flow field is plotted in Figure 4.8, which coherently suggests shrinking in the middle region and stretching in protrusions. The total computation time for one 3D flow field was 101 seconds, with 49, 44 and 8 seconds for super-pixel generation, 3D patch matching, and flow interpolation, respectively.

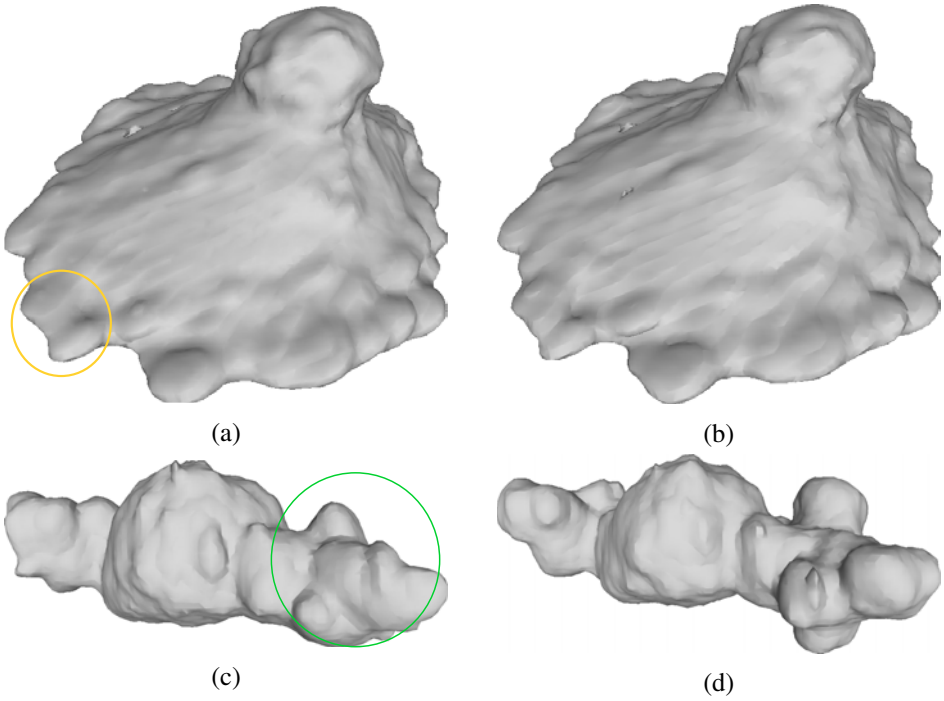


FIGURE 4.6 – (a) source and (b) target volumes for the first sequence with volume of size $241 \times 240 \times 101$, (c) and (d) for the second sequence with volume of size $256 \times 256 \times 143$ for two different MV3 cells. The colored circles roughly correspond to the regions highlighted in Fig.4.7 and 4.8 with respective color. Input microscopy images are courtesy of Danuser Lab, UTSW.

4.2.2.4 Remarks

Furthermore, we have presented a method for efficient 3D optical flow estimation in 3D LM image volumes. It is efficient in the sense that it does not have to scan and match all the voxels in the volume, thus reducing computation time. It involves the interpolation of a sparse 3D flow field obtained by matching super-pixels per slice. We have developed a 3D PatchMatch algorithm to find correspondences. For densification, we have designed a weighted interpolation taking intensity edges and local flow variations into consideration. Tests performed on synthetic cases show its effectiveness. The 3D computed flow field could be further refined by post-processing it with a variational algorithm.

4.2.3 Voxel Representation

Though, the super-pixel representation allows efficient execution of PatchMatch, it does not perform well when low-texture regions and small displacements are involved. An additional challenge is posed by the structure of the dynamic content. For example, a meaningful segmentation of collagen fiber is difficult to achieve with super-pixels. Besides, it does not produce

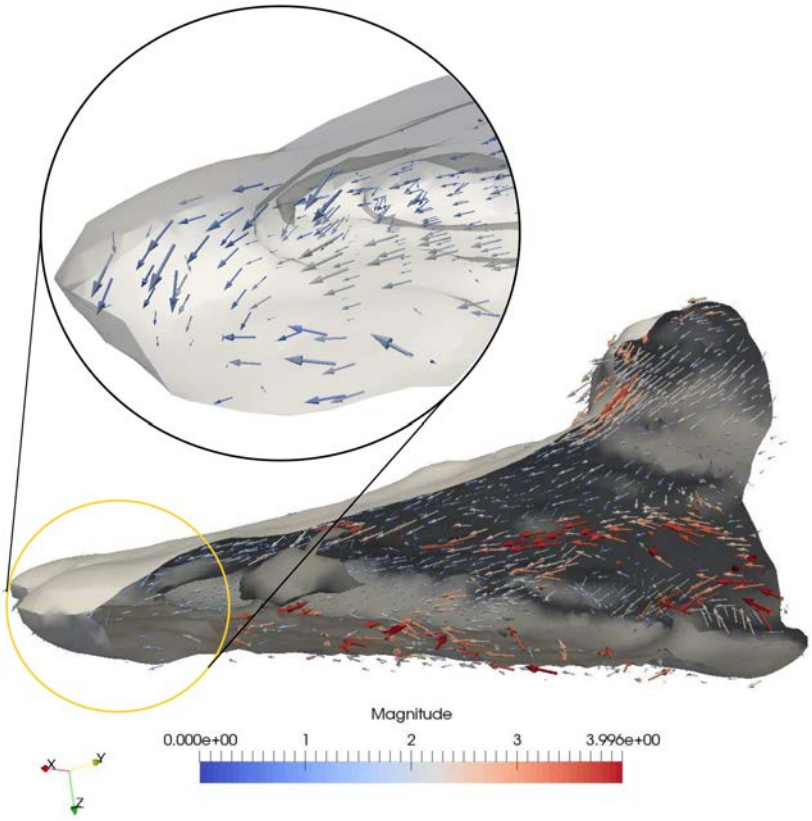


FIGURE 4.7 – Cross section view of flow field for the 1st sequence.

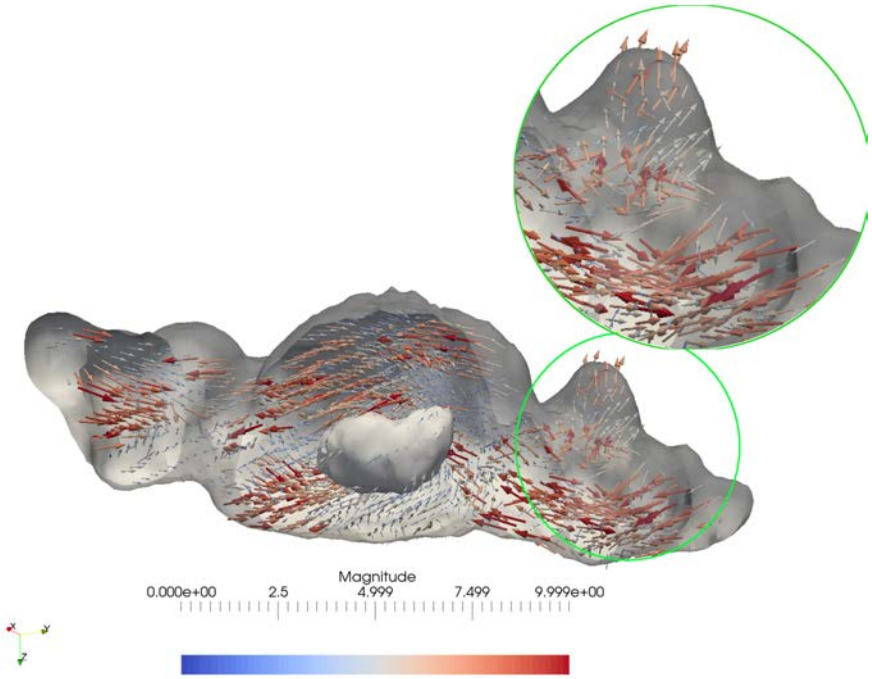


FIGURE 4.8 – Cross section view of flow field for the 2nd sequence.

smooth and regularized enough flow fields when complex motion is involved. This motivates us to investigate denser scanning at voxel level in the volume. To gain in execution speed and reduce local minima, we introduce coarse-to-fine 3D PatchMatch.

4.2.3.1 Coarse-to-Fine 3D PatchMatch

The efficiency of coarse-to-fine scheme in 2D PatchMatch paradigm has been demonstrated in Hu *et al.* (2016) and Li *et al.* (2018). Unlike these works, where seed points were selected from a uniform grid, we perform scanning at each voxel. Choosing seed points from a regular grid in a mostly void 3D volume might lead to loss of information. Therefore, we decide to scan every voxel in the volume.

In the coarsest level, the initialization is done randomly. After few iteration, the 3D PatchMatch algorithm produces a global motion field at the coarsest level. This motion field is then upscaled and up-sampled to be used as initialization in next finer stage. No warping of image is done. Another benefit of coarse-to-fine scheme for PatchMatch is that it can make use of relatively a smaller random search region within each consecutive finer level. We assign the search region to be half the size of image volume at the coarsest level and adjust it in finer levels in accordance with the maximum motion magnitude computed in preceding coarse level. This allows for significant speed boost in the execution of the method.

4.2.3.1.1 Outlier Removal : At each pyramid level, we perform outlier removal by forward-backward consistency check. We reverse the role of source and target frames to compute backward flow $\boldsymbol{w}_b$. We remove the flow vectors that are inconsistent when $|\boldsymbol{w}(\boldsymbol{x}) + \boldsymbol{w}_b(\boldsymbol{x} + \boldsymbol{w}(\boldsymbol{x}))| > \epsilon$. We then use median filtering to possibly fill in flow vectors at deleted locations. This results in a semi-dense flow field. Though the 3D PatchMatch algorithm itself is not parallelizable, we can run the forward and backward pass of the 3D PatchMatch in parallel. However, the memory footprint doubles.

4.2.3.2 Similarity Cost

In pursuit of appropriate similarity cost for 3D fluorescence images, we compare three conventional similarity costs : (i) Sum of squared difference (SSD), (ii) Truncated zero normalized cross-correlation (TZNCC), (iii) Hamming distance of Census signature (CS).

A bit of Census signature is the binarized form of image derivative, given as :

$$H(r_q) = \begin{cases} 0, & \text{if } r_q \geq 0 \\ 1 & \text{otherwise,} \end{cases} \tag{4.5}$$

Similarity Cost	Formulation
SSD	$\sum_{\boldsymbol{q}\in\boldsymbol{P}}(I_1(\boldsymbol{q}) - I_2(\boldsymbol{q}+\boldsymbol{w}(\boldsymbol{p})))^2$
CS*	$\sum_{\boldsymbol{q}\in\boldsymbol{P}}(H(r_{\boldsymbol{q}}) \oplus H_c(r_{\boldsymbol{q}+\boldsymbol{w}(\boldsymbol{p})})$
$TZNCC$**	$1 - \frac{1}{\sigma_{\boldsymbol{P}}\sigma_{\boldsymbol{P}^c}} \sum_{\boldsymbol{q}\in\boldsymbol{P}}(I_1(\boldsymbol{q}) - \bar{\boldsymbol{P}})(I_2(\boldsymbol{q}+\boldsymbol{w}(\boldsymbol{p})) - \bar{\boldsymbol{P}^c})$

* $H(r)$ and $H_c(r)$ refers to bits of Census signatures of patches $\boldsymbol{P}$ and $\boldsymbol{P}^c$, respectively. The function $H(r)$ is defined in eq.(5.1).
** $\bar{\boldsymbol{P}}$, $\bar{\boldsymbol{P}^c}$, $\sigma_{\boldsymbol{P}}$ and $\sigma_{\boldsymbol{P}^c}$ are the mean intensity values and standard deviations of $\boldsymbol{P}$ and $\boldsymbol{P}^c$ patches respectively.

TABLE 4.2 – Table : Similarity costs

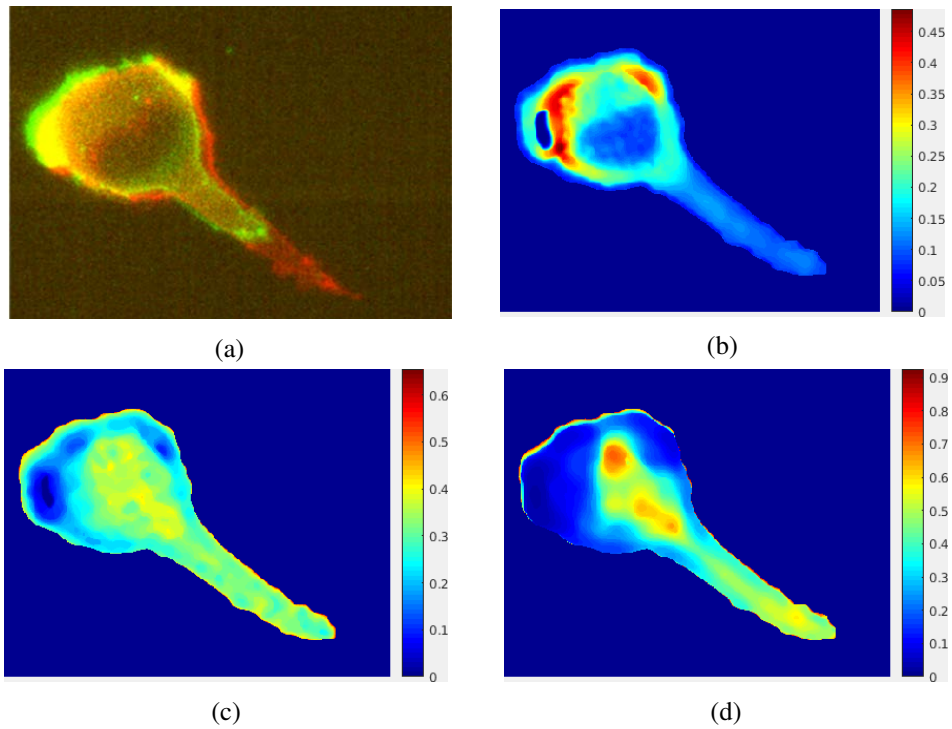


FIGURE 4.9 – (a) Overlapped slices of source (red) and target (green) volume pair in Sequence **2**. Residual matching cost using (b) SSD, (c) CS, (d) TZNCC based measure. The color code for the residual matching cost is given in the right side of the respective figures.

where $r_{\boldsymbol{q}} = I_1(\boldsymbol{p}) - I_2(\boldsymbol{q})$ and $\boldsymbol{q}$ belongs to the neighbourhood $\mathcal{N}(\boldsymbol{p})$ of $\boldsymbol{p}$. Formulation of the tested similarity measures is given in Table 4.2. For comparison purpose, we embed the three appearance-based criteria within our 3D PatchMatch framework. We apply them to 3D volumes depicting mid-range to long-range motion. The method produces semi-dense correspondences between a pair of volumes.

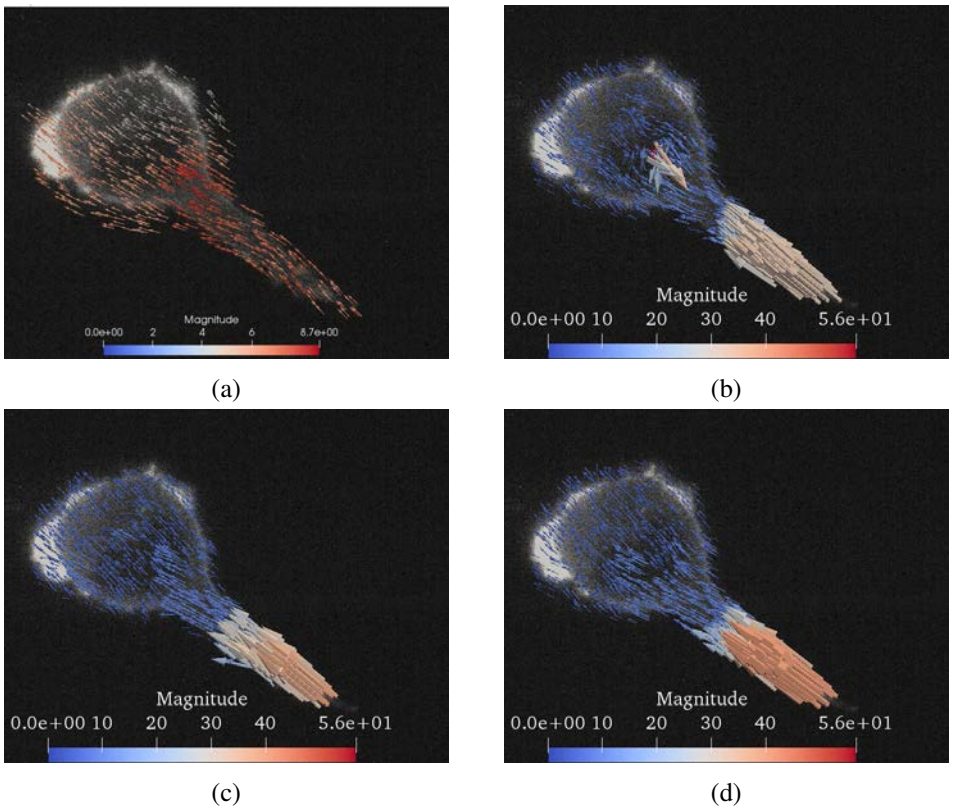


FIGURE 4.10 – 3D velocity vector estimated using (a) Amat's method, and 3D PatchMatch with (b) SSD, (c) CS, and (d) TZNCC similarity measures respectively. The color code for the magnitude of the motion is given in the bottom of the respective figures. The length of the arrow also denotes the amplitude of the motion.

4.2.3.3 Residual Matching Cost

We define the residual matching cost as the similarity cost estimated for the matched pair in the final iteration of 3D PatchMatch. We use this cost to compare the three similarity measures introduced in Section 4.2.3.2. The cost is of importance as the flow field estimated by 3D PatchMatch is due to the propagation of displacements precisely having low residual matching cost. Trustworthy measurements are expected in the (fluorescence) labeled regions which are the ones of key interest for biologists. Residual matching cost helps us evaluate matches in specific regions.

Figure 4.9 illustrates the evaluation. The lower the similarity cost, the better the confidence. The ranges of matching cost for TZNCC is $[0, 2]$, SSD is $[0, \infty)$ and CS is $[0, |\boldsymbol{P}|]$. Even though the costs are not directly comparable, their juxtaposition in Fig.4.9 reveals how confidently the final matches were made when estimating the 3D motion depicted in Fig. 4.10. The raw data display the labeled region with high intensity and the unlabeled cytosol as noisy interior. TZNCC and CS exhibit good confidence in the labeled region in contrast to the cytosolic region.

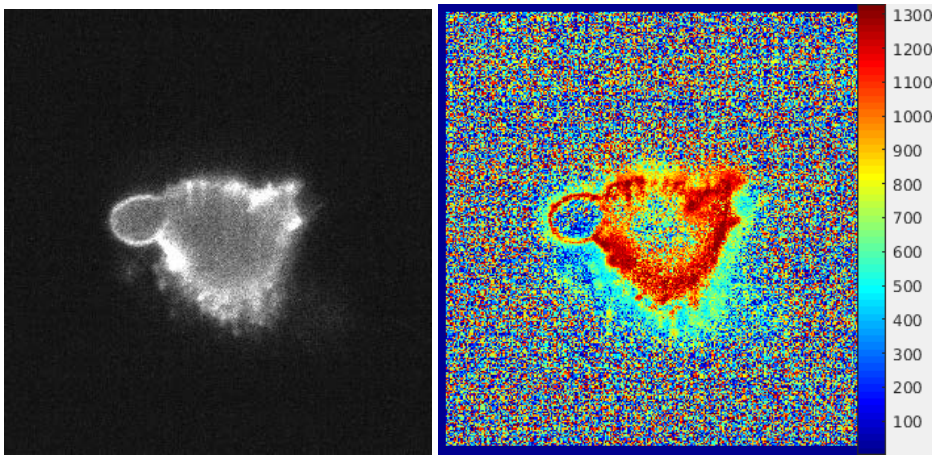


FIGURE 4.11 – Left : A raw slice of a frame in sequence **2**. Right : The rank transform. The color code for the rank of patch is given in the right side.

Surprisingly, SSD is assertive in unlabeled cytosolic region, and unreliable in the labeled region. This means that the displacements computed using SSD are propagated from unlabeled region rather than trustworthy labeled region. Figures 4.10 displays 3D flow fields computed using the Amat's method, and our 3D PatchMatch using SSD, CS and TZNCC measures. Clearly, Amat's method fails to capture large displacement of the protrusion. Our method proves to be effective here. Further, the flow fields computed using CS and TZNCC are less noisy than that of SSD.

Robustness of TZNCC comes from the normalization of the intensities with the average and local standard deviation. Robustness of CS comes from the composition of the Census signature which captures the underlying structure of the patch by the arrangement of bits representing the binarized directional derivatives of the intensity. Granularity of the intensity distribution of patches from these regions can be visualized by the rank transform (Zabih and Woodfill 1994). Rank of a voxel is the total number of voxels in its neighbourhood which have smaller intensity values than itself. Figure 4.11 illustrates that the labeled regions have high and uniform rank, whereas the background and unlabeled region inside the cell have non-uniform rank distribution. The uniformity and the high rank in patches of labeled region suggest less number of unexpected bit flips when comparing two labeled patches, and hence, smaller Hamming distances in these region. Consequently, the propagation is driven by the displacements in trustworthy labeled region rather than those in the unlabeled regions. In conclusion, we found CS and TZNCC to be more discriminative than SSD. The computational complexity of TZNCC measure is quadratic with respect to the patch size, whereas that of CS measure is linear. Therefore, we choose CS over TZNCC to carry further with the variational refinement method in Chapter 5. Additionally, the overall confidence of CS appears to be more coherent within the cell than that of TZNCC.

In our experiments, the flow field computed with our coarse-to-fine 3D PatchMatch was

more informative than that computed using our super-pixel based PatchMatch. The smoothness and discontinuities were more apparent in the results produced by the latter one. The agnostic nature of the former to cellular structures makes it more generic than the previous one. We are motivated to use the coarse-to-fine version in the pipeline that will be discussed in the next chapter.

4.3 Triangular Mesh Representation

So far, we have only used the intensity information of a 3D volumetric image and not the geometric features. Now, we aim to investigate one of such geometric features, namely Heat Kernel Signature (HKS) which represent relevant feature to compute 3D correspondences. Driscoll *et al.* (2018) used mean curvature of 3D triangular mesh representing melanoma cell to detect different morphological structures in the cell surface. We use spectral descriptor, namely heat kernel sigature (HKS) which is robust to reconstruction noise. Further, this descriptor is known to be invariant to isometric transformation, to be affine invariant and stable (Sun *et al.* 2009). HKS has been used to generate correspondences between isometric shapes (Ovsjanikov *et al.* 2010). Dey *et al.* (2010) used it for multi-scale matching of non-rigid shapes. We performed experiments with HKS on 3D surface generated for blebbing of MV3 cell and investigated the possibility to generate correspondences from HKS feature.

4.3.1 Heat Kernel Signature

Heat Kernel Signature (HKS) was introduced by Sun *et al.* (2009) to describe geometrical characteristics of a point in a shape, based on the properties of the heat diffusion process on the shape. It captures the notion of dissipation of heat from a point to its neighbourhood over time. For a manifold $\boldsymbol{M}$, the heat kernel $\mathcal{K}_{\tilde{t}}$ can be written as :

$$\mathcal{K}_{\tilde{t}}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=0}^{N-1} e^{-\lambda_i \tilde{t}} \phi_i(\boldsymbol{x}) \phi_i(\boldsymbol{y}), \tag{4.6}$$

where λ_i and ϕ_i are the i^{th} eigenvalue and eigenvector of Laplace-Beltrami operator (Sorkine 2005) on $\boldsymbol{M}$. The heat kernel $\mathcal{K}_t(\boldsymbol{x}, \boldsymbol{y})$ is the amount of heat transfered at time $\tilde{t}$ from point $\boldsymbol{x}$ to point $\boldsymbol{y}$ in $\boldsymbol{M}$ when a unit heat is applied at $\boldsymbol{x}$. $\tilde{t}$ is referred to as the heat diffusion time and not to be confused with t which denotes the time of the volume sequence. $\mathcal{K}_{\tilde{t}}(\boldsymbol{x}, \boldsymbol{y})$ was termed "Heat affinity" by Yi *et al.* (2011).

HKS, on the other hand, is defined over temporal domain as :

$$\text{HKS}(\boldsymbol{x}) : \mathbb{R}^+ \rightarrow \mathbb{R}, \text{HKS}(\boldsymbol{x}, \tilde{t}) = \mathcal{K}_{\tilde{t}}(\boldsymbol{x}, \boldsymbol{x}) \tag{4.7}$$

4.3.2 Discrete Mesh Laplacian

In order to obtain the eigen-components, we build a discrete mesh Laplacian (Sorkine 2006) of the 3D surface f. We write the graph of the mesh as $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{F})$, where $\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{F}$ are the set of vertices, edges and faces in the mesh. Let N be the number of vertices.

We have chosen cotangent-weight Laplacian (Wardetzky *et al.* 2007) as the discrete approximation of the Laplace-Beltrami operator defined as :

$$\Delta f_i = n_i \sum_j (\cot(\alpha_j) + \cot(\beta_j))(f_i - f_j), \tag{4.8}$$

where n_i normalizes the cotangent weights, j denotes the 1-ring neighbours of vertex i, and α and β are the two angles opposite to the edge ij as shown in Fig. 4.12. For computation, we

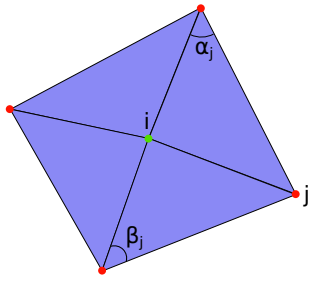


FIGURE 4.12 – 1-ring neighbourhood of vertex i

turn 4.8 into matrix $\boldsymbol{L}$ as follows :

$$L(i,j) = \begin{cases} -1, & \text{if } i = j \\ n_i(\cot(\alpha_j) + \cot(\beta_j)), & \text{if } ij \in E \\ 0, & \text{otherwise} \end{cases}$$

By construction, the matrix $\boldsymbol{L}$ is a $N \times N$ symmetric, semi-positive definite matrix. Then, the eigen-components, λ_i and ϕ_i that we are looking for are well behaved : the lowest eigenvalue $\lambda_0 = 0$ and $\lambda_N \geq \lambda_{N-1} \ldots \lambda_2 \geq \lambda_1 > 0$, while ϕ_i are real-valued functions. The complexity of eigen-decomposition increases with the size of vertices. For a mesh with large size, the eigen-decomposition could get computationally expensive. Sorkine (2006) showed that the lower spectrum of the eigenvectors, while less oscillating in nature, contain low-level details, and she used it for various mesh processing applications.

4.3.3 Heat Kernel Signature of MV3 cell surface

We applied this representation to real LM volumes to evaluate its relevance. We used the blebbing sequence of MV3 cell expressing EGFP-Tractin, which binds to actin filament. The

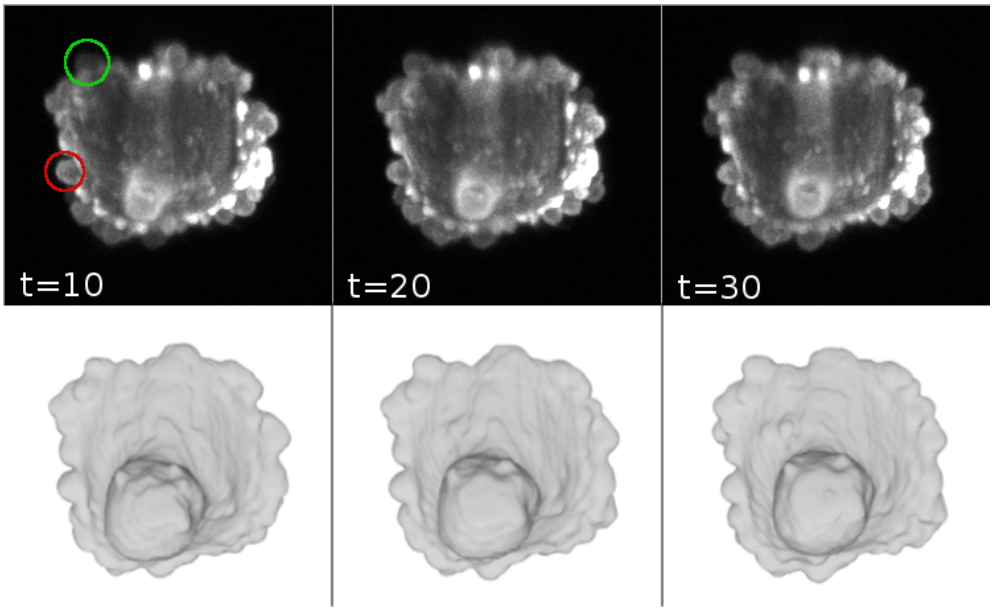


FIGURE 4.13 – First row : MIP in xy plane of the data at time t = 10,20 and 30. Second row : Triangular mesh reconstructed using the marching cube algorithm. The green and red circle highlights growing and shrinking blebs, respectively.

sequence was acquired by DiaSLM at around 3 Hz of acquisition rate. Let us recap that the life cycle of a bleb is characterized by its rapid expansion and slow retraction. The expanding bleb and the retracting bleb are circled in green and red, respectively in Fig. 4.13. The other brighter regions pertain to significantly retracted blebs where actin filaments have almost been restored.

Figure 4.13 shows three time frames of the sequence at different interval, and its mesh reconstruction using marching cube algorithm (Lorensen and Cline 1987) provided in *Fiji* software. Practically, we compute the HKS on the mesh surface of the MV3 cell. The 3D model has over 18000 vertices, which varies between frames as the cell undergoes complex deformation. Thus, it can have variable surface area between frames. Therefore, the mesh sequences are not temporally coherent. We computed the Laplacian matrix $\boldsymbol{L}$ for each meshes. We used MATLAB's implementation of Arnoldi-Lanczos iterative algorithm (Saad 2003) to compute few low-order eigen-components of $\boldsymbol{L}$, which has been suggested in the work of Sorkine (2006).

In Table 4.3, the 2^{nd} and the 3^{rd} columns display the HKS$(\boldsymbol{x}, \tilde{t})$ using *JET* (see Fig. 4.15) color mapped to corresponding vertices for the frames 1 and 2 of the MV3 cell, respectively. The hotter (reddish) the region is, the higher the value of HKS, and the colder (blueish) the region is, the smaller the value of HKS. We can see higher values of HKS near highly curved regions. It is also noticeable that local maxima are mostly seen in blebs across all heat diffusion time.

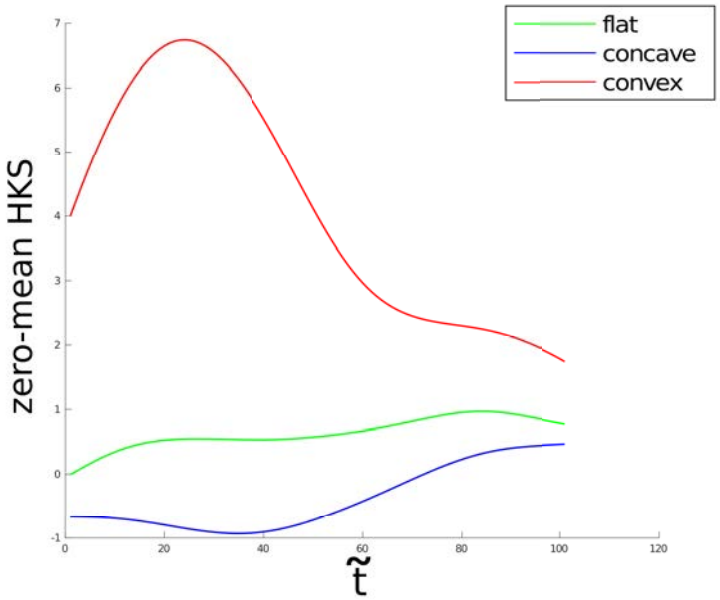


FIGURE 4.14 – zero-mean HKS vs heat diffusion time $\tilde{t}$

Comparing across the heat diffusion time, we observe more variations in smaller time scale than larger ones. To explain this, we put two things into consideration : (1) the low-pass nature of the term $e^{-\lambda_i \tilde{t}}$ in Eq. 4.7, and (2) the oscillating nature of ϕ_i. The weight $e^{-\lambda_i \tilde{t}}$ for higher eigen-components are significant when $\tilde{t}$ is small. Further, the oscillating behaviour of the higher order eigenvectors also causes fluctuating values of HKS across the surface at lower time scale.

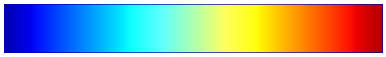


FIGURE 4.15 – Jet color code : Warmer color corresponds to higher values.

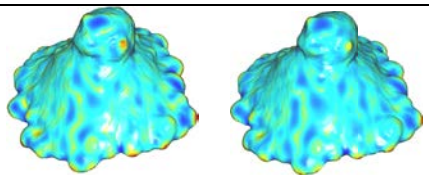
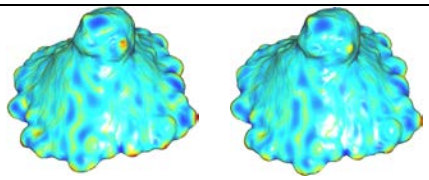
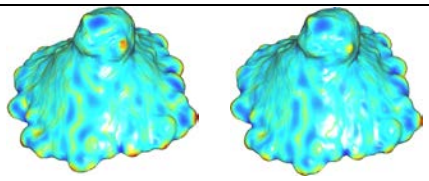
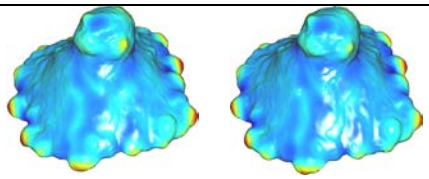
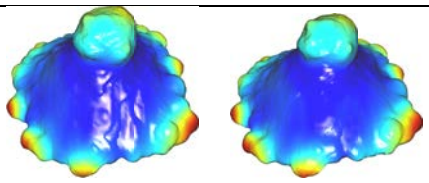
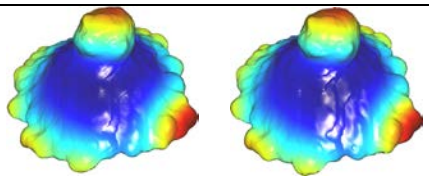
This experiment shows that the extrema of the HKS of the surfaces are quite consistent across the frames. Figure 4.14 shows the evolution of zero-mean HKS with time-scale corresponding to vertices in flat, concave and convex regions. The signatures of vertices in convex regions are significantly positive, while those of concave regions start off with negative value. The vertices in flat regions have small positive signatures.

In this study, we are interested in convex regions as blebs and other dynamic protrusions manifest as convex regions.We now intend to use them as features to compute possible correspondences. There are two choices to be made :

— The number of eigen-components : We use 500 components in our experiments.

— Choice of time steps : We use 150 time steps in logarithmic scale.

TABLE 4.3 – Rendering HKS as color

$\tilde{t}$	Frame 1	Frame2
0.05		
0.5		
5		
50		
200		
500		

Because of the $O(N^3)$ complexity of the eigen-decomposition, we do not perform the full eigen-decomposition. Instead, we rely of Arnoldi-Lanczos iterative algorithm to obtain first few-eigen components. It is to be noted that, the meshes in the sequence do not have same number of vertices. Hence, the size of $\boldsymbol{L}$ varies. The time to compute the eigen decomposition for each $\boldsymbol{L}$ may vary between meshes. Larger number of time steps is not necessary. First of all, it only increases the vector length. This results in a larger memory footprint. Secondly, from the Fig. 4.14 it can be seen that the value of the elements in HKS, after some period of time $\tilde{t}$, tends to be in equilibrium. Those values do not contribute to the features that are discriminating enough. In this manner, HKS for a vertex on the surface of the cell can be computed. We used

500 low-order eigen vectors of a discrete laplacian and 150 heat diffusion time steps.

The HKS is a vector of length defined by the number of heat diffusion time steps. We subtract from the elements of the vector, the mean value of the vector itself, to form the zero-mean HKS. We refer to the zero-mean HKS as HKS in the following subsection.

4.3.3.1 Matching Blebs in 3D+t Mesh Sequence

Driscoll *et al.* (2018) used standard geometric features such as perimeter, mean curvature, shape diameter function (Shapira *et al.* 2007). A training dataset was prepared, where labeling of blebs and protrusions was done with the aid of experts. Support vector machine (SVM) (Hearst 1998) was used to train and classify different protrusion types. Regarding the training and the classification, our experiments falters as we do not have large dataset and expert's labels. However, we proceed to match the blebs which already appears as convex regions between time steps using nearest-neighbour search in the mesh coordinate system.

In this case, we first need to find the local maxima of bleb regions. For this, we follow the method proposed by Yi *et al.* (2011) while using the magnitude of HKS instead of mean of heat kernel. The method proceeds in two steps : (1) Finding the vertices with maximum local HKS ,and (2) segmentation of surface by clustering vertices around these maximum points. The method requires to input the total number of clusters beforehand. Following the observation of Li *et al.* (2004), Yi *et al.* (2011) counted the ascending series of λ, until a dramatic increment in the value is found. The technique generally works in the segmentation of objects like humans, chairs, quadrupled animals, where clusters are less in number and predictable. However, we did not find a distinct increment in λ that could be correlated to the segments of blebs observed. Therefore, we resorted to manual counting of the blebs.

Since, the convex region has highly positive HKS, we find a vertex $\boldsymbol{x}$ in highly curved region which has the highest HKS amplitude. We label it as the first bleb center. Then we find the vertex $\boldsymbol{y}$ with the least heat affinity to $\boldsymbol{x}$, according to Eq. 4.6. We label $\boldsymbol{y}$ as the second bleb center. Excluding the labeled vertices, we find a new vertex with least heat affinity to the bleb centers and having the highest HKS amplitude. Here, the heat affinity is exploited to get geodesic distances (Sun *et al.* 2009) to make sure that the farthest possible vertices from the current center are being tested. The algorithm 1 describes the procedure.

After the bleb centers are obtained, we segment the bleb regions using K-Means clustering. We initialized the seeds with the bleb centers. Using the HKS feature vector for each vertices, we compute the segments.

Algorithm 1 Finding vertices with max local HKS

mesh **G**, eigen-components (λ, ϕ), Number of cluster N_c
Output array of vertices with maximum HKS C

procedure MAXIMUM HKS POINT
  $\boldsymbol{x} \leftarrow \underset{\boldsymbol{x} \in \boldsymbol{G}}{argmax} |HKS(\boldsymbol{x}, \tilde{t})|$ ▷ $\boldsymbol{x}$ is the vertex with maximum HKS amplitude
  $C \leftarrow C \cup \boldsymbol{x}$ ▷ add the current maxima to a list
  for `j=1 to N` **do** ▷ for each vertex in $\boldsymbol{G}$
    $affninity(\boldsymbol{y}_j) \leftarrow \mathcal{K}(\boldsymbol{x}, \boldsymbol{y}_j)$ ▷ Prepare array affinity(.) by computing heat affinity to $\boldsymbol{x}$ for each vertex $\boldsymbol{y}_j$, with Eq. 4.6
    $label(j) \leftarrow 1$ ▷ label the vertex with cluster index
  for `i=2 to` N_c **do** ▷ for each remaining bleb maxima
    $\boldsymbol{x} \leftarrow \underset{\boldsymbol{x}' \in \boldsymbol{G}}{argmin}(affninity(\boldsymbol{x}'))$ ▷ $\boldsymbol{x}'$ is the vertex with least affinity to $\boldsymbol{x}$
    for `j=1 to N` **do**
      if $\mathcal{K}(\boldsymbol{x}, \boldsymbol{y}_j) \geq affinity(\boldsymbol{y}_j)$ **then** ▷ In case the vertex is more affine to the new possible center
        $affinity(\boldsymbol{y}_j) \leftarrow \mathcal{K}(\boldsymbol{x}, \boldsymbol{y}_j)$
        $label(j) \leftarrow i$ ▷ label the vertex with new cluster index
        $\tilde{\boldsymbol{x}} = \{\boldsymbol{y}_i\}$ ▷ Prepare a set of vertices with only the current cluster label
        $\boldsymbol{x} \leftarrow \underset{\boldsymbol{x} \in \tilde{\boldsymbol{x}}}{argmax} |HKS(\boldsymbol{x}, \tilde{t})|$ ▷ a new possible center with max HKS
    $C \leftarrow C \cup \boldsymbol{x}$ ▷ add the current maxima to the list
    for `j=1 to N` **do** ▷ for each vertex in $\boldsymbol{G}$
      $affninity(\boldsymbol{y}_j) \leftarrow \mathcal{K}(\boldsymbol{x}, \boldsymbol{y}_j)$
  return C

4.3.3.2 Experimental Results

To match blebs between the volumes, we first need to define the number of blebs we want to detect. It is difficult to discern the exact number of dynamica blebs in the cell surface. In sequence MV3 B, we observed around 12 protrusions to be consistent throughout the reconstructed meshes. With 4 additional clusters to cope with new protrusions in the sequence which includes the cell body itself, we decided to compute 16 segments. Here, a cluster represents vertices of a bleb. The montage in Fig. 4.16 presents a sequence of 16 meshes, where each matched bleb is assigned to a color.

Figure 4.16 depicts that the persistent protrusions are matched throughout the time frames. The submeshes of the matched blebs are colored with the same color. We now focus on two interesting blebs highlighted by the green (Bleb A) and red (Bleb B) circles. The Bleb A exhibits protrusion and retraction within the time frames considered. The Bleb B undergoes retraction only.

The centroids of the blebs are computed for each matched segment. Figure 4.17a plots the track of the centroids of the Bleb A. The plot conveys that the onset of protrusion is drastic,

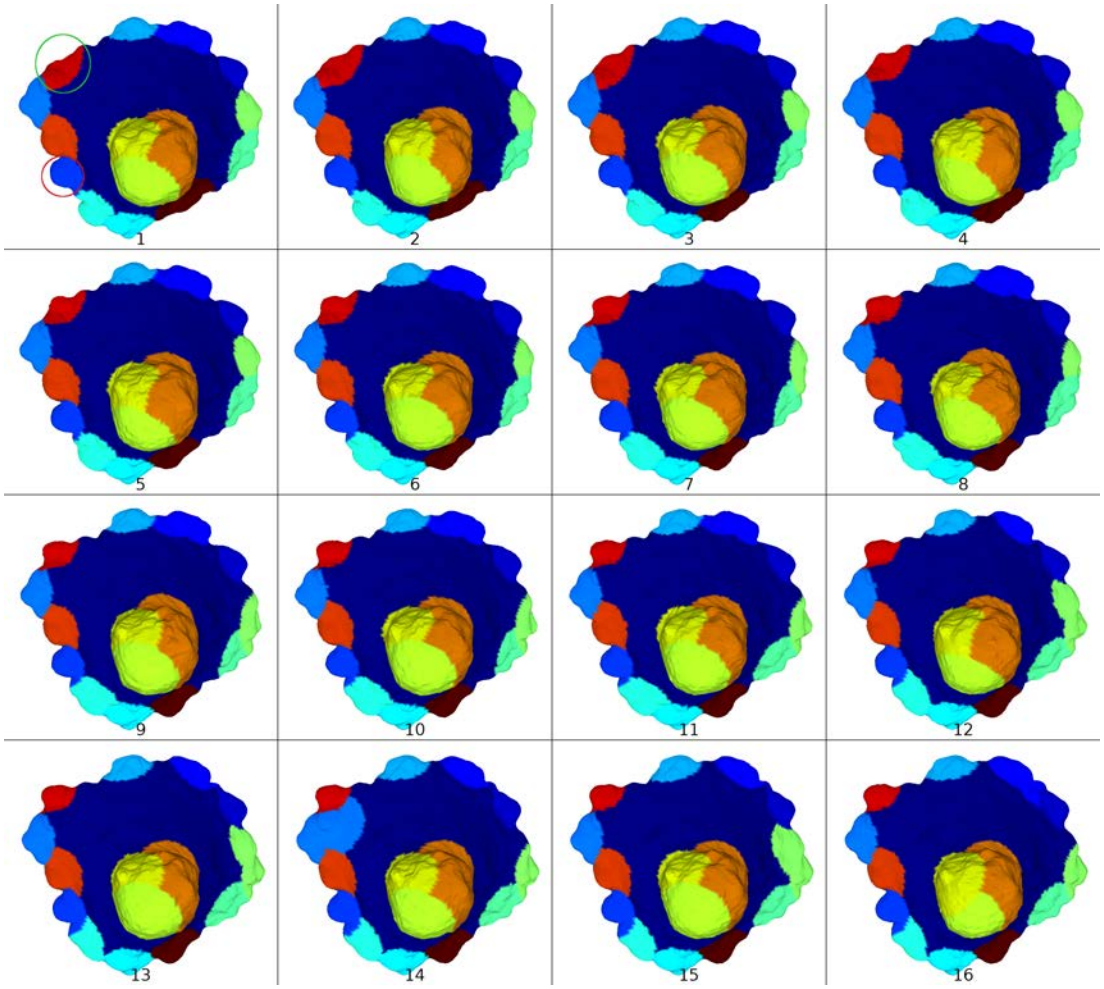


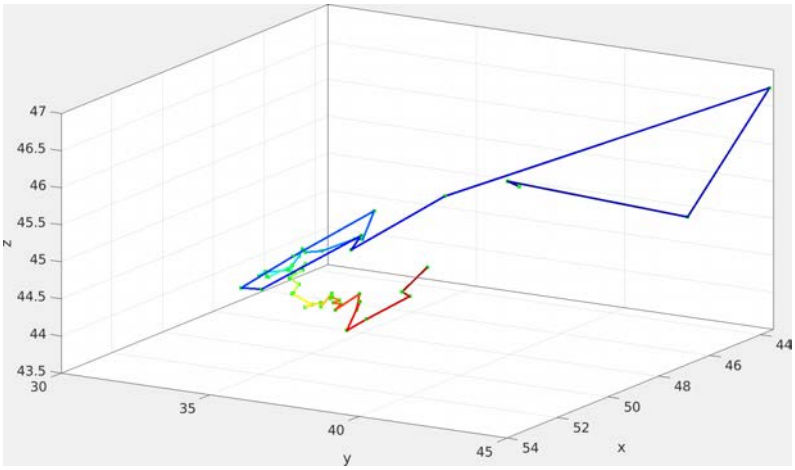
FIGURE 4.16 – Matched segments of blebs in MV3 cell mesh sequence. Each color represents a matched region. Displaying only the first 16 frames.

while the retraction is slow. Figure 4.18a shows the amplitude of the displacements of the centroids between two consecutive time frames. The retraction phase starts at around the 15^{th} frame, and has motion amplitudes less than a unit displacement per 0.34 seconds.

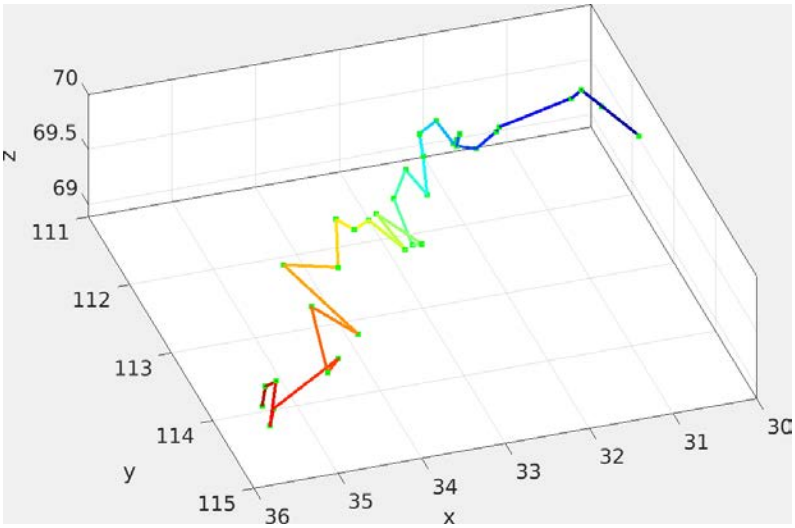
Figure 4.17b depicts the track of the centroids of Bleb B. At this observed state, Bleb B had already reached its maximum size. The slow retraction is evident from the depicted tracks. Figure 4.18b depicts the motion amplitude of centroids between two consecutive time frames. As in Bleb A, here the retraction is also characterized by small displacements.

4.4 Conclusion

We have developed three different methods for matching in 3D volumes acquired by light microscopy. The first method segments the volume slices to super-pixel segments, and matches



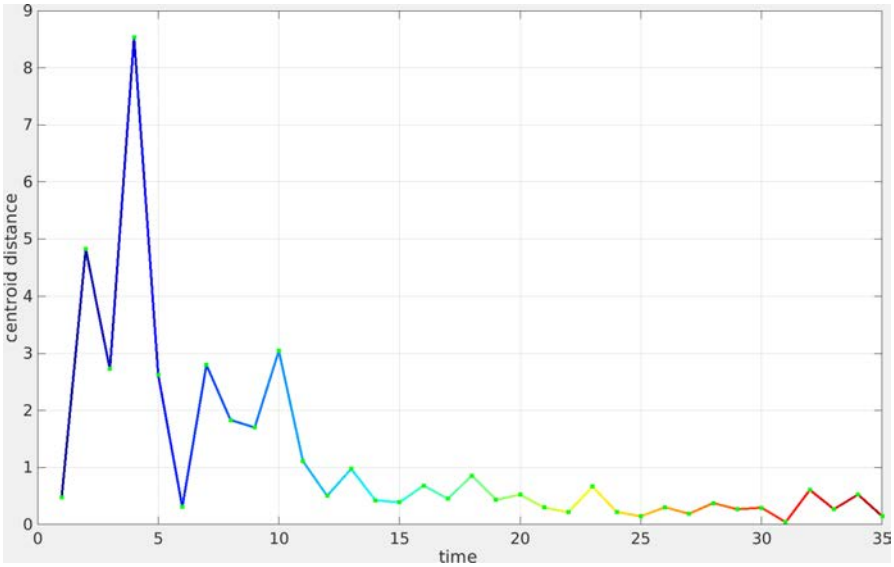
(a) Track of the centroids of the bleb that protrudes and retracts.



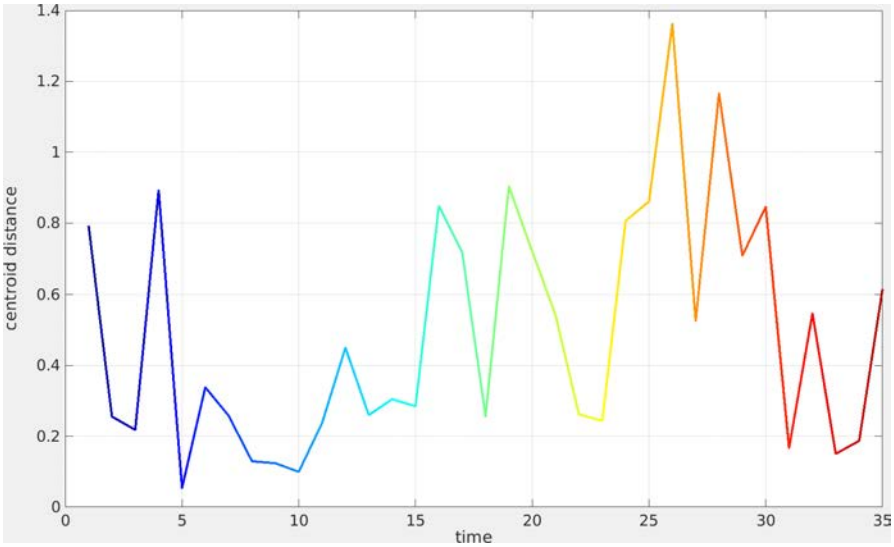
(b) Track of the centroids of the bleb that retracts only.

FIGURE 4.17 – Tracks of the centroids across time of (a) Bleb A, (b) Bleb B. The color of the line represents the time index, coded in *jet* color code. The blue color represents the start and the red color represents the end of the sequence.

them with our extension of 3D PatchMatch. The method performed well when applied to synthesized sequences. However, there are real sequences where cellular structures do not allow for meaningful super-pixel segmentation. This motivated us to develop coarse-to-fine 3D PatchMatch, which performs matching for each voxel. We tested three different matching cost and found Census signature (CS) to be well suited. CS is then considered in variational approach which will be discussed in the next chapter. Alongside the matching in image-based representation, we also explored matching in mesh-based representation. We investigated HKS and found it to be well-suited for protrusion, and bleb detection and tracking. We went on to match the bleb regions between time frames. We found that the centroid of a bleb has around



(a) Distance between the centroids of the Bleb A between two successive frames.



(b) Distance between the centroids of the Bleb B between two successive frames.

FIGURE 4.18 – Distance between the centroids of blebs between two successive frames.

unit motion magnitude in the retraction phase. Our method of bleb segmentation and matching could lead to other interesting applications like defining region of interests or tracking cellular morphologies. However, this approach is mainly dedicated to analyze surface deformation. Consequently it cannot be combined with optical flow evaluation where dense points have to be matched between volumes of arbitrary morphology.

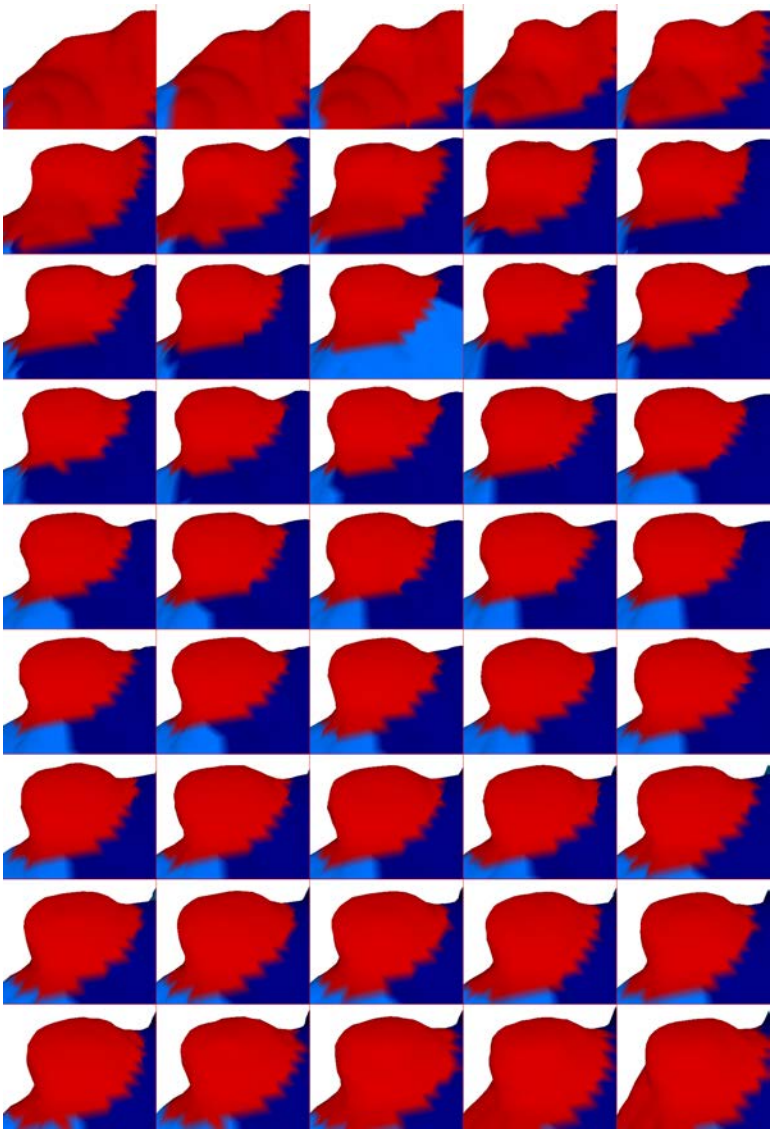


FIGURE 4.19 – Protrusion and retraction of a bleb. The frames succeeds from left to right and top to bottom.

Chapitre 5

Variational Method

5.1 Introduction

In this chapter, we present a 3D variational optical flow method, which exploits the temporal of Census signature (CS). First, we define the continuous approximation of CS following Hafner *et al.* (2013) and extend the formulation to 3D images. The continuous formulation facilitates the linear approximation of CS in the continuous setting. We propose a 3D coarse-to-fine variational optical flow method based on CS constancy assumption. We test our method qualitatively and quantitatively, and compare the results to those obtained with Amat's method and a version of 3D Horn-and-Schunck method. For quantitative evaluation, we have proposed a novel measure called SAE (Structural Angular Error) based on the local structure tensor and the backward warping.

5.2 Census Constancy Assumption

We recall that a bit of CS is a binarized form of discrete image derivative, given as :

$$H(r_{\boldsymbol{q}}) = \begin{cases} 0, & \text{if } r_{\boldsymbol{q}} \geq 0, \\ 1 & \text{otherwise,} \end{cases} \tag{5.1}$$

where $r_{\boldsymbol{q}} = I_1(\boldsymbol{p}) - I_2(\boldsymbol{q})$ and $\boldsymbol{q}$ belongs to the neighbourhood $\mathcal{N}(\boldsymbol{p})$ of $\boldsymbol{p}$. To form the CS of the voxel at $\boldsymbol{p}$, all the voxels at $\boldsymbol{q}$'s in its neighbourhood must be considered, and their binarized derivatives have to be arranged in a particular order to form a vector of fixed length.

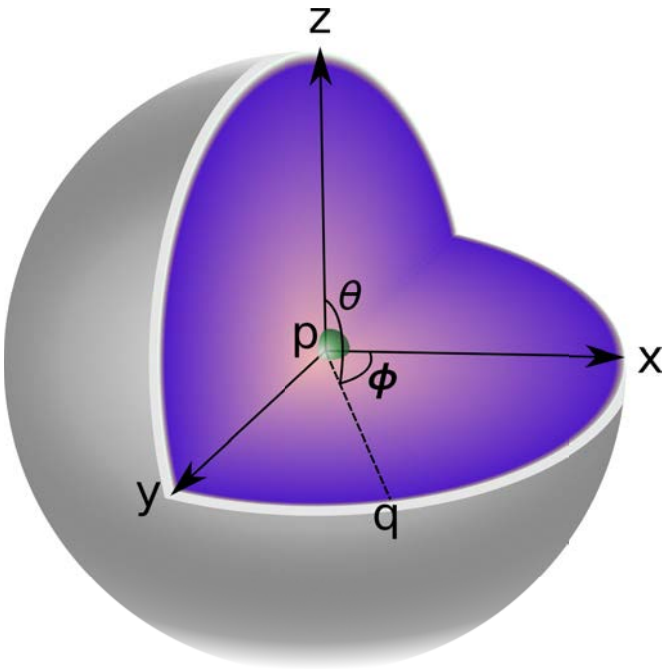


FIGURE 5.1 – Continuous Census signature : Here $\boldsymbol{p}$ is the central voxel. The Census bit is computed for $\boldsymbol{q}$ which lies at $(1, \theta, \phi)$, as specified by the directional derivative operator ∂_θ^ϕ. The Census elements are computed for all points in the unit sphere, and after arranging them in an order, the collection is referred to as the Census Signature of $\boldsymbol{p}$.

5.2.1 Census Transform in the continuous setting

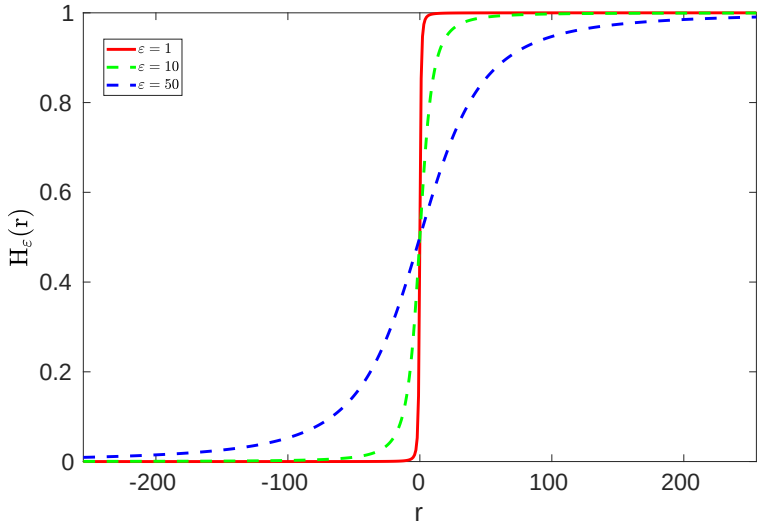
In the continuous setting, following Hafner *et al.* (2013), instead of binary bits, we assign continuous values to the directional derivatives, which we call the "Census element". The step function $H(r_q)$ can be approximated as follows :

$$H_\epsilon(r_{\boldsymbol{q}}) = \frac{1}{2}\left(1 + \frac{r_{\boldsymbol{q}}}{\sqrt{r_{\boldsymbol{q}}^2 + \epsilon^2}}\right), \tag{5.2}$$

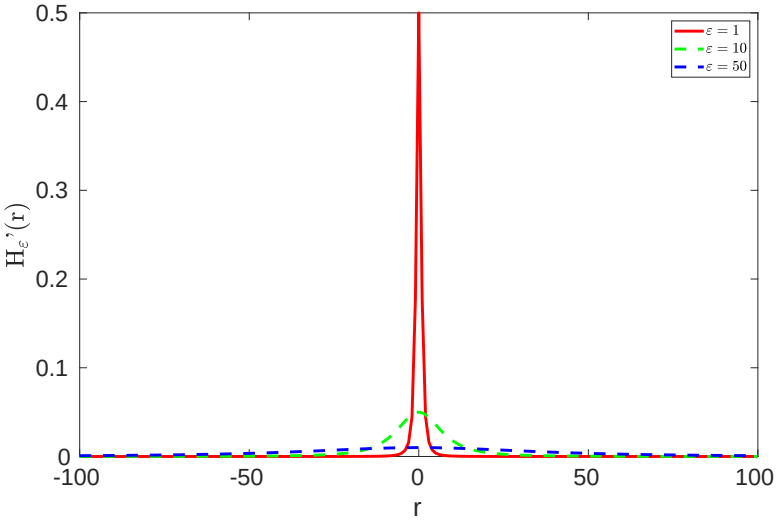
and the first-order derivative is given by :

$$H'_\epsilon(r_{\boldsymbol{q}}) = \frac{\epsilon^2}{2(r_{\boldsymbol{q}}^2 + \epsilon^2)^{3/2}}, \tag{5.3}$$

where ϵ is the step-function approximation factor. In what follows, $r_{\boldsymbol{q}} = \partial_\theta^\phi I(\boldsymbol{p})$, where ∂_θ^ϕ acts as a spatial directional derivative operator with direction specified by the spherical angles, (ϕ, θ), with $\phi \in [0, 2\pi)$ and $\theta \in [0, \pi]$, in a unit sphere around central voxel $\boldsymbol{p}$, as illustrated in Fig. 5.1. The 3D unit sphere, centered at $\boldsymbol{p}$ is denoted as $\mathbb{S}^2(\boldsymbol{p}, 1)$ which is a $n-1$-sphere (i.e., the sphere defined by the set of points precisely on the surface of the sphere, and not by the points inside of the volume). Now, any $\boldsymbol{q} \in \mathbb{S}^2(\boldsymbol{p}, 1)$ can be parameterized by $(1, \phi, \theta)$.



(a) Approximations of $H_\epsilon(r_{\boldsymbol{q}})$ with ϵ



(b) Corresponding derivatives $H'_\epsilon(r_{\boldsymbol{q}})$

FIGURE 5.2 – approximation of $H_\epsilon(r_{\boldsymbol{q}})$ for different values of ϵ.

In the continuous setting, the Census elements are computed for all $\boldsymbol{q} \in \mathbb{S}^2(\boldsymbol{p}, 1)$. This results in the continuous CS of infinite length. Practically, the values of ϕ and θ are discretized to form a CS of finite length. Figure 5.2 shows the approximations of $H_(r_{\boldsymbol{q}})$ as $H_\epsilon(r_{\boldsymbol{q}})$ and their respective derivatives $H'_\epsilon(r_q)$ for different values of ϵ .

To put the Census signature constancy assumption into perspective, we take a pair of corresponding points $(x, y, z, t)^\top$ and $(x + u, y + v, z + w, t + 1)^\top$ from two volume pairs, I_t and I_{t+1}. Let us denote $\boldsymbol{w} = (u, v, w, 1)^\top$ as the motion vector to be estimated between the two images. In short, we denote the space and time coordinates together as quadruplets. Let

$\boldsymbol{p} = (x, y, z, t)^\top$. According to the constancy assumption of the CS, we have for all pairs of (ϕ, θ),

$$H_\epsilon(\partial_\theta^\phi I_{t+1}(\boldsymbol{p} + \boldsymbol{w}(\boldsymbol{p}))) - H_\epsilon(\partial_\theta^\phi I_t(\boldsymbol{p})) \approx 0. \tag{5.4}$$

We linearize Eq. 5.4 in two stages. First, $\partial_\theta^\phi I(\boldsymbol{x} + \boldsymbol{w})$ is linearized with respect to $\boldsymbol{p}$ as

$$\partial_\theta^\phi I(\boldsymbol{p} + \boldsymbol{w}(\boldsymbol{p})) = \partial_\theta^\phi I(\boldsymbol{p}) + \boldsymbol{w}(\boldsymbol{p})^\top \nabla(\partial_\theta^\phi I(\boldsymbol{p})), \tag{5.5}$$

$$H_\epsilon(\partial_\theta^\phi I(\boldsymbol{p}) + \boldsymbol{w}(\boldsymbol{p})^\top \nabla(\partial_\theta^\phi I(\boldsymbol{p}))) - H_\epsilon(\partial_\theta^\phi I(\boldsymbol{p})) \approx 0, \tag{5.6}$$

with ∇ as a gradient operator. Second, we linearize H_ϵ with respect to $\partial_\theta^\phi I(\boldsymbol{p})$ using the first order Taylor series expansion as,

$$H_\epsilon(\partial_\theta^\phi I(\boldsymbol{p}) + \boldsymbol{w}(\boldsymbol{p})^\top \nabla(\partial_\theta^\phi I(\boldsymbol{p}))) = H_\epsilon(\partial_\theta^\phi I(\boldsymbol{p})) + H'_\epsilon(\partial_\theta^\phi I(\boldsymbol{p})).\boldsymbol{w}(\boldsymbol{p})^\top \nabla(\partial_\theta^\phi I(\boldsymbol{p}))). \tag{5.7}$$

From Eqs. 5.6 and 5.7, we can write :

$$H'_\epsilon(\partial_\theta^\phi I(\boldsymbol{p})).\boldsymbol{w}(\boldsymbol{p})^\top \nabla(\partial_\theta^\phi I(\boldsymbol{p}))) \approx 0. \tag{5.8}$$

The sum of the elements on the left-hand side of Eq. 5.8, for every neighbouring points at $(1, \phi, \theta)$ in $\mathbb{S}^2(\boldsymbol{p}, 1)$ approximates the Hamming distance in the continuous setting. Therefore, the data term based on Census constancy assumption using Eq. 5.8 can be written as :

$$D(I(\boldsymbol{p}), \boldsymbol{w}(\boldsymbol{p})) = \int_0^{2\pi} \int_0^{\pi} H'_\epsilon(\partial_\theta^\phi I(\boldsymbol{p})).(\boldsymbol{w}(\boldsymbol{p})^\top \nabla(\partial_\theta^\phi I(\boldsymbol{p})))d\theta d\phi. \tag{5.9}$$

We use L_2 norm for both the data and the regularization terms to make the solution tractable, yielding the following energy functional to be minimized :

$$E(\boldsymbol{w}) = \int_\Omega \|D(I, \boldsymbol{w})\|_2^2 + \alpha \|\nabla \boldsymbol{w}\|_2^2 \, d\boldsymbol{p}. \tag{5.10}$$

5.2.2 Euler Lagrange Equations

Applying the calculus of variation, the minimizer to the following energy function :

$$E(u, v, w) = \int_\Omega F(x, y, z, u, v, w, u_x, u_y, u_z, v_x, v_y, v_z, w_x, w_y, w_z)dxdydz \tag{5.11}$$

with respect to $\boldsymbol{w}$ can be found by solving the associated Euler Lagrange (E.L.) equations. The associated E.L equations of Eq. 5.11 are defined by the following partial differential equations (PDEs) :

$$F_u - \frac{\partial}{\partial x}F_{u_x} - \frac{\partial}{\partial y}F_{u_y} - \frac{\partial}{\partial z}F_{u_z} \approx 0, \tag{5.12}$$

$$F_v - \frac{\partial}{\partial x}F_{v_x} - \frac{\partial}{\partial y}F_{v_y} - \frac{\partial}{\partial z}F_{v_z} \approx 0, \tag{5.13}$$

$$F_w - \frac{\partial}{\partial x}F_{w_x} - \frac{\partial}{\partial y}F_{w_y} - \frac{\partial}{\partial z}F_{w_z} \approx 0, \tag{5.14}$$

where F is defined from Eq. 5.10 as :

$$F = \|D(I, \boldsymbol{w})\|_2^2 + \alpha \|\nabla \boldsymbol{w}\|_2^2, \tag{5.15}$$

For practical reason, we discretize the data-term by quantization of ϕ and θ angles, such that a fixed number of $\boldsymbol{q}$'s are chosen at fixed positions. Let n be the index of a discretized point $\boldsymbol{q}$, such that $\boldsymbol{q} \in \mathcal{N}(\boldsymbol{p})$. We choose a 26-connected neighbourhood, as it is a close approximation of a 3D sphere. In discrete setting, we use ∂_n instead of ∂_θ^ϕ. This allows us to write the discrete data term D as :

$$D(I(\boldsymbol{p}), \boldsymbol{w}(\boldsymbol{p})) = \frac{1}{N} \sum_{n=1}^{N} H'_\epsilon(\partial_n I(\boldsymbol{p})).(\boldsymbol{w}(\boldsymbol{p})^\top \nabla(\partial_n I(\boldsymbol{p})), \tag{5.16}$$

where $N = 26$. To solve the PDEs in Eqs. 5.12, 5.13, 5.14, we need to compute :

$$F_u = \frac{2}{N} \sum_{n=1}^{N} H'^2_\epsilon(\partial_n I(\boldsymbol{p})).\partial_n I_x(\boldsymbol{p}).(\boldsymbol{w}(\boldsymbol{p})^\top \nabla(\partial_n I(\boldsymbol{p}))^2, \tag{5.17}$$

$$F_v = \frac{2}{N} \sum_{n=1}^{N} H'^2_\epsilon(\partial_n I(\boldsymbol{p})).\partial_n I_y(\boldsymbol{p}).(\boldsymbol{w}(\boldsymbol{p})^\top \nabla(\partial_n I(\boldsymbol{p}))^2, \tag{5.18}$$

$$F_w = \frac{2}{N} \sum_{n=1}^{N} H'^2_\epsilon(\partial_n I(\boldsymbol{p})).\partial_n I_z(\boldsymbol{p}).(\boldsymbol{w}(\boldsymbol{p})^\top \nabla(\partial_n I(\boldsymbol{p}))^2, \tag{5.19}$$

and

$$F_{u_x} = 2\alpha u_x, \quad F_{u_y} = 2\alpha u_y, \quad F_{u_z} = 2\alpha u_z, \tag{5.20}$$

$$F_{v_x} = 2\alpha v_x, \quad F_{v_y} = 2\alpha v_y, \quad F_{v_z} = 2\alpha v_z, \tag{5.21}$$

$$F_{w_x} = 2\alpha w_x, \quad F_{w_y} = 2\alpha w_y, \quad F_{w_z} = 2\alpha w_z. \tag{5.22}$$

Putting all the terms together and rewriting the PDEs in discrete form, we get :

$$\sum_{\boldsymbol{p} \in \Omega} \left(\frac{1}{N} \sum_{n=1}^{N} H'^2_\epsilon(\partial_n I(\boldsymbol{p})).\partial_n I_x(\boldsymbol{p}).(\boldsymbol{w}(\boldsymbol{p})^\top \nabla_4(\partial_n I(\boldsymbol{p}))^2 - \alpha \underbrace{(u_{xx} + u_{yy} + u_{zz})}_{\Delta u} \right) = 0, \tag{5.23}$$

$$\sum_{\boldsymbol{p} \in \Omega} \left(\frac{1}{N} \sum_{n=1}^{N} H'^2_\epsilon(\partial_n I(\boldsymbol{p})).\partial_n I_y(\boldsymbol{p}).(\boldsymbol{w}(\boldsymbol{p})^\top \nabla_4(\partial_n I(\boldsymbol{p}))^2 - \alpha \underbrace{(v_{xx} + v_{yy} + v_{zz})}_{\Delta v} \right) = 0, \tag{5.24}$$

$$\sum_{\boldsymbol{p} \in \Omega} \left(\frac{1}{N} \sum_{n=1}^{N} H'^2_\epsilon(\partial_n I(\boldsymbol{p})).\partial_n I_z(\boldsymbol{p}).(\boldsymbol{w}(\boldsymbol{p})^\top \nabla_4(\partial_n I(\boldsymbol{p}))^2 - \alpha \underbrace{(w_{xx} + w_{yy} + w_{zz})}_{\Delta w} \right) = 0. \tag{5.25}$$

In order to compute the Laplacian of the three flow components, we take the usual finite element approximation. A uniform cuboid grid is drawn with the unit cell of side lengths h_x,

h_y and h_z in x, y, z axes, respectively. For a point at position (x, y, z), we compute :

$$\begin{aligned}\Delta u = &\frac{u(x+1,y,z)-u(x,y,z)}{hx^2}+\frac{u(x-1,y,z)-u(x,y,z)}{hx^2}+\\ &\frac{u(x,y+1,z)-u(x,y,z)}{hy^2}+\frac{u(x,y-1,z)-u(x,y,z)}{hy^2}+\\ &\frac{u(x,y,z+1)-u(x,y,z)}{hz^2}+\frac{u(x,y,z-1)-u(x,y,z)}{hz^2},\end{aligned} \tag{5.26}$$

5.2.2.1 Numerical Solver

To solve Eqs. 5.23-5.25, we use a fixed-point algorithm with Successive Over-Relaxation (SOR) technique (David M. Young 1971). This leads to nested iterations with two loops : (1) outer loop of warping to remove implicit non linearity, (2) solver loop for fixed point problem. To derive the equations for SOR iterations, we denote the first term in Eq. 5.23 as :

$$A_u(\boldsymbol{p}) = \frac{1}{N}\sum_{n=1}^{N} H_\epsilon'^2(\partial_n I(\boldsymbol{p})).\partial_n I_x(\boldsymbol{p}).(\boldsymbol{w}^\top \nabla_4(\partial_n I(\boldsymbol{p}))^2. \tag{5.27}$$

Similarly, the first terms in Eqs. 5.24 and 5.25 are denoted as A_v and A_w. For further simplification of Eq. 5.27, we write :

$$B_u^n(\boldsymbol{p}) = H_\epsilon'^2(\partial_n I(\boldsymbol{p})).\partial_n I_x(\boldsymbol{p}). \tag{5.28}$$

The Laplacian terms are written as :

$$S_u(\boldsymbol{p}) = \sum_l \left(\sum_{\boldsymbol{q}\in\mathcal{N}_l^-(\boldsymbol{p})} u^{k+1}(\boldsymbol{q}) + \sum_{\boldsymbol{q}\in\mathcal{N}_l^+(\boldsymbol{p})} u^k(\boldsymbol{q}) \right), \tag{5.29}$$

$$S_v(\boldsymbol{p}) = \sum_l \left(\sum_{\boldsymbol{q}\in\mathcal{N}_l^-(\boldsymbol{p})} v^{k+1}(\boldsymbol{q}) + \sum_{\boldsymbol{q}\in\mathcal{N}_l^+(\boldsymbol{p})} v^k(\boldsymbol{q}) \right), \tag{5.30}$$

$$S_w(\boldsymbol{p}) = \sum_l \left(\sum_{\boldsymbol{q}\in\mathcal{N}_l^-(\boldsymbol{p})} w^{k+1}(\boldsymbol{q}) + \sum_{\boldsymbol{q}\in\mathcal{N}_l^+(\boldsymbol{p})} w^k(\boldsymbol{q}) \right), \tag{5.31}$$

where, $\mathcal{N}_l^-(\boldsymbol{p})$ and $\mathcal{N}_l^+(\boldsymbol{p})$ are causal and non causal neighbourhood of $\boldsymbol{p}$ in axis l (here we consider x, y and z axes), while considering a raster scan order, and k is the number of solver iterations. The iterative solution for u (in similar manner for v, w) is given as :

$$u(\boldsymbol{p})^{k+1} = (1-\omega)u(\boldsymbol{p})^{k+1} - \omega \frac{\frac{1}{N}\sum_{n=0}^{N-1} B_u^n(\boldsymbol{p}) \left(v^k(\boldsymbol{p})\frac{\partial}{\partial y}\partial_n I(\boldsymbol{p}) + w^k(\boldsymbol{p})\frac{\partial}{\partial z}\partial_n I(\boldsymbol{p}) + \frac{\partial}{\partial t}\partial_n I(\boldsymbol{p}) \right) + \alpha S_u(\boldsymbol{p})}{\frac{2}{N}\sum_{n=0}^{N-1} B_u^n(\boldsymbol{p})\frac{\partial}{\partial x}\partial_n I(\boldsymbol{p})}, \tag{5.32}$$

where $\omega \in [1, 2]$ is the relaxation parameter. Using the updated scalar field of u, we solve for the scalar fields v and then w.

5.2.3 CS approximation factor

CS approximation factor ϵ controls the response of the H_ϵ, given the image gradients. A very small value of ϵ causes noisy gradients to significant contribute to CS. Thus, the linearization of the data term becomes invalid. A large value of ϵ does not serve the purpose as the step function. Therefore, ϵ should be sufficiently greater than 0. We experimentally define the range of values for ϵ in the following section.

5.3 Pyramidal approach/Coarse-to-Fine strategy

The variational methods are essentially valid for small displacements. To address large displacements, one of the approaches is to apply in coarse-to-fine (CTF) strategy. For this a pyramid of volumes with decreasing spatial resolution is built. At the smallest resolution, or the coarsest level, the displacements become smaller, and detectable by the variational approach. The flow field is upsampled and upscaled by the factor of resolution ratio in the next finer level. At each level, the flow field is used to warp the target volume, which brings the volume (the source and the new target) pair closer and increases the convergence rate.

Digital warping for images is described in Wolberg (1994) with two schemes : forward and backward warping schemes. Forward warping, because it requires copying each input pixel from the source image to output positions in the target image with arbitrary displacements, the spatial transformation becomes inappropriate. Holes and overlaps may occur in the warped image. As a remedy, it has been suggested to use Coon patches (Coons 1967) which represents a pixel as a square patch that can be transformed into an arbitrary quadrilateral in the warped image.

Instead of this costly approach, a more simpler approach of backward warping is widely accepted in the optical flow community. With backward warping, each output coordinate is mapped to input image with an inverse map. The discrete location of output image is mapped to real valued position in input image. This requires interpolation which could be bi-cubic or linear interpolation for 2D images. Tri-linear interpolation can be used for 3D images. We proceed with backward warping.

Algorithm 3 describes the procedure of constructing the volume pyramid, computing flow field in each pyramid level using Algorithm 2, and warping the computed flow field in the successive levels.

Algorithm 2 Compute single scale optical flow

Input I_1 = Source volume , I_2 = Target volume, $\boldsymbol{w}$ = Initial flow field,
α = smoothness parameter, ϵ = CS approximation factor, η = stopping threshold,
N_{WARP} = number of warpings (Outer loop) , N_{SOR} = maximum number of solver iterations (Solver loop).
Output $\boldsymbol{w}$ = Updated flow field.

procedure $SOR\_CSL2(I_1, I_2, \boldsymbol{w}, \alpha, \epsilon, \eta, N_{warps}, N_{SOR\_iter})$
  Compute directional derivatives ∂I_{1_n} for source volume.
  for `n=1 to` N_{WARP} **do**
    Compute $I_2(\boldsymbol{p}+\boldsymbol{w}(\boldsymbol{p}))$, and its image gradients $I_{2x}(\boldsymbol{p}+\boldsymbol{w}(\boldsymbol{p}))$, $I_{2y}(\boldsymbol{p}+\boldsymbol{w}(\boldsymbol{p}))$, $I_{2z}(\boldsymbol{p}+\boldsymbol{w}(\boldsymbol{p}))$.
    Compute $\partial_n I_2, \partial_n I_{2x}, \partial_n I_{2y}, \partial_n I_{2z}$
    Compute $H'_\epsilon(\partial_n I_{2_n})$ using Eq. 5.2.
    $\boldsymbol{w}^n \leftarrow \boldsymbol{w}$.
    Reset $\boldsymbol{w}$ to zero displacement.
    $k \leftarrow 0$
    while $k < N_{SOR}$ `and stop_criterion >` η **do** ▷ Fixed point iteration
      Solve Eq. 5.32 and the respective Eqs. for v and w.
      $stop\_criterion \leftarrow \left\|\boldsymbol{w}^k - \boldsymbol{w}^{k+1}\right\|_2$
      $k \leftarrow k + 1$
    $\boldsymbol{w} \leftarrow \boldsymbol{w}^n + \boldsymbol{w}$ ▷ Increment flow field
  return $\boldsymbol{w}$

Algorithm 3 Compute CTF opical flow

Input I_1 = Source volume , I_2 = Target volume, α = smoothness parameter, f = downscaling factor, ϵ = CS approximation factor, η = stopping threshold , N_s = CTF levels, N_{warps} = number of warpings (Outer loop) , N_{SOR_iter} = maximum number of solver iterations (Solver loop).
Output $\boldsymbol{w}^s$ = Updated flow field.

procedure $multiscaleOF(I_1, I_2, \boldsymbol{w}, \alpha, \epsilon, \eta, N_{WARP}, N_{SOR})$
  Normalize I_1, I_2 between 0 and 255. Initialize flow fields with zero displacements.
  for `s=1 to` N_s **do**
    Downscale I_1, I_2 by f as I_1^s, I_2^s, respectively.
  Initialize flow field $\boldsymbol{w}$ with 0
  for $s = N_s - 1$ `to` 0 **do** ▷ Decrement from Coarse to Fine
    $SOR\_CSL2(I_1^s, I_2^s, \boldsymbol{w}^s, \alpha, \epsilon, \eta, N_{WARP}, N_{SOR})$ ▷ Algorithm 2
    if s > 0 **then**
      $\boldsymbol{w}^{s-1}(\boldsymbol{p}) \leftarrow \frac{1}{f}\boldsymbol{w}^s(f\boldsymbol{p})$
  return $\boldsymbol{w}^s$

TABLE 5.1 – Values or range of values for the parameters of our method.

Parameter	Description	Value
ϵ	CS approximation factor	20-80
α	smoothness factor	0.0001-6
N_s	number of pyramid levels	1-3
N_{warps}	number of warpings per level	5-10
N_{SOR_iter}	number of maximum SOR iterations	50
f	downsampling factor	0.5

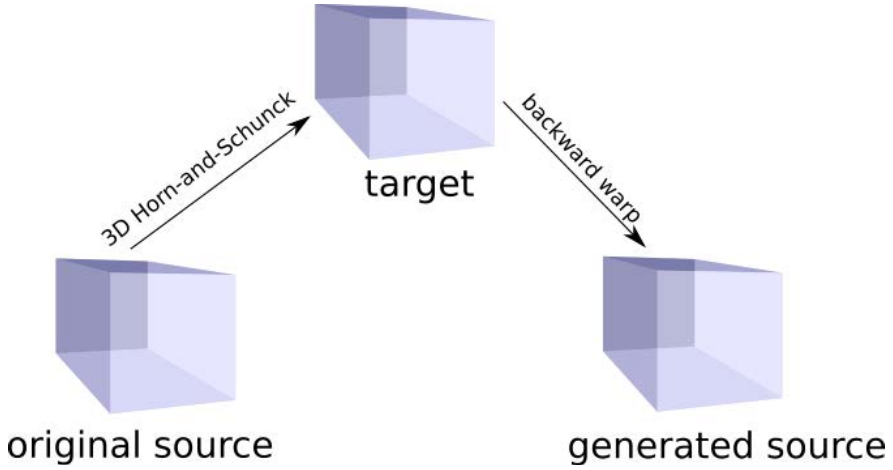


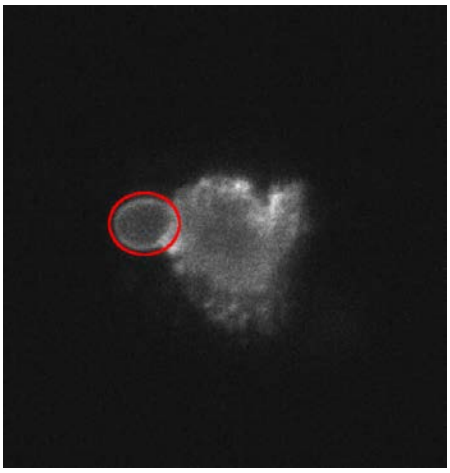
FIGURE 5.3 – A 3D flow field is computed for the original source and the target volumes using the 3D Horn-and-Schunck method. The computed flow field is then used to backward warp the target volume to obtain a generated source. The generated source and the target volumes with the computed flow field serve as the ground-truth set.

5.4 Experiments

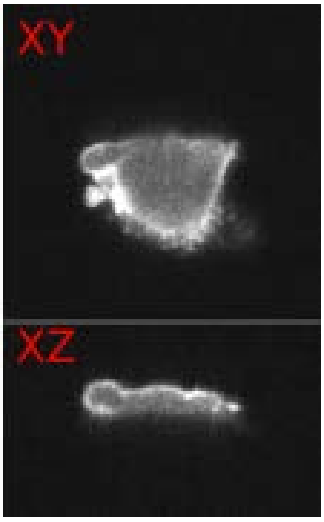
In this section, we describe the experiments that we conducted in order to assess the performance of our method. Comparisons are made against Amat's method and our extension Horn-and-Schunck method to 3D. We also assess the behaviour of the proposed SAE measure. Tests have been performed in generated and real test sequences.

5.4.1 Pseudo-ground truth generation

To make a quantitative evaluation of our method, we require ground-truth flow fields. Currently, there are no velocity measurements of real sequences available. Simulation of flow field specific to structures of the cell is also challenging. One can generate arbitrary field, but may not reflect the realistic motion. However, a motion estimation method can be employed that can generate realistic enough motion. We call this motion field, the pseudo ground-truth



(a) The bleb in motion is circled in red.



(b) xy and xz slices of the original source volume.

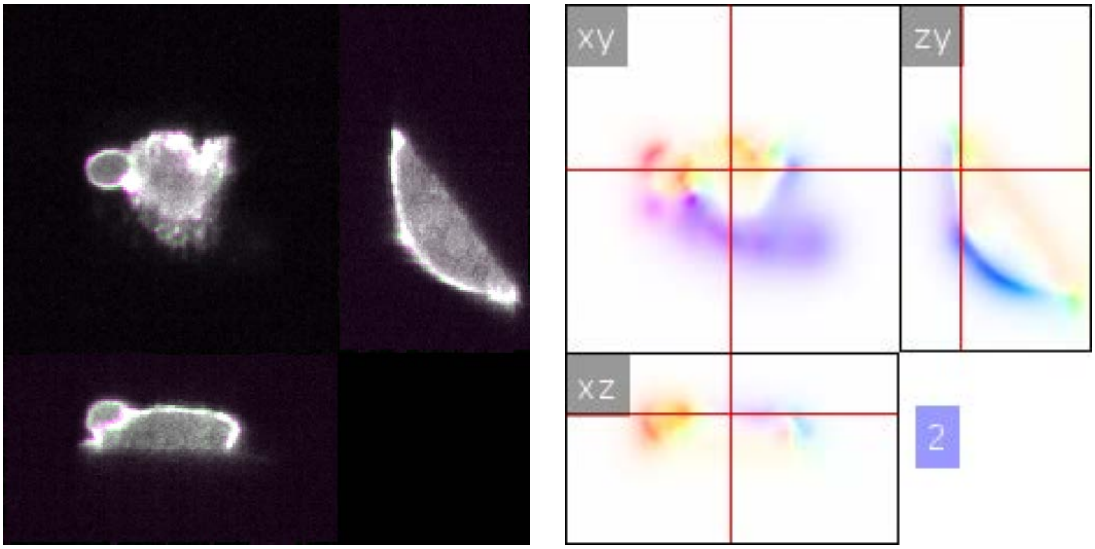
FIGURE 5.4 – The blebbing MV3 cell in MV3 A acquired with DiaSLM. The region circled in red depicts a bleb in motion. The sequence is used to generate Dataset 1 and 2.

flow field. The computed motion can be used to backward warp the target volume to obtain a generate source. The pair of the generated source and the target volumes, along with the computed flow field serve as the (pseudo) ground-truth set. The scheme is depicted in Fig. 5.3, where 3D Horn-and-Schunck method (as an example) has be employed.

We choose, the sequence MV3 A for the ground-truth generation. The sequence exhibits small, smooth and mainly translational motion of a bleb as shown in Fig. 5.4. For the following experiments, we generate few pairs of volumes of size $144 \times 150 \times 83$. To reduce the testing time, we have downscaled the original volumes in XY plane by a factor of 2.

5.4.1.1 Dataset 1

Dataset 1 is generated using 3D Horn-and-Schunck method. To generate the pairs, a source volume is picked at time $t = 0$, and 10 successive volumes at $t = \{1, 2, 3, ..., 10\}$ are chosen as the target volumes from the aforementioned sequence. Thus, the maximum of the motion magnitude between the source and the target volumes increases with time step t. To avoid generation of overly smooth flow fields, we set a low value for the smoothing parameter of the 3D Horn-and-Schunck method. After the computation of 3D flow field for the pairs, each target volumes are backward warped to obtain 10 new source volumes. The new source and the target volumes, along with the computed flow field, then serve as the ground-truth set. This dataset is used to tune the parameters of our method. We also evaluate the performance of our method and Amat's method with this dataset. Figure in 5.5b presents the side-by-side view of the generated



(a) (b)

FIGURE 5.5 – (a) Overlap of xy slice of the generated volume pair in Dataset 1. Source in red and target in green channel. (b) Corresponding 3PHS map of ground-truth flow field. The number inside the blue box on the bottom left of the 3PHS map denotes the maximum of the range of motion magnitude upto which the Saturation space is normalized to.

volume pairs and 3PHS map of the corresponding 3D flow fields. Figure 5.6 depicts the 3PHS rendering of 6 flow fields in this dataset.

5.4.1.2 Dataset2

Similar to Dataset 1, for the generation of Dataset 2, we choose a volume I_0 at time 0 as a source. We then select a set of volumes as target I_t at incrementing time interval. 12 target volumes at time t ={1, 2, 3, .., 12} are selected (see Fig. 5.7). The motion appears to be translational which makes the evaluation of the motion amplitude predictable over the selected volume pairs. We compute flow fields $\tilde{\boldsymbol{w}}_{i=1}, \tilde{\boldsymbol{w}}_{i=2}$ using Amat's and our methods, respectively. By backward warping I_t, we reconstruct source volumes $\tilde{I}_{t,i}$ for time $t = \{1, 2, 3, ..., 12\}$ with the flow fields at $i = \{1, 2\}$. We compute SAE of the flow fields for each voxel in I_0. The maps $S_{t,i} \in \mathbb{R}^3$ are then generated for each flow fields.

Consider a pair of volumes : I_0 and I_t. Now we drop t from the subscript for the sake of brevity. With the motivation of compositing a 3D flow field that consists of the best 3D velocity vectors corresponding to any point in the volume, we use SAE measure to prune the candidates from $\{\boldsymbol{w}_1, \boldsymbol{w}_2\}$. We consider a vector to be the best if it can make the most faithful reconstruction of the two. For any point $\boldsymbol{p}$ in I_0, we choose the 3D velocity vector $\boldsymbol{w}_{c=i}(\boldsymbol{p})$ from the two flow

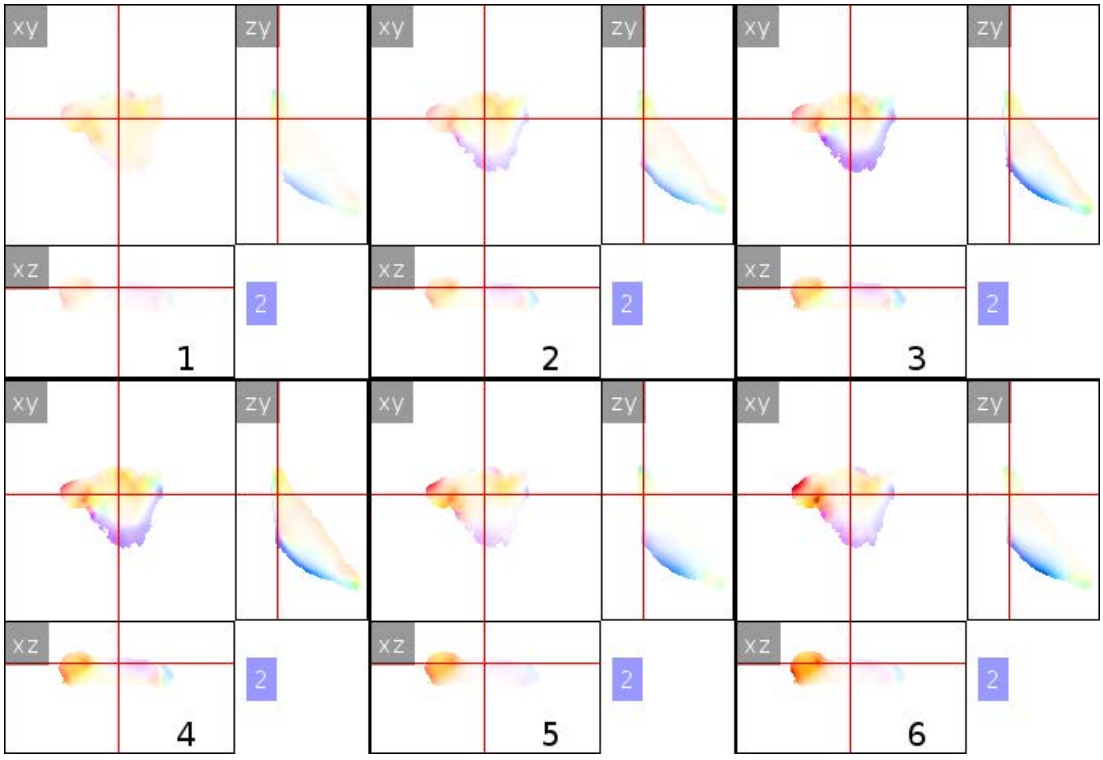


FIGURE 5.6 – 3PHS map of the flow field computed by 3D Horn-and-Schunck method, and used as the ground truth flow field in Dataset 1. The time step is marked at the mid-bottom of each sub-figure.

fields such that $S_c(\boldsymbol{p})$ is the minimum of the two SAE maps, i.e., more specifically, we define :

$$\boldsymbol{w}_c(\boldsymbol{p}) = \boldsymbol{w}_i(\boldsymbol{p}) | \boldsymbol{w}_i \in \{\boldsymbol{w}_1(\boldsymbol{p}), \boldsymbol{w}_2(\boldsymbol{p}))\}, \tag{5.33}$$

$$S_i(\boldsymbol{p}) = \min(S_1(\boldsymbol{p}), S_2(\boldsymbol{p})). \tag{5.34}$$

The composite flow field is generated for each time steps. Let $\boldsymbol{w}_{t,c}$ denote the composite flow field. To ensure a smooth flow field, we use median filter over each scalar component of $\boldsymbol{w}_{t,c}$. Now we generate a new source volume $\tilde{I}_{t,c}$ by backward warping I_t using $\boldsymbol{w}_c$. Ideally, all $\tilde{I}_{t,c}$ should be the same as I_0 volume, if the composite flow fields are the same as the unknown ground-truth flow fields. The new sources for each time step can be seen in Fig. 5.8.

Now, we consider $\boldsymbol{w}_{t,c}$ as the ground truth flow field between pairs $\tilde{I}_{t,c}$ and I_t. The composite/ground truth flow fields for each time steps are depicted in Fig. 5.9 as the 3PHS map.

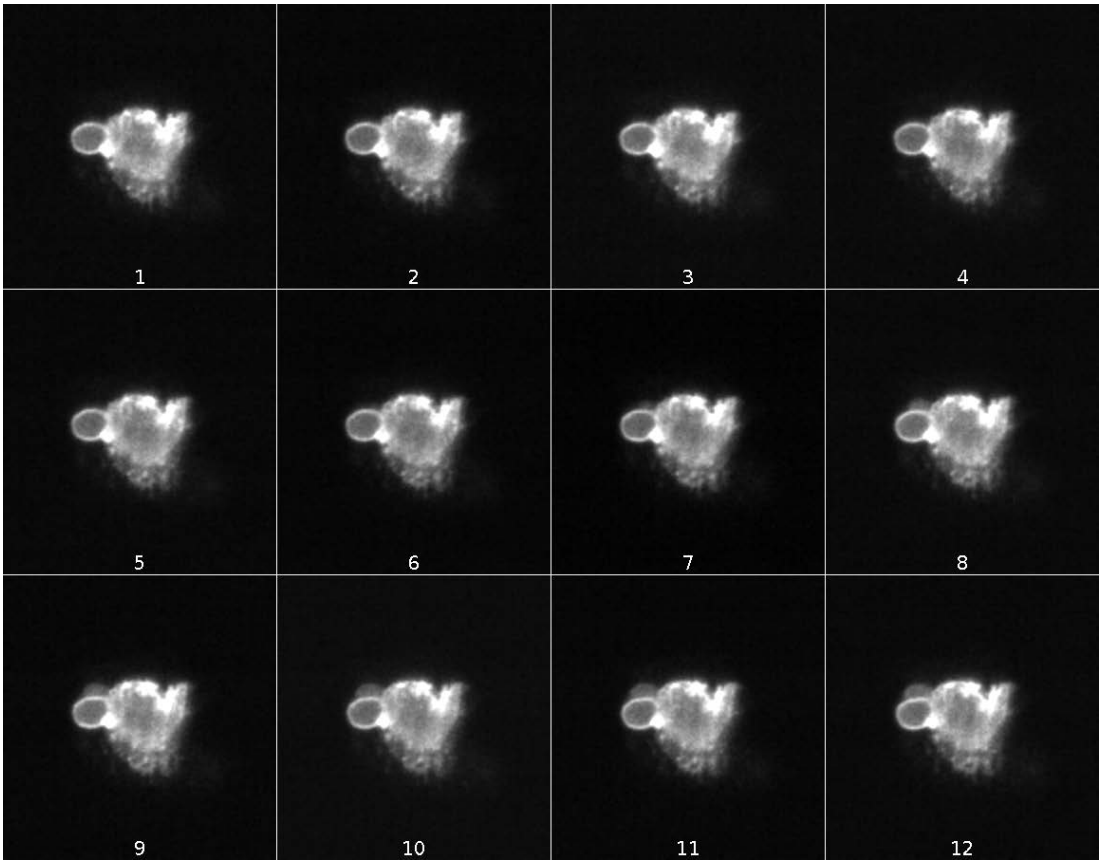


FIGURE 5.7 – A slice of 12 target volumes for Dataset 2. The time step is given in the mid-bottom on each sub-figure.

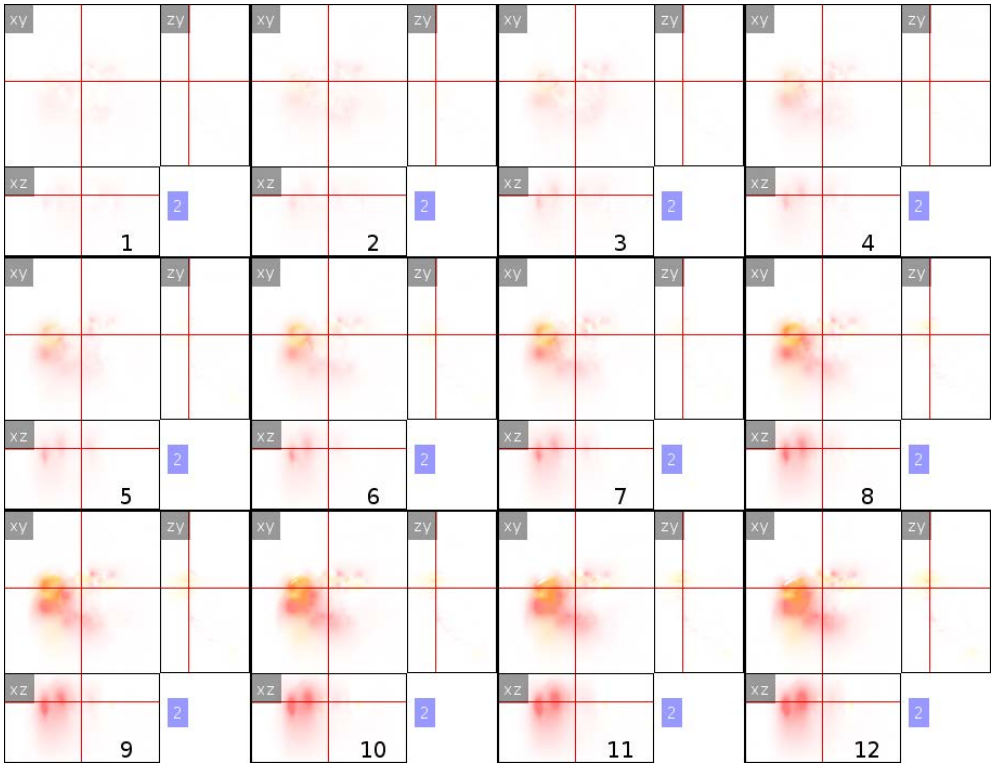


FIGURE 5.9 – 3PHS map of ground truth flow for Dataset 2 at each time step. The time step is marked at the mid-bottom of each sub-figure.

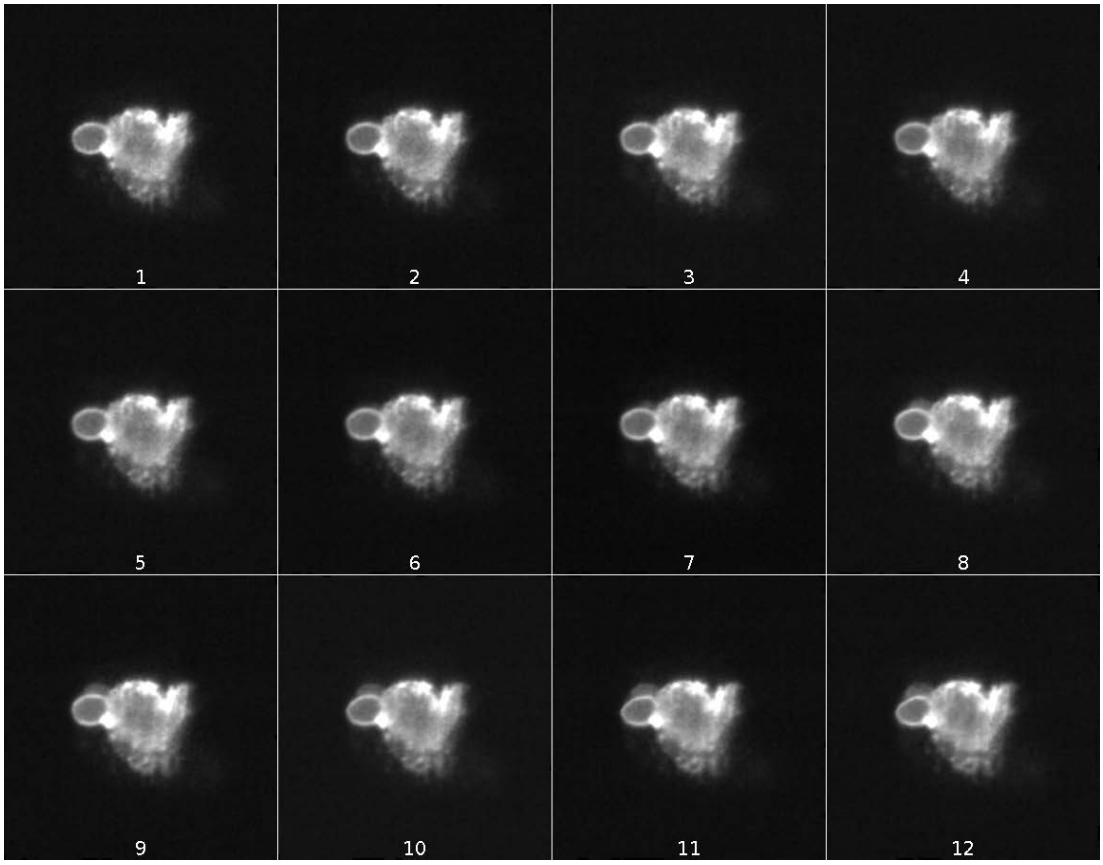


FIGURE 5.8 – A slice of 12 reconstructed source volumes by the composite flow fields for Dataset 2. The time step is given in the mid-bottom on each sub-figure.

5.4.2 Parameterization

The values/range of values for all the parameters associated to our method that we have investigated with during the experiments are presented in Table 5.1. We now test different values of the CS approximation factor ϵ and the smoothness parameter α to compute flow fields with our method. We recall that End-Point-Error (EPE) is the Euclidean distance between the estimated and the ground truth vectors. The average of EPE (AEPE) is used to measure the accuracy of the computed flow fields. We select a pair of volumes from the Dataset 1, and then compute flow field using our method for exponentially decreasing values of α and a set of $\epsilon = \{20, 50, 80\}$. The volume pair and the corresponding ground-truth flow field are depicted in Fig. 5.5. Figure 5.10 plots the accuracy of the computed flow field for different values of α and ϵ. The larger values of ϵ allows the method to compute flow fields with better accuracy (low average EPE) even with the smaller values of α. This prevents the over-smoothing of flow fields, which is desirable in a sequence where mostly background voxels are present. Figure 5.11 depicts the 3PHS maps of the flow fields for small, mid and high values of α and ϵ. One

can notice that with the high value of α, for all values of ϵ, the method produces smoother flow fields. Mid values of α and ϵ produce favourable flow fields, which exhibit discontinuities in the labeled actin region and the unlabeled cytosol region. For smallest values of both parameters, the flow fields are noisy, and have unrealistic motion amplitude. For smaller values of ϵ, the noisy gradients also contribute to the CS, which causes the data-term linerization to fail. Thus, the computed motion is not reliable.

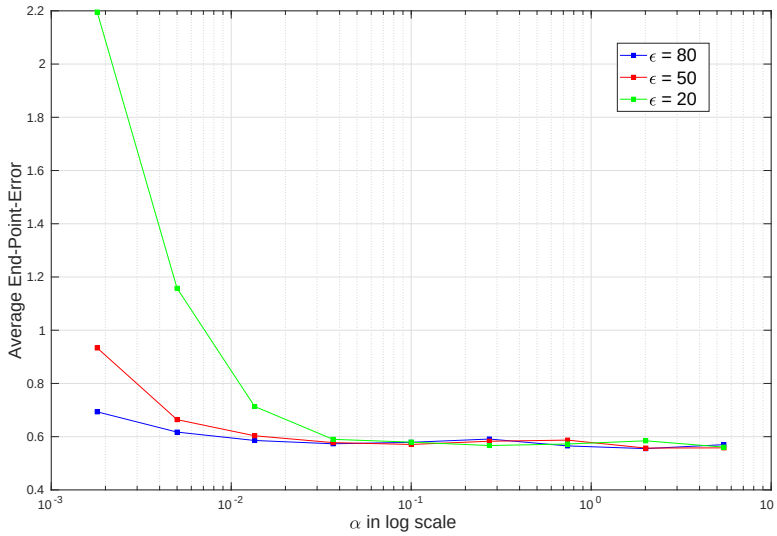


FIGURE 5.10 – α vs AEPE for different values of ϵ.

5.4.3 Assessment of ASAE behaviour

We test Amat's method and our method by computing 3D flow fields in the Dataset 1. For Amat's method, the supervoxel width is set to 9, the smoothness parameter as 50 with 3 pyramid levels. For our method, we use 26-neighbourhood for CS computation, $\epsilon = 80, \alpha = 0.011$, 2 pyramid levels with 10 warpings per level, and 30 SOR iterations. The respective parameters, as well as the initial conditions for both methods are the same for each volume pairs in the dataset. Therefore, higher error in the computed flow field is expected with larger time steps.

Figure 5.12a shows the AEPE of the method over the sequence. Clearly, our method outperforms Amat's method. The plot in Fig. 5.12b shows the similar trend of ASAE to that of average EPE. Like the AEPE, ASAE reveals that the accuracy of both methods drop for volume pairs at bigger time steps. 3PHS maps of the ground-truth flow fields and the results of our method and Amat's method are presented in Figs. 5.6,5.14a and 5.14b. The resemblance between our computed flow fields and the ground-truth flow fields are visible. On the other hand, the 3PHS map of the flow fields computed by Amat's method is piece-wise constant in the super-voxel segments. This patch like local vector field in super-voxel regions makes Amat's

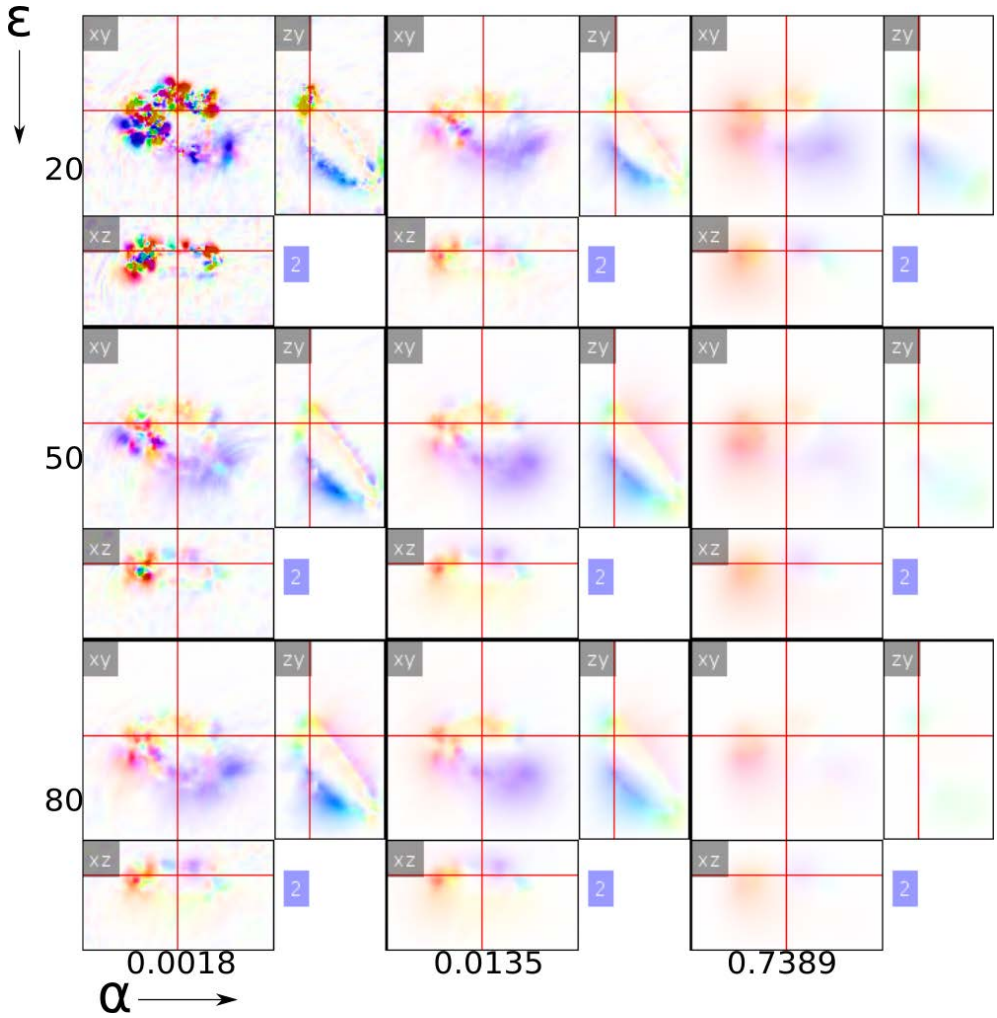


FIGURE 5.11 – Display of 3PHS maps of flow fields computed with our method using three different values of ϵ and α for a volume pair in Dataset 1.

method inferior compared to our method. Tables 5.2 and 5.3 reports the percentile and average errors of the EPE and the SAE measures of the two methods, respectively.

We also perform the test with the Dataset 2, where we compute and assess the flow fields with 3D Horn-and-Shunck method. The intensities of the volumes are normalized between 0 – 255. We consider 3 pyramid levels, while using 5 warpings per level. The smoothness parameter is set to 55 with 50 fixed point iterations. We also compute flow fields with Amat's and our method for this dataset.

The computed flow fields of each time step are depicted as 3PHS map in Figs. 5.15a, 5.15b, 5.16 with the three methods. The flow field computed with Amat's method is visually patchy. Because of the foreground mask used during computation, the method does not compute flow field outside the foreground segmentation. On the other hand, 3D Horn-and-Schunck method

TABLE 5.2 – End-Point-Error : Percentile and average errors for the Dataset 1

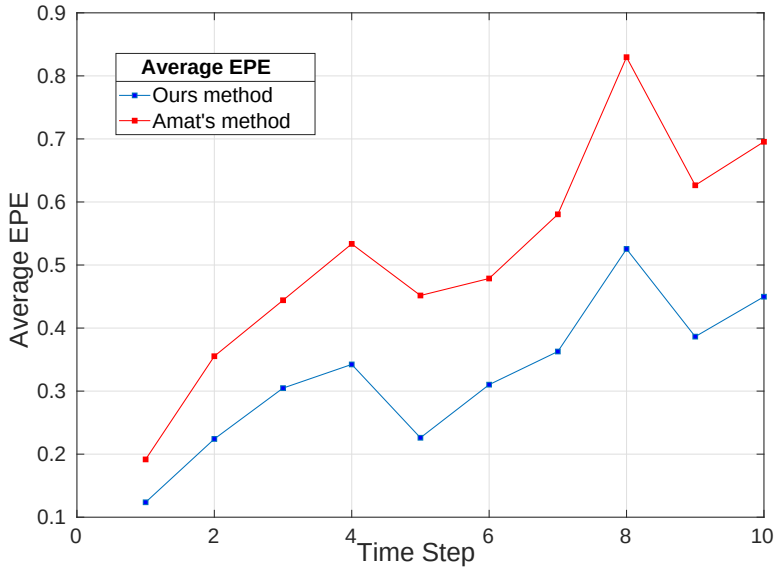
Time Step	Amat's method				Our method			
	90%	95%	99%	Mean	90%	95%	99%	Mean
1	0.4382	0.6264	1.1269	0.1917	0.2532	0.3587	0.6799	**0.1237**
2	0.7761	1.1291	2.7410	0.3553	0.3977	0.5491	1.3267	**0.2241**
3	0.9444	1.3608	3.3245	0.4441	0.5534	0.7377	1.7847	**0.3048**
4	1.1617	1.6334	4.0799	0.5335	0.6222	0.8556	2.1611	**0.3423**
5	1.0408	1.4602	2.7193	0.4517	0.5024	0.6944	1.2492	**0.2260**
6	1.1278	1.7025	3.3711	0.4786	0.6358	0.8779	1.8720	**0.3102**
7	1.4597	2.1316	4.2987	0.5804	0.7422	1.0578	2.3148	**0.3628**
8	1.9364	2.6369	5.9004	0.8297	0.9450	1.3820	3.9761	**0.5254**
9	1.5825	2.4362	4.7762	0.6265	0.8043	1.1341	2.3198	**0.3864**
10	1.7317	2.6051	5.7586	0.6955	0.8781	1.2143	3.0211	**0.4497**

TABLE 5.3 – SAE : Percentile and average errors for the Dataset 2

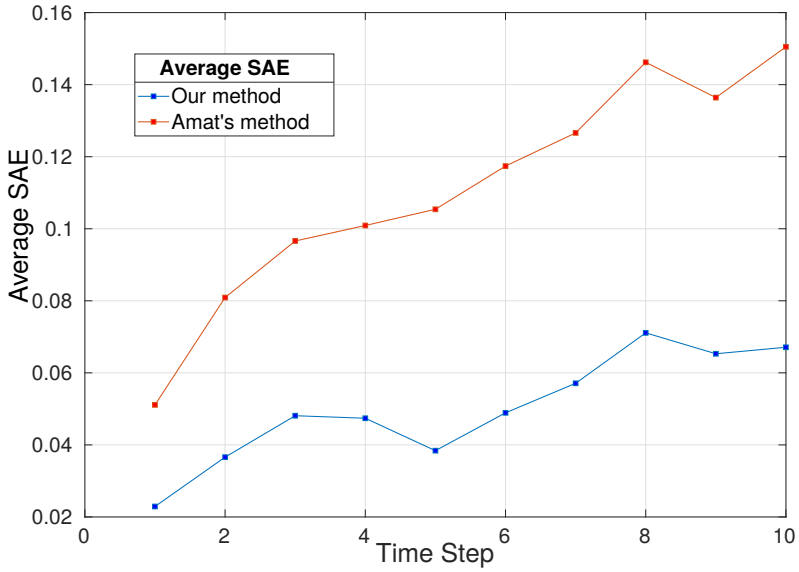
Time Step	Amat's method				Our method			
	90%	95%	99%	Mean	90%	95%	99%	Mean
1	0.1169	0.1898	0.4695	0.0511	0.0493	0.0768	0.1907	**0.0229**
2	0.1855	0.3086	0.7945	0.0809	0.0802	0.1216	0.2889	**0.0366**
3	0.2263	0.3849	0.9691	0.0966	0.1035	0.1590	0.4025	**0.0481**
4	0.2397	0.3814	0.8915	0.1009	0.1011	0.1534	0.3715	**0.0474**
5	0.2466	0.4136	0.9964	0.1054	0.0876	0.1326	0.3153	**0.0384**
6	0.2808	0.4679	1.1921	0.1174	0.1076	0.1689	0.4440	**0.0489**
7	0.3119	0.5285	1.2221	0.1266	0.1275	0.2021	0.5201	**0.0571**
8	0.3510	0.5813	1.2449	0.1462	0.1572	0.2501	0.6489	**0.0711**
9	0.3496	0.5862	1.3142	0.1364	0.1469	0.2327	0.6522	**0.0653**
10	0.3873	0.6601	1.3369	0.1505	0.1488	0.2440	0.6614	**0.0671**

computes significant motion in regions where no large motion vectors are present in the ground truth. Our method outperforms the two for each time steps with respect to AEPE and ASAE. This could be accounted to the bias induced by the more contribution of vectors in composite flow field by our method than Amat's method. The percentile error and average values for these measure are presented in Table 5.4 and 5.5. One can notice the 90% and above, the SAE measure is higher than the average. This suggests the presence of outliers, which can be visualized in the SAE maps depicted in Fig. 5.13. The abrupt errors (outliers) pertain to the unlabeled regions in the maps corresponding to the both methods. The errors in the motile bleb region in each pair are relatively lower for our results than Amat's results. To remove the effect of outliers, one can precisely segment the labeled and unlabeled cytosol regions in the cell channel, and apply the segmentation mask over the computed SAE map. However, we do not have a robust enough method for now, which can perform such segmentation. Therefore, we continue with the straightforward segmentation map of the cell body itself. This is not an issue with the collagen

channel because it does not contain the cytosol that entails the noisy region.



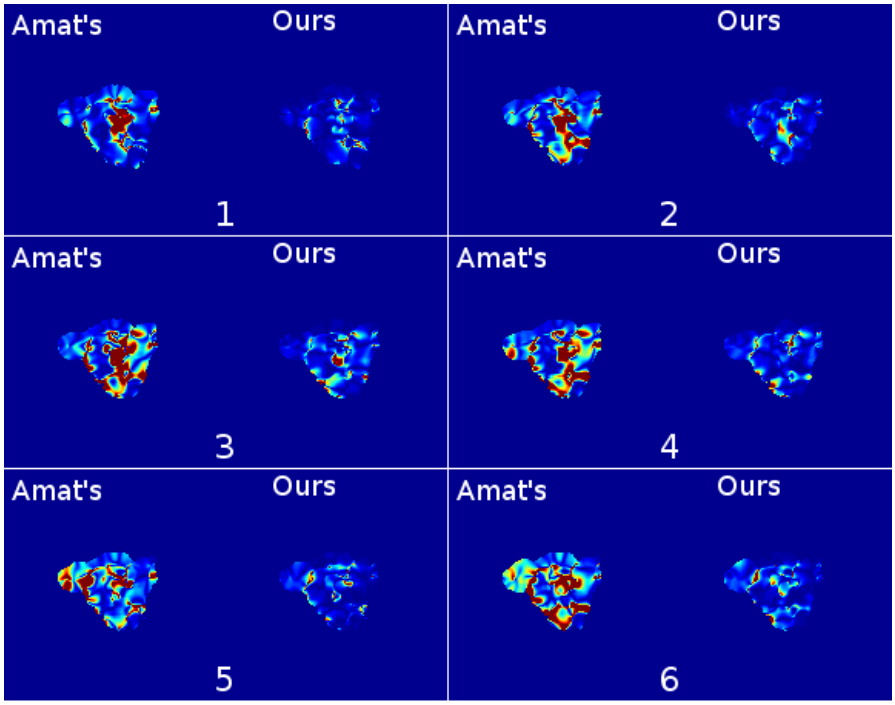
(a) Average End-Point-Error (EPE) of the flow fields computed by our method and Amat's method for pairs of volumes separated by time steps mentioned in x-axis of the plot.



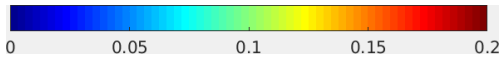
(b) Average Structural Angular Error (SAE) of the flow fields computed by our method and Amat's method for pairs of volumes separated by time steps mentioned in x-axis of the plot.

FIGURE 5.12 – Plots of AEPE and ASAE for Dataset 1.

The plot in Fig. 5.17 shows the evolution of ASAE of 3D Horn-and-Schunck. It turns out that ASAE follows the trend of AEPE. Like the trend depicted in the plots in Fig. 5.12, the ASAE



(a)



(b)

FIGURE 5.13 – (a) SAE maps for the slices of 3D flow fields computed by Amat's method and our method for Dataset 1. (b) The color code for the SAE maps.

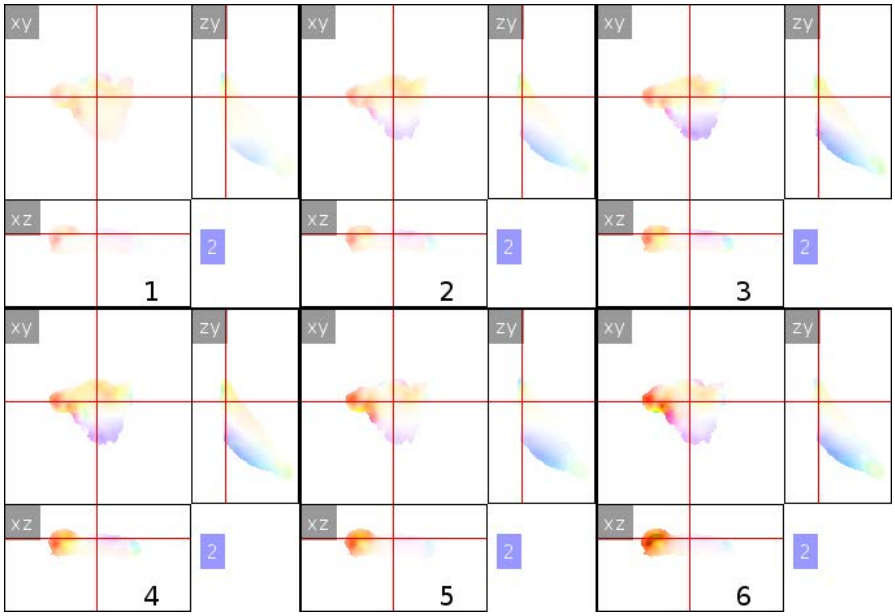
behaves very similar to AEPE. Though, these error measures are not directly commensurable, in absence of ground truth flow field, ASAE becomes a relevant measure to estimate the accuracy of the computed flow field.

This concludes the tests with (pseudo) ground-truth flow fields. Note that these tests have enabled us to set parameters of our method, and assess the reliability of our proposed ASAE measure.

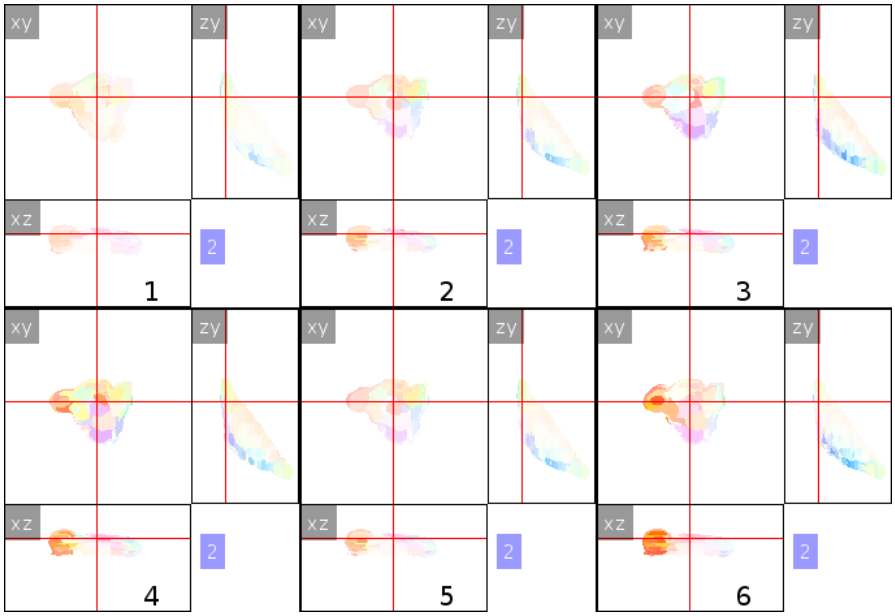
5.4.4 Results in real sequences

5.4.5 Blebbing MV3 cell

We perform a set of experiments on the sequence MV3 A. The same sequence has been used for (pseudo) ground truth generation. For this, we select four volumes as source at discrete times t_1, t_2, t_3, t_4. We then create four different sequences with 12 target volumes at incrementing time steps. We refer to these sequences as dataset 3. We have selected the source and the duration of time such that the motion of the bleb appears to be translational and smooth overtime.



(a) 3PHS map of flow computed by our method for Dataset 1. The time step is marked at the mid-bottom of each sub-figure.

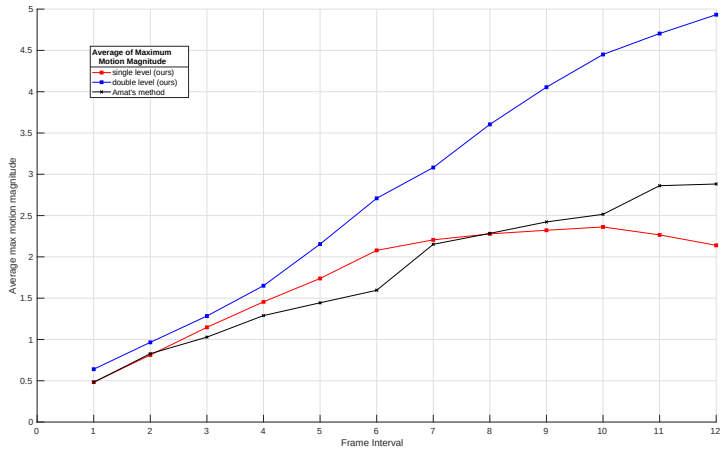


(b) 3PHS map of flow computed by Amat's method for Dataset 1. The time step is marked at the mid-bottom of each sub-figure.

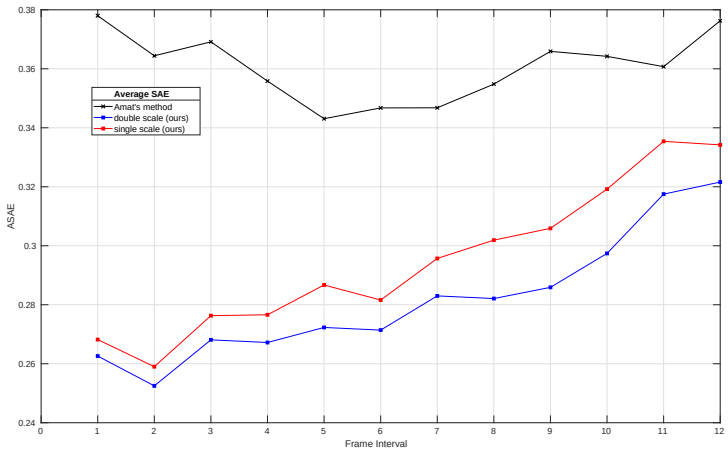
FIGURE 5.14 – 3PHS maps of flow fields computed using our method and Amat's method

We process the dataset with our method using single and double level pyramids, and with Amat's method. Since, there is no ground-truth motion for the dataset, we compare the performance of the two methods based on the ASAE measure. Figure 5.18a shows the average (over the four sequences) of maximum motion magnitude in the bleb region at each time step.

We do not consider the motion magnitude at the centroid of the bleb, nor the average magnitude in the region. Because of the spherical shape of the bleb, the centroid lies in the unlabeled and larger cytosolic region. Likewise, the average magnitude of the region would be representative of the vectors in cytosolic region. By observing the 3PHS map (for e.g., Fig. 5.9), one can say that the maximum motion magnitude pertains to voxels in the actin region. For these reasons, we opt for the maximum of the motion magnitude in the bleb region.



(a) Average maximum motion magnitude computed in the bleb region for pairs of volumes separated by time steps mentioned in x-axis of the plot.



(b) Average Structural Angular Error (SAE) computed in the bleb region for pairs of volumes separated by time steps mentioned in x-axis of the plot.

FIGURE 5.18 – Evaluation of motion computed using Amat's and our method applied to real sequences in dataset 1.

With increasing time steps, the motion of the bleb also increases in amplitude. We recall

from Chapter 4, subsection 1.3.3.2, that the retracting bleb in the MV3 cell had around unit voxel displacement between two successive frames in the experiment. The current sequence has been acquired with the same microscope, and the same acquisition rate as the one in Chapter 4. Fig. 5.18a also suggests around a unit voxel displacement between two immediate frame pairs. As the time step between the frame pair increases, the motion magnitude also increases (seemingly linear).

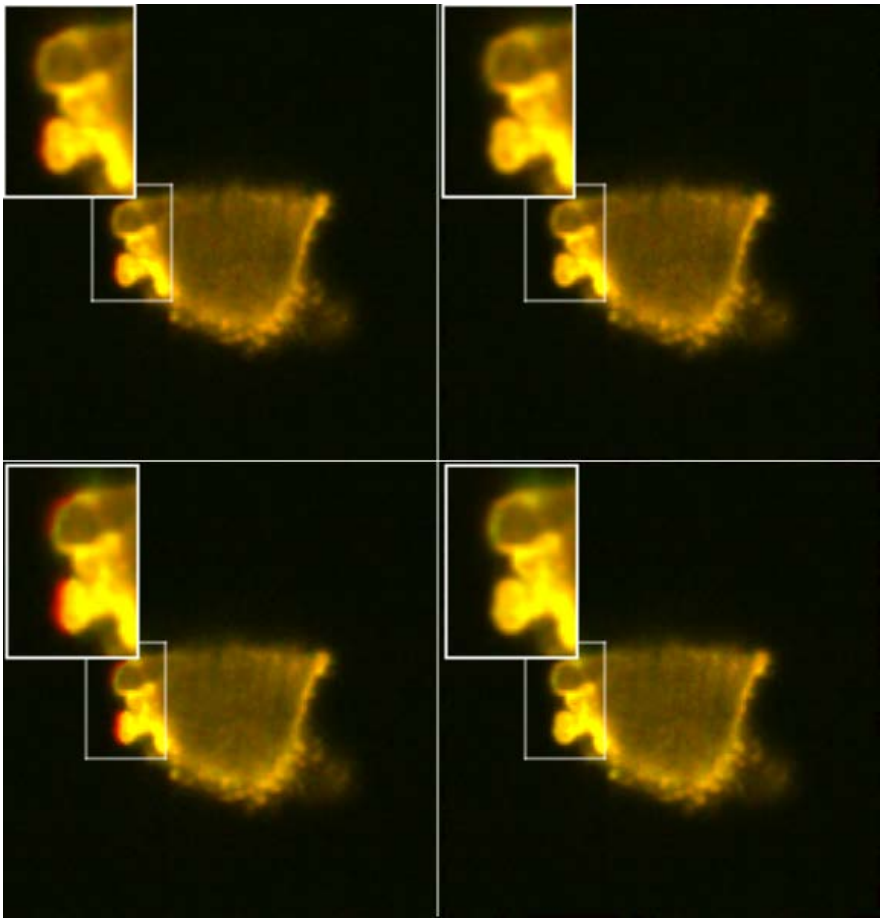


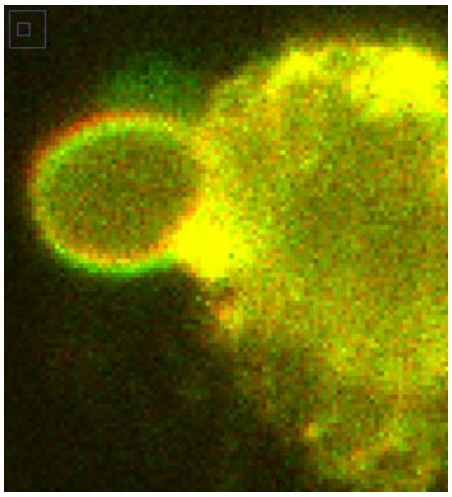
FIGURE 5.19 – Overlap of a slice of the original source and the optical flow reconstructed source by Amat's method (left) and our method (right). The original source's slice is in red channel and the reconstructed ones are in the green channel. A complete overlap of the original and reconstructed slices results in yellow pixels. The other color reflects the error in the optical flow field. The rows corresponds to the computed flow fields between the pairs at time interval of 5 and 9 steps. At larger time intervals, Amat's method clearly fails to accurately compute the motion of the bleb. The bleb regions have been zoomed, which are highlighted by the white rectangles.

For our method, with single pyramid level, the computed motion magnitude for the pairs at different time steps are linearly increasing until it peaks at around 2.5 unit of amplitude. We

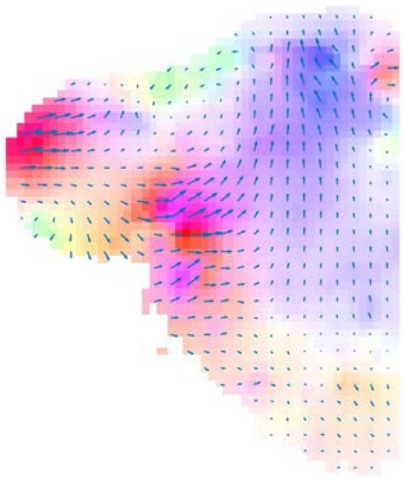
consider this magnitude of the motion as the break-point of our single level implementation. The results obtained with Amat's method also show the linearly increasing motion magnitude. Using double pyramid levels, our computes linear flow fields in volume pairs corresponding to the larger time steps.

As seen in Fig. 5.18b, our method (both single and double level pyramids) outperforms Amat's method with respect to average SAE. However, the average SAE measure for the results of our method increases gradually with the larger time steps. The average SAE measures for the results of Amat's method are relatively higher in all time steps. The inaccuracy of the methods can be further explained with visualization of the reconstruction by the backward warping of computed the flow fields as see in Fig. 5.19. The figure depicts overlap of slices of the original source and the optical flow reconstructed source computed by Amat's method and our method. The original source's slice is in red channel and the reconstructed ones are in the green channel. The reconstruction throughout the time intervals should ideally be the same for each pair in the sequence, as all of the target volumes share the same source. A complete overlap of the original and reconstructed slices results in yellow pixels. The other color reflects the mismatch. Clearly, at larger time intervals, the results of Amat's method fail to compute the motion of the bleb. The reconstructions exhibit gradually shrinkage of the bleb, which is visible at the $9th$ interval. On the other hand, reconstruction with the flow field computed by our method is fairly stable which is in agreement with the average SAE measure. Once again, our method prevails. However, none of the methods detect the motion of another bleb (behind the bleb in focus) which has faint fluorescence signal. The bleb seems to have started expressing the tractin as it undergoes retraction.

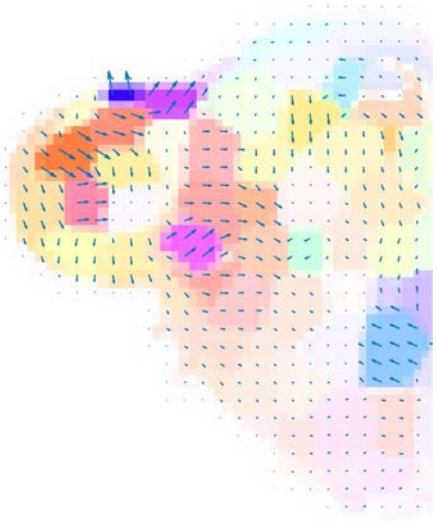
Now, lets take a closer look at the flow fields in the bleb region for a successive volume pair in MV3 A, computed by the three methods. Figure 5.20 presents the zoomed region in 3PHS maps overlaid with arrows to elaborate on the smoothness and regularity of the computed flow fields. Visually, the bleb moves to the bottom right direction in the XY plane. 3D Horn-and-Schunck fails to capture this motion. On the other hand, Amat's method and our method are able to compute the observed motion. However, the piecewise constant nature of the flow field in super-voxel segments are apparent in the results obtained by Amat's method. Our method is able to compute smooth motion which is coherent to what is observed.



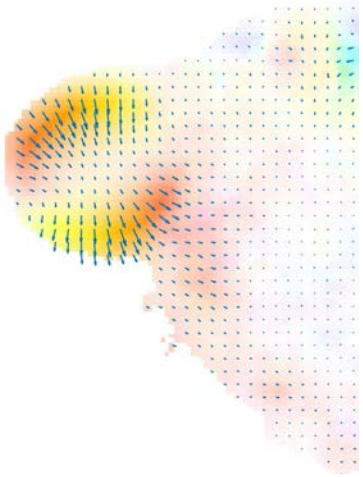
(a) Overlap of slices of source (red) and target (green) volumes.



(b) 3PHS map for 3D Horn-and-Schunck's result



(c) 3PHS map of the computed flow field with Amat's method



(d) 3PHS map of the computed flow field with our method (single scale)

FIGURE 5.20 – 3PHS map zoomed in bleb region of the cell in MV3 A. Arrows overlaid on the XY-plane of 3PHS map.

5.4.6 Collagen network in motion

We now test our method on real sequences involving collagen motion. We have selected two such sequences with 4 frames each, depicting collagen as the extra cellular matrix (ECM) for (1) migrating MV3 cell via large scale retraction of filopodium (Collagen 1), and (2) blebbing MV3 cell (Collagen 2). Collagen 1 includes volumes of size $512 \times 512 \times 101$, and exhibits both small and large scale motion (see Fig. 5.21). Collagen 2 comprises volumes of size $256 \times 256 \times 101$,

which exhibits mainly small motion (see Fig. 5.25).

Collagen sequences are interesting because the scene contains fiber-like thin objects that move with diverse range of motion amplitude. The discontinuities are present in both intensity and motion along the fibers. Unlike the cell channel, no complex deformation or change in topology is observed.

We now report the results of our method while using different numbers of pyramid levels for Collagen 1. From the experiments reported in the previous Section 5.4.5, we concluded that with a single scale, our method peaks at around 2.5 units of motion amplitude. Similar results with the single level pyramid were obtained in this experiment. The maximum motion amplitude of 2.7 units is computed which is far from the observed motion (see Fig. 5.21). For this sequence, we have to rely on 4 levels of pyramids in order to obtain a flow field with good accuracy. The right column of Fig. 5.22 shows the reconstruction produced with flow field computed with $1-4$ levels of pyramid.

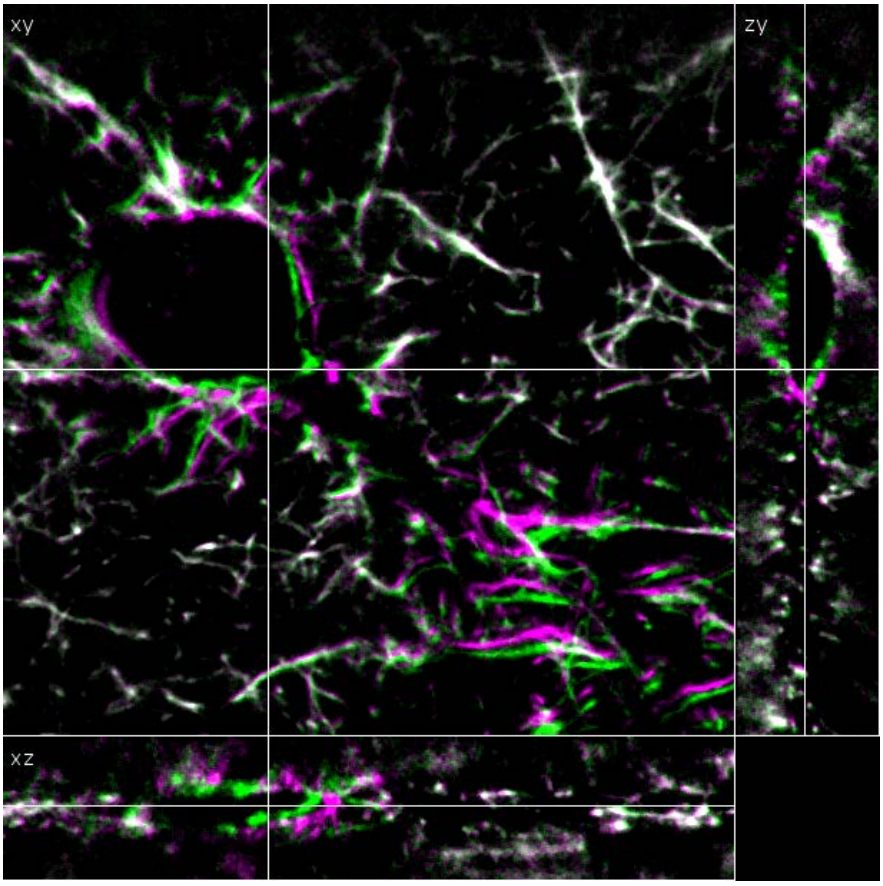


FIGURE 5.21 – Overlap of slices of Collagen 1 : source in magenta, target in green.

Super-voxel segmentation of the scene is challenging because of the thin, uneven sized foreground objects and large percentage of background voxels. The convergence and the stability

of Amat's method largely depends on the SLIC segmentation and the foreground mask quality. Figure 5.23a shows the 3PHS map of the flow field computed using Amat's method without the foreground mask with super-voxel width of 9 units. Outliers can be observed in its 3PHS map.

Figure 5.24b shows the MIP-flow map (introduced in Chapter 3) of the same collagen sequence. The map here, can provide better summary than 3PHS map. One can observe the dynamic collagen strands rendered with their flow color code. Since only the motion vector of the points with maximum intensities are being projected, the over-smoothness of the flow field does not hinder to discern the mobile collagen strands. Here, the maximum range of motion magnitude to which the saturation level has been normalized to is set to 16 units.

We display the 3PHS maps of the flow fields for Collagen 2 in Fig. 5.25. For both methods, the maps have been presented after background masking. The ASAE measures of both methods are reported in Table 5.6. Our method outperforms Amat's method only by smaller margins in the last two frames. During the tests, it was not necessary to tune the parameters of our method. However, for Amat's method, the tuning of SLIC segmentation, the threshold values for foreground mask generation and smoothness parameter required many trials. Same set of parameters did not work across the sequence. Compared to our results, it fails to capture the motion of smaller strands throughout the sequence. This can be seen at the top-left portion of XY plane of the 3PHS map in Fig. 5.25. This could be accounted to the failure of SLIC segmentation to form super-voxels in these regions. However, the errors in the smaller regions are not reflected by the average of SAE.

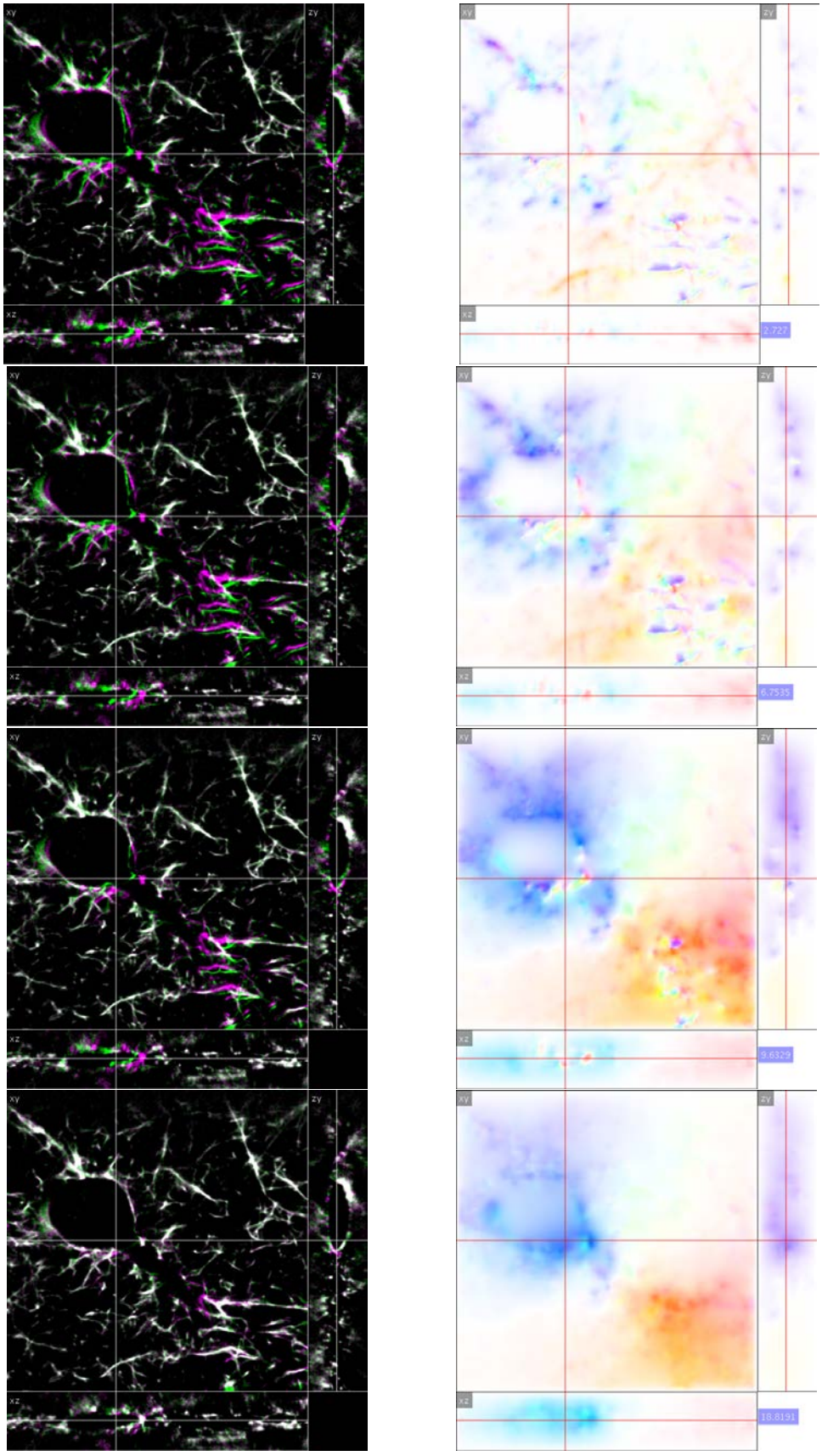
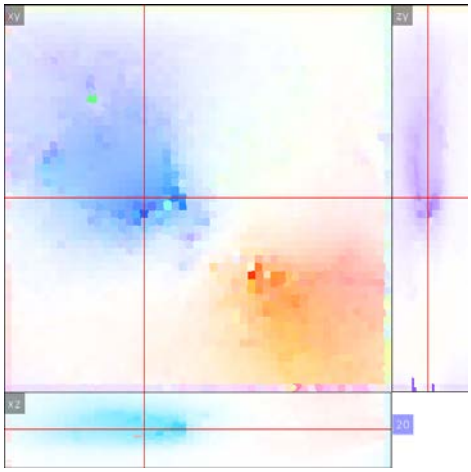
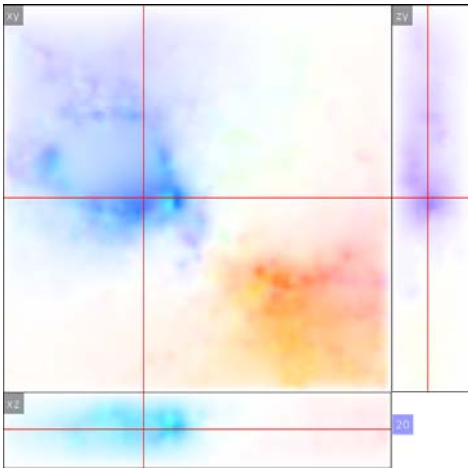


FIGURE 5.22 – Collagen motion computed by our method with 1 (top) to 4 (bottom) pyramid levels. Left column : Overlap of slices of source (magenta) and reconstructed source (green). The whiter the voxels are, the better the match is (i.e. motion vector). Right column : The respective 3PHS maps of the computed flow fields. The maximum ranges of the motion magnitude to which the saturation levels are normalized to are : 2.7, 6.75, 9.6, 18.8 units, from top to bottom, respectively.

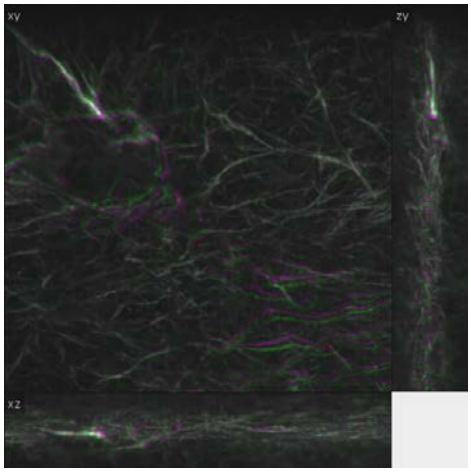


(a) The results of Amat's method without foreground mask.

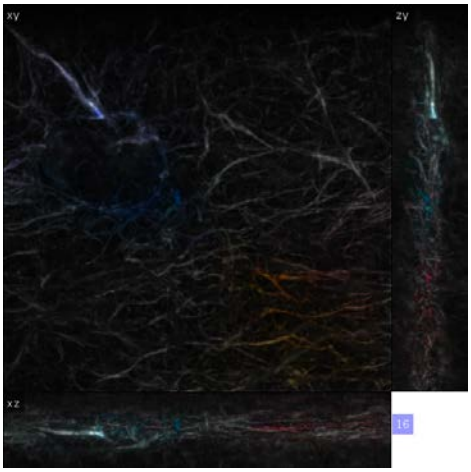


(b) The results of our method without foreground mask.

FIGURE 5.23 – 3PHS map of the collagen motion.



(a)



(b)

FIGURE 5.24 – (a) Maximum intensity projection (MIP) of source (green) and target (magenta) (b) MIP-flow map of our result for one of the pair in Collagen 1.

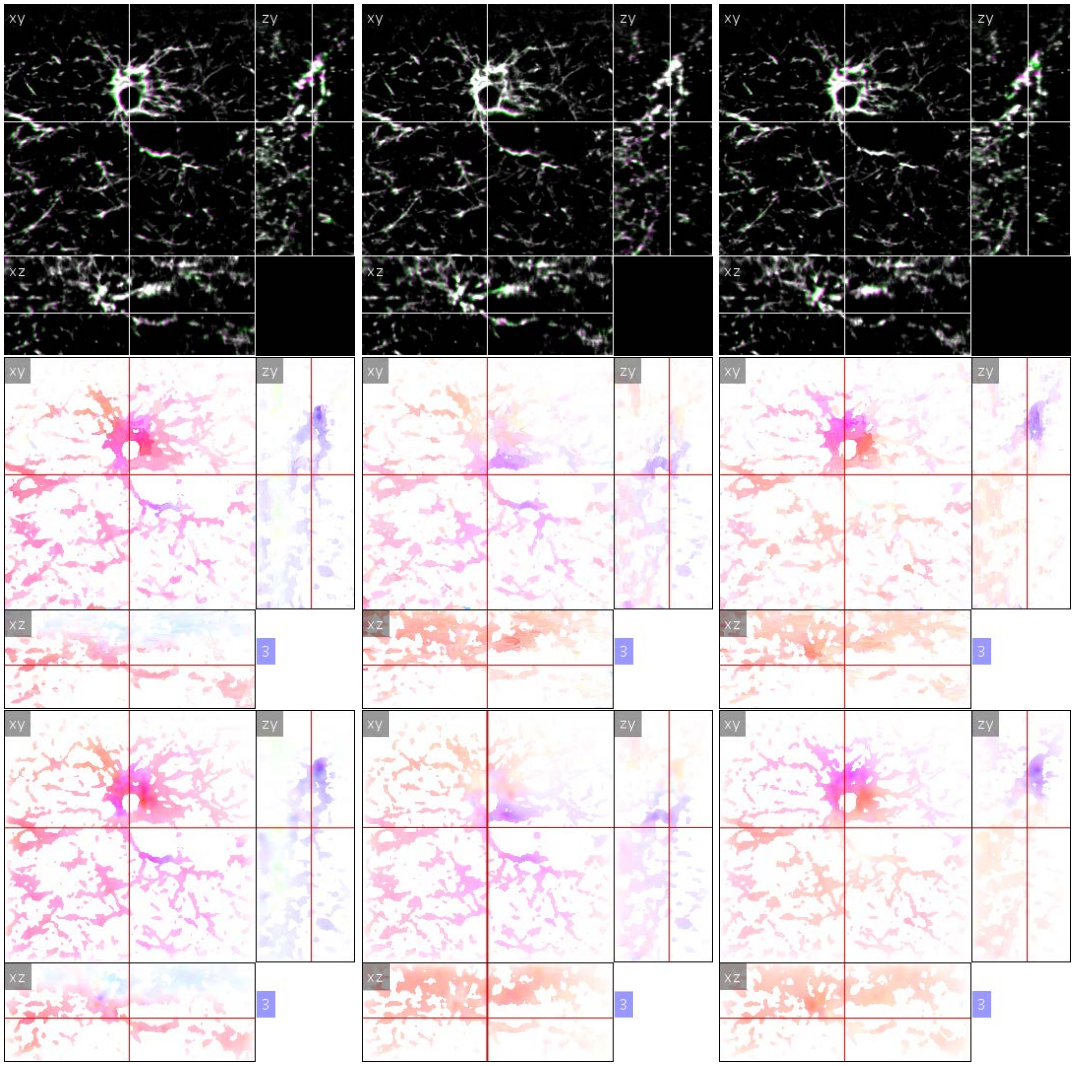
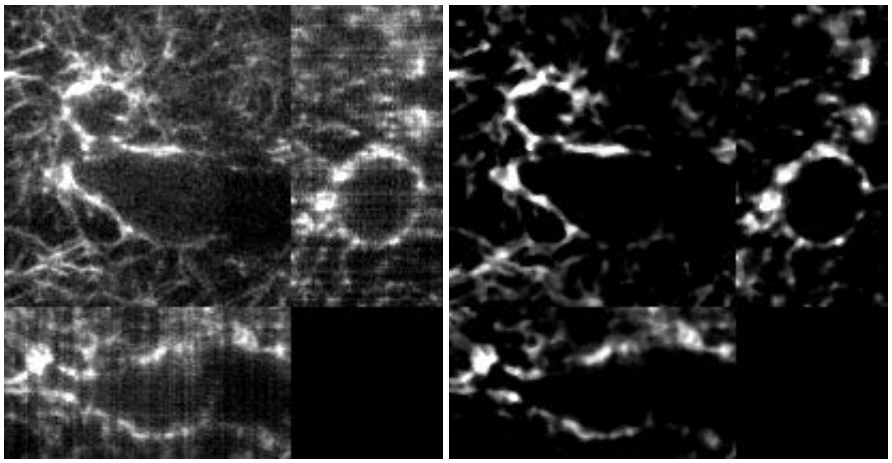


FIGURE 5.25 – Sequence Collagen 2. Top row : The overlap of slices of source (in green) and target (in magenta) for each volume pairs in Collagen 1 along the row. Mid row : The 3PHS maps of flow field for each pair computed using Amat's method. Bottom row : The 3PHS map of each of flow field for each pair computed using our method. The flow fields were masked with source foreground.

TABLE 5.6 – ASAE measures for volume pairs in Collagen 1.

Collagen 1 Pair	Amat's method	Our method
1	0.352	0.2638
2	0.2266	0.2254
3	0.2222	0.2206

5.4.6.1 Foreground masking of flow fields



(a) Raw data of Collagen 1 (b) After z-mean adjustment

FIGURE 5.27 – The effect of z-mean adjustment.

We have used the L_2-norm for regularization, which penalizes larger deviations heavily. Since we impose small gradient on the motion field, the discontinuities are blurred by the $L2$-norm. Keeping MIP-flow aside, 3PHS map can reveal the discontinuities in the field. Discontinuities not only improve user's visual feedback for understanding the dynamics of the scene, but also helps in motion segmentation, algorithmically. Sub-figures in 5.23 show the blurry flow fields computed by Amat's method and our method without application of foreground mask. The flow fields are able to summarize the general motion direction in the scene but requires foreground mask to depict precise motion of individual collagen strands. Foreground masking comes as the pre-processing step in Amat's method and the post-processing step in our method. It requires user's observation and thresholding to get a good foreground segmentation.

We provide a variant of the method to deal with background voxels without requiring user intervention. The technique assumes the objects are thin, and primarily oriented in XY plane. The idea is to subtract the mean intensity along the z-axis of the volumes, and add it back after the negative intensities are nullified. In this manner, we remove the unlabeled noisy voxels, and voxels that take arbitrary intensity values in each slice because of the faulty camera sensor, as seen in Figs. 5.27a and 5.28.

The background region in the resulting volume only consists of zero intensity voxels. Usually, during the computation of variational optical flow, the initial flow field is set to zero displacements everywhere. In textureless or smooth intensity regions, where data-term is less reliable, the regularization of the flow field provides the motion vector. If the smoothness

parameter is not set to a large value, and less number of minimization iterations are used, the background region will retain their initial zero displacement vector at the end of the iterations. The resulting flow field is the desired discontinuous flow field as seen in Fig. 5.29b.

The algorithm, which we refer to as z-mean adjustment is given in Algorithm 4.

Algorithm 4 z-mean adjustment

INPUT : 3D volumetric image stack I
OUTPUT : 3D volumetric image stack I'

1: **for** `each row i and each column j` **do**
2: $\quad \bar{I}(i,j) = \frac{1}{Z}\sum_{k=0}^{Z} I(i,j,k)$ ▷ Z = Numer of z-stacks
3: **for** `each row i and each column j and each slice k` **do**
4: $\quad I'(i,j,k) = min(I(i,j,k) - \bar{I}(i,j), 0)$
5: **for** `each row i and each column j and each slice k` **do**
6: $\quad$ **if** $I'(i,j,k) > 0$ **then**
7: $\quad\quad I'(i,j,k) = I(i,j,k) + \bar{I}(i,j)$

We use this technique sparingly, and only for the collagen motion sequences. It is only suited for the volumes with contents which are thick along z-axis. For instance, in cell channel, where the cell body is thick along z-axis, the subtraction of the mean along this axis might create null voxels even in the labeled region. This would become problematic for the data-term linearization.

5.5 Robust Regularization

Flow fields with discontinuities can be better computed with the L_1 regularization. Instead of relying on L_2 norm and foreground mask at post processing stage, we can integrate a robust L_1 penalty term in the variational energy function. We recall that $\psi(.)$ and $\psi'(.)$ are the robust penalty function and its derivative, respectively. Using the previous data term of Eq. 5.10 and involving the robust regularization, we get :

$$F = \|D(I, \boldsymbol{w})\|_2^2 + \alpha\psi(\|\nabla \boldsymbol{w}\|) \tag{5.35}$$

The related Euler-Lagrange equations are as follows :

$$\sum_{\boldsymbol{p}\in\Omega}\left(\sum_{n=1}^{N} H_\epsilon'^2(\partial_n I(\boldsymbol{p})).\partial_n I_x(\boldsymbol{p}).(\boldsymbol{w}(\boldsymbol{p})^\top \nabla_4(\partial_n I(\boldsymbol{p})) - \alpha \mathrm{div}(\psi'.\nabla u)\right) = 0, \tag{5.36}$$

$$\sum_{\boldsymbol{p}\in\Omega}\left(\sum_{n=1}^{N} H_\epsilon'^2(\partial_n I(\boldsymbol{p})).\partial_n I_y(\boldsymbol{p}).(\boldsymbol{w}(\boldsymbol{p})^\top \nabla_4(\partial_n I(\boldsymbol{p})) - \alpha \mathrm{div}(\psi'.\nabla v)\right) = 0, \tag{5.37}$$

$$\sum_{\boldsymbol{p}\in\Omega}\left(\sum_{n=1}^{N} H_\epsilon'^2(\partial_n I(\boldsymbol{p})).\partial_n I_z(\boldsymbol{p}).(\boldsymbol{w}(\boldsymbol{p})^\top \nabla_4(\partial_n I(\boldsymbol{p})) - \alpha \mathrm{div}(\psi'.\nabla w)\right) = 0. \tag{5.38}$$

The terms with divergence are computed using finite element method as :

$$div(\psi'\nabla u) = \frac{\partial(\psi' u_x)}{\partial x} + \frac{\partial(\psi' u_y)}{\partial y} + \frac{\partial(\psi' u_z)}{\partial z}$$
$$+ \frac{(\psi')_{x+1,y,z} + (\psi')_{(x,y,z)}}{2} . \frac{u_{x+1,y,z} - u_{(x,y,z)}}{h_x^2} + \frac{(\psi')_{x-1,y,z} + (\psi')_{(x,y,z)}}{2} . \frac{u_{x-1,y,z} - u_{(x,y,z)}}{h_x^2}$$
$$+ \frac{(\psi')_{x,y+1,z} + (\psi')_{(x,y,z)}}{2} . \frac{u_{x,y+1,z} - u_{(x,y,z)}}{h_y^2} + \frac{(\psi')_{x,y-1,z} + (\psi')_{(x,y,z)}}{2} . \frac{u_{x,y-1,z} - u_{(x,y,z)}}{h_y^2} \quad (5.39)$$
$$+ \frac{(\psi')_{x,y,z+1} + (\psi')_{(x,y,z)}}{2} . \frac{u_{x,y,z+1} - u_{(x,y,z)}}{h_z^2} + \frac{(\psi')_{x,y,z-1} + (\psi')_{(x,y,z)}}{2} . \frac{u_{x,y,z-1} - u_{(x,y,z)}}{h_z^2}, \quad (5.40)$$

where (x, y, z) is the location of current voxel in the finite element grid.

We solve the E.L. equations using SOR iterations. We implement it in the coarse-to-fine manner. During our experiments, more number of warpings and iterations were required to reach results similar to those obtained using L_2 regularization. Figure 5.30 shows 3PHS maps of flow fields computed for Collagen 1. Unlike the previous L_2 regularization, the flow field is not overly smooth in the background region even without z-mean adjustment. However, the regularization is the foreground region poor. This would require further tuning of the parameters. Instead of pursuing L_1 regularization in the presented energy minimization scheme, we have decided to use a more efficient total variational framework as described in (Rudin *et al.* 1992). This will be discussed in the next chapter.

5.6 Time Complexity

Time complexity of our algorithm with L_2 regularization is similar to that of 3D Horn-and-Schunck as for both implementation we use SOR iterations. The time complexity of SOR method is $O(kN)$, where N is the size of the volume and k is the number of iterations. However, additional complexity of $O(nN)$ arises from the computation of Census Signature (CS), where n is the size of the discrete neighbourhood. The aspects that can be parallelized are the computation of CS and its derivatives. For this, each elements of the CS has to be stored as a 3D matrix, which increases the memory requirement. For our algorithm with L_1 regularization, additional complexity of $O(N)$ occurs because of the computation of ψ'. The minimization can also be parallelized if we use splitting methods instead of SOR algorithm.

5.7 Conclusion

In this chapter, we have proposed a 3D variational method that uses Census signature based data term. We have adopted a pyramidal approach to compute motion of small to large amplitudes. We compared our method against 3D Horn-and-Schunck and Amat's method. For visual assessment, we have put forward 3PHS and MIP-flow visualization schemes. We also defined the SAE measure, computed as an optical flow reconstruction error to assess the accuracy of the computed flow field. The SAE map can provide good intuition to where in the slices of volumes, the flow field might have been erroneous. The SAE map along with the average SAE (ASAE) is a means to perform quantitative evaluation in absence of the ground-truth flow fields. We have also drawn parallels between the standard optical flow error measure, EPE and the proposed error measure SAE. We found them to be similar in revealing the performance of the aforementioned optical flow methods. In our experimental study, we devised a pipeline to generate composite flow fields that can serve as a ground-truth flow field set. For the composition of the flow field, we employed our SAE measure so as to select relevant 3D vectors from a set of computed flow fields using known methods.

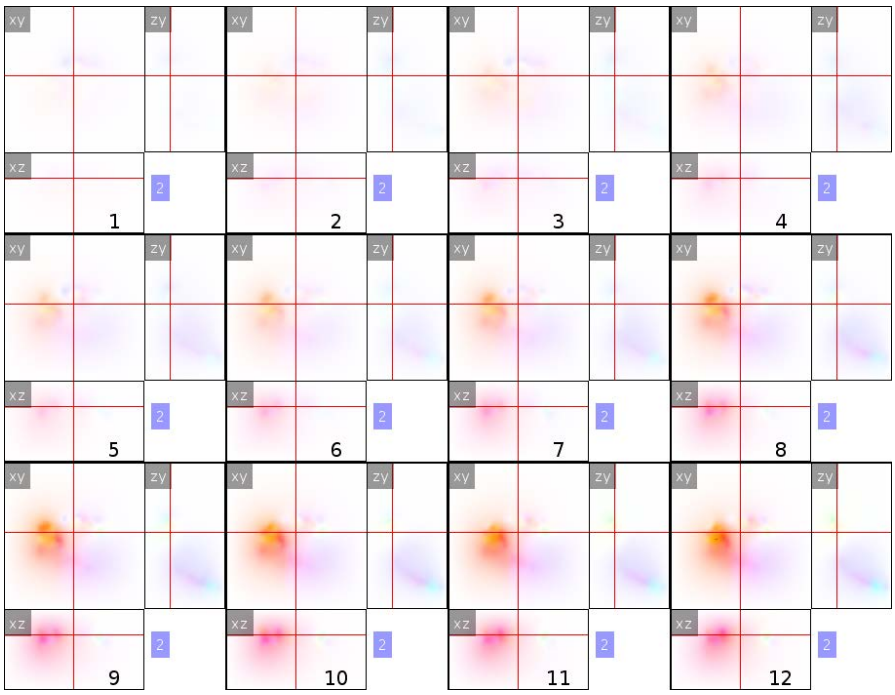
We have tested real sequences of the cell and the collagen motion, and used the SAE and the ASAE to assess the performance of the methods. We have also proposed a pre-processing of data term as z-mean adjustment to get cleaner flow fields in the foreground region while using our method with L_2 regularization for small number of iterations. We have also explored the potential of L_1 regularization which is able to produce discontinuous flow fields even without the foreground mask. However, the visual inspection of the computed flow field's 3PHS map and the ASAE measure suggested that the L_2-regularized flow fields performed better.

TABLE 5.4 – Percentile and average SAE of the three methods for the Dataset 2.

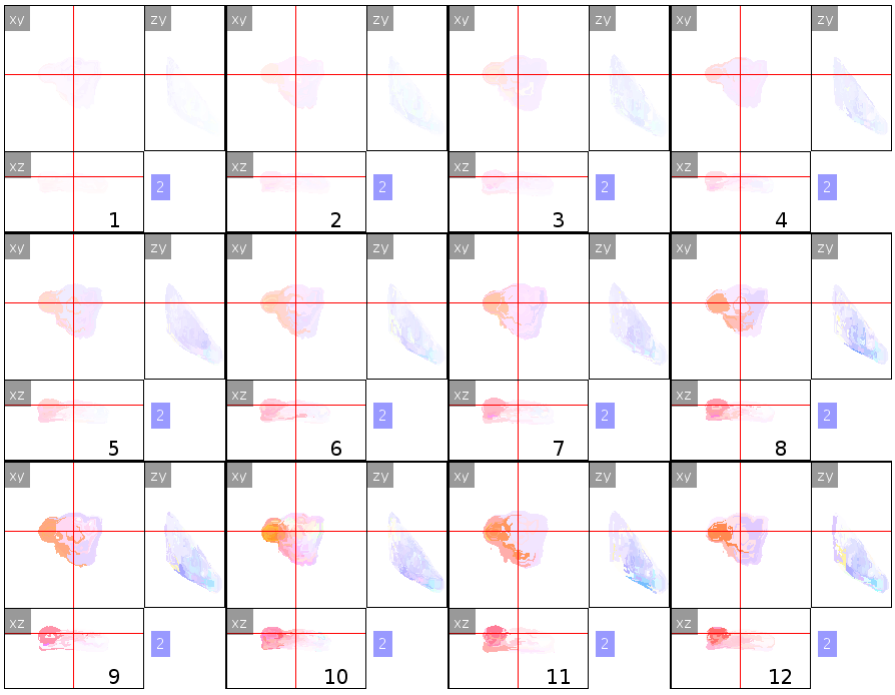
Time Step	Amat's method				Our method				3D Horn-and-Schunck			
	90%	95%	99%	Mean	90%	95%	99%	Mean	90%	95%	99%	Mean
1	0.0400	0.0619	0.1519	0.0187	**0.0311**	**0.0504**	**0.1280**	**0.0142**	0.0625	0.0981	0.2536	0.0372
2	0.0565	0.0867	0.2177	0.0268	**0.0418**	**0.0668**	**0.1700**	**0.0194**	0.1224	0.1982	0.5188	0.0594
3	0.0821	0.1241	0.2941	0.0376	**0.0566**	**0.0937**	**0.2450**	**0.0262**	0.1618	0.2597	0.6661	0.074
4	0.0737	0.1129	0.2670	0.0351	**0.0463**	**0.0721**	**0.1742**	**0.0217**	0.1926	0.2970	0.7438	0.0985
5	0.0751	0.1207	0.2851	0.0354	**0.0492**	**0.0796**	**0.2074**	**0.0235**	0.2310	0.3670	0.8769	0.1168
6	0.0833	0.1322	0.3322	0.0394	**0.0600**	**0.0961**	**0.2455**	**0.0281**	0.2395	0.3767	0.8804	0.1226
7	0.1145	0.1740	0.4124	0.0525	**0.0728**	**0.1139**	**0.3068**	**0.0342**	0.1939	0.3016	0.7735	0.0888
8	0.1457	0.2314	0.6179	0.0666	**0.0918**	**0.1447**	**0.4139**	**0.0421**	0.2406	0.3867	0.9539	0.107
9	0.1630	0.2598	0.6544	0.0742	**0.1009**	**0.1560**	**0.3887**	**0.0452**	0.2707	0.4403	1.0953	0.1129
10	0.1771	0.2827	0.7535	0.0799	**0.1061**	**0.1636**	**0.3956**	**0.0462**	0.3328	0.5324	1.1968	0.1512
11	0.1899	0.3175	0.8493	0.085	**0.1021**	**0.1581**	**0.4010**	**0.0465**	0.2642	0.4146	1.0474	0.1164
12	0.2200	0.3556	0.8618	0.0945	**0.1088**	**0.1663**	**0.4557**	**0.0502**	0.3205	0.4977	1.1377	0.1305

TABLE 5.5 – Percentile and average EPE of the three methods for the Dataset 2.

Time Step	Amat's method				Our method				3D Horn-and-Schunck			
	90%	95%	99%	Mean	90%	95%	99%	Mean	90%	95%	99%	Mean
1	0.1034	0.1284	0.1830	0.0599	**0.0699**	**0.0856**	**0.1203**	**0.0393**	0.2112	0.3014	0.5167	0.1005
2	0.1600	0.1827	0.2415	0.0987	**0.0946**	**0.1106**	**0.1436**	**0.0514**	0.3948	0.5223	0.8839	0.1808
3	0.2057	0.2437	0.3558	0.1383	**0.1270**	**0.1435**	**0.1803**	**0.0672**	0.5702	0.7677	1.4087	0.2577
4	0.1754	0.1969	0.2385	0.1688	**0.2692**	**0.3193**	**0.4482**	**0.0951**	0.8997	1.2744	2.5811	0.4317
5	0.2926	0.3143	0.3972	0.1994	**0.2442**	**0.2639**	**0.2950**	**0.1389**	1.1819	1.6533	3.2126	0.5632
6	0.3258	0.4227	0.5735	0.2122	**0.2292**	**0.2542**	**0.2922**	**0.1252**	1.3442	1.8666	3.3963	0.6554
7	0.4027	0.4888	0.7029	0.2692	**0.2877**	**0.3187**	**0.3750**	**0.157**	1.1518	1.6061	2.5673	0.4505
8	0.5396	0.6336	0.9344	0.3232	**0.3324**	**0.3722**	**0.4379**	**0.1829**	1.5812	2.2250	3.7033	0.6290
9	0.6330	0.7574	1.0325	0.3612	**0.3722**	**0.4228**	**0.5011**	**0.2026**	1.9034	2.6358	4.3211	0.7400
10	0.5580	0.7041	1.0407	0.3641	**0.3752**	**0.4249**	**0.5213**	**0.2017**	2.2076	3.0059	5.0898	0.9682
11	0.6679	0.8438	1.2138	0.3914	**0.4228**	**0.4742**	**0.5631**	**0.2221**	2.0086	2.8359	4.8273	0.7996
12	0.7631	0.9173	1.2433	0.4103	**0.4262**	**0.4795**	**0.5820**	**0.2321**	2.2608	3.2341	5.4237	0.9198



(a) 3PHS map of flow computed by our method for Dataset 2. The time step is marked at the mid-bottom of each sub-figure.



(b) 3PHS map of flow computed by Amat's method for Dataset 2. The time step is marked at the mid-bottom of each sub-figure.

FIGURE 5.15 – 3PHS map of the computed flow fields for Dataset 2.

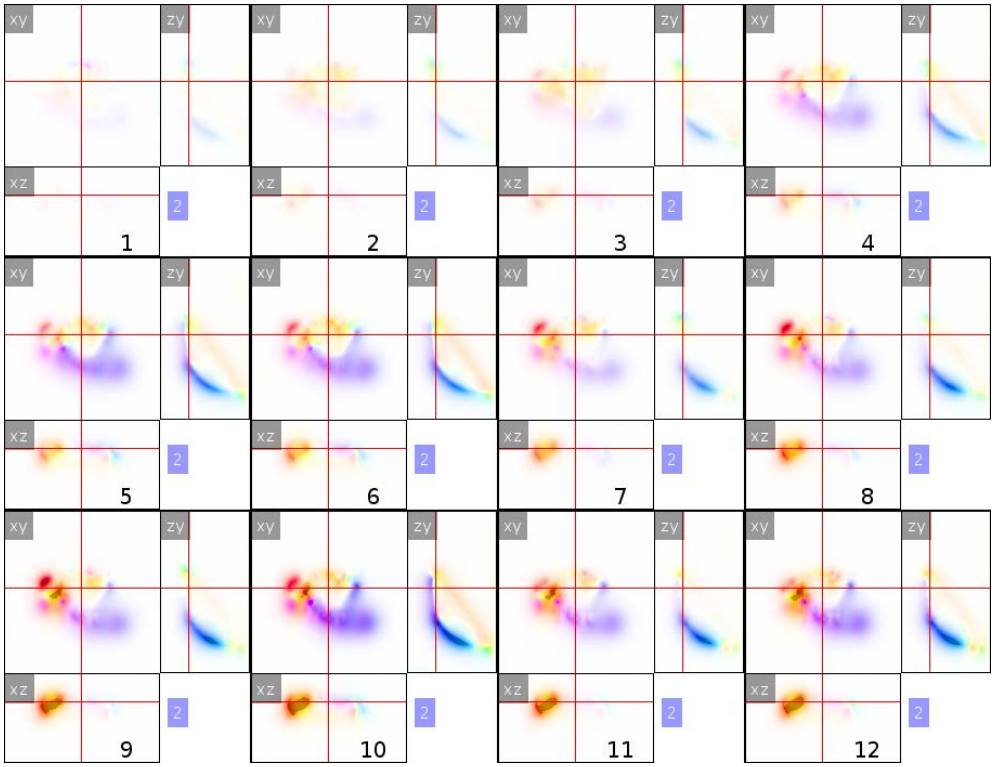
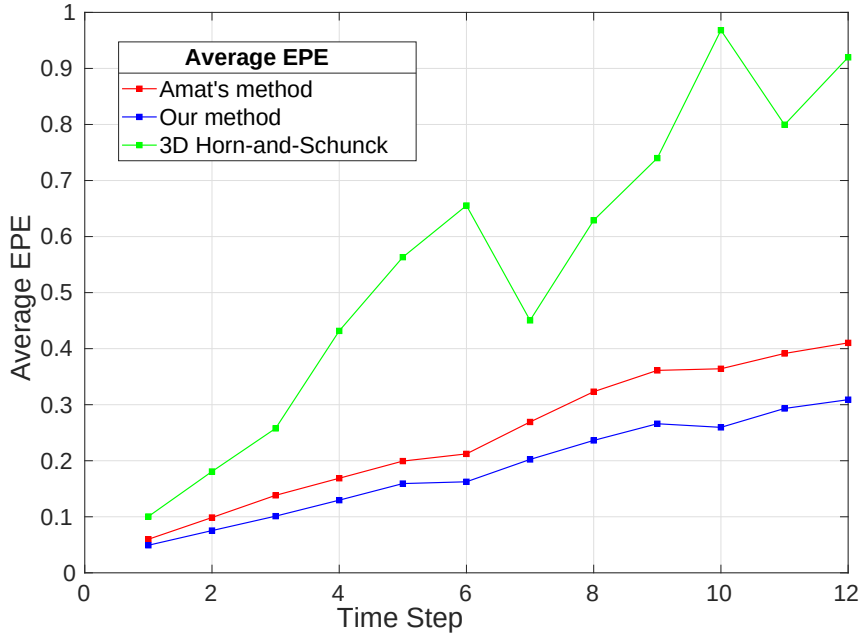
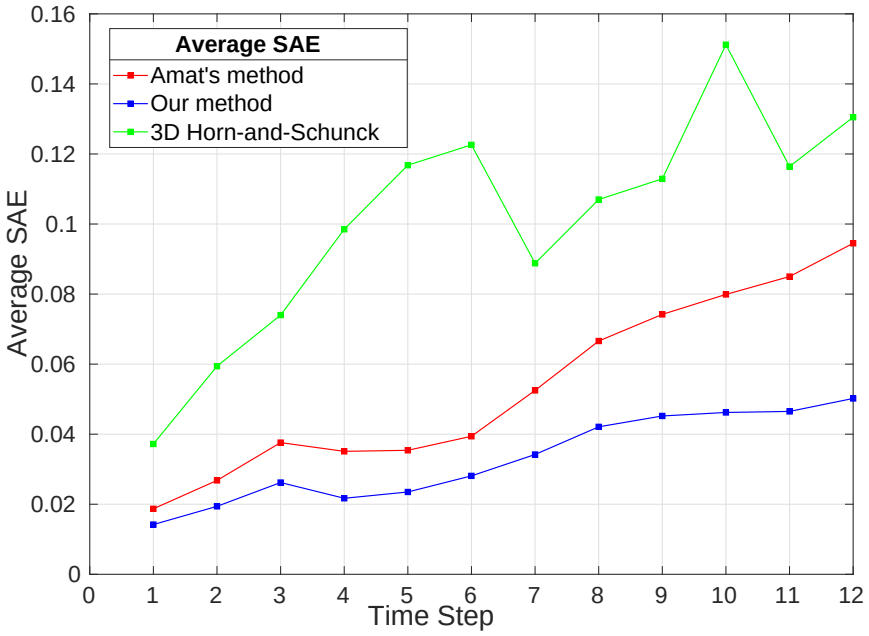


FIGURE 5.16 – 3PHS map of flow computed by 3D Horn-and-Schunck method for Dataset 2. The time step is marked at the mid-bottom of each sub-figure.

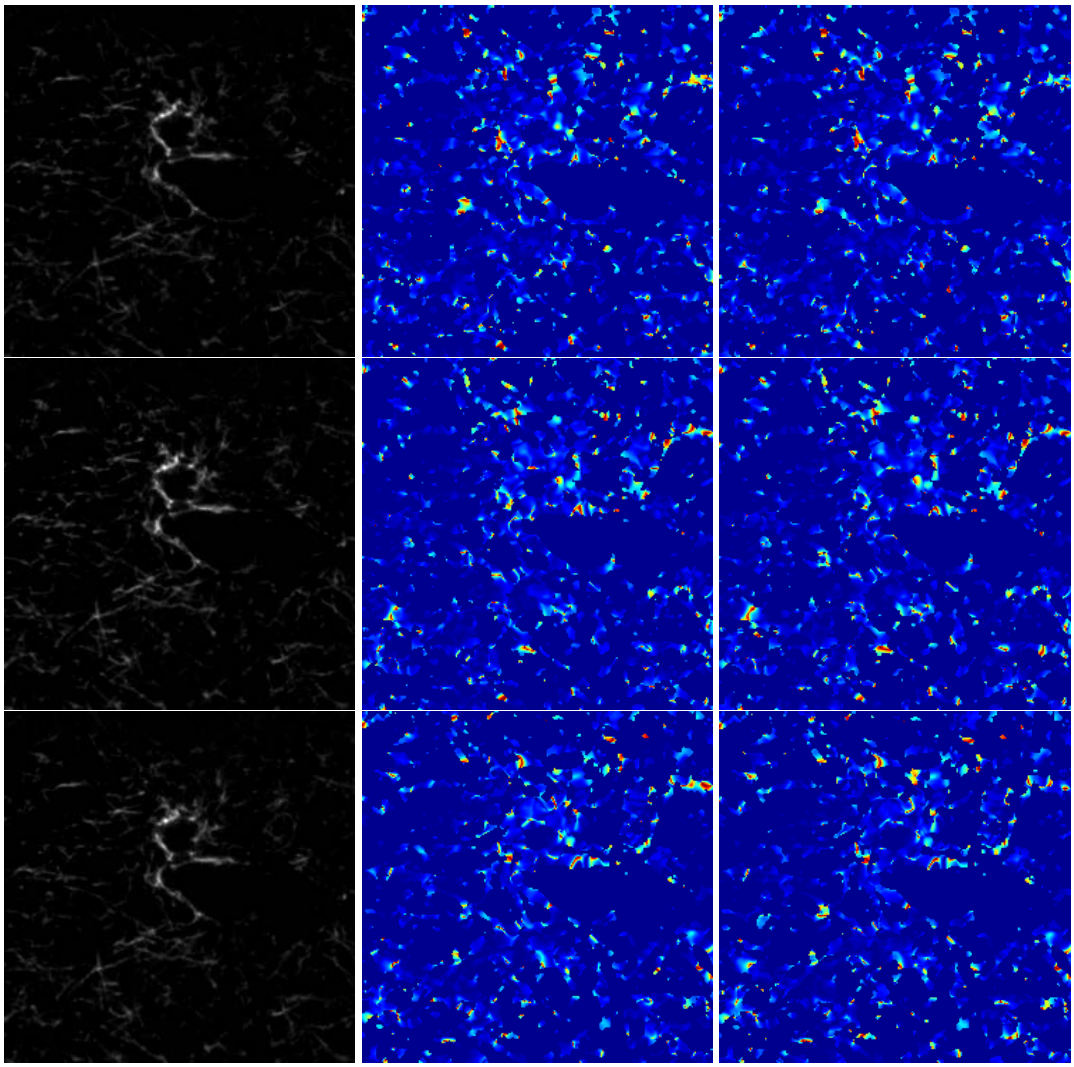


(a) Average End-Point-Error (EPE) of the flow fields computed by the three aforementioned methods for pairs of volumes separated by time steps mentioned in x-axis of the plot.



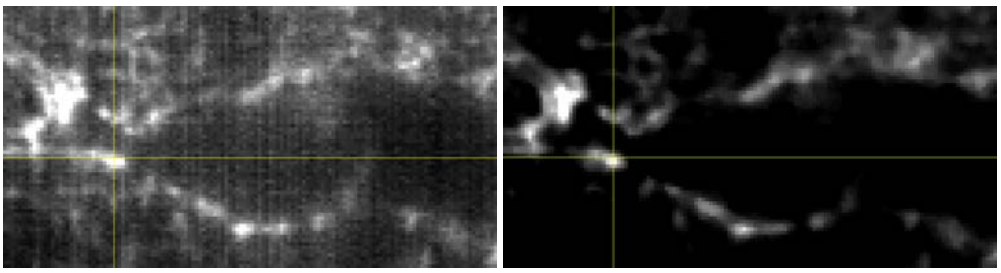
(b) Average Structural Angular Error (SAE) of the flow fields computed by the three aforementioned methods for pairs of volumes separated by time steps mentioned in x-axis of the plot.

FIGURE 5.17 – ASAE and AEPE for Dataset 2.

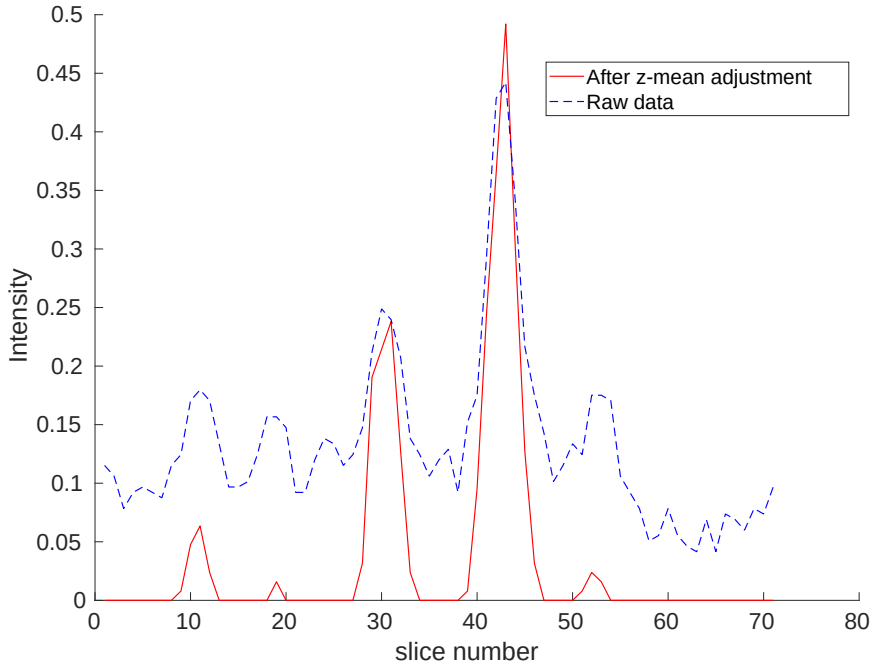


(a) Slices of source volumes (b) SAE of Amat's results (c) SAE of our results

FIGURE 5.26 – First column : Slices of XY plane for each source volume in Collagen 1. Second column : Heat map of SAE for Amat's method. Third column : Heat map of SAE for our method. The color code for the heat map is given at the bottom of the figure.

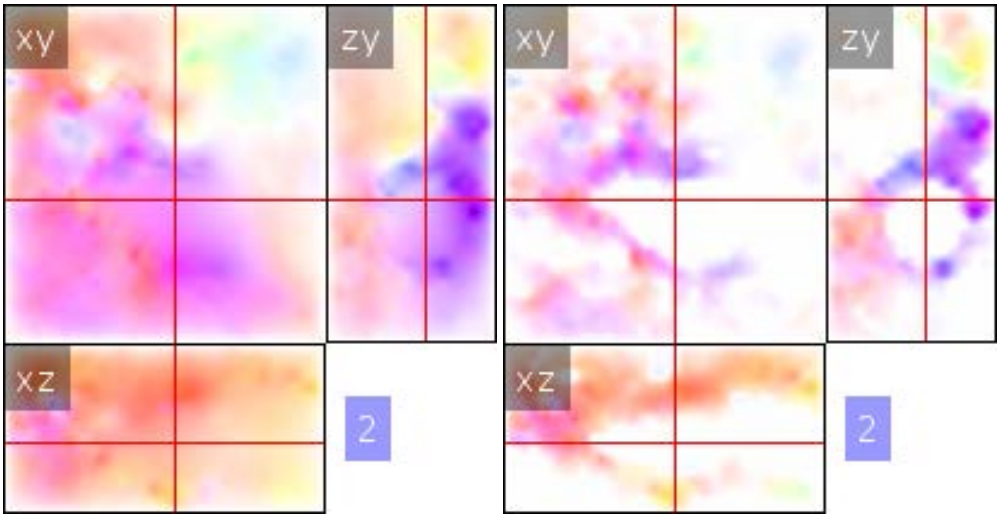


(a) XY plane of the raw data ; (b) After z mean adjustment



(c) Intensity profile along z-axis indicated by the yellow vertical line drawn in Figs. 5.28a and 5.28b

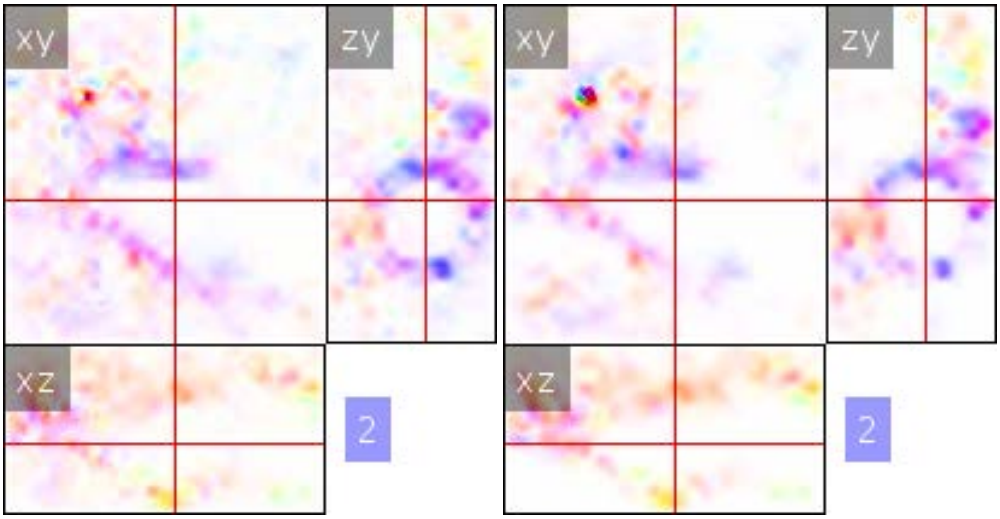
FIGURE 5.28 – The intensity profile after z-mean adjustment. The intensity values are normalized in range of [0, 1] for the plot.



(a) Without z-mean adjustment.
ASAE = 0.2593

(b) With z-mean adjustment.
ASAE = 0.2571

FIGURE 5.29 – 3PHS maps of the flow fields computed by our method with L_2 regularization.



(a) Without z-mean adjustment.
ASAE = 0.269

(b) With z-mean adjustment.
ASAE = 0.2634

FIGURE 5.30 – 3PHS map of the flow field computed by our method L_1 regularization.

Chapitre 6

Combination of the Matching and the Variational methods

6.1 Introduction

Often, a sequence contains large and small motions. 3D PatchMatch is able to handle large displacements but fails where sub-voxelic motion is present. Conversely, the variational method on its own is only robust for small displacements. We have demonstrated the coarse-to-fine approach to deal with large motion using variational method in Chapter 5. However, the number of levels to effectively scale down large motion is not easy to set a priori. Further, this approach fails to capture motion of smaller objects as they get lost at coarsest levels and subsequently in warping stages. In contrast, 3D PatchMatch does not face such issues. To deal with all possible large displacements, initial search region at the coarsest level for 3D PatchMatch is set to the whole volume domain. Hence, the number of coarser levels are not critical. There is no warping performed between levels, which preserves smaller objects to be present in successive finer levels.

In this chapter, we leverage the results of variational method with the flow fields obtained from 3D PatchMatch. We present two different ways to do so.

The first method combines the developments we have made so far in Chapter 4 and 5, i.e., coarse-to-fine 3D PatchMatch and our variational method with L_2 regularization. The flow field computed from 3D PatchMatch method is used as a preprocessing step, where a new target volume is formed that exhibits smaller displacements relative to the original source volume.

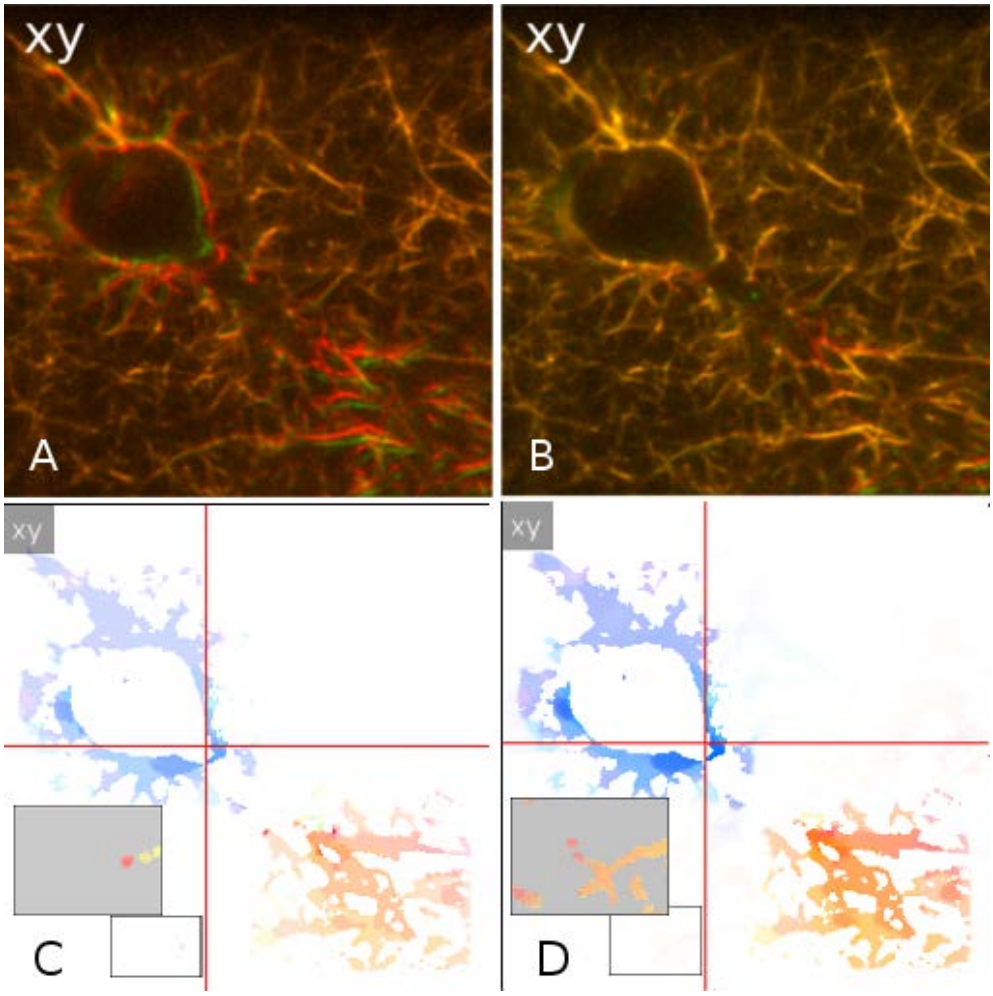


FIGURE 6.1 – (A) Overlap of slices of the 2^{nd} volume pair in the sequence Collagen 1 : source in red, target in green (showing only XY-plane). (B) Source in red, reconstructed source after 3D PatchMatch and our variational refinement in green ; the overlapped voxels appear in yellow. (C) 3PHS motion map using 3D PatchMatch alone. The small motion is not captured. (D) 3PHS motion map after 3D PatchMatch and our variational refinement. The red rectangle highlights the small motion captured by the combination. Value of the colors in the highlighting rectangle has been decreased to make the colors corresponding to small vectors more apparent. Foreground mask has been applied to the flow fields.

The new pair is then used as inputs to our variational method where they are processed to deal with smaller displacements.

Our second method is a novel 3D optical flow algorithm that uses the linearized Census signature constancy assumption as the data term and total variational regularization. Unlike the first method, this method directly accommodates the flow field obtained from 3D PatchMatch into the variational framework. The method is described in the second part of the chapter.

Algorithm 5 Combining 3D PatchMatch and our variational (L_2 regularized) method

INPUT : A pair of 3D images I_1 (source) and I_2 (target)
OUTPUT : 3D flow field $\boldsymbol{w}$

1: $\boldsymbol{w}_l \leftarrow 3DPatchMatch(I_1, I_2)$ ▷ $\boldsymbol{w}'$ is the semi dense flowfield
2: $\boldsymbol{w}_i \leftarrow GaussianInterpolation(\boldsymbol{w}_l)$ ▷ densify $\boldsymbol{w}_l$
3: $I_r(\boldsymbol{p}) \leftarrow I_2(\boldsymbol{p} + \boldsymbol{w}_i(\boldsymbol{p})), \forall \boldsymbol{p} \in \Omega$ ▷ Warp I_2
4: $\boldsymbol{w}_s \leftarrow \text{CSOR}(I_1, I_r)$ ▷ compute flow field $\boldsymbol{w}_s$ with Algorithm 2 in Chapter 5
5: $\boldsymbol{w} \leftarrow \boldsymbol{w}_s + \boldsymbol{w}_l$ ▷ sum up $\boldsymbol{w}_l$ and $\boldsymbol{w}_s$

6.1.1 Combining 3D PatchMatch with our variational method

To deal with mixed motion, we adopt an incremental approach as in Revaud *et al.* (2015). A step-by-step description of our implementation is provided as follows :

(1) The original the source volume I_1 and the target volume I_2 is used for computing large integer displacements using 3D PatchMatch method. Let the computed flow field be denoted by $\boldsymbol{w}_l$.

(2) $\boldsymbol{w}_l$ is smoothened and densified using the gaussian interpolation. This step also fills in the holes in w_l that might have been introduced during the outlier removal in 3D PatchMatch stage. Let the interpolated flow field be denoted by w_i.

(3) The target volume I_2 is backward warped with w_i, which reconstructs a close approximation of the source volume. Let the reconstructed volume be denoted by I_r. Between I_1 and I_r, there remains smaller displacements.

(4) Our variational method with L_2 regularization is used to compute the smaller displacements between I_1 and I_r. Let the computed flow field be denoted by w_s.

(5) The final flow field containing both large and small displacements is computed by the vector summation of w_l and w_s.

Algorithm 5 presents the steps involved.

6.1.2 Experimental Results

To demonstrate the use of the aforementioned combination of our methods, we experiment with the sequence Collagen 1. The sequence exhibits both large and small ranges of motion. 3D PatchMatch is used to compute the auxiliary flow field with patch of size $9 \times 9 \times 3$, three pyramid levels with 7, 5 and 1 iterations (coarse to fine levels, respectively) and the Hamming distance between the Census signatures of patches as the similarity measure. For the variational method, the smoothness parater $\alpha = 0.05$ is used with 50 SOR iterations.

Figure 6.1 (A) shows the overlap of slices of a source and a target volume in the sequence

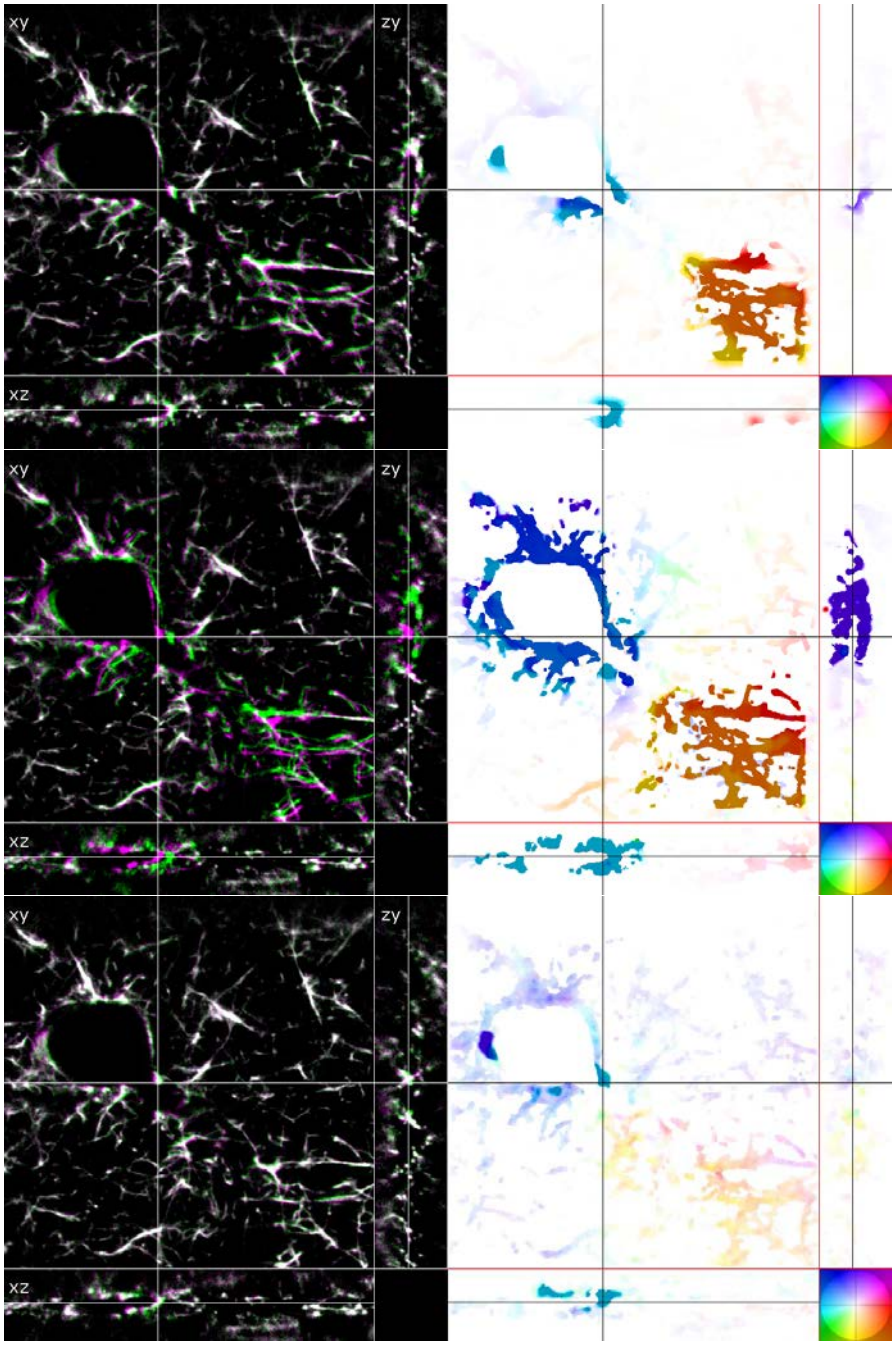


FIGURE 6.2 – Left : Overlap of slices of source(in red) and target(in green) volumes in the sequence Collagen 1. The correctly overlapping voxels are now rendered in white. The successive pairs in the sequence are placed from the top to the bottom of the figure. Right : The 3PHS map of the flow field computed by the combination of our 3D PatchMatch and our L_2 regularized method. In order to visualize smaller motion, the maximum range of the motion magnitude to normalize the saturation of the map is set to 3 units. Foreground mask has been applied to the flow fields.

which exhibits the largest displacement of collagen network. This is the event that takes place when the cell in the sequence MV3 C detaches itself from the network to move itself. The large displacements are observed in the networks of collagen to which the cell had adhered to. Small displacements in the nearby networks can be observed as well. Such smaller motion are not captured by 3D PatchMatch (see Fig. 6.1) (C). To compute both large and small motion, we rely on the combination of our 3D PatchMatch and the variational method. Figure 6.1 (B) depicts the overlap of slices of the original source and the reconstructed source by the flow field computed by the combination of our methods. The 3PHS map of the corresponding flow field is depicted in 6.1 (D), where we highlight the collagen strand having the small displacement which went unaccounted for in the results of 3D PatchMatch. Figure 6.2 shows the flow fields computed for all volume pairs in Collagen 1. The maximum range of motion to which the saturation of the 3PHS map has been normalized, is set to 3 units of displacement. This allows us to appreciate the smaller motion of the collagen strands. The sequence starts with smaller motion of the collagen strands where the cell is attached to. The large displacements occur in the second frame pair, when the cell detaches itself from the collagen. The detached collagen displaces by around 10 units. The motion is propagated to the near by collagen strands, which then exhibit smaller displacements. We are able to capture these displacements as well. The last frame pair also exhibits smaller motion, where the collagen strands follow the direction from the previous frames.

6.2 Combining 3D PatchMatch and the novel TV-based method

Our motivation for developing the TV-regularized optical flow algorithm is three fold : (i) integration of the flow field obtained from a matching algorithm (in our case, 3D PatchMatch), (ii) robust regularization of the flow field for discontinuity preservation, and (iii) faster execution time as it can be implemented in parallel manner. To our knowledge, a continuous version of Census signature (CS) has not been integrated with TV-regularization for the optical flow computation. On that ground, we present a novel TV algorithm for 3D motion estimation in fluorescence images which uses CS-based data term.

Total variation (TV) algorithm are known for being robust to noise. Rudin *et al.* (1992) proposed a TV algorithm for image denoising. The objective to denoise an image amounts to minimizing an energy function of the following form :

$$\min_{g} \left\{ \int_{\Omega} |\nabla g(\boldsymbol{p})| d\boldsymbol{p} + \frac{1}{2\theta} \int_{\Omega} (g(\boldsymbol{p}) - f(\boldsymbol{p}))^2 d\boldsymbol{p} \right\}, \tag{6.1}$$

where g is the denoised image, θ is a regularization parameter and $\boldsymbol{p} \in \Omega$. If $\theta \to 0$, then the output g is less likely to be denoised. The model uses total variation $|\nabla g|$ for regularization

and L_2 norm for the data fidelity term. Zach *et al.* (2007) reformulated the variational optical flow using the aforementioned model. Instead of minimizing for recovering an image, they minimized two scalar fields u, v, i.e., the components of 2D vector field. The motivation to do so was to produce smooth, yet boundary preserving flow fields. The data-fidelity term in the original model in Eq. 6.1 was treated as an auxiliary term in their formulation. We write their formulation as :

$$E(\boldsymbol{w}, \tilde{\boldsymbol{w}}) = E_{data} + E_{auxiliary} + E_{smooth}, \tag{6.2}$$

$$E_{data} = \int_{\Omega} |I(\boldsymbol{p} + \boldsymbol{w}(\boldsymbol{p})) - I(\boldsymbol{p})| d\boldsymbol{p}, \tag{6.3}$$

$$E_{auxiliary} = \int_{\Omega} \frac{1}{2\theta} (\boldsymbol{w}(\boldsymbol{p}) - \tilde{\boldsymbol{w}}(\boldsymbol{p}))^2 d\boldsymbol{p}, \tag{6.4}$$

$$E_{smooth} = \int_{\Omega} \alpha(\|\boldsymbol{w}(\boldsymbol{p})\|) d\boldsymbol{p}, \tag{6.5}$$

where $\tilde{\boldsymbol{w}}$ is the auxiliary field which has similar role to f in Eq. 6.1. Therefore, $\tilde{\boldsymbol{w}}$ field should be a close approximation of $\boldsymbol{w}$. Here, the brightness constancy assumption was used to form the data term. In contrast, we propose to integrate the 3D continuous Census signature as the data term, as described in Chapter 5. We recall our data term as :

$$E_{data} = \int_{\Omega} \sum_{n=0}^{N-1} H'_{\epsilon}(\partial_n I(\boldsymbol{p})).(\boldsymbol{w}(\boldsymbol{p})^{\top} \nabla(\partial_n I(\boldsymbol{p}))) d\boldsymbol{p}, \tag{6.6}$$

where $N = 26$, i.e., the size of 26-connected neighbourhood of $\boldsymbol{p}$, and ∂_n is the directional derivative operator. The energy related to the data term can now be separated from the gradients of the flow field. This enables us to split the minimization of Eq. 6.2 into two halves. The first half solves for $\boldsymbol{w}$ while the second half solves for $\tilde{\boldsymbol{w}}$.

The objective function corresponding to the first half is given as :

$$J_1(u, \tilde{u}) = \int_{\Omega} \left(\alpha |\nabla u(\boldsymbol{p})| + \frac{1}{2\theta} (u(\boldsymbol{p}) - \tilde{u}(\boldsymbol{p}))^2 \right) d\boldsymbol{p}, \tag{6.7}$$

where u and $\tilde{u}$ are the scalar components of 3D flow fields $\boldsymbol{w}$ and $\tilde{\boldsymbol{w}}$. For the remaining two scalar components v and w, similar objective functions are defined. The minimization is done with respect to $\boldsymbol{w}$. We name this minimization Sub-problem #1.

The objective function corresponding to the second half is given as :

$$J_2(\boldsymbol{w}, \tilde{\boldsymbol{w}}) = \int_{\Omega} \left(\|D(I(\boldsymbol{p}), \tilde{\boldsymbol{w}}(\boldsymbol{p}))\|^2 + \frac{1}{2\theta} \|\boldsymbol{w}(\boldsymbol{p}) - \tilde{\boldsymbol{w}}(\boldsymbol{p})\|^2 \right) d\boldsymbol{p}, \tag{6.8}$$

where D is the discretized data term corresponding to CS constancy assumption, which was developed in Chapter 5. Here, D is linearized with respect to $\tilde{\boldsymbol{w}}$ and given as :

$$D(I(\boldsymbol{p}), \tilde{\boldsymbol{w}}(\boldsymbol{p})) = \frac{1}{N} \sum_{n=1}^{N} H'_{\epsilon}(\partial_n I(\boldsymbol{p})).(\tilde{\boldsymbol{w}}(\boldsymbol{p})^{\top} \nabla(\partial_n I(\boldsymbol{p})), \tag{6.9}$$

The minimization is done with respect to $\tilde{\boldsymbol{w}}$. We name this minimization problem (related to Eq. 6.8) Sub-problem #2.

Hereafter, we describe the solution to the aforementioned sub-problems.

6.2.1 Sub-problem #1

We rewrite the energy function for Sub-problem #1 as :

$$J_1(u,\tilde{u}) = \int_{\Omega} \left(\alpha |\nabla u(\boldsymbol{p})| + \frac{1}{2\theta}(u(\boldsymbol{p}) - \tilde{u}(\boldsymbol{p}))^2 \right) d\boldsymbol{p}. \tag{6.10}$$

The Eq. 6.10 resembles the total variation denoising model presented in Eq. 6.1, which can be solved by Chambolle's duality-based algorithm (Chambolle 2004). This solution has been suggested in (Zach *et al.* 2007), (Wedel *et al.* 2009) (Mohamed and Mertsching 2012). For the sake of completeness, we show the derivation of the algorithm for u component (v, w follow similarly).

The E.L. equation related to Eq. 6.10 is given by :

$$(u(\boldsymbol{p}) - u(\tilde{\boldsymbol{p}})) - \alpha_1 \mathrm{div}\left(\frac{\nabla u(\boldsymbol{p})}{|\nabla u(\boldsymbol{p})|} \right) = 0, \tag{6.11}$$

where $\alpha_1 = \alpha.\theta$. The equation is undefined for $|\nabla u(\boldsymbol{p})| = 0$. Therefore, we use the robust norm as $\psi(|\nabla u(\boldsymbol{p})|) = \sqrt{|\nabla u(\boldsymbol{p})|^2 + \epsilon}$ with ϵ as a very small positive number.

Let $\boldsymbol{x}_u, \boldsymbol{x}_v, \boldsymbol{x}_w$ be the dual vector fields defined as :

$$\boldsymbol{x}_u(\boldsymbol{p}) = \frac{\nabla u(\boldsymbol{p})}{\psi(|\nabla u(\boldsymbol{p})|)}, \tag{6.12}$$

$$\therefore \boldsymbol{x}_u(\boldsymbol{p})\psi(|\nabla u(\boldsymbol{p})|) - \nabla u(\boldsymbol{p}) = 0. \tag{6.13}$$

$\boldsymbol{x}_v, \boldsymbol{x}_w$ are defined similarly. Now, we rewrite Eq. 6.11 as :

$$(u(\boldsymbol{p}) - \tilde{u}(\boldsymbol{p})) - \alpha_1 \mathrm{div}(\boldsymbol{x}_u(\boldsymbol{p})) = 0 \implies u(\boldsymbol{p}) = \alpha_1.\mathrm{div}(\boldsymbol{x}_u(\boldsymbol{p})) + \tilde{u}(\boldsymbol{p}). \tag{6.14}$$

From Eqs. 6.13 and 6.14, we get :

$$\boldsymbol{x}_u(\boldsymbol{p})\psi(\|\nabla(\mathrm{div}(\boldsymbol{x}_u(\boldsymbol{p})) + \tilde{u}(\boldsymbol{p})/\alpha_1)\|) - \nabla(\mathrm{div}(\boldsymbol{x}_u(\boldsymbol{p})) + \tilde{u}(\boldsymbol{p})/\alpha_1) = 0, \tag{6.15}$$

which is solved by using fixed-point iterations as :

$$\boldsymbol{x}_u^{k+1}(\boldsymbol{p}) = \frac{\boldsymbol{x}_u^k(\boldsymbol{p}) + \tau\nabla(\mathrm{div}(\boldsymbol{x}_u^k(\boldsymbol{p})) + \tilde{u}(\boldsymbol{p})/\alpha_1)}{1 + \tau\psi(\left\|\nabla(\mathrm{div}(\boldsymbol{x}_u^k(\boldsymbol{p})) + \tilde{u}(\boldsymbol{p})/\alpha_1)\right\|)}. \tag{6.16}$$

6.2.2 Sub-problem #2

Solution to Sub-problem #1 leads to refined $\boldsymbol{w}$, thanks to its close approximate $\tilde{\boldsymbol{w}}$. However, $\tilde{\boldsymbol{w}}$ can be further refined using the CS constancy assumption. For fixed $\boldsymbol{w}$, we minimize the following energy function :

$$J_2(\boldsymbol{w}, \tilde{\boldsymbol{w}}) = \int_{\Omega} \left(\|D(I(\boldsymbol{p}), \tilde{\boldsymbol{w}}(\boldsymbol{p}))\|^2 + \frac{1}{2\theta} \|\boldsymbol{w}(\boldsymbol{p}) - \tilde{\boldsymbol{w}}(\boldsymbol{p})\|^2 \right) d\boldsymbol{p}. \tag{6.17}$$

We drop notation $\boldsymbol{p}$ (for the sake of brevity), and set the first-order partial derivatives of Eq. 6.17 to zeros as :

$$\frac{\partial J_2(\boldsymbol{w}, \tilde{\boldsymbol{w}})}{\partial \tilde{u}} = \frac{1}{N} \sum_{n=0}^{N-1} H_\epsilon'^2(\tilde{u}\partial_n(I_x) + \tilde{v}\partial_n(I_y) + \tilde{w}\partial_n(I_z) + \partial_n I_t)\partial_n(I_x) + \theta^{-1}(\tilde{u} - u) = 0, \tag{6.18}$$

$$\frac{\partial J_2(\boldsymbol{w}, \tilde{\boldsymbol{w}})}{\partial \tilde{v}} = \frac{1}{N} \sum_{n=0}^{N-1} H_\epsilon'^2(\tilde{u}\partial_n(I_x) + \tilde{v}\partial_n(I_y) + \tilde{w}\partial_n(I_z) + \partial_n I_t)\partial_n(I_y) + \theta^{-1}(\tilde{v} - v) = 0, \tag{6.19}$$

$$\frac{\partial J_2(\boldsymbol{w}, \tilde{\boldsymbol{w}})}{\partial \tilde{w}} = \frac{1}{N} \sum_{n=0}^{N-1} H_\epsilon'^2(\tilde{u}\partial_n(I_x) + \tilde{v}\partial_n(I_y) + \tilde{w}\partial_n(I_z) + \partial_n I_t)\partial_n(I_z) + \theta^{-1}(\tilde{w} - w) = 0. \tag{6.20}$$

The solutions to the system of the equations can be found using Jacobi iterations. The iterative solution is given by the following set of equations :

$$\tilde{u}^{k+1} = u^k - \frac{\sum_{n=0}^{N-1} H_\epsilon'^2(u^k\partial_n(I_x) + v^k\partial_n(I_y) + w^k\partial_n(I_z) + \partial_n I_t)\partial_n(I_x)}{\sum_{n=0}^{N-1} H_\epsilon'^2(\partial_n(I_x)^2 + \partial_n(I_y)^2 + \partial_n(I_z)^2) + N\theta^{-1}}, \tag{6.21}$$

$$\tilde{v}^{k+1} = v^k - \frac{\sum_{n=0}^{N-1} H_\epsilon'^2(u^k\partial_n(I_x) + v^k\partial_n(I_y) + w^k\partial_n(I_z) + \partial_n I_t)\partial_n(I_y)}{\sum_{n=0}^{N-1} H_\epsilon'^2(\partial_n(I_x)^2 + \partial_n(I_y)^2 + \partial_n(I_z)^2) + N\theta^{-1}}, \tag{6.22}$$

$$\tilde{w}^{k+1} = w^k - \frac{\sum_{n=0}^{N-1} H_\epsilon'^2(u^k\partial_n(I_x) + v^k\partial_n(I_y) + w^k\partial_n(I_z) + \partial_n I_t)\partial_n(I_z)}{\sum_{n=0}^{N-1} H_\epsilon'^2(\partial_n(I_x)^2 + \partial_n(I_y)^2 + \partial_n(I_z)^2) + N\theta^{-1}}. \tag{6.23}$$

The auxiliary flow field can be obtained using a coarse-to-fine scheme or correspondences from a matching algorithm. We have used 3D PatchMatch to obtain $\tilde{\boldsymbol{w}}$. We use Algorithm 6, to solve the aforementioned sub-problems for the global minimization. At the end of each fixed-point iteration, 3D median filter is applied to remove the outliers (Sun *et al.* 2010),(Wedel *et al.* 2009). Few warpings are required (in the same level of volume resolution) in order to remove non-linearities.

The complexity of the algorithm has increased with respect to the one proposed in Chapter 5, as dual vector fields and their divergences have to be computed in each fixed-point iteration. However, each sub problems can be solved in a parallel manner, which can accelerate the execution speed of the method.

Algorithm 6 Minimization with TV

Input I_1 = Source volume , I_2 = Target volume, $\tilde{\boldsymbol{w}}$ = Initial flow field, α_1 = smoothness parameter, ϵ = CS approximation factor, θ^{-1} = coupling factor, η = stopping threshold, N_{WARP} = number of warpings (Outer loop) , N_{SOR} = maximum number of solver iterations (Solver loop).
Output $\boldsymbol{w}$ = final flow field.

Initialize $\boldsymbol{w} \leftarrow \tilde{\boldsymbol{w}}$
Initialize $\boldsymbol{x}_u \leftarrow 0, \boldsymbol{x}_v \leftarrow 0, \boldsymbol{x}_w \leftarrow 0$
for `s=1 to` N_{WARP} **do** ▷ Outer loop
    Compute $I_2(\boldsymbol{p}+\boldsymbol{w}(\boldsymbol{p}))$, and its image gradients $I_{2x}(\boldsymbol{p}+\tilde{\boldsymbol{w}}(\boldsymbol{p}))$, $I_{2y}(\boldsymbol{p}+\tilde{\boldsymbol{w}}(\boldsymbol{p}))$, $I_{2z}(\boldsymbol{p}+\tilde{\boldsymbol{w}}(\boldsymbol{p}))$.
    Compute $\partial_n I_2, \partial_n I_{2x}, \partial_n I_{2y}, \partial_n I_{2z}$
    Compute $H'_\epsilon(\partial_n I_{2_n})$
    while $k < N_{SOR}$ `and stop_criterion >` η **do** ▷ Fixed point iterations
        Solve for $\tilde{u}^k, \tilde{v}^k, \tilde{w}^k$ using Eqs. 6.21,6.22,6.23
        $u \leftarrow \tilde{u}^k + \alpha_1 \mathrm{div}(\boldsymbol{x}_u)$, similarly for v, w.
        Solve Eq. 6.16 and the respective Eqs. for $\boldsymbol{x}_v$ and $\boldsymbol{x}_w$.
        $stop\_criterion \leftarrow \left\| \boldsymbol{w}^k - \boldsymbol{w}^{k+1} \right\|_2$
        $k \leftarrow k+1$
    $\boldsymbol{w} \leftarrow$ 3Dmedian-filter($\boldsymbol{w}$)
return $\boldsymbol{w}$

6.2.3 Preliminary Results

We have tested the proposed TV-based method on two collagen sequences. The first experiment demonstrates the improvements we have made using our TV-based method compared to our previous L_1 and L_2-regularization-based methods (in Chapter 5). The second experiment demonstrates the integration of the results of 3D PatchMatch as the auxiliary flow field in our TV-regularization-based method.

First, we compare the 3D flow field computed by our methods with L_1, L_2 and TV-regularization as well as our extension of (Zach *et al.* 2007) method to 3D (Zach3D) for Collagen 2. Zach3D, though uses TV-regularization, it exploits the brightness constancy assumption in the data term as shown in Eq. 6.3. This is where our TV-regularized method differs from Zach3D, as we exploit the CS constancy assumption using Eq. 6.6. Figure 6.3 juxtaposes the 3PHS maps of the flow field computed by the aforementioned methods. The ASAE measure is reported in the caption of the respective figures. The flow field has been computed for the volume pair without z-mean adjustment (discussed in Chapter 5). Also, no foreground mask has been applied to generate the 3PHS map.

Now, let us focus on subfigures of Fig. 6.3 (b-d). We intend to demonstrate the improvements we have achieved using TV-regularization. We clarify that these results were produced using the same data term but different regularization schemes. Over-smoothing of the flow field is

TABLE 6.1 – Execution time for a single outer-loop enclosing 10 fixed-point iterations for our methods.

Our Methods	Time per outer-loop (10 SOR iterations)
L_1 regularized	510 seconds
L_2 regularized	482 seconds
TV regularized (4 CPU cores)	260 seconds

evident for the results obtained with L_2-regularization, as the vectors in the unlabelled region also exhibit some motion (see Fig. 6.3 (b)). Our method with L_1-regularization is able to produce the flow field while preferring discontinuities (see Fig. 6.3 (c)). However, the flow field requires improvement of smoothness in the labelled region. Our method with TV-regularization produces the flow field with desired traits of being discontinuous between the labelled and the unlabelled regions, and being smooth in the labelled region (see Fig. 6.3 (d)). Furthermore, it improves the ASAE measure in the demonstrated sequence.

Now, we focus on subfigures of Fig. 6.3 (d-e) corresponding to our TV-regularized method and Zach3D. We reiterate that between these methods, only the data-term differs. Figure 6.3e depicts the 3PHS map of the flow field computed by Zach3D. Though the computed flow field is discontinuous, the computation of ambiguous vectors in unlabelled region demonstrates that the BCA-based data term is inferior to our proposed Census signature-based data term.

We have also computed flow fields for the sequence using Amat's method. The method employs a foreground mask that produces flow fields with sharp boundaries. This makes its comparison with the aforementioned methods unfair. Sub-figures 6.4a and 6.4b show the result with and without using the foreground mask, respectively. Piecewise constant flow fields in the super-voxel regions are evident in both cases.

We report the execution time for our three methods for a single outer-loop iteration enclosing 10 fixed-point iterations in Table 6.1. A volume pair of size $290 \times 182 \times 115$ was used. The methods were coded in C++ and executed in a computer with Intel Core i7 CPU (2.80 Hz), 16 GB of memory running Ubuntu 16.04 OS. Four CPU cores were employed to execute our TV-regularization-based method, while the other two methods have been implemented for single core CPU.

In our next experiment, we combine 3D PatchMatch and the proposed TV-regularization-based method. We integrate the flow field obtained from 3D PatchMatch as the auxiliary flow field $\tilde{\boldsymbol{w}}$ of the proposed method. The infuence of input $\tilde{\boldsymbol{w}}$ on the output $\boldsymbol{w}$ is controlled by θ^{-1}. The higher the value of θ^{-1} is, closer the solution $\boldsymbol{w}$ is in agreement to $\tilde{\boldsymbol{w}}$. However, it is difficult to determine the parameter. Zach *et al.* (2007) do not explain how to choose the value or range

of values. Pock *et al.* (2007), in context of MRI image registration empirically set it to 50, while Hermann and Werner (2014) set it to 10. These methods rely on the coarse-to-fine scheme to estimate the auxiliary fields in different scales of image pair. We also resort to an empirical approach to set the values.

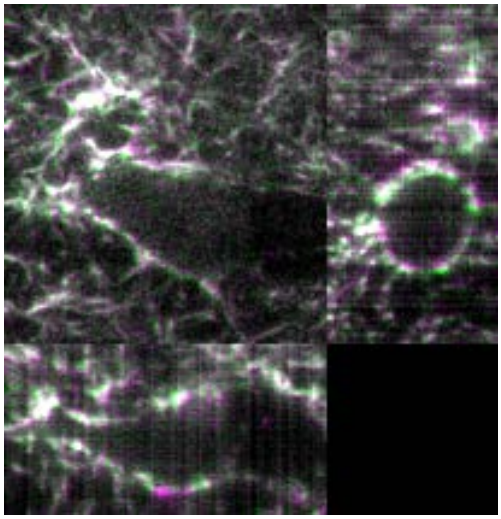
For this, we consider a volume pair in Collagen 1. The flow field computed with 3D PatchMatch is used as the auxiliary flow field. For the variational step with our TV-regularization, no coarse-to-fine method is applied. However, 5 outer loops to deal with possible linearities are used. 15 fixed-point iterations are employed per outer loop. The CS approximation factor and the smoothness parameter are set as $\epsilon = 80$ and $\alpha = 0.0015$, respectively. We tested several values of θ^{-1}, however we report three different values of θ^{-1} : 0.01 as small, 0.1 as mid and 1 as high values. Figure 6.5 depicts the 3PHS maps of the computed flow fields. The maximum range of motion amplitude to which the saturation of all the maps have been normalized to is 3 units. Figure 6.5 (A) corresponds to the auxiliary flow field computed by 3D PatchMatch. Figure 6.5 (B) corresponds the the flow field computed by our TV-regularized method for $\theta^{-1} = 0.01$. Here, the method is able to compute smaller displacements. However, irregularities appear in the resulting flow field. As θ^{-1} takes a small value, the minimizing solution is no longer required to be close to the auxiliary flow field. In this case, it is up to the regularization term, which is now uncoupled from the data term, to compute the flow field. For $\theta^{-1} = 0.1$, as seen in Fig. 6.5 (C), the smaller displacements are captured, and the flow fields are smooth. For $\theta^{-1} = 1$, smaller displacements could not be captured as seen in Fig. 6.5 (D). The solution is highly constrained to be similar to the auxiliary flow field which does not possess small displacements in the first place.

Figure 6.6a shows another collagen network in motion with the slices of source (in green) and target (in magenta) of the sequence. In this experiment, we compute discrete flow field using our 3D PatchMatch method with parameters defined in Subsection 6.1.2. The computed flow field is depicted in Fig. 6.6b. The flow field is refined by the proposed TV-based method for $\theta^{-1} = 0.1$ as depicted in Fig. 6.6c. The refined flow field is smoother than that computed by 3D PatchMatch. Once again, the failure of 3D PatchMatch to compute small motion vector is evident when we compare the corresponding 3PHS maps in ZY-plane.

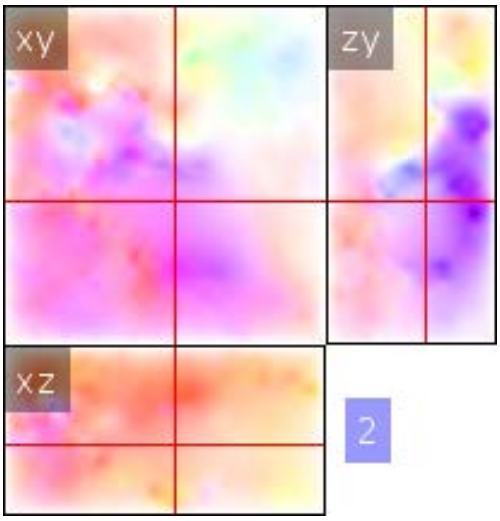
6.3 Conclusion

In this chapter, we have described two ways of combining 3D flow fields computed with 3D PatchMatch and a refinement based on the variational method. In the first method, we first compute large displacements with 3D PatchMatch. The resulting flow field is interpolated and used for warping the target volume to reconstruct the source volume. The reconstructed source

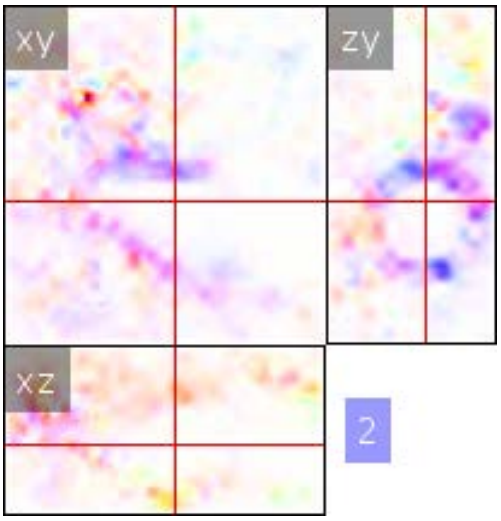
and the original source volumes are then used as inputs to our variational method described in Chapter 5 to properly handle the smaller displacements. The final flow field is the sum of the interpolated flow field and the output of the variational method. Our second method uses the interpolated flow field of the matching stage as an auxiliary flow field. It uses TV-regularization which enables discontinuity preservation in the 3D flow fields. The preliminary tests show better flow fields than that obtained with our previous L_1 and L_2-regularization-based methods and 3D PatchMatch.The method is parallelization friendly and we could improve the execution time to compute flow fields for larger volumes.



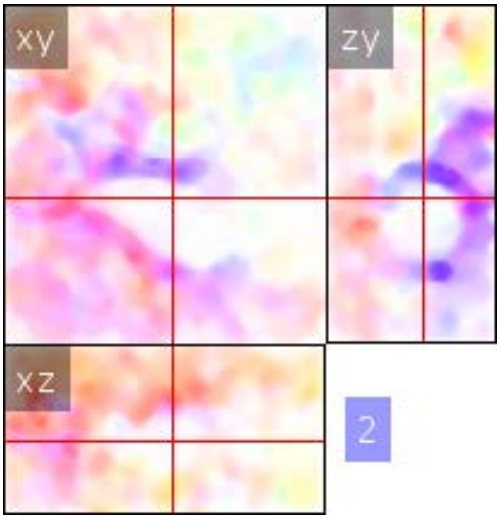
(a) Overlap of slices of source and target volumes.



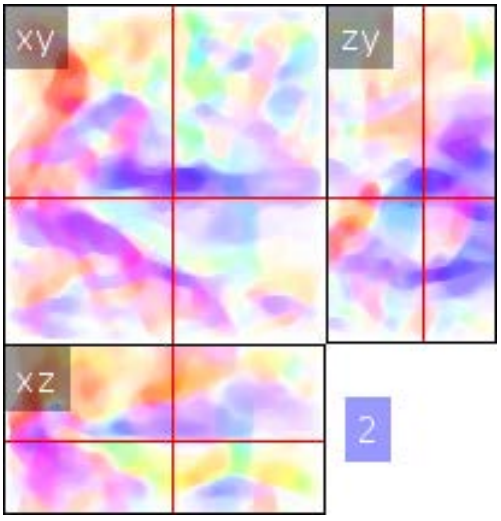
(b) With our L_2-regularization.
ASAE = 0.2593



(c) With our L_1-regularization.
ASAE = 0.269

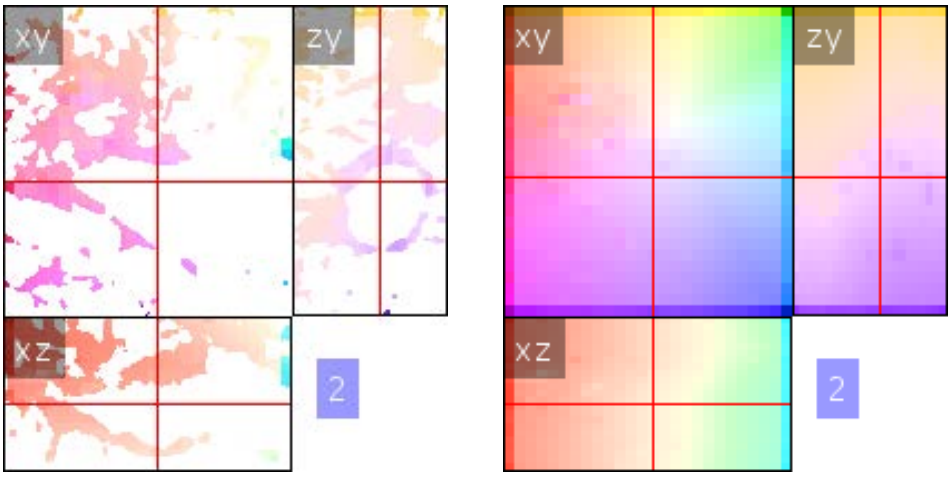


(d) With our TV-regularization.
ASAE = 0.233



(e) With Zach3D.
ASAE = 0.3249

FIGURE 6.3 – (a) Overlap of orthogonal slices of the source (green) and target (magenta) volumes in Collagen 2. (b-f) 3PHS map of the corresponding motion field computed by our method with L_2, L_1, and TV regularization, and Zach3D, respectively.



(a) with foreground mask. (b) without foreground mask.

FIGURE 6.4 – 3PHS map of the flow field computed using Amat's method (a) with foreground mask (b) without foreground mask.

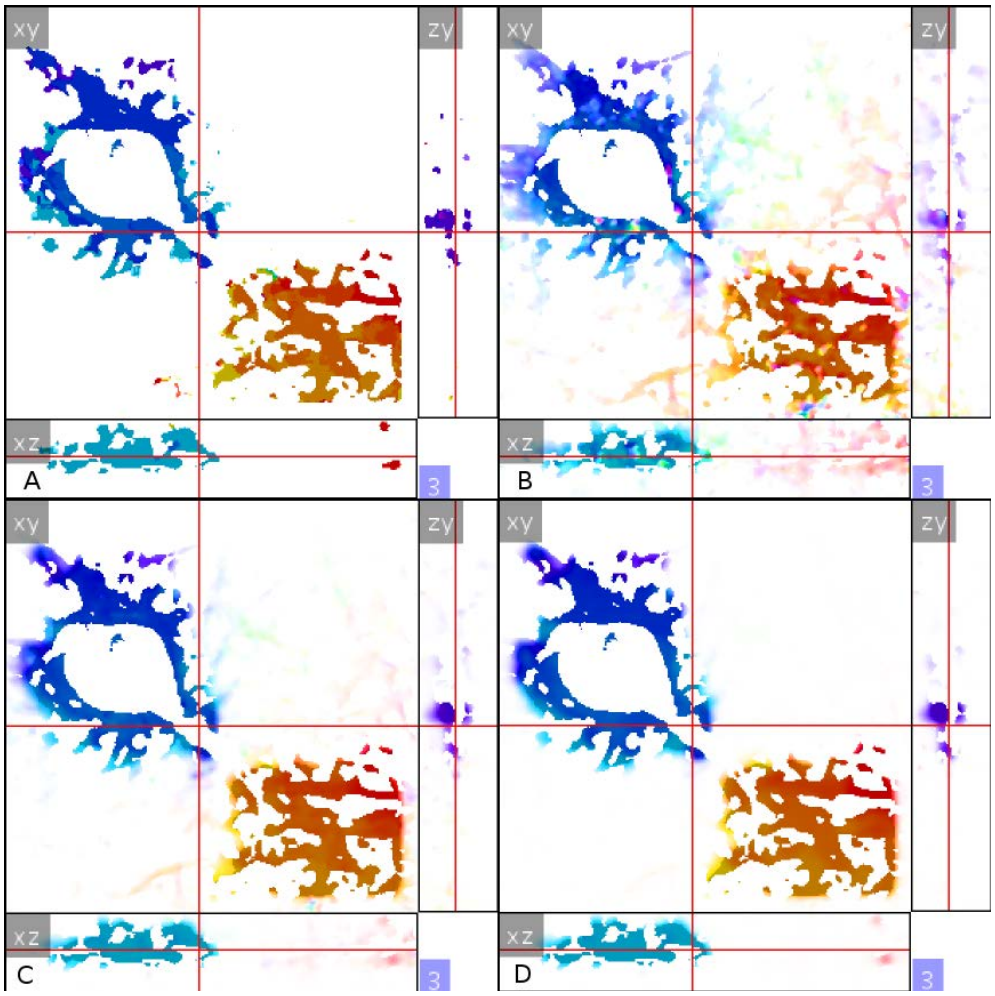
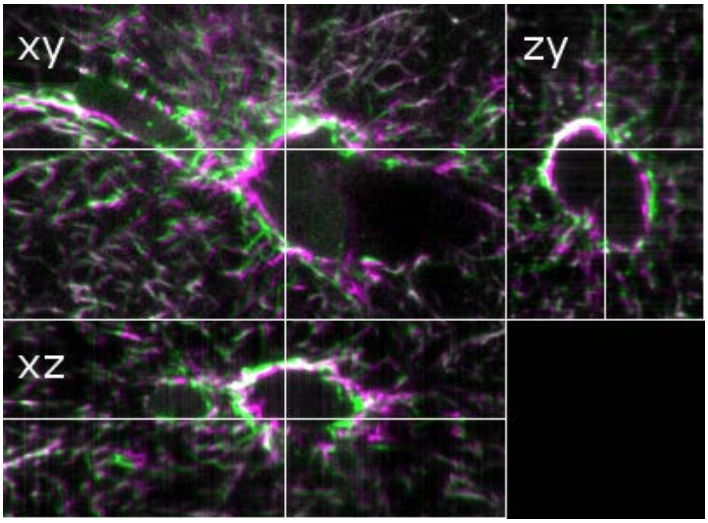
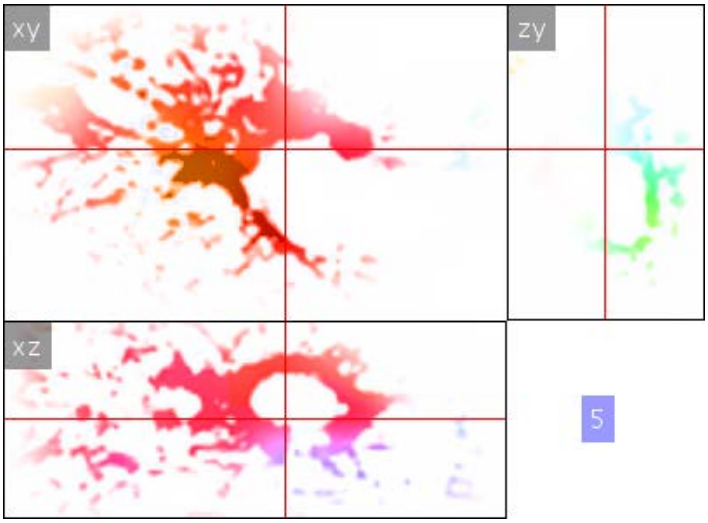


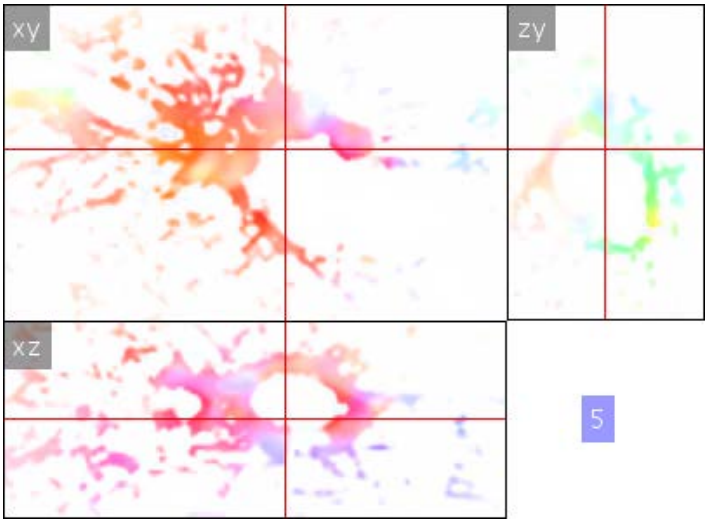
FIGURE 6.5 – 3PHS map of the computed flow fields using (A) 3D PatchMatch, our TV-regularized method for (B) $\theta^{-1} = 0.01$ (C) $\theta^{-1} = 0.1$, (D) $\theta^{-1} = 1$. Foreground mask has been applied to the flow fields.



(a) Overlap of slices of source and target volumes.



(b) 3PHS map of the compute flow field with 3D PatchMatch



(c) 3PHS map of the compute flow field with the proposed method

FIGURE 6.6 – (a) Overlap of orthogonal slices of the source (green) and target (magenta) volumes of a collagen network. (b) 3PHS map of 3D flow field computed by our method with TV-regularization. Foreground mask has been applied to the flow fields. (Best viewed in electronic form.)

CHAPITRE 7

CONCLUSION AND PERSPECTIVES

Let us restate the main aim of our research : definition of a 3D motion estimation method and assessment tools, which can allow us to study the dynamics of generic biological objects from 3D light microscopy image sequences. Hereafter, we summarize our contribution on that topic, and describe short-term and long-term perspectives.

7.1 Research Contributions

Towards the achievement of our aim, we first reviewed the literature on motion estimation in both 2D natural images and 3D fluorescence images. Though the 2D optical flow methods have matured in performance and accuracy, the same thing cannot be said about 3D optical flow for 3D fluorescence images. We group the 2D optical flow methods in three classes : (1) Dense and semi-dense matching of pixels between a pair of images (2) Variational methods, and (3) Deep-learning methods.

Thereafter, we devised 3D PatchMatch methods for matching voxels in 3D fluorescence images. We started with super-pixel-based 3D PatchMatch. However, it became apparent that a denser approach was required to compute flow fields in thin fibrous specimen like collagen. Therefore, we developed a coarse-to-fine 3D PatchMatch method. We investigated three similarity measures based on intensity of the patches : SSD, TZNCC and Census signature (CS). Among them, CS was found to be the optimally performing one. As a consequence, we went on to develop 3D variational methods with a continuous version of CS as a data term, and quadratic and robust regularization for smooth flow fields. We compared our 3D variational

method with the 3D version of Horn-and-Schunck method and Amat's method. Qualitatively and quantitatively, our method outperformed both methods. We concluded that CS based data-term is better than the direct intensity-based data-term for computing optical flow. Further, we proceeded to combine 3D PatchMatch with variational method to handle large and small displacements. In the process, we developed a total variation based 3D optical flow method which uses our proposed CS-based data term. With this, not only were we able to integrate the flow field computed from 3D PatchMatch with variational method, but also accelerate the computation time with multi-CPU implementation.

However, making the qualitative and the quantitative assessment were not straightforward tasks. The usual arrow-based plots for flow fields are not commensurable, nor well-suited for analysis in prototyping of methods. Therefore, we relied on color coding of the flow field. We developed three different schemes for color coding, namely : 3DHSV, 3PHS and MIP-flow. 3DHSV scheme maps Hue, Saturation and Value space of color to the direction, the magnitude and the off-the-plane motion direction of 3D flow vectors, respectively. In 3PHS scheme, the 3D motion is projected to three orthogonal planes, where motion can be viewed in 2D form. The discontinuities and diffusion of flow vectors are readily visible in all three planes. Therefore, it provides a better way to scrutinize the flow field. We have used it extensively during the development of our methods. The last scheme called MIP-flow, allows rendering of color coded flow vectors coherent to the image structures. For this, only maximum intensity points along a chosen axis are selected and their corresponding vectors are projected to a plane perpendicular to the chosen axis. This method provides the best cut off of the flow vectors corresponding to noisy, low signal, and background voxels.

The standard End-Point-Error (EPE) and Average Angle Error (AAE) are of no use for quantitative evaluation when the ground truth is not available as for 3D fluorescence image sequences. To overcome this problem, as well as provide a confidence measure of the computed flow field, we devised a novel error measure called SAE, which is based on local structure tensors of the original source volume and the reconstructed source volume using compute flow field. Average of SAE exhibited similar behavior to EPE when comparing flow fields computed in a dataset with generated ground-truth flow fields. Further, the SAE map of our error measure can be used for an up-close analysis of the computed flow fields.

Additionally, we explored triangular mesh-based representation of cell surface. We were able to localize protruding blebs, segment them and track them across time.

We summarize our main contributions to the existing knowledge of 3D motion estimation and assessment in fluorescence image data as follows :

1) Census signature constancy assumption is better than the brightness constancy assumption when computing optical flow for 3D fluorescence image sequences, both for the

discrete matching and the variational motion estimation.

2) Color-coded visualization of a 3D flow field has to be done smartly. We provide three different ways to do so. Projection of the 3D flow fields to 2D planes can be helpful to better understand the flow field.

3) Quantitative assessment of 3D optical flow can be done even without the ground-truth flow fields. For this, we provide SAE measure based upon the principal orientation of structures present in the original source volume and the reconstructed source volume.

7.2 Perspectives

In this section, we point out the items that we believe could be the extensions to our works. We present the short-term and long-term perspectives of the work.

7.2.1 Short-term perspectives

7.2.1.1 Parallelization of our method

Our 3D variational method can be implemented in the paradigm of splitting methods. Split-Bregman optimization method for optical flow shows faster convergence and high affinity to parallelization. An efficient multi-CPU or GPU-based implementation of the method could be an extension to our 3D variational method.

7.2.1.2 Sparse variation for regularization

Our 3D variational methods penalize the smoothness of the flow field using L_2, L_1 and TV norms. These regularizations, however, do not take care of the sparseness of the data. To create sparse but regularized flow field in the foreground regions, sparse variation (Eickenberg *et al.* 2015) for regularization might become useful.

7.2.1.3 Visualization tools

Our visualization tools can be complemented with widgets for user interactivity. The 3DHSV map of the whole volume is computed in one-pass and requires inputs : upper bound for amplitude normalization and range of off-the-plane motion. This map can be viewed in already available softwares like *Fiji*. The 3PHS map, however, requires plane selection by the user for slicing. A widget would be necessary for this interaction. Similarly, MIP-flow, which we believe is well-suited for the biologists and/or the biophysicists for viewing, also requires selection of an axis to compute projection of maximum intensity. Figure 7.1 shows selection of arbitrary plane rotated by ϕ angle around z-axis. Such a plane can be chosen using graphics library like

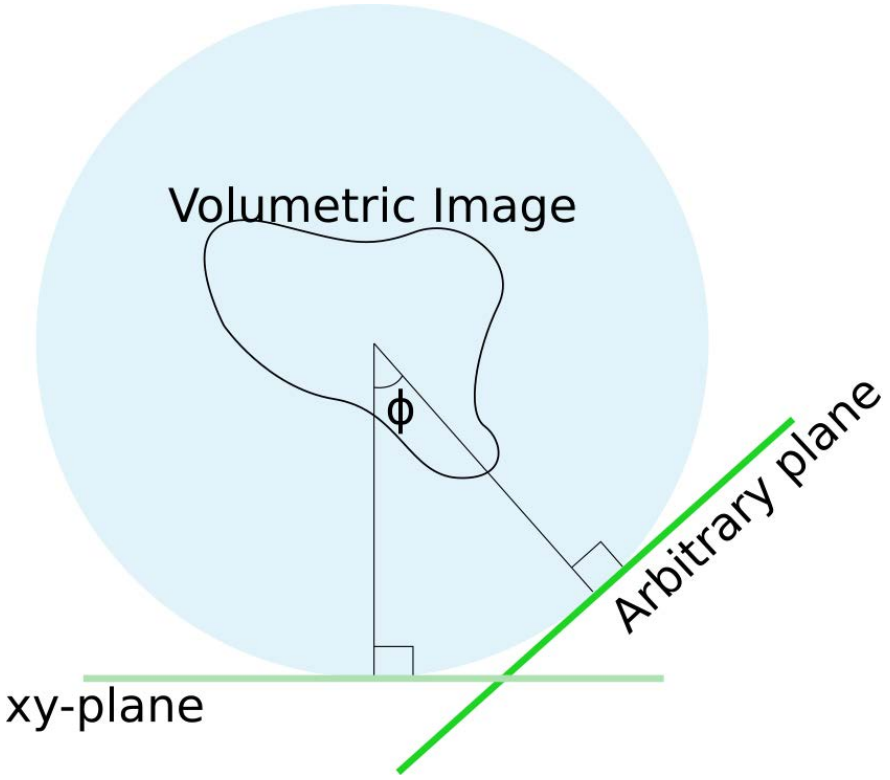


FIGURE 7.1 – Choosing arbitrary plane rotated by ϕ angle around z-axis of the 3D viewport (top view depicted by the circle). The maximum intensity points have to be chosen along the axis perpendicular to the arbitrary plane.

OpenGL. A computer routine has to be written, however, to interpolate the voxel intensities along the line perpendicular to the arbitrary plane and passing through the volumetric image data. Thus, the maximum of the intensities and their position in 3D can be selected for building the MIP-flow map.

7.2.2 Long-term perspectives

7.2.2.1 Triangular mesh-based representation

Triangular mesh-based representation of a cell surface allows a number of geometric measurements, such as curvature, geodesics, centroid, etc., to be computed easily, which is not straightforward in image-based representation. Although the intensity information is lost, these measurements and features derived from them can be useful to compute the dynamics of the cell morphology. We have used the heat kernel signature to detect distinct vertices in blebs, and track them across time. We were not able to match the vertices between the tracked bleb regions. Since the mesh sequence is temporally incoherent by construction, the vertices do not automatically match. Additionally, the surface area of a bleb changes with time, which changes

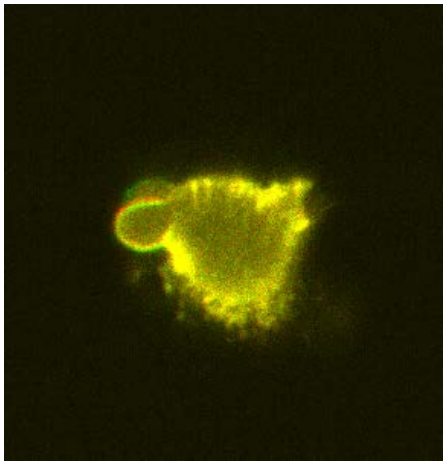
the number of vertices in its region. A future work would be to investigate matching of these vertices in the tracked bleb regions. The matching will not always be from vertices to vertices, but vertices to a point in a triangular face. This makes the work challenging. This would allow us to recover the non-rigid motion of the blebs in a cell.

7.2.2.2 Towards Deep-learning optical flow methods

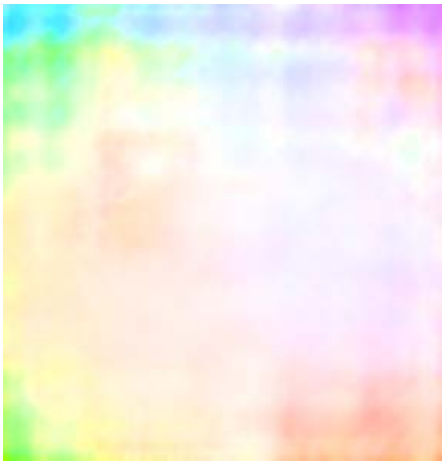
In our thesis work, we have delved, however not exhaustively, into deep-learning approaches for optical flow methods. This approach is interesting as the supervised deep learning methods for optical flow estimation produce the most accurate results in 2D optical flow benchmark dataset. Further, their GPU-based implementation makes them the fastest methods. The main hindrance to adopt this approach for 3D motion estimation in fluorescence image sequences is the unavailability of training sets or easier way to generate them. Additional issue arises from the size of the training architecture and the huge GPU memory requirement, given the large size of the 3D volumetric data.

One could synthesize arbitrary flow fields, but they do not always well represent cellular or sub-cellular dynamics. A direction towards generating realistic enough ground-truth flow field is to compute many flow fields using methods with different models (data-term, matching cost, regularization and other constraints), and selecting the vectors from the set of the computed flow fields that has the least SAE value. This ensures that the selected vector can reconstruct structure that is closest in resemblance to the source. Thus, composed flow fields might serve as ground-truth set for training a 3D version of existing 2D deep-learning optical flow methods (Philipp *et al.* 2015; Sun *et al.* 2017; Simon *et al.* 2017).

To get around the memory issue while training, as well as to take the advantage of already trained architectures for 2D optical flow, we propose to compute multiple 2D flow fields from the slice pairs of volumes. The slices should be obtained as the projections of the volume from multiple views to account for off-the-plane motion. Feeding multiple 2D slices is much more memory efficient than feeding the entire stack. The problem then becomes very similar to the one solved by Chan and Liebling (2015). The authors computed 2D flow fields in slice pairs obtained from multiple views of a volume sequence exhibiting blood flow. They use radial basis weights for the divergence-free interpolation of 2D flow fields to obtain 3D flow field. We can imagine tuning one of the aforementioned deep-learning architectures to customize them to fluorescence microscopy images. Figures 7.2 and 7.3 shows 2D flow field computed for a slice pair in the cell and the collagen channels using PWC-net (without tuning). The network computes false motion in background regions. There, the motion is smooth and high in amplitude. Fine tuning the network using fluorescence images for training might be able to resolve this issue. Adding more layers to learn interpolation weights that impose constraints

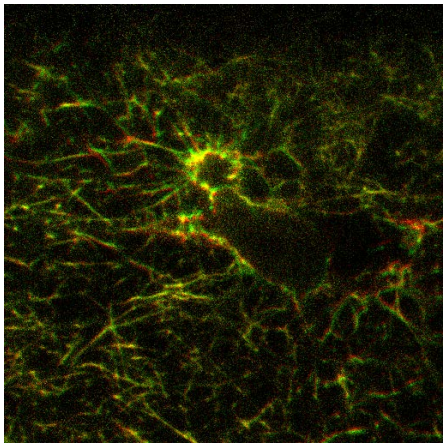


(a) Overlapped source and target slices



(b) Computed flow field

FIGURE 7.2 – (a) Slices of source in red, target in green. Both slices were obtained at same depth in z-axis of the MV3 blebbing sequence.(b) 2D flow field computed for the slice-pair with PWC-net (Sun *et al.* 2017)



(a) Overlapped source and target slices

(b) Computed flow field

FIGURE 7.3 – (a) Slices of source in red, target in green. Both slices were obtained at same depth in z-axis of the collagen sequence.(b) 2D flow field computed for the slice-pair with PWC-net (Sun *et al.* 2017)

like divergence-free motion or incompressibility, might add flexibility to compute motion of various specimen.

Unsupervised learning would require a loss function that is ground-truth independent. Our experiments show good results when using CS as the data-term. It has been demonstrated by the authors of the well known UnFlow architecture (Simon *et al.* 2017) that a good data-term can be used as the loss function for unsupervised learning of 2D optical flow. Therefore, we suggest 3D CS-based loss function for a possible 3D extension to the known architectures. The SAE

measure is another potential candidate for the loss function. However, SAE requires expensive eigen-decomposition of 3×3 matrix attached to each voxel. Fortunately, deep-learning packages provide GPU-based algorithm for the eigen-decomposition, which makes the computation a lot more efficient.

Deep-learning can also benefit matching voxels. We have used intensity-based information for matching patches. We can imagine CNN-based features of the patches that can be matched using some similarity metric. Han *et al.* (2015) used a dataset containing 1.5 million labeled 2D patches with 50% of them having matches, to train their network for feature and metric learning. Let us now focus on the training data. The patches in the aforementioned training dataset, were extracted using Difference of Gaussian interest point detector or multi-scale Harris corner detector. These detectors and many others are available in many 2D computer vision libraries. However, implementations of such detectors are not readily available for 3D fluorescence image data. During the prototyping of methods, the development of these detectors take considerable amount of time. This brings us to the perspective described in the next subsection.

7.2.2.3 A 3D computer vision library for fluorescence volumetric data

A 3D computer vision library with implementation of robust detectors can benefit multiple domains concerned with fluorescence imaging. For example, a collagen network contains many joints and junctions which might be interesting to the biophysicists. There are cases where a researcher manually labels these interesting structures in 3D, which is a tiresome job. A library containing such detectors for 3D images would be valuable to the research community.

CHAPITRE 8

APPENDIX

8.1 SAE analysis of MV3 C

Figure 8.1 presents the SAE analysis of optical flow computed by Amat's method and our 3D PatchMatch for large retraction in MV3 C sequence. We report ASAE for Amat's method and our method to be 0.46 and 0.42, respectively.

Clearly, Amat's method fails to capture the large motion in filopodium, despite the usage of triple level pyramids for their coarse-to-fine optical flow estimation. The SAE map in Fig. 8.1 C depicts the error. On the other hand, our method is capable of capturing the large motion. However, the apparent gap in their performance is not marked by a large gap in ASAE measure. This is due to the larger SAE values in cytosolic region for both the methods. Due to the averaging of SAE values, the gap between the methods appear low.

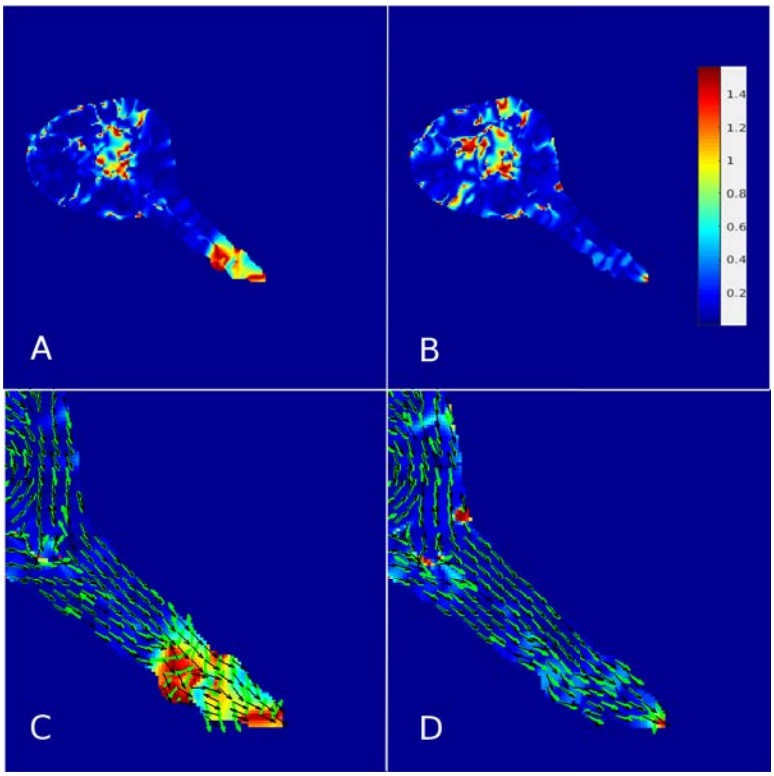


FIGURE 8.1 – SAE map for the optical flow computed with (A) Amat's method (B) Our 3D PatchMatch method. The color code is given in the right side of (B). Principal orienta- tion of the structures denoted by arrows in black for original source volume, and in green for the reconstructed source volumes by the optical flow computed using (C) Amat's method and (D) Our 3D PatchMatch method.

8.2 List of publications

8.2.1 Conference with abstract

Sandeep Manandhar, Patrick Bouthemy, Philippe Roudot, Charles Kervrann, *3D motion estimation in 3D light microscopy image sequences : application to cell migration*, Quantitative Bioimaging, 2019, Rennes

8.2.2 Conference papers

— Sandeep Manandhar, Patrick Bouthemy, Erik Welf, Philippe Roudot, Charles Kervrann, *A sparse-to-dense method for 3D optical flow estimation in 3D light microscopy image sequences*, IEEE 15th International Symposium on Biomedical Imaging, Apr 2018, Washington DC, United States. pp.952-956

— Sandeep Manandhar, Patrick Bouthemy, Erik Welf, Philippe Roudot, Charles Kervrann, *3D optical flow estimation combining 3D Census signature and total variation regularization*, IEEE 16th International Symposium on Biomedical Imaging, 2020 (submitted)

8.2.3 Journal paper

Sandeep Manandhar, Patrick Bouthemy, Erik Welf, Gaudenz Danuser, Philippe Roudot, Charles Kervrann, *3D Flow Field Estimation and Assessment for Live Cell Fluorescence Microscopy*, Bioinformatics (Oxford, England), 2019, In press.

References

Abramoff, M., Niessen, W., and A. Viergever, M. (2000). Objective quantification of the motion of soft tissues in the orbit. *IEEE trans. on Med. Imag.*, **19**, 986–95.

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Susstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, **34**(11), 2274–2282.

Ahmad, R. K. *et al.* (2015). 3D structure tensor analysis of light microscopy data for validating diffusion MRI. *NeuroImage*, **111**, 192–203.

Alvarez, L., Lefébure, M., and Sánchez, J. (1999). A pde model for computing the optical flow. In *In Congreso de Ecuaciones Diferenciales y Aplicacione*, pages 1349–1356, LasPalmas de Gran Canaria.

Alvarez, L., Weickert, J., and Sánchez, J. (2000). Reliable estimation of dense optical flow fields with large displacements. *Int. J. Comput. Vision*, **39**(1), 41–56.

Amat, F. *et al.* (2013). Fast and robust optical flow for time-lapse microscopy using super-voxels. *Bioinformatics*, **29**(3), 373–380.

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

Aria, A. and Ioannis, P. (2016). Unsupervised convolutional neural networks for motion estimation. *CoRR*, **abs/1601.06087**.

Bailer, C., Varanasi, K., and Stricker, D. (2017). Cnn-based patch matching for optical flow with thresholded hinge embedding loss. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2710–2719.

Baker, S. *et al.* (2011a). A database and evaluation methodology for optical flow. *Int. J. Computer Vision*, **92**(1), 1–31.

Baker, S., Scharstein, D., Lewis, J. P., Roth, S., Black, M. J., and Szeliski, R. (2011b). A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, **92**(1), 1–31.

Bao, L., Yang, Q., and Jin, H. (2014). Fast edge-preserving patchmatch for large displacement optical flow. *IEEE Transactions on Image Processing*, **23**(12), 4996–5006.

Barbara, Z. and Jan, F. (2003). Image registration methods : a survey. *Image and Vision Computing*, **21**(11), 977 – 1000.

Barnes, C. *et al.* (2009). PatchMatch : A randomized correspondence algorithm for structural image editing. In *Proc. of Special Interest Group on Computer Graphics and Interactive Techniques*, pages 24 :1–24 :11, New Orleans.

Baron (2004). Experience with 3D optical flow on gated MRI cardiac datasets. In *Canadian Conference on Computer and Robot Vision*, pages 370–377.

Baron, J. L. (2004). Experience with 3D optical flow on gated MRI cardiac datasets. In *First Canadian Conference on Computer and Robot Vision*, pages 370–377.

Barron, J. L., Fleet, D. J., and Beauchemin, S. S. (1994). Performance of optical flow techniques. *International Journal of Computer Vision*, **12**(1), 43–77.

Behbahani, S., Asadi, S., Ashtiyani, M., and Maghooli, M. (2007). Analysing optical flow based methods. In *2007 IEEE International Symposium on Signal Processing and Information Technology*, pages 133–137.

Betzig, E. (2005). Excitation strategies for optical lattice microscopy. *Opt. Express*, **13**(8), 3021–3036.

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

Boquet-Pujadas, A. *et al.* (2017a). BioFlow : a non-invasive, image-based method to measure speed, pressure and forces inside living cells. *Scientific Reports*, **7**(1), 9178.

Boquet-Pujadas, A., Lecomte, T., Manich, M., Thibeaux, R., LabruyÃÍre, E., GuillÃĺn, N., Olivo-Marin, J.-C., and Dufour, A. (2017b). Bioflow : A non-invasive, image-based method to measure speed, pressure and forces inside living cells. *Scientific Reports*, **7**.

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. (1993). Signature verification using a "siamese" time delay neural network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, pages 737–744, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Brox, T. *et al.* (2004). High accuracy optical flow estimation based on a theory for warping. In *Proc. of European Conference on Computer Vision*, pages 25–36, Heidelberg.

Brox, T. and Malik, J. (2011). Large displacement optical flow : descriptor matching in variational motion estimation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **33**(3), 500–513.

Bruhn, A., Weickert, J., and SchnÃűrr, C. (2005). Lucas/kanade meets horn/schunck : Combining local and global optic flow methods. *International Journal of Computer Vision*, **61**(3), 211–231.

Budde, M. and Frank, J. (2012). Examining brain microstructure using structure tensor analysis of histological sections. *NeuroImage*, **63**, 1–10.

Burger, M., Dirks, H., and Frerking, L. (2016). *On optical flow models for variational motion estimation : In Imaging and Geometric Control*.

Butler, D. J., Wulff, J., Stanley, G. B., and Black, M. J. (2012). A naturalistic open source movie for optical flow evaluation. In *European Conf. on Computer Vision*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag.

Byrd, R., Lu, P., Nocedal, J., and Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, **16**(5), 1190–1208.

Chalfie, M., Tu, Y., Euskirchen, G., Ward, W., and Prasher, D. (1994). Green fluorescent protein as a marker for gene expression. *Science*, **263**(5148), 802–805.

Chambolle, A. (2004). An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, **20**(1), 89–97.

Chan, K. and Liebling, M. (2015). Estimation of divergence-free 3D cardiac blood flow in a zebrafish larva using multi-view microscopy. In *2015 IEEE 12th International Symposium on Biomedical*

Imaging, pages 385–388.

Chandramouli, P., Heitz, D., Laizet, S., and Mémin, E. (2018). Coarse large-eddy simulations in a transitional wake flow with flow models under location uncertainty. *Computers and Fluids*, **168**, 170–189.

Charras, G. (2008). A short history of blebbing. *J. of Microscopy*, **231**(3), 466–478.

Chen, J. *et al.* (2018). Fast optical flow estimation based on the split-bregman method. *IEEE Trans. on Circuits and Systems for Video Technology*, **28**(3), 664–678.

Cohen, I. (1993). Nonlinear variational method for optical flow computation. In *Scandinavian Conf. on Image Analysis*, volume 1, pages 523–530, Tromso.

Colin, P., Mistriotis, P., and Konstantopoulos, K. (2016). Cancer cell motility : lessons from migration in confined spaces. *Nature reviews. Cancer*, **17**(2), 131–140.

Coons, S. A. (1967). Surfaces for computer-aided design of space forms. Technical report, Cambridge, MA, USA.

Davenport, D., Nicol, J. A. C., and Russell, F. S. (1955). Luminescence in hydromedusae. *The Royal Society of London. Series B - Biological Sciences*, **144**(916), 399–411.

David M. Young, J. (1971). Iterative solution of large linear systems. Academic Press NY.

Dean, K. M. *et al.* (2015). Deconvolution-free Subcellular Imaging with Axially Swept Light Sheet Microscopy. *Biophysical J.*, **12**(108), 2807–15.

Dean, K. M. *et al.* (2016). Diagonally Scanned Light-Sheet Microscopy for Fast Volumetric Imaging of Adherent Cells. *Biophysical J.*, **6**(110), 1456–65.

Delon, J. and Rougé, B. (2007). Small baseline stereovision. *J. Math. Imaging Vis.*, **28**(3), 209–223.

Delpiano, J. *et al.* (2012). Performance of optical flow techniques for motion analysis of fluorescent point signals in confocal microscopy. *Machine Vision and Applications*, **23**(4), 675–689.

Demetz, O., Hafner, D., and Weickert, J. (2013). The complete rank transform : A tool for accurate and morphologically invariant matching of structures.

Dey, T., Li, K., Luo, C., Ranjan, P., Safa, I., and Wang, Y. (2010). Persistent heat signature for pose-oblivious matching of incomplete models. *Computer Graphics Forum*, **29**(5), 1545–1554.

Dhandayuthapani, S., Via, L., Thomas, C., Horowitz, P., Deretic, D., and Deretic, V. (1995). Green fluorescent protein as a marker for gene expression and cell biology of mycobacterial interactions with macrophages. *Molecular Microbiology*, **17**(5), 901–912.

Dhruv, Â., Alex, E., and Charles, W. (2016). On the quantification of cellular velocity fields. *Biophysical Journal*, **110**(7), 1469–1475.

Domingues, B., Manuel, J., Soares, P., and Helena, P. (2018). Melanoma treatment in review. *Immunotargets and Therapy*, **7**, 35–49.

Driscoll, M. K., Welf, E. S., Dean, K. M., Fiolka, R., and Danuser, G. (2018). Cell morphological motif detector for high-resolution 3D microscopy images. *bioRxiv*.

Drulea, M. and Nedevschi, S. (2013). Motion Estimation Using the Correlation Transform. *IEEE Trans. on Image Processing*, **22**, 3260–3270.

Duan, Q., Angelini, E., Herz, S., Ingrassia, C., Olivier, G., Costa, K. D., Holmes, J., and Laine, A. (2005). Evaluation of optical flow algorithms for tracking endocardial surfaces on three-dimensional ultrasound data. *Proceedings of SPIE - The International Society for Optical Engineering*, **5750**.

Eberly, D. (2018). A robust eigensolver for 3×3 symmetric matrices.

Eickenberg, M., Dohmatob, E., Thirion, B., and Varoquaux, G. (2015). Total variation meets sparsity : statistical learning with segmenting penalties. In *Medical Image Computing and Computer Aided Intervention*, Proceedings of MICCAI 2015, München, Germany.

Fleet, D. J. and Jepson, A. D. (1990). Computation of component image velocity from local phase information. *International Journal of Computer Vision*, **5**(1), 77–104.

Fonck, E., Feigl, G., Fasel, J., Sage, D., Unser, M., Rüfenacht, D., and Stergiopulos, N. (2009). Effect of aging on elastin functionality in human cerebral arteries. *Stroke*, **40**(7), 2552–2556.

Förstner, W. and Gülch, E. (1987). A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS intercommission conference on fast processing of photogrammetric data*, pages 281–305, Interlaken.

Fortun, D., Bouthemy, P., and Kervrann, C. (2015). Optical flow modeling and computation : A survey. *Computer Vision and Image Understanding*, **134**, 1–21.

Frantz, C., Stewart, K. M., and Weaver, V. M. (2010). The extracellular matrix at a glance. *Journal of cell science*, **123(24)**, 4195–4200.

Friedl, P. and Alexander, S. (2011). Cancer invasion and the microenvironment : plasticity and reciprocity. *Cell*, **147**, 992–1009.

Gaggioli, C. *et al.* (2007). Fibroblast-led collective invasion of carcinoma cells with differing roles for RhoGTPases in leading and following cells. *Nature Cell Biology*, **9**, 1392–1400.

Ginkel, M. V. (2002). *Image analysis using orientation space based on steerable filters*. PhD dissertation, Delft University of Technology, The Netherlands.

Giraud, R., Ta, V., Bugeau, A., Coupé, P., and Papadakis, N. (2019). Superpatchmatch : an algorithm for robust correspondences using superpixel patches. *CoRR*, **abs/1903.07169**.

Gkamas, T. and Nikou, C. (2011). Guiding optical flow estimation using superpixels. In *Digital Signal Processing*, pages 1–6.

Goldstein, T. and Osher, S. (2009). The split bregman method for L1-regularized problems. *SIAM J. Img. Sci.*, **2**(2), 323–343.

Grewenig, S., Weickert, J., and Bruhn, A. (2010). From box filtering to fast explicit diffusion. In *Pattern Recognition*, pages 533–542, Berlin, Heidelberg.

Guckenberger, R., Kosslinger, C., Gatz, R., Breu, H., Levai, N., and Baumeister, W. (1988). A scanning tunneling microscope (stm) for biological applications : design and performance. *Ultramicroscopy*, **25**(2), 111 – 121.

Gwosdek, P., Zimmer, H., Grewenig, S., Bruhn, A., and Weickert, J. (2012). A highly efficient gpu implementation for variational optic flow based on the euler-lagrange framework. In *Trends and Topics in Computer Vision*, pages 372–383, Berlin, Heidelberg.

Hafner, D. *et al.* (2013). Why is the census transform good for robust optic flow computation ? In *Proc. of Scale Space and Variational Methods in Computer Vision*, pages 210–221, Graz.

Hamming, R. W. (1973). *Numerical Methods for Scientists and Engineers*. McGraw-Hill, Inc., New York, NY, USA.

Han, X., Leung, T., Jia, Y., Sukthankar, R., and Berg, A. C. (2015). Matchnet : Unifying feature and metric learning for patch-based matching. In *Proc. of IEEE Conference on Computer Vision and*

Pattern Recognition, pages 3279–3286, Boston.

Hannah, M. (1974). *Computer Matching of Areas in Stereo Images*. PhD dissertation, Stanford University.

Haralick, R. and Lee, J. (1983). The facet approach to optic flow. In *Proceedings of image understanding workshop*, pages 84–93, Arlington Va.

Harris, C. and Stephens, M. (1988). A combined corner and edgedetector. In *Proc. Alvey Vision Conference*, pages 147–151, Manchester.

Hearst, M. A. (1998). Support vector machines. *IEEE Intelligent Systems*, **13**(4), 18–28.

Heitz, F. and Bouthemy, P. (1993). Multimodal estimation of discontinuous optical flow using markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**(12), 1217–1232.

Hermann, S. and Werner, R. (2014). TV-L1-based 3D medical image registration with the census cost function. In *Image and Video Technology*, pages 149–161, Berlin, Heidelberg.

Hess, R. (2010). *Blender - a 3D modelling and rendering package*. Blender Foundation, Blender Institute, Amsterdam.

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

Hu, Y. *et al.* (2016). Efficient coarse-to-fine PatchMatch for large displacement optical flow. In *Computer Vision and Pattern Recognition*, pages 5704–5712, Las Vegas.

Hu, Y. *et al.* (2017). Robust interpolation of correspondences for large displacement optical flow. In *Computer Vision and Pattern Recognition*, pages 4791–4799, Honolulu.

Hubený, J. *et al.* (2007). Estimating large local motion in live-cell imaging using variational optical flow towards motion tracking in live cell imaging using optical flow. In *Proc. of Computer Vision Theory and Applications*, pages 542–548, Barcelona.

Huber, P. J. (1964). Robust estimation of a location parameter. *Annals of Mathematical Statistics*, **35**, 73–101.

Hui, T.-W. *et al.* (2018). Liteflownet : A lightweight convolutional neural network for optical flow estimation. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 8981–8989, Salt Lake City.

Huisken, J., Swoger, J., Del Bene, F., Wittbrodt, J., and Stelzer, E. H. K. (2004). Optical sectioning deep inside live embryos by selective plane illumination microscopy. *Science*, **305**(5686), 1007–1009.

Ilg, E. *et al.* (2017). Flownet 2.0 : Evolution of optical flow estimation with deep networks. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1647–1655, Honolulu.

Jaqaman, K. *et al.* (2008). Robust single-particle tracking in live-cell time-lapse sequences. **5**(8), 695–702.

Jure, Z. and Yann, L. (2014). Computing the stereo matching cost with a convolutional neural network. *CoRR*, **abs/1409.4326**.

Kappe, C. P. *et al.* (2016). Reconstruction and visualization of coordinated 3D cell migration based on optical flow. *IEEE Trans. Visualization and Computer Graphics*, **22**(1), 995–1004.

Khalili, A. A. and Ahmad, M. R. (2015). A review of cell adhesion studies for biomedical and biological applications. *International journal of molecular sciences*, **16(8)**, 18149–18184.

Khan, A. R., Cornea, A., Leigland, L. A., Kohama, S. G., Jespersen, S. N., and Kroenke, C. D. (2015). 3D structure tensor analysis of light microscopy data for validating diffusion MRI. **111**, 192–203.

Krishnendu, C. *et al.* (2018). Recent progress in light sheet microscopy for biological applications. *Applied Spectroscopy*, **72**(8), 1137–1169.

Krizhevsky, A., Sutskever, I., and E. Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, **25**.

Krzic, U., Gunther, S., Saunders, T., Streichan, S., and Hufnagel, L. (2012). Multiview light-sheet microscope for rapid in toto imaging. *Nature Methods*, **9**, 730–733.

Li, W., Ng, W., and Lim, E. (2004). Spectral analysis of text collection for similarity-based clustering. In *Advances in Knowledge Discovery and Data Mining*, pages 389–393, Berlin, Heidelberg.

Li, Y., Hu, Y., Song, R., Rao, P., and Wang, Y. (2018). Coarse-to-fine patchmatch for dense correspondence. *IEEE Transactions on Circuits and Systems for Video Technology*, **28**(9), 2233–2245.

Liu, C., Yuen, J., Torralba, A., Sivic, J., and Freeman, W. T. (2008). Sift flow : Dense correspondence across different scenes. In *European Conference on Computer Vision*, pages 28–42, Berlin, Heidelberg.

Lorensen, W. E. and Cline, H. E. (1987). Marching cubes : A high resolution 3D surface construction algorithm. In *Computer Graphics and Interactive Techniques*, SIGGRAPH '87, pages 163–169, New York.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157.

Lu, J., Li, Y., Yang, H., Min, D., Eng, W., and Do, M. N. (2013). Patch match filter : Efficient edge-aware filtering meets randomized search for fast correspondence field estimation. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1854–1861, Washington DC.

Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679.

Maintz, J. and Viergever, M. (1998). A survey of medical image registration. *Medical Image Analysis*, **2**(1), 1–36.

Manuel, W., Thomas, P., and Horst, B. (2010). Motion estimation with non-local total variation regularization. *Computer Vision and Pattern Recognition*, pages 2464–2471.

Marion, S., Guillen, N., Bacri, J.-C., and Wilhelm, C. (2005). Acto-myosin cytoskeleton dependent viscosity and shear-thinning behavior of the amoeba cytoplasm. *European Biophysics Journal*, **34**(3), 262–272.

Maska, M. *et al.* (2014). A benchmark for comparison of cell tracking algorithms. *Bioinformatics*.

Mazzarello, P. (1999). A unifying concept : the history of cell theory. *Nature Cell Biology*, **1**(1), E13–E15.

Mémin, E. and Pérez, P. (1997). Adaptative Multigrid and Variable Parameterization for Optical-flow Estimation. Research Report RR-3102, INRIA.

Mémin, E. and Pérez, P. (2002). Hierarchical estimation and segmentation of dense motion fields. *International Journal of Computer Vision*, **46**(2), 129–155.

Mémin, E., Pérez, P., Machecourt, D., and Temis, P. (1996). Dense estimation and object-oriented segmentation of the optical flow with robust techniques. *IEEE Transactions on Image Processing*, **7**(5), 703–719.

Menze, M. and Geiger, A. (2015). Object scene flow for autonomous vehicles. In *Computer Vision and*

Pattern Recognition, pages 3061–3070, Boston.

Mohamed, M. A. and Mertsching, B. (2012). TV-L1 optical flow estimation with image details recovering based on modified census transform. In *Advances in Visual Computing*, pages 482–491.

Mostafa, E., Harald, K., and Ulrich, R. (2007). 3D optical flow computation using a parallel variational multigrid scheme with application to cardiac c-arm ct motion. *Image and Vision Computing*, **25**(9), 1482–1494.

Muja, M. and Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In *Proc. of Computer Vision Theory and Applications*, pages 331–340, Lisbon.

Müller, T., Rabe, C., Rannacher, J., Franke, U., and Mester, R. (2011). Illumination-robust dense optical flow using census signatures. In *Pattern Recognition*, pages 236–245, Berlin, Heidelberg.

Nagel, H. H. and Enkelmann, W. (1986). 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.

Nan, W., Yee-Hong, Y., and R., P. (1996). 3D surface reconstruction using optical flow for medical imaging. In *IEEE Nuclear Science Symposium. Conference Record*, volume 3, pages 1845–1849.

Ovsjanikov, M., Mérigot, Q., Mémoli, F., and Guibas, L. (2010). One point isometric matching with the heat kernel. *Computer Graphics Forum*, **29**(5), 1555–1564.

Papenberg, N., Bruhn, A., Brox, T., Didas, S., and Weickert, J. (2006). Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision*, **67**(2), 141–158.

Pathak, A. and Kumar, S. (2012). Independent regulation of tumor cell migration by matrix stiffness and confinement. *Biophysics and Computational Biology, Engineering*, **109(26)**, 10334–10339.

Petitjean, L., Reffay, M., Grasland-Mongrain, E., Poujade, M., Ladoux, B., Buguin, A., and Silberzan, P. (2010). Velocity fields in a collectively migrating epithelium. *Biophysical Journal*, **98**(9), 1790 – 1800.

Philipp, F., Alexey, D., Eddy, I., Philip, H., Caner, H., Vladimir, G., Patrick, V., Daniel, C., and Thomas, B. (2015). Flownet : Learning optical flow with convolutional networks. *CoRR*, **abs/1504.06852**.

Planchon, T., Gao, L., Milkie, D., Davidson, M., Galbraith, j., Galbraith, C., and Betzig, E. (2011). Rapid three-dimensional isotropic imaging of living cells using bessel beam plane illumination. **8**, 417–423.

Plant, A., Bhadriraju, K., Spurlin, T., and Elliott, J. (2009). Cell response to matrix mechanics : Focus on collagen. *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research*, **1793**(5), 893 – 902.

Pock, T., Urschler, M., Zach, C., Beichel, R., and Bischof, H. (2007). *A duality based algorithm for TV-L1-optical-flow image registration*, volume 4792 LNCS, pages 511–518. Springer, Brisbane.

Pop, S., Dufour, A. C., Le Garrec, J.-F., Ragni, C. V., Cimper, C., Meilhac, S. M., and Olivo-Marin, J.-C. (2013). Extracting 3D cell parameters from dense tissue environments : application to the development of the mouse heart. *Bioinformatics*, **29**(6), 772–779.

Qianglong, Z., Hua, Y., and Zhouping, Y. (2017). An optical flow algorithm based on gradient constancy assumption for PIV image processing. *Measurement Science and Technology*, **28**(5).

Raffel, M., Willert, C., and Kompenhans, J. (1998). Particle image velocimetry. a practical guide. *Journal of Fluid Mechanics*, **377**, 374–381.

Ranftl, R., Gehrig, S. K., Pock, T., and Bischof, H. (2012). Pushing the limits of stereo using variational stereo estimation. *2012 IEEE Intelligent Vehicles Symposium*, pages 401–407.

Ranjan, A. and Black, M. J. (2016). Optical flow estimation using a spatial pyramid network. *CoRR*, **abs/1611.00850**.

Ren, Z., Yan, J., Ni, B., Liu, B., Yang, X., and Zha, H. (2017). Unsupervised deep learning for optical flow estimation. In *AAAI Conference on Artificial Intelligence*, San Francisco.

Revaud, J. *et al.* (2015). EpicFlow : Edge-Preserving Interpolation of Correspondences for Optical Flow. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Boston.

Rezakhaniha, R., Agianniotis, A., Schrauwen, J. T. C., Griffa, A., Sage, D., Bouten, C. V. C., van de Vosse, F. N., Unser, M., and Stergiopulos, N. (2012). Experimental investigation of collagen waviness and orientation in the arterial adventitia using confocal laser scanning microscopy. *Biomechanics and Modeling in Mechanobiology*, **11**(3), 461–473.

Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D : Nonlinear Phenomena*, **60**(1), 259 – 268.

Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition.

Schnorr, C. (1994). Segmentation of visual motion by minimizing convex non-quadratic functionals. In *Proceedings of 12th International Conference on Pattern Recognition*, volume 1, pages 661–663.

Schnörr, C. and Sprengel, R. (1994). A nonlinear regularization approach to early vision. *Biological Cybernetics*, **72**(2), 141–149.

Shapira, L., Shamir, A., and Cohen-Or, D. (2007). Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer*, **24**, 249–259.

Shimomura, O., Johnson, F., and Saiga, Y. (1962). Extraction, purification and properties of aequorin, a bioluminescent protein from the luminous hydromedusan, aequorea. *Journal of Cellular and Comparative Physiology*, **59**(3), 223–239.

Siedentopf, H. and Zsigmondy, R. (1902). Uber sichtbarmachung und größenbestimmung ultramikoskopischer teilchen, mit besonderer anwendung auf goldrubingläser. *Annalen der Physik*, **315**(1), 1–39.

Simon, M., Junhwa, H., and Stefan, R. (2017). Unflow : Unsupervised learning of optical flow with a bidirectional census loss. *CoRR*, **abs/1711.07837**.

Sorkine, O. (2005). Laplacian Mesh Processing. In *Eurographics 2005 - State of the Art Reports*. The Eurographics Association.

Sorkine, O. (2006). Differential representations for mesh processing. *Comput. Graph. Forum*, **25**, 789–807.

Sorokin, D. V. *et al.* (2018). Filogen : A model-based generator of synthetic 3-D time-lapse sequences of single motile cells with growing and branching filopodia. *IEEE Transactions on Medical Imaging*, **37**(12), 2630–2641.

Sotiras, A., Davatzikos, C., and Paragios, N. (2013). Deformable Medical Image Registration : A Survey. *IEEE Transactions on Medical Imaging*, **32**(7), 1153–1190.

Stein, F. (2004). Efficient computation of optical flow using the census transform. In *Pattern Recognition*, pages 79–86, Berlin, Heidelberg.

Steinbrucker, F., Pock, T., and Cremers, D. (2009a). *Advanced Data Terms for Variational Optic Flow Estimation.*, pages 155–164.

Steinbrucker, F., Pock, T., and Cremers, D. (2009b). Large displacement optical flow computation without warping. *IEEE International Conference on Computer Vision*, pages 1609–1614.

Stewart, G. W. (2001). A krylov-schur algorithm for large eigenproblems. *SIAM J. Matrix Anal. Appl.*, **23**(3), 601–614.

Style, R., Boltyanskiy, R., German, G. K., Hyland, C., MacMinn, C., Mertz, A., Wilen, L., Xu, Y., and Dufresne, E. (2014). Traction force microscopy in physics and biology. *Soft matter*, **10**, 4047–4055.

Sun, D. *et al.* (2010). Secrets of optical flow estimation and their principles. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2432–2439, San Francisco.

Sun, D., Roth, S., Lewis, J. P., and Black, M. J. (2008). Learning optical flow. In *European Conference on Computer Vision*, pages 83–97, Berlin, Heidelberg.

Sun, D., Yang, X., Liu, M., and Kautz, J. (2017). Pwc-net : Cnns for optical flow using pyramid, warping, and cost volume. *CoRR*, **abs/1709.02371**.

Sun, J., Ovsjanikov, M., and Guibas, L. (2009). A concise and provably informative multi-scale signature based on heat diffusion. In *Proceedings of the Symposium on Geometry Processing*, SGP '09, pages 1383–1392, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Teh, I., McClymont, D., Zdora, M.-C., Whittington, H. J., Davidoiu, V., Lee, J., Lygate, C. A., Rau, C., Zanette, I., and Schneider, J. E. (2017). Validation of diffusion tensor MRI measurements of cardiac microstructure with structure tensor synchrotron radiation imaging. *Journal of Cardiovascular Magnetic Resonance*, **19**(1), 31.

Tektonidis, M. and Rohr, K. (2017). Diffeomorphic multi-frame non-rigid registration of cell nuclei in 2D and 3D live cell images. *IEEE Trans. Image Processing*, **26**(3), 1405–1417.

Thirion, J.-P. (1998). Image matching as a diffusion process : an analogy with maxwell's demons. *Medical Image Analysis*, **2**(3), 243 – 260.

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

Torr, P. (1997). Geometric motion segmentation and model selection. *Philos. Trans. Roy. Soc. A*, **356**, 1321–1340.

Uras, S., Girosi, F., Verri, A., and Torre, V. (1988). A computational approach to motion perception. *Biological Cybernetics*, **60**(2), 79–87.

Vallmitjana, A., Civera-Tregón, A., Hoenicka, J., Palau, F., and Benítez, R. (2017). Motion estimation of subcellular structures from fluorescence microscopy images. In *Engineering in Medicine and Biology Society*, pages 4419–4422, Jeju, Korea.

Vettenburg, T. *et al.* (2014). Light-sheet microscopy using an airy beam. *Nature Methods*, **11**, 541–544.

Vogel, C. *et al.* (2013). An evaluation of data costs for optical flow. In *Proc. of Pattern Recognition*, pages 343–353, Heidelberg.

Wallis, J. W., Miller, T. R., Lerner, C. A., and Kleerup, E. C. (1989). Three-dimensional display in nuclear medicine. *IEEE Transactions on Medical Imaging*, **8**(4), 297–230.

Wardetzky, M., Mathur, S., Kaelberer, F., and Grinspun, E. (2007). Discrete Laplace operators : No free lunch. In *Geometry Processing*. The Eurographics Association.

Wasserman, L. (2010). *All of statistics : a concise course in statistical inference*. Springer, New York.

Wedel, A., Pock, T., Zach, C., Bischof, H., and Cremers, D. (2009). An improved algorithm for TV-L1 optical flow. In *Statistical and Geometrical Approaches to Visual Motion Analysis*, pages 23–45, Berlin, Heidelberg.

Weickert, J. and Schnörr, C. (2001). A theoretical framework for convex regularizers in pde-based computation of image motion. *International Journal of Computer Vision*, **45**(3), 245–264.

Weickert, J., Bruhn, A., Brox, T., and Papenberg, N. (2006). *A Survey on Variational Optic Flow Methods for Small Displacements*, pages 103–136. Berlin, Heidelberg.

Weiger, M. C., Vedham, V., Stuelten, C., Shou, K., Herrera, M., Sato, M., Losert, W., and Parent, C. (2013). Real-time motion analysis reveals cell directionality as an indicator of breast cancer progression. In *PloS one*, volume 8.

Weinzaepfel, P., Revaud, J., Harchaoui, Z., and Schmid, C. (2013). DeepFlow : Large displacement optical flow with deep matching. In *IEEE Intenational Conference on Computer Vision*, Sydney, Australia.

Wills, J., Agarwal, S., and Belongie, S. (2006). A feature-based approach for dense segmentation and estimation of large disparity motion. *International Journal of Computer Vision*, **68**(2), 125–143.

Wolberg, G. (1994). *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos, CA, USA, 1st edition.

Wulff, J., Butler, D. J., Stanley, G. B., and Black, M. J. (2012). Lessons and insights from creating a synthetic optical flow benchmark. In *ECCV Workshop on Unsolved Problems in Optical Flow and Stereo Estimation*, Part II, LNCS 7584, pages 168–177. Springer-Verlag.

Wyckoff, J. B., Pinner, S. E., Gschmeissner, S., Condeelis, J. S., and Sahai, E. (2006). Rock- and myosin-dependent matrix deformation enables protease-independent tumor-cell invasion in vivo. *Current Biology*, **16**(15), 1515 – 1523.

Yi, F., Mengtian, S., Minhyong, K., and Karthik, R. (2011). Heat-mapping : A robust approach toward perceptually consistent mesh segmentation. In *Computer Vision and Pattern Recognition*, pages 2145–2152, Colorado Springs.

Zabih, R. and Woodfill, J. (1994). Non-parametric local transforms for computing visual correspondence. In *Proc. of European Conference on Computer Vision*, pages 151–158, Stockholm.

Zach, C. *et al.* (2007). A duality based approach for realtime TV-L1 optical flow. In *Pattern Recognition*, pages 214–223.

Zagoruyko, S. and Komodakis, N. (2015). Learning to compare image patches via convolutional neural networks. *CoRR*, **abs/1504.03641**.

Zbytek, B., Carlson, A., Granese, J., Ross, J., Mihm, J. M., and Slominski, A. (2008). Current concepts of metastasis in melanoma. *Expert review of dermatology*, **3**(5), 569–585.

Zhang, Q., Xu, L., and Jia, J. (2014). 100+ times faster weighted median filter (wmf). In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2830–2837.

Zhang, W., Fehrenbach, J., Desmaison, A., Lobjois, V., Ducommun, B., and Weiss, P. (2016). Structure tensor based analysis of cells and nuclei organization in tissues. *IEEE Transactions on Medical Imaging*, **35**(1), 294–306.

Zhao, S., Li, S., and Bourahla, O. (2017). Deep optical flow estimation via multi-scale correspondence

structure learning. *CoRR*, **abs/1707.07301**.

Zhu, X. and Milanfar, P. (2010). A no-reference image content metric and its application to denoising. In *Proceedings of the International Conference on Image Processing*, pages 1145–1148, Hong Kong.

Zimmer, H., Bruhn, A., and Weickert, J. (2011). Optic flow in harmony. *International Journal of Computer Vision*, **93**(3), 368–388.

Zweig, S. and Wolf, L. (2016). Interponet, A brain inspired neural network for optical flow dense interpolation. *CoRR*, **abs/1611.09803**.

Titre : Estimation de mouvement 3D et évaluation dans des séquences volumiques de microscopie de fluorescence

Mots clés : Flot optique 3D, séquences d'images volumiques, microscopie à fluorescence

Résumé : Ce travail de thèse porte sur l'estimation et l'évaluation de champs de vitesse 3D dans des séquences d'images 3D de microscopie à fluorescence. Nous nous sommes intéressés aux méthodes de mise en correspondance et aux méthodes variationnelles pour l'estimation du mouvement entre volumes 3D de la séquence. Pour l'appariement, nous avons développé deux extensions 3D originales de PatchMatch, les deux incorporant une mesure de similarité exploitant la signature de Census discrète : une méthode exploitant les super-pixels et qui procède couche par couche dans le volume, et une méthode multi-résolution s'appliquant directement aux volumes. Nous avons par ailleurs conçu une méthode de segmentation des protubérances sur la surface de la cellule et une étape d'appariement des éléments segmentés reposant sur une représentation en mailles triangulaires. En ce qui concerne l'estimation dense des flots 3D, nous avons élaboré plusieurs méthodes variationnelles. Si toutes exploitent la signature Census continue des voxels dans le terme d'attache aux données, elles se distinguent par la nature du terme de régularisation : L2, L1 ou TV.

Nous avons également combiné la méthode 3D PatchMatch avec la méthode variationnelle pour appréhender à la fois des mouvements de grande et de petite amplitude.

Pour l'évaluation visuelle, nous avons proposé trois techniques différentes de visualisation des flots 3D par code couleur. Elles offrent des vues synthétiques 2D pertinentes du flot 3D calculé, respectivement couche par couche dans le volume, selon des projections tri-planaires, ou par affichage sur l'image obtenue par projection d'intensité maximale. De plus, nous avons défini une nouvelle mesure d'erreur quantitative pour évaluer la précision du flot 3D estimé, lorsqu'aucune vérité-terrain n'est disponible. Elle s'exprime comme la différence angulaire entre l'orientation principale locale dans le volume source et celle correspondante dans le volume rétro-reconstruit à partir du flot 3D calculé. Nous avons testé nos méthodes sur des séquences d'images microscopiques réelles contenant des cellules de mélanome MV3 dans un environnement de collagène. En comparant avec la méthode d'Amat et al. et notre extension 3D de la méthode classique de Horn-et-Schunck, nous avons pu en déduire que nos méthodes sont les plus performantes.

Title : 3D motion estimation and assessment in fluorescence microscopy volume sequences

Keywords : 3D optical flow, volumetric image sequences, fluorescence microscopy

Abstract : The thesis work deals with the computation and the assessment of 3D motion fields in 3D fluorescence microscopy image sequences. We have investigated 3D matching and variational methods for 3D flow field estimation between two consecutive volumes. For matching, we have developed two original 3D extensions of PatchMatch both involving the discrete Census similarity measure: a super-pixel based method that proceeds slice by slice, and a coarse-to-fine method directly applied to the volumes. We have also designed a protrusion segmentation method on the cell surface along with a matching stage relying on a triangular mesh-based representation. Regarding the dense estimation of 3D flow fields, we have adopted a variational approach, while exploiting the continuous Census signature of voxels in the data term. We have tested three regularization terms: L2, L1, and TV-based regularization.

We have also combined the 3D PatchMatch method with the variational method to be able to handle simultaneously large and small motion magnitude.

For visual assessment, we have proposed three different color-coded visualization techniques of 3D flow fields. They offer 2D summaries of the 3D flow field, respectively, slice-by-slice, with tri-planar projections, and after maximum intensity point projection. In addition, we have defined a new quantitative error measure for assessing the accuracy of the estimated flow field when no ground truth is available. It involves the angular difference between the local principal orientation of the original source volume and the corresponding one in the volume backward-warped with the 3D computed flow field. We have tested our methods on real microscopy image sequences containing MV3 melanoma cells in collagen environment. When comparing with the state-of-the-art method of Amat et al., and our 3D extension of the classical Horn-and-Schunck method, we found our proposed methods to be the best performing ones.