Université du Québec

Institut National de la Recherche Scientifique

Centre Énergie Matériaux & Télécommunications

Estimation Directe et Conjointe du Flot de Scène et de la Profondeur à Partir d'une Séquence Monoculaire d'Images

Par

Yosra Mathlouthi

Thèse présentée pour l'obtention

du grade de philosophiae doctor (Ph.D.)

President du Jury	Leszek Szczecinski INRS-ÉMT
Examinateur Externe	Carlos Vázquez ÉTS
Examinateur Externe	Matthew Toews ÉTS
Directeur de recherche	Amar Mitiche INRS-ÉMT
Codirecteur de recherche	Ismail Ben Ayed ÉTS

Remerciements

En premier lieu, je tiens à remercier énormément mon directeur de recherche Pr. Amar Mitiche non seulement pour m'avoir donné l'opportunité de travailler avec lui, pour son encadrement professionnel et pour sa contribution majeure dans cette thèse mais aussi pour ses conseils précieux et son soutien moral continue pendant cette route d'apprentissage et de perfectionnement. C'était une chance pour moi d'avoir travaillé avec un grand nom dans le domaine et une personne de haute qualités humaines et scientifiques.

Je tiens aussi à remercier mon co-directeur Pr. Ismail Ben Ayed pour sa supervision de qualité, pour ses commentaires instructives et ses suggestions enrichissantes, pour sa bienveillance pour le raffinement et le perfectionnement des travaux réalisés au cours de cette thèse, pour son aide continue qui m'a permit d'améliorer mes compétences, pour m'avoir offert la chance de travailler avec Pr. Amar Mitiche et m'ouvrir la porte sur un meilleur avenir en me proposant un stage à l'INRS.

Je remercie les membres de jury Pr. Leszek Szczecinski, Pr. Carlos Vázquez et Pr. Matthew Toews pour avoir accepté évaluer et juger mon travail.

Je remercie mes amis et mes collègues pour leurs soutiens continus sur les plans moral et scientifiques tout au long de cette belle expérience de quatre ans.

Un très grands merci à mon petit frère, notre rayon de soleil qui ne cessent de m'encourager et de me donner du support moral ainsi que des ondes positives et de l'énergie pour aller vers l'avant dans les situations difficiles.

J'offre cette thèse à ma très chère mère à qui tous les mots de gratitude du monde ne suffiraient jamais pour la remercier. C'est grâce à elle que je suis à ce niveau d'éducation.

Je dédie cette thèse également à l'âme de mon père qui a toujours cru en moi et à mes compétences. C'est le seul qui a su comment m'encourager pour faire face à toutes mes peurs et mes difficultés d'une manière humoriste.

Table des matières

Table des figures

Liste des tableaux

Résumé

Dans cette thèse, on étudie l'estimation conjointe du flot de scène dense et de la profondeur relative à partir d'une séquence d'images monoculaire. On commence par développer un schéma de base qui permet de poser le problème sous une forme variationnelle par une fonctionnelle composée de deux termes : un terme de conformité aux données spatiotemporelles de la séquence d'images et un terme de régularisation. Le terme de données relie la vitesse tridimensionnelle (3D) et la profondeur en termes de variations spatiotemporelles visuelles. Ce terme s'obtient en remplaçant les coordonnées du vecteur de vitesse optique dans la contrainte du gradient du flot optique de Horn et Schunck par leur expressions en termes du flot de scène et de la profondeur. Sous cette forme, l'énoncé de notre problème est analogue à l'estimation classique du flot optique proposée par Horn et Schunck, quoiqu'elle implique ici le flot de scène et la profondeur au lieu du mouvement de l'image.

En premier lieu, on utilise un terme de régularisation L^2 qui assure une solution lisse partout dans l'image. La discrétisation des équations d'Euler-Lagrange correspondantes à notre fonctionnelle forme un système creux à grande échelle d'équations linéaires. On écrit explicitement ce système et on ordonne ses équations de façon que sa matrice soit symétrique positive définie. Ceci implique que les itérations de Gauss-Seidel convergent point par point ou bloc par bloc, et offre un moyen très efficace pour résoudre les équations d'Euler-Lagrange.

En second lieu, une amélioration de la méthode étudiée est proposée par une

version qui préserve les frontières du mouvement et des objets dans la scène. Le terme de régularisation L^1 permet le lissage de la solution à l'intérieur des zones uniformes et l'inhibe sur les frontières de mouvement et de profondeur. La discrétisation des équations d'Euler-Lagrange correspondantes à la fonctionnelle de régularisation du type L^1 donne un grand système creux d'équations non-linéaires que l'on peut résoudre en alternant des approximations linéaires avec les itérations de Gauss-Seidel.

On considère aussi le problème inverse mal posé du calcul des dérivées spatio-temporelles qui sont nécessaires pour notre problème de flot de scène. On aborde ce calcul par une approche variationnelle, où la fonctionnelle objectif qu'on propose traite l'approximation d'une dérivée de l'image par la minimisation de la somme de deux termes : un terme d'adéquation de l'intégrale des dérivées à l'image et un terme de régularisation. Le terme de données utilise un opérateur d'anti-différentiation ce qui contraint la fonction recherchée à approximer les dérivées de l'image. Le terme de régularisation L^2 contraint la dérivée à être lisse sur tout le domaine de l'image. La discrétisation des équations d'Euler-Lagrange développées pour la minimisation de la fonctionnelle objectif donne lieu à un grand système creux d'équations linéaires que l'on peut résoudre par la méthode de Gauss-Seidel.

Le problème de calcul des dérivées spatio-temporelles est aussi amélioré par une version qui préserve les frontières des objets dans l'image en utilisant une fonction de régularisation L^1. Cette amélioration définit les dérivées de l'image comme des fonctions qui, lorsqu'elles sont intégrées, redonnent l'image à une constante additive près, et qui sont lisses partout sur le domaine de l'image sauf sur les frontières des objets dans l'image. Un grand système creux d'équations non-linéaires découle de la discrétisation des équations d'Euler-Lagrange correspondantes à la fonctionnelle du problème de calcul des dérivées spatio-temporelles amélioré. Ce système est résolu par approximations linéaires successives avec la méthode de Gauss-Seidel.

Nous prèsentons les résultats qualitatifs et quantitatifs de plusieurs tests, avec des

images synthétiques et réelles, qui montrent la validité et l'efficacité des méthodes proposées.

Chapitre 1

Introduction

1.1 Mise en contexte

Le traitement et l'analyse de données tridimensionnelles (3D) est un domaine qui suscite beaucoup d'intérêt en vision artificielle. L'importance de l'information 3D se manifeste dans plusieurs domaines d'application comme l'imagerie médicale [1–3], la robotique [4–6], les applications militaires et spatiales [7–9], et l'industrie du divertissement [10–12]. L'information 3D permet la détection, la capture et le suivi de mouvement, la modélisation de la forme des objets réels et la création d'effets visuels spéciaux. Dans le domaine de la robotique, par exemple, l'information 3D est nécessaire pour la navigation des robots mobiles autonomes, pour la manipulation par des robots d'objets réels et pour établir un schéma du relief de l'environnement des robots [13]. Dans les applications médicales, la forme et le comportement dynamique des organes sont souvent nécessaires au diagnostic des maladies. Les outils d'imagerie médicale 3D permettent des mesures et des analyses quantitatives ainsi que des modélisations géométriques et cinématiques. Aujourd'hui, les techniques d'imagerie 3D sont au service de l'échographie, la chirurgie endoscopique, la microscopie chirurgicale ainsi que l'enseignement et la formation. L'application des nouvelles tech-

nologies d'imagerie 3D dans le domaine médical a contribué à l'amélioration de la précision chirurgicale et de la sécurité des patients, et à la réduction du temps des opérations [14].

Les informations 3D peuvent être récupérées à partir d'une entrée visuelle par plusieurs méthodes, telles que la stéréoscopie, le mouvement et l'analyse de la forme de l'ombrage [15, 16]. Ceci est similaire aux principes du système visuel humain. Les informations 3D peuvent aussi être récupérées à partir d'autres techniques, qui ne sont pas similaires aux principes du système visuel humain, mais qui ont aussi été utilisées avec succès dans plusieurs applications de vision industrielle, telles que la lumière structurée [17]. Plusieurs méthodes permettant l'extraction de l'information 3D ont été étudiées et comparées pour explorer leurs mérites et leurs limites relatives [17–22].

Le sujet de l'analyse du mouvement a suscité plusieurs études de recherche en vision par ordinateur [23–28]. Parmi ces études, certaines ont abordé des problèmes similaires à ceux traités en recherche sur la vision humaine [29–31], y compris les premières préoccupations de Helmholtz et de Gibson concernant la perception du mouvement [32–34]. L'analyse du mouvement joue un rôle essentiel dans plusieurs applications de vision par ordinateur comme la description et la compression des vidéos, l'analyse de l'activité humaine et la robotique. En médecine, en sport, en kinésiologie et en surveillance par vidéo, l'analyse du mouvement humain est devenue un outil important d'investigation et de diagnostic [26, 35, 36]. L'analyse du mouvement humain peut être divisée en trois catégories : la reconnaissance d'activité humaine, le suivi du mouvement humain et l'analyse du mouvement de corps ou des parties de corps humain. En procédé de fabrication, par exemple, l'analyse de mouvement peut servir à surveiller et à analyser les lignes d'assemblage et les machines de production pour détecter les inefficacités ou les dysfonctionnements [37]. Plusieurs procédures sont réalisés sous le volet de l'analyse du mouvement tels que la détection du mouvement [38–40], la segmentation basée sur le mouvement [41–43], le suivi du

mouvement et [44–46] l'interprétation 3D du mouvement [47–49]. L'amélioration de ces procédures dépend alors de la précision du calcul du mouvement dans l'image.

Problème fondamental en vision artificielle, l'interprétation 3D du mouvement est une étape importante en analyse de séquences d'images. Le flot optique est le champ de vecteurs de vitesse optique des surfaces environnementales projetées sur l'image quand un système de visualisation se déplace relativement à l'environnement visualisé. Par conséquent, le flot optique comporte des informations sur les surfaces d'image et leurs mouvements. L'analyse 3D du flot optique consiste à récupérer les structures des objets visibles et leurs mouvements 3D relatifs dans la scène visualisée, et à segmenter l'image en se basant sur le mouvement [34]. L'interprétation 3D du mouvement joue un rôle important dans une variété d'applications, comme la manipulation guidée visuellement, la locomotion et la navigation robotique. Plusieurs études théoriques et expérimentales en vision par ordinateur ont révélé le lien entre le mouvement dans l'image et les variables 3D (le relief et le mouvement 3D) [34]. Cette constatation a encouragé plusieurs chercheurs à étudier les différentes méthodes d'extraction des informations 3D à partir du flot optique [50–57].

1.2 État de l'art

Dans le contexte de l'analyse 3D du mouvement, le sujet abordé dans cette thèse est l'étude d'une méthode qui permet l'estimation du flot de scène et de la profondeur relative (variables 3D) d'une scène à partir d'une seule séquence d'images. Le flot de scène est le champ des vecteurs de vitesse 3D des surfaces environnementales visibles dans le domaine de l'image. C'est seulement les surfaces visibles qui interviennent dans la définition du flot de scène car uniquement elles, et non pas les surfaces cachées, qui comportent de l'information visuelle. Il est donc important de connaitre les différentes méthodes d'analyse 3D du mouvement et d'identifier leurs mérites et limites relatives.

Les méthodes d'analyse 3D du mouvement peuvent être divisées en deux catégories : l'interprétation éparse [58–63] et l'interprétation dense [64–66]. Pour l'interprétation éparse, l'estimation des variables 3D (la profondeur et le mouvement 3D) est réalisée en quelques points. Ce sont les points saillants de l'image environnementale qui peuvent être facilement et systématiquement identifiés dans des vues distinctes de l'environnement. Contrairement à l'interprétation éparse, en interprétation dense, l'estimation de la profondeur et du mouvement 3D se fait en tous les points des surfaces visibles. L'apparition des méthodes d'interprétation éparse en [34, 58, 63, 67–70] a précédé celle des méthodes denses car cette dernière a nécessité l'apparition de la relation fondamentale de projection point par point entre l'environnement et son image. L'interprétation dense est considérée comme la plus complexe. Cependant, grâce au modèle de Longuet-Higgins et Prazdny [65], à l'algorithme de Horn et Schunck [71] et aux formulations variationnelles récentes des problèmes d'analyse 3D [34, 72–74], les traitements sont devenus plus efficaces. Pour simplifier le cas de l'interprétation dense, plusieurs études ont traité le cas d'un système de vision en mouvement dans un environnement statique. La segmentation de l'environnement selon le mouvement a aussi aidé à simplifier le problème [74–77].

Deux catégories de méthodes d'analyse 3D du mouvement ont été aussi considérées. Celles où le système de vision est en mouvement dans un environnement statique sont les plus simples car le problème revient à récupérer un seul mouvement 3D, qui est celui du système de visualisation [51, 78–88]. Celles où le système de vision et les objets observés de l'environnement sont en mouvement simultané et indépendant, ont été abordées dans plusieurs études de méthodes denses [53–56, 66, 75–77, 89]. Dans le cas général, il est indispensable de prendre en compte les frontières du mouvement dans l'interprétation de telle sorte que les objets en mouvement peuvent être délimités avec précision. La préservation des frontières du mouvement est un enjeu majeur dans l'interprétation 3D du flot optique [34].

Les méthodes d'analyse 3D du mouvement peuvent être classées en : directes et indirectes. L'interprétation est dite indirecte quand le flot optique est calculé et utilisé explicitement comme une donnée par le processus d'estimation des variables 3D [90–92]. Le flot optique peut être estimé indépendamment de l'interprétation 3D ou en parallèle avec elle. Des études psychophysiques ont montré que le système visuel humain procède d'une manière indirecte où l'environnement est visualisé et le champ du mouvement est traité dans l'image rétinienne avant l'interprétation 3D de la scène [33, 34, 93, 94]. Contrairement aux méthodes indirectes, les méthodes directes expriment les variables 3D d'une manière directe et explicite dans la formulation, sans recours au calcul préalable du flot optique [78, 95, 96]. Par exemple, en [65, 97, 98], les paramètres du flot optique sont remplacés par les variables du modèle 3D. Dans ce cas, l'interprétation est dite directe. Puisque le mouvement 3D est une fonction de la profondeur et du flot optique [90], l'estimation directe a été étudié dans le contexte des séquences d'images à points de vue multiples. Ces séquences permettent l'utilisation des contraintes à partir du flot optique à points de vue multiples et fournissent des moyens pour le calcul de la profondeur par correspondance [99, 100]. L'estimation directe du flot de scène a été abordée pour la première fois en [99]. Cette étude a d'abord focalisé sur un cas spécial où la profondeur et la correspondance stéréoscopique sont connues. Le cas général a été traité en utilisant la tesselation par voxel de l'espace et des tests photométriques de visibilité.

Le flot de scène peut être calculé selon deux méthodes : paramétrique et non-paramétrique. Les méthodes paramétriques, comme leur nom l'indique, utilisent une forme paramétrique pour définir les coordonnées du flot de scène à estimer. Par contre, les méthodes non-paramétriques calculent le champ de vecteurs du flot de scène directement sans recours à une représentation intermédiaire pour les mouvements ou les surfaces environnementales. Les méthodes non-paramétriques sont très utiles et présentent un avantage dans le cas où un modèle pratique réalisable pour

le mouvement ou pour la structure des surfaces n'est pas disponible. C'est souvent le cas du mouvement articulé humain ou animal [101]. En général les études du flot de scène paramétrique supposent que les objets environnementaux sont rigides, ce qui permet de décomposer le flot de scène en paramètres 3D de translation et de rotation [34, 53, 55, 74] et, aussi, d'utiliser des descriptions locales affines [102]. Les équations fondamentales de Longuet-Higgins et Prazdny [97] qui relient les paramètres du mouvement rigide, la profondeur et le flot optique sont utilisées dans la plupart des études pour l'estimation des variables 3D. Dans ce cas, le problème se résume à l'estimation de la profondeur et des paramètres du mouvement rigide en tant que variables 3D inconnues.

On peut aussi diviser les méthodes d'analyse 3D en deux catégories : variationnelles et non-variationnelles. Toutes les méthodes variationnelles utilisent des fonctionnelles composées d'un terme de données basé sur le modèle de Longuet-Higgins et Prazdny pour le mouvement rigide et un terme de régularisation qui prend en considération les discontinuités 3D pour préserver les frontières de mouvement 3D et de la profondeur [53–56, 66, 89]. Les méthodes non-variationnelles [75–77] supposent que le flot optique est déjà donné et segmentent le champ visuel en différents objets rigides en mouvement par des processus de regroupement, tels que les régions croissantes par mouvement 3D [76], le regroupement du mouvement 3D par des modèles de mixture [75] et le regroupement par projection orientée du flot optique [77]. Les méthodes variationnelles diffèrent dans la maniére dont elles décrivent les frontières à préserver par le terme de régularisation dans la fonctionnelle objectif. Par exemple, dans [66], la méthode discrète basée sur le principe de longueur de description minimale (MDL : Minimum Description Length) applique la segmentation par bloc constant MDL de Leclerc [103] à l'interprétation 3D du flot optique. En encodage MDL, les courbes délimitent des bords locaux plutôt que les frontières dans l'image. Ce manque d'information explicite sur les limites globales des régions conduit

généralement à une segmentation fragmentée. Dans le cas de formulation continue en interprétation 3D, les frontières peuvent être préservées en utilisant un terme de régularisation par longueur qui permet un lissage selon la direction tangente des frontières, mais pas selon la direction orthogonale aux frontières. Par exemple, dans ce contexte, la formulation de [54] minimise sur le domaine de l'image une intégrale contenant un terme de données basé sur le modèle de Longuet-Higgins et Prazdny pour mouvement rigide, et un terme de régularisation par diffusion anisotrope pour préserver les discontinuités de la profondeur. Dans ce cas, la segmentation basée sur le mouvement n'est pas abordée explicitement. Différemment, les frontières de mouvement peuvent être prises en compte par une fonctionnelle de courbes actives pour la segmentation 3D et l'interprétation 3D conjointe du flot optique [53, 73, 104]. Dans ce cas, la segmentation se réfère explicitement aux courbes actives pour représenter les frontières. Une telle approche a été étudiée dans [55] où la fonctionnelle objectif contient un terme de données pour chaque région à segmenter et des termes de régularité pour les frontières des régions et pour la profondeur. La minimisation de la fonctionnelle objectif conduit simultanément à une segmentation par évolution de courbes et à une estimation non linéaire de la profondeur relative. Dans le même contexte, l'estimation du flot optique et l'interprétation 3D simultanée a été abordée dans le cadre de la segmentation par courbe active dans [53,56]. L'expression linéarisée du modèle de mouvement rigide de Longuet-Higgins et Prazdny dans le terme des données utilisé pour l'estimation conjointe a conduit à une estimation linéaire de mouvement 3D dans les régions à segmenter. La segmentation s'est fondée sur le mouvement 3D dans [53] et sur le flot optique [56].

L'interprétation 3D du mouvement peut être réalisée à partir de séquences d'images stéréoscopiques, aussi nommées épipolaires, ou à partir d'une seule séquence d'images monoculaire. L'analyse 3D à partir d'une séquence monoculaire a été peu étudiée [64, 65, 105–107]. Dans les méthodes épipolaires, le champ des vitesses de mouve-

ment 3D peut être représenté conjointement par le champ de vitesse du mouvement 2D (le flot optique) et le champ de disparité. Dans ce cas, le mouvement 3D est déterminé par ces champs par le biais d'une contrainte stéréocinématique entre la vitesse optique et la disparité [90–92]. La vitesse optique et la disparité peuvent alors être estimées en même temps à partir de séquences d'images épipolaires par le biais de contraintes stéréocinématiques [108–115]. Puisque le flot de scène est relié à la profondeur par le biais du flot optique dans le cas de séquence d'images monoculaire [34], l'interprétation 3D du flot optique peut se faire par l'estimation du flot de scène, où aucun modèle de mouvement, rigide ou autre, n'est nécessaire. L'estimation non-pramétrique du flot de scène a été généralement traitée dans le contexte de la stéréoscopie [99,100,102,108–110,115–117], malgré que son étude peut être réalisée par l'analyse des séquences d'images monoculaires, indépendamment de la stéréoscopie et sans avoir recours aux contraintes stéréocinématiques [34]. Dans cette thèse, nous allons décrire des méthodes qui permettent l'estimation non-pramétrique du flot de scène à partir d'une séquence d'images monoculaire. Ces méthodes profitent du lien entre le flot de scène et la profondeur assuré par le biais du flot optique et l'appliquent dans un schéma qui rappelle la méthode de Horn et Schunck pour l'estimation du flot optique.

1.3 Défis

Dans ce travail, on s'intéresse à l'interprétation 3D du mouvement à partir d'une séquence d'images monoculaire. Notre méthode fait partie des méthodes d'analyse 3D basées sur le mouvement de l'image provenant du déplacement des objets dans la scène filmée ou du déplacement du système de visualisation par rapport à l'environnement filmé. On propose une méthode dense où la profondeur et le mouvement 3D sont estimés en chaque point de la grille d'échantillonnage du domaine de l'image.

Il s'agit d'une méthode variationnelle, directe et non paramétrique. La particularité de ce travail provient de la combinaison unique de ces caractéristiques qui présentent plusieurs difficultés et limitations intrinsèques :

(i) Il s'agit d'un problème mal posé au sens d'Hadamard puisqu'il n'existe pas une seule solution de structure et de mouvement 3D qui correspond aux variations spatiotemporelles de l'image ;

(ii) L'utilisation du flot optique impose aux déplacements réalisés entre les deux images successives traitées d'être petits. Dans le cas contraire, les processus de multi-résolutions et/ou multi-grilles peuvent être utiles ;

(iii) L'estimation dense de la structure et du mouvement 3D de la scène réelle mène à une grande complexité de calcul ;

(iv) La profondeur ne peut pas être récupérée pour les surfaces d'objets qui ne sont pas en mouvement relatif par rapport au système de vision et aussi pour les surfaces qui ont de faibles textures ou qui sont sans textures.

1.4 Contributions

Le but de cette thèse est l'estimation conjointe du flot de scène dense et de la profondeur à partir d'une *seule* séquence d'images et l'estimation des dérivées partielles de l'image avec une formulation variationnelle. La caractéristique du traitement monoculaire et non-paramétrique du problème et l'estimation conjointe du flot de scène et de la profondeur rend ce travail distinct des autres méthodes qui ont utilisés des séquences d'images stéréoscopiques et non pas des séquences d'images monoculaires.

Le travail réalisé dans cette thèse se divise essentiellement en quatre parties :

1− *L'estimation directe et conjointe du flot de scène et de la profondeur à partir d'une séquence monoculaire utilisant une régularisation* L^2. Une approche de base est développée selon un enoncé variationnel qui rappelle la méthode de Horn et

Schunck [71] connue comme référence dans le contexte de l'estimation du flot optique. Ce schéma minimise une fonctionnelle à deux termes : un terme de conformité conjointe du flot de scène et de la profondeur aux données spatiaux-temporelles de la séquence d'images qui relie la vitesse 3D et la profondeur et un terme de régularisation quadratique (L^2) qui assure une solution lisse. Le terme de données s'obtient en remplaçant les coordonnées du vecteur de vitesse optique dans la contrainte du gradient du flot optique de Horn et Schunck [71] par leurs expressions en termes du flot de scène et de la profondeur. Par conséquent, l'estimation du flot de scène sous cet énoncé du problème est analogue à l'estimation classique du flot optique proposée par Horn et Schunck, quoiqu'elle implique ici le flot de scène et la profondeur au lieu du mouvement de l'image.

2– *L'estimation des dérivées spatiales régularisées d'une image suivant une approche variationnelle d'anti-différentiation avec une régularisation* L^2. La différentiation est souvent approximée par des différences finies, lesquelles sont très sensibles au bruit. Généralement, la procédure de dé-bruitage ne s'avère pas efficace face au bruit. On propose une méthode variationnelle qui peut être capable de calculer les dérivées d'image d'une manière beaucoup plus précise. Les dérivées obtenues sont non seulement conformes à l'image globalement, mais aussi régularisées en conférant des propriétés à leurs variations. La méthode proposée vise à régulariser le processus de différentiation afin d'éviter les erreurs de calcul qui dérivent des données bruitées. Cette approche variationnelle consiste à minimiser une fonctionnelle à deux termes : un terme d'adéquation des dérivées à l'image et un terme de régularisation par lissage. Le terme des données utilise un opérateur d'anti-différentiation. Le terme de lissage est une régularisation quadratique (L^2).

3– *L'amélioration du calcul de l'estimation des dérivées partielles pour traiter une version qui préserve les frontières en utilisant une régularisation* L^1 : Cette partie se base sur une méthode variationnelle de régularisation L^1 pour l'estimation des

dérivées partielles régularisées. Dans ce cas, aussi, on minimise une fonctionnelle à deux termes : un terme d'adéquation des dérivées à l'image par un opérateur d'anti-différentiation et un terme de régularisation L^1. Celle ci permet le lissage à l'intérieur des zones uniformes et l'inhibe à travers les frontières. Ce type de lissage préserve les frontières des objets dans l'image.

4– *L'amélioration du calcul de l'estimation du flot de scène pour traiter une version de la méthode qui préserve les frontières en utilisant une régularisation* L^1 : Cette partie se base sur une méthode variationnelle de régularisation L^1 pour l'estimation monoculaire du flot de scène et de la structure 3D. De la même manière que ce qui est fait dans la première partie, on minimise une fonctionnelle à deux termes : un terme de conformité de données et un terme de régularisation L^1. Ce type de régularisation préserve les frontières des mouvements 3D et des profondeurs des objets dans l'image puisqu'il permet un lissage à l'intérieur des zones uniformes et l'inhibe à travers les frontières.

1.5 Plan de la thèse

1.5.1 Chapitre 2

Le chapitre 2 développe une approche variationnelle et non-paramétrique proposée pour l'estimation dense, directe et conjointe du flot de scène et de la profondeur à partir d'une séquence monoculaire où un mouvement simultané des objets dans la scène et de la caméra peut avoir lieu.

L'objectif de l'interprétation 3D du flot optique est l'estimation de la structure et du mouvement des surfaces environnementales visibles et la segmentation de l'environnement en différents objets en mouvement. Dans ce contexte, on propose dans ce chapitre une procédure qui permet de récupérer conjointement le flot de scène et la

profondeur relative. Le flot de scène est le champ de vitesse 3D des surfaces visibles de l'environnement. Il représente un élément fondamental dans l'analyse 3D des scènes puisqu'il décrit le mouvement des objets environnementaux réels. Le calcul du flot de scène dense à partir d'une séquence d'images représente un défi [34, 117] malgré l'évolution considérable réalisée dans l'estimation du flot optique [118,119]. Contrairement au flot optique qui a été l'objet de plusieurs études pendant les trente dernières années [53,114,119–121], ce n'est que récemment que les travaux de recherche se sont consacrés pour l'étude du flot de scène [34, 99, 108–110, 116, 117, 122].

Dans ce chapitre, le problème de l'estimation dense, directe et conjointe du flot de scène et de la profondeur à partir d'une séquence monoculaire est posé sous une forme variationnelle. La formulation minimise une fonctionnelle à deux termes : un terme de conformité aux données spatiotemporelles de la séquence d'images qui relie la vitesse 3D et la profondeur et un terme de régularisation L^2 pour assurer une solution lisse. Le terme de données s'obtient en remplaçant les coordonnées du vecteur de vitesse optique dans la contrainte du gradient du flot optique de Horn et Schunck [71] par leur expressions en termes du flot de scène et de la profondeur. Par conséquent, l'estimation du flot de scène sous cet énoncé du problème est analogue à l'estimation classique du flot optique proposée par Horn et Schunck, quoiqu'elle implique ici le flot de scène et la profondeur au lieu du mouvement de l'image. Puis, un système creux à grande échelle d'équations linéaires qui découle de la discrétisation des équations d'Euler Lagrange correspondantes à la fonctionnelle est résolu d'une manière itérative.

Soit $I : (x, y, t) \rightarrow I(x, y, t)$ une séquence d'images, où (x, y) sont les coordonnées spatiales définies dans le domaine borné Ω et $t \in \mathbb{R}^+$ est la coordonnée temporelle. Soit u et v les fonctions coordonnées du flot optique. La contrainte du gradient du flot optique de Horn et Schunck qui relie u et v aux variations spatiotemporelles est :

$$I_x u + I_y v + I_t = 0 \tag{1.1}$$

Où I_x, I_y et I_t sont les dérivées spatiotemporelles de l'image.

Soit **P** un point dans l'espace, (X, Y, Z) sont ses coordonnées 3D et (x, y) sont ses coordonnées dans l'image. La géométrie du modèle du système de vision est montrée dans la figure (1.1).

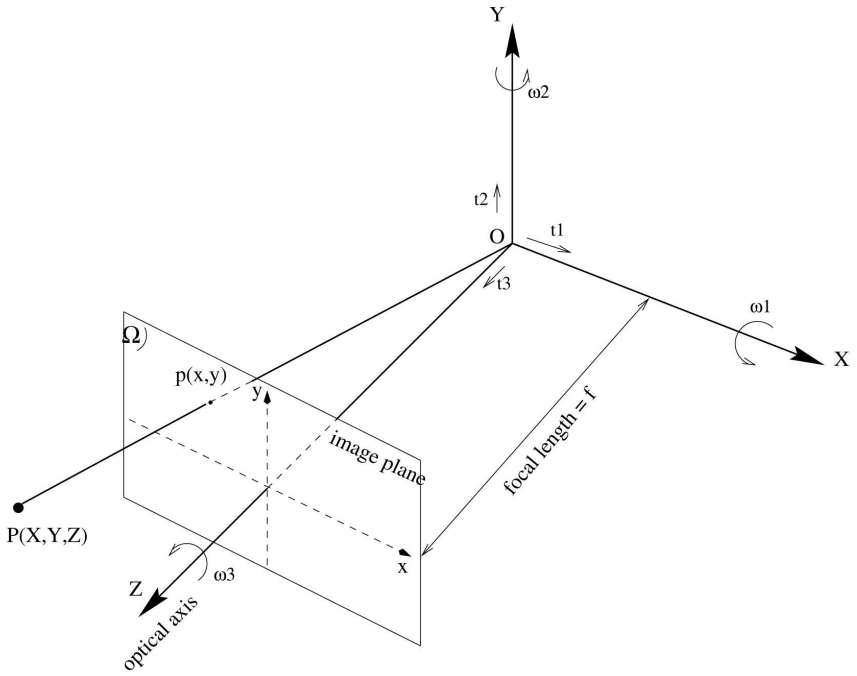


FIGURE 1.1 – Le système de vision est symbolisé par un système de référence cartésien $(\mathbf{O}; \mathbf{X}, \mathbf{Y}, \mathbf{Z})$, où $\mathbf{X}, \mathbf{Y}$ et $\mathbf{Z}$ sont les vecteurs unitaires selon les axes X, Y et Z, et par une projection centrale à travers l'origine **O**. L'axe Z est l'axe des profondeurs. Le plan d'image π est orthogonal à l'axe des profondeurs à une distance f (c'est la distance focale) du centre **O**.

La dérivée temporelle des équations des projections du point **P** ($x = f\frac{X}{Z}$ et $y = f\frac{Y}{Z}$, où f est la distance focale), donne les coordonnées u et v de la vitesse optique en fonction du flot de scène et de la profondeur : $u = \frac{dx}{dt} = \frac{fU - xW}{Z}$; $v = \frac{dy}{dt} = \frac{fV - yW}{Z}$,

où Z est la profondeur (Fig. 1.1) et $(U, V, W) = (\frac{dX}{dt}, \frac{dY}{dt}, \frac{dZ}{dt})$ est le flot de scène en $\mathbf{P}$. La substitution des coordonnées u et v du flot optique par leurs expressions dans l'équation de contrainte du gradient (1.1), suivie par sa multiplication par $Z \neq 0$, donne la contrainte linéaire suivante qui relie le flot de scène et la profondeur aux dérivées spatiotemporelles :

$$fI_xU + fI_yV - (xI_x + yI_y)W + I_tZ + I_tZ_0 = 0, \tag{1.2}$$

Où Z_0 est la profondeur relative au plan fronto-parallèle $\Pi_{Z_0} : Z = Z_0$. C'est une profondeur arbitraire positive qui sert à fixer de l'échelle de l'interprétation.

La solution du problème de l'estimation conjointe du flot de scène et de la profondeur relative à partir d'une séquence d'images monoculaire minimise la fonctionnelle objectif suivante :

$$\begin{aligned} \mathbf{E}(U, V, W, Z|I) &= \frac{1}{2}\int_\Omega (fI_xU + fI_yV - (xI_x + yI_y)W + I_tZ + I_tZ_0)^2 dxdy \\ &+ \frac{\alpha}{2}\int_\Omega (\|\nabla U\|^2 + \|\nabla V\|^2 + \|\nabla W\|^2)dxdy + \frac{\beta}{2}\int_\Omega \|\nabla Z\|^2 dxdy, \end{aligned} \tag{1.3}$$

où α et β sont des constantes positives qui balancent la contribution du terme de lissage dans la fonctionnelle et ∇ est le gradient spatial.

La discrétisation des équations d'Euler Lagrange correspondantes à la minimisation de la fonctionnelle (1.3) produit un système creux à grande échelle d'équations linéaires qui peut s'écrire sous une forme matricielle :

$$\mathbf{Aq} = \mathbf{r} \tag{1.4}$$

où $\mathbf{A}$ est une matrice de taille $4N \times 4N$ dont les éléments sont exprimés en fonction des valeurs des positions et des variations spatiotemporelles des pixels dans l'image, $\mathbf{q}$ est un vecteur de taille $4N$ qui contient les valeurs inconnues du flot de scène et de la profondeur, et $\mathbf{r}$ est un vecteur de taille $4N$ dont les éléments exprimés sont en

fonction des valeurs des positions et des variations spatiotemporelles des pixels dans l'image.

La résolution itérative désignée pour les matrices creuses est la méthode la mieux adaptée pour ce type de système [123, 124].

1.5.2 Chapitre 3

Le but du chapitre 3 est l'estimation des dérivées spatiales régularisées d'une image suivant une approche variationnelle avec une régularisation L^2. Les dérivées d'images sont présentes dans plusieurs problèmes de traitement d'images et de vision par ordinateur, comme l'estimation et la détection du mouvement [34, 71, 101, 119], le recalage d'image [125, 126], l'estimation d'images intrinsèques à partir d'une seule image [127], et la reconnaissance de formes et d'objets 3D [128]. Par conséquence, il est important de réaliser un schéma d'estimation de dérivées d'images avec une haute précision.

En vision par ordinateur et particulièrement en analyse du mouvement [34,71,101, 119], les dérivées spatiotemporelles d'une image sont généralement calculées par une somme locale des différences finies de l'image. L'approximation de la fonction dérivée par des différences finies est un problème mal posé puisque même des petites perturbations dans les valeurs de la fonction peuvent causer des changements importants et arbitraires de la dérivée [129–131]. Par conséquent, le bruit dans une image peut affecter d'une manière significative la qualité des interprétations des images traitées ultérieurement par des processus qui utilisent ces dérivées d'images. Le pré-traitement des images par des filtres pour réduir le bruit ne résout pas le problème mal posé de l'approximation de la fonction dérivée par des différences finies. Ce traitement serait alors une opération inefficace et futile [131]. En plus, la procédure de dé-bruitage est compliquée, lourde et peu efficace.

Pour réduire l'effet indésirable du bruit dans une image quand on calcule ses

dérivées, les algorithmes utilisés en vision par ordinateur se servent souvent des moyennes locales de différences finies [71, 101, 119, 125–127]. Par exemple, en analyse du mouvement, pour réduire l'effet du bruit, le problème de l'estimation des dérivées spatiotemporelles I_x, I_y et I_t à partir de deux images consécutives dans une séquence est généralement traité en utilisant la formule citée dans le papier de Horn and Schunck [71]. Cette formule reste entièrement basée sur la notion de l'approximation par des différences finies. Par conséquence, cette définition ne correspondraient pas aux structures générales du bruit d'image et le problème mal posé est réduit mais il est encore présent.

Dans ce chapitre, on propose une méthode variationnelle qui peut calculer les dérivées d'image d'une manière beaucoup plus précise. Les dérivées recherchées sont non seulement conformes à l'image globalement, mais aussi régularisées en conférant des propriétés à leurs variations. La méthode proposée vise à régulariser le processus de différentiation afin d'éviter les erreurs de calculs qui dérivent des données bruitées. Les dérivées partielles spatiales de l'image vont être estimées par une méthode variationnelle qui minimise une fonctionnelle à deux termes : un terme d'adéquation des dérivées à l'image et un terme de régularisation par lissage. Le terme des données utilise un opérateur d'anti-différentiation qui a été utilisé jusqu'ici dans la communauté de mathématique computationnelle pour les fonctions réelles à une seule variable [131,132], mais non pas pour le traitement d'image. À notre connaissance, notre méthode est la première à exploiter l'anti-différentiation pour lr traitement d'image. Le terme de pénalité est une régularisation L^2 qui permet d'obtenir une solution lisse.

Soit $I : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ la fonction image, avec n la dimension de l'image. On va présenter le cas de $n = 2$ mais ce travail peut s'étendre directement à une dimension arbitraire n. Soit I_x et I_y les dérivées partielles spatiales de I dans le cas bidimentionnel (2D). Dans ce qui suit, on décrit la méthode de calcul pour I_x. En utilisant la transposée de l'image, la dérivée I_y peut alors être résolue de la même manière que

I_x. Puisque l'axe des temps est échantillonné uniquement en deux points, la formulation ne va pas être appliquée à la dérivée temporelle I_t. Cependant, I_t peut être simplement estimée par la formule de Horn and Schunck [71] .

La dérivée partielle I_x qu'on cherche, minimise la fonctionnelle suivante :

$$E(g) = \frac{1}{2}\int_{\Omega} \left(\|Dg - I\|^2 + \lambda\|\nabla g\|^2\right) dxdy \tag{1.5}$$

où ∇ est le gradient spatial, λ est une constante positive et D est l'opérateur intégrale de l'anti-différentiation, défini par :

$$Dg(x,y) = \int_0^x g(z,y)dz. \tag{1.6}$$

On obtient les conditions nécessaires pour un minimum de (1.5) par la résolution des équations d'Euler-Lagrange suivantes :

$$D^*(Dg - I) - \lambda\nabla^2 g = 0 \tag{1.7}$$

où D^* est l'opérateur adjoint de D, défini par :

$$D^*g(x,y) = \int_x^l g(z,y)dz. \tag{1.8}$$

La discrétisation de l'équation (1.7) produit un système d'équations linéaires creux à grande échelle. Pour simplifier les notations, les symboles des opérateurs linéaires D et D^* en (1.7) vont être réutilisés pour leurs matrices discrétisées correspondantes. On utilise la méthode des trapèzes pour l'approximation des intégrales [133] pour définir les matrices D et D^*. Le Laplacien dans (1.7) peut être approximé par $\lambda\sum_{j\in\mathbf{N}_i}(g_j - g_i)$. Le système d'équations linéaires à résoudre peut s'écrire sous une forme matricielle :

$$(D^*D - L)\,\mathbf{g} = D^*I, \tag{1.9}$$

où L est la matrice de l'opérateur Laplacien. Ceci est un système d'équations linéaires creux à grande échelle qui peut être résolu d'une manière efficace en utilisant les méthodes itératives de Gauss-Seidel [134].

1.5.3 Chapitre 4

L'article présenté au chapitre 4 est une preuve de concept qui regroupe quelques idées du deuxième et troisième chapitre. Dans le chapitre 4, on rappelle la méthode d'estimation du flot de scène proposée dans le chapitre 2, ainsi que la méthode de calcul des dérivées partielles expliqué dans le chapitre 3. On applique la méthode des dérivées partielles régularisées à l'algorithme du calcul du flot de scène et on évalue expérimentalement son effet en utilisant des exemples de séquences d'images synthétiques et réelles. Les résultats obtenus montrent que l'estimation du flot de scène avec des dérivées régularisées est plus performante en termes de précision que celle qui utilise une moyenne de différences finies pour le calcul des dérivées. En tant que deuxième auteur, ma contribution était au niveau de la formulation numérique et son implémentation expérimentale.

L'approche variationnelle et non-paramétrique proposée dans le chapitre 2 pour l'estimation dense, directe et conjointe du flot de scène et de la profondeur à partir d'une séquence d'images monoculaire minimise la fonctionnelle objectif suivante selon le flot de scène (U, V, W) et la profondeur Z :

$$\begin{aligned} \mathbf{E}(U,V,W,Z|I) &= \frac{1}{2}\int_{\Omega}(fI_xU + fI_yV - (xI_x + yI_y)W + I_tZ + I_tZ_0)^2 dxdy \\ &+ \frac{\alpha}{2}\int_{\Omega}(\|\nabla U\|^2 + \|\nabla V\|^2 + \|\nabla W\|^2)dxdy + \frac{\beta}{2}\int_{\Omega}\|\nabla Z\|^2 dxdy. \end{aligned} \tag{1.10}$$

où α et β sont des constantes positives qui balancent la contribution du terme de lissage dans la fonctionnelle et ∇ est le gradient spatial. La fonctionnelle (1.10) fait intervenir les dérivées partielles de l'image (I_x, I_y et I_t). Pour améliorer les résultats de l'estimation du flot de scène, les dérivées partielles de l'image figurant dans (1.10) sont calculées par la méthode proposée dans le chapitre 3. Il s'agit d'une méthode variationnelle qui calcule les dérivées de l'image d'une manière beaucoup plus performante en termes de précision que celle des moyennes des différences finies, qui seraient non

seulement conformes à l'image globalement, mais qui seraient aussi régularisées en conférant des propriétés à leurs variations. La méthode proposée vise à régulariser le processus de différentiation afin d'éviter les erreurs de calcul qui dérivent des données bruitées en minimisant la fonctionnelle suivante selon la dérivée partielle g :

$$E(g) = \frac{1}{2} \int_{\Omega} \left(\|Dg - I\|^2 + \lambda \|\nabla g\|^2 \right) dxdy \tag{1.11}$$

où ∇ est le gradient spatial, λ est une constante positive et D est l'opérateur intégrale de l'anti-différentiation défini par $Dg(x, y) = \int_0^x g(z, y)dz$.

1.5.4 Chapitre 5

Dans le chapitre 5, nous étudions la différentiation d'images par une méthode variationnelle qui permet de préserver les frontières des objets dans l'image. Il s'agit d'une amélioration du problème abordé dans le chapitre 3, où on a utilisé un terme de conformité de données sous forme d'un opérateur d'anti-différentiation et un terme de régularisation Tikhonov qui permet le lissage de la dérivée calculée partout sur le domaine de l'image. Dans le chapitre 5, on va étendre la formulation à une version de la fonctionnelle objectif qui préserve les frontières en utilisant une fonction de régularisation L^1. Par conséquent, cette amélioration définie les dérivées de l'image comme des fonctions qui, lorsqu'elles sont intégrées, redonnent l'image à une constante additive près, et qui sont lisses partout sur le domaine de l'image sauf sur les frontières des objets dans l'image. Trois raisons justifient l'utilisation de la régularisation L^1 pour préserver les frontières : (i) sa capacité à préserver des frontières aiguës et nettes en pénalisant les oscillations, (ii) elle peut être implémentée par des approximations de calculs efficaces sans affecter d'une manière notable la précision des résultats et, (iii) il existe une littérature importante qui la supporte [135].

Soit $I : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ une image et I_x et I_y ses dérivées partielles spatiales. Ici, on décrit la méthode de calcul pour I_x, la derivée I_y peut être résolue de la même

manière en utilisant la transposée de l'image. Puisque l'axe du temps est échantillonné uniquement en deux points, la formulation ne va pas être appliquée à I_t. Cependant, I_t peut être simplement estimée par des moyennes locales des différences finies de l'image données comme en [71]. On part de la fonctionnelle avec régularisation L^2 proposée dans le chapitre 3 :

$$E(g) = \frac{1}{2}\int_\Omega \left(\|Dg - I\|^2 + \lambda\|\nabla g\|^2\right) dxdy, \tag{1.12}$$

où g est la fonction qui présente l'une des deux dérivées de l'image (I_x ou I_y), $\nabla g = (g_x, g_y)$ est le gradient spatial, λ est une constante positive et D est l'opérateur intégrale de l'anti-différentiation selon x. On remplace la régularisation Tikhonov par une régularisation L^1, la nouvelle fonctionnelle à minimiser par rapport à la fonction g est :

$$E(g) = \frac{1}{2}\int_\Omega \left(\|Dg - I\|^2 + \lambda\|\nabla g\|\right) dxdy, \tag{1.13}$$

L'opérateur L^1 dans le terme de régularisation est défini par $\|\nabla g\| = \left(g_x^2 + g_y^2\right)^{\frac{1}{2}}$. Les équations d'Euler-Lagrange correspondantes à (1.13) sont :

$$D^*(Dg - I) - \lambda\frac{\partial}{\partial x}\frac{g_x}{\left(g_x^2 + g_y^2\right)^{\frac{1}{2}}} - \lambda\frac{\partial}{\partial y}\frac{g_y}{\left(g_x^2 + g_y^2\right)^{\frac{1}{2}}} = 0, \tag{1.14}$$

où D^* est l'opérateur adjoint de D. Pour éviter la non-différentiabilité qui peut être causée par les dénominateurs en (1.14), dans la pratique, on peut remplacer l'expression de ces dénominateurs par $\left(g_x^2 + g_y^2 + \epsilon\right)^{\frac{1}{2}}$, où ϵ est une petite valeur positive. En général, cette opération n'affecte pas les résultats de calcul de manière significative. Les équations d'Euler-Lagrange correspondantes à (1.13) sont non-linéaire. Cependant, elles peuvent être résolus d'une manière efficace par des approximations linéaires successives : au cours de l'itération courante, les termes non linéaires calculés à l'itération précédente sont utilisés comme données, ce qui mène à résoudre une équation linéaire. Plus précisément, à l'itération actuelle k, on considère l'équation

suivante :

$$D^*(Dg^k - I) - \frac{\lambda}{\left((g_x^{k-1})^2 + (g_y^{k-1})^2 + \epsilon\right)^{\frac{1}{2}}}\nabla^2 g^k = 0, \tag{1.15}$$

En pratique, à l'itération k, le terme du Laplacien en (1.15) est discrétisé au point i comme une proportion fixe de $\sum_{j\in\mathcal{N}_i}(g_j^k - g_i^k)$. La discrétisation de (1.15) produit à chaque itération k, un système d'équations linéaires creux et à grande échelle qui peut être résolu en utilisant la méthodes itératives de Gauss-Seidel [134].

1.5.5 Chapitre 6

Dans Le chapitre 6, nous étudions l'estimation directe et conjointe du flot de scène et de la profondeur à partir d'une séquence monoculaire par une méthode variationnelle qui permet de préserver les frontières du mouvement 3D et de la profondeur. Il s'agit d'une amélioration du problème abordé dans le chapitre 2. Ce schéma minimise une fonctionnelle à deux termes : un terme de conformité de données qui relie la vitesse 3D et la profondeur en termes de variations spatiotemporelles visuelles et un terme de régularisation L^1 pour préserver les frontières. Les formulations variationnelles basiques utilisent une régularisation L^2 (Tikhonov) qui impose à la solution un lissage sur tout le domaine de l'image, ce qui mène à des équations d'Euler-Lagrage linéaires : c'est le cas qu'on a décrit dans le chapitre 2 et qu'on va améliorer dans le chapitre 6.

Vu qu'elle impose à la solution un lissage sur tout le domaine de l'image, la régularisation L^2 estompe les frontières du flot de scène et de la profondeur calculées. En général, une certaine forme de régularisation qui préserve les frontières est nécessaire. Ceci est vrai particulièrement lorsque le mouvement des objets environnementaux est indépendant de celui du système de visualisation car la variation du mouvement et de la structure, dans ce cas, peut être brusque, acérée et significative aux frontières occlues de ces objets. Par conséquent et pour plus de précision, ces

frontières doivent être préservées par l'opérateur de régularisation.

L'estimation du flot de scène et de la profondeur avec conservation des frontières peut être spécifiée de différentes manières [136]. Par exemple, au lieu d'une régularisation L^2, on peut utiliser la fonction de Aubert et al. [137, 138], ou la régularisation L^1. On peut également empêcher le lissage à travers les frontières par l'estimation conjointe du mouvement 3D et de la segmentation [53]. Dans le chapitre 6, on a choisit d'appliquer la régularisation L^1 pour trois raisons :

(i) La capacité de la contrainte L^1 à préserver les frontières acérées, tout en pénalisant les oscillations. Ceci est bien adapté pour les "images en blocs" qui tendent à être lisses à l'intérieur des régions dont les frontières sont significativement acérées. En général c'est le cas typique des champs de mouvement ;

(ii) En pratique, elle peut être implémentée par des approximations de calculs efficaces sans affecter d'une manière notable la précision des résultats et ;

(iii) Il existe une littérature importante qui la supporte, en particulier dans la restauration d'images [135].

Soit $I : (x, y, t) \rightarrow I(x, y, t)$ une séquence d'images, où (x, y) sont les coordonnées spatiales définies dans le domaine borné Ω et $t \in \mathbb{R}^+$ est la coordonnée temporelle. Soit (X, Y, Z) les coordonnées 3D d'un point $\mathbf{P}$ dans l'espace et (x, y) sont les coordonnées de sa projection sur l'image. Le système de coordonnées et la géométrie du modèle de système de vision sont montrés dans la figure (1.1). Soit U, V et W les coordonnées fonctions du flot de scène. La contrainte du gradient linéaire du flot de scène et de la profondeur (expliquée dans le chapitre 2) qui relie les coordonnées du flot de scène U, V, W et la profondeur Z aux variations spatiotemporelles de l'image est :

$$fI_xU + fI_yV - (xI_x + yI_y)W + I_tZ + I_tZ_0 = 0 \tag{1.16}$$

Où I_x, I_y et I_t sont les dérivées spatiotemporelles de l'image qui vont être estimées dans le chapitre 6 par la méthode proposée dans le chapitre 5, Z est la profondeur (Fig.

1.1), $(U,V,W) = (\frac{dX}{dt}, \frac{dY}{dt}, \frac{dZ}{dt})$ est le flot de scène au point $\mathbf{P}$ et Z_0 est la profondeur relative au plan fronto-parallèle $\Pi_{Z_0} : Z = Z_0$. Dans la formulation du chapitre 2, le flot de scène et la profondeur relative résulte de la minimisation de la fonctionnelle régularisée L^2 suivante :

$$\begin{aligned}\mathbf{E}(U,V,W,Z|I) &= \tfrac{1}{2}\int_\Omega (fI_xU + fI_yV - (xI_x + yI_y)W + I_tZ + I_tZ_0)^2 dxdy \\ &+ \tfrac{\alpha}{2}\int_\Omega (\|\nabla U\|^2 + \|\nabla V\|^2 + \|\nabla W\|^2)dxdy + \tfrac{\beta}{2}\int_\Omega \|\nabla Z\|^2 dxdy,\end{aligned} \tag{1.17}$$

où α et β sont des constantes positives qui balancent la contribution relative du terme de lissage.

Pour la version qui préserve les frontières, on remplace pour chaque variable la régularisation L^2 par une régularisation L^1, à savoir :

$$\int_\Omega \|\nabla Q\| dxdy = \int_\Omega \left(Q_x^2 + Q_y^2\right)^{\frac{1}{2}} dxdy, \tag{1.18}$$

où $Q \in \{U,V,W,Z\}$. La fonctionnelle objectif est alors :

$$\begin{aligned}\mathbf{E}(U,V,W,Z|I) &= \tfrac{1}{2}\int_\Omega (fI_xU + fI_yV - (xI_x + yI_y)W + I_tZ + I_tZ_0)^2 dxdy \\ &+ \tfrac{\alpha}{2}\int_\Omega ((U_x^2 + U_y^2)^{\frac{1}{2}} + (V_x^2 + V_y^2)^{\frac{1}{2}} + (W_x^2 + W_y^2)^{\frac{1}{2}})dxdy \\ &+ \tfrac{\beta}{2}\int_\Omega (Z_x^2 + Z_y^2)^{\frac{1}{2}} dxdy.\end{aligned} \tag{1.19}$$

Les équations d'Euler-Lagrange correspondantes à la fonctionnelle (1.19) sont non-linéaires. La discrétisation de ces équations produit un système creux à grande échelle d'équations non-linéaires. En analyse numérique, pour résoudre de tels systèmes d'équations non-linéaires on utilise généralement une méthode itérative où les termes non linéaires sont évalués à l'itération précédente. Ces termes sont alors traités comme données à l'itération courante. Dans ce cas, les équations à résoudre sont linéaires.

Le système qui en découle est alors un système creux à grande échelle d'équations linéaires. De la même manière proposée dans le chapitre 2, ce système peut être résolu par la méthode itérative de Gauss-Seidel [123, 124, 133].

1.6 Liste de publications

1. **Y. Mathlouthi**, A. Mitiche, and I. Ben Ayed, "Monocular, boundary preserving joint recovery of scene flow and depth," *Frontiers in ICT, 2016* (accepté).
2. **Y. Mathlouthi**, A. Mitiche, and I. Ben Ayed, "Boundary preserving variational image differentiation," *Proc. GCPR,* LNCS 9796, pp. 355-364, 2016, Springer.
3. **Y. Mathlouthi**, A. Mitiche, and I. Ben Ayed, "Calcul variationnel des dérivées d'une image et application l'estimation du flot optique et du flot de scène," *Proc. RFIA,* 2016.
4. **A. Mitiche**, Y. Mathlouthi, and I. Ben Ayed, "Monocular Concurrent Recovery of Structure and Motion Scene Flow," *Frontiers in ICT,* vol. 2, p. 16, 2015.
5. **Y. Mathlouthi**, A. Mitiche, and I. Ben Ayed, "Direct Estimation of Dense Scene Flow and Depth from a Monocular Sequence," *Advances in Visual Computing, Proc. ISVC,* LNCS 8887, pp. 107-117, 2014, Springer..

Bibliographie

[1] C. Studholme, D. L. Hill, and D. J. Hawkes, "An overlap invariant entropy measure of 3D medical image alignment," *Pattern recognition*, vol. 32, no. 1, pp. 71–86, 1999.

[2] T. Heimann and H.-P. Meinzer, "Statistical shape models for 3D medical image segmentation : a review," *Medical image analysis*, vol. 13, no. 4, pp. 543–563, 2009.

[3] W.-C. Lin, C.-C. Liang, and C.-T. Chen, "Dynamic elastic interpolation for 3D medical image reconstruction from serial cross sections," *IEEE transactions on medical imaging*, vol. 7, no. 3, pp. 225–232, 1988.

[4] S. May, B. Werner, H. Surmann, and K. Pervolz, "3D time-of-flight cameras for mobile robotics," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 790–795.

[5] J.-F. Lalonde, N. Vandapel, D. F. Huber, and M. Hebert, "Natural terrain classification using three-dimensional ladar data for ground robot mobility," *Journal of field robotics*, vol. 23, no. 10, pp. 839–861, 2006.

[6] M. Johnson-Roberson, O. Pizarro, S. B. Williams, and I. Mahon, "Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys," *Journal of Field Robotics*, vol. 27, no. 1, pp. 21–51, 2010.

[7] M. S. T. TON and P. HOWARD, "3D wavefront image formation for niitek gpr," in *Proceedings of SPIE, the International Society for Optical Engineering*.

Society of Photo-Optical Instrumentation Engineers, 2009.

[8] C. Samson, C. English, A. Deslauriers, I. Christie, F. Blais, and F. Ferrie, "Neptec 3D laser camera system : From space mission sts-105 to terrestrial applications," in *2002 ASTRO Conference*, 2004.

[9] U. Soergel, K. Schulz, U. Thoennessen, and U. Stilla, "Integration of 3D data in sar mission planning and image interpretation in urban areas," *Information Fusion*, vol. 6, no. 4, pp. 301–310, 2005.

[10] B. Lange, C.-Y. Chang, E. Suma, B. Newman, A. S. Rizzo, and M. Bolas, "Development and evaluation of low cost game-based balance rehabilitation tool using the microsoft kinect sensor," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2011, pp. 1831–1834.

[11] K. Aitpayev and J. Gaber, "Creation of 3D human avatar using kinect," *Asian Transactions on Fundamentals of Electronics, Communication & Multimedia*, vol. 1, no. 5, pp. 1–3, 2012.

[12] B. Mendiburu, *3D movie making : stereoscopic digital cinema from script to screen*. CRC Press, 2012.

[13] F. Cheng and X. Chen, "Integration of 3D stereo vision measurements in industrial robot applications," in *International Conference on Engineering & Technology*, 2008.

[14] N. Ayache, "Medical computer vision, virtual reality and robotics," *Image and Vision Computing*, vol. 13, no. 4, pp. 295–313, 1995.

[15] C. H. Esteban and F. Schmitt, "Silhouette and stereo fusion for 3D object modeling," *Computer Vision and Image Understanding*, vol. 96, no. 3, pp. 367–392, 2004.

[16] Z. Zhang and O. Faugeras, *3D dynamic scene analysis : a stereo based approach*.

Springer Science & Business Media, 2012, vol. 27.

[17] A. M. Waxman and S. Ullman, "Surface structure and three-dimensional motion from image flow kinematics," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 72–94, 1985.

[18] D. Hoffman, "Inferring shape from motion fields." DTIC Document, Tech. Rep., 1980.

[19] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography : a factorization method," *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, 1992.

[20] R. Szeliski and S. B. Kang, "Recovering 3D shape and motion from image streams using nonlinear least squares," *Journal of Visual Communication and Image Representation*, vol. 5, no. 1, pp. 10–28, 1994.

[21] C. J. Poelman and T. Kanade, "A paraperspective factorization method for shape and motion recovery," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 3, pp. 206–218, 1997.

[22] H. Sekkati and A. Mitiche, "Dense 3D interpretation of image sequences : A variational approach using anisotropic diffusion," in *Image Analysis and Processing, 2003. Proceedings. 12th International Conference on.* IEEE, 2003, pp. 424–429.

[23] M. Irani and S. Peleg, "Motion analysis for image enhancement : Resolution, occlusion, and transparency," *Journal of Visual Communication and Image Representation*, vol. 4, no. 4, pp. 324–335, 1993.

[24] W. Wolf, "Key frame selection by motion analysis," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, vol. 2. IEEE, 1996, pp. 1228–1231.

[25] C. Jauffret and D. Pillon, "Observability in passive target motion analysis," *IEEE*

Transactions on Aerospace and Electronic Systems, vol. 32, no. 4, pp. 1290–1300, 1996.

[26] J. K. Aggarwal and Q. Cai, "Human motion analysis : A review," in *Nonrigid and Articulated Motion Workshop, 1997. Proceedings., IEEE.* IEEE, 1997, pp. 90–102.

[27] J. Y. Wang and E. H. Adelson, "Layered representation for motion analysis," in *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR'93., 1993 IEEE Computer Society Conference on.* IEEE, 1993, pp. 361–366.

[28] R. N. Stauffer, E. Y. Chao, and R. C. Brewster, "Force and motion analysis of the normal, diseased, and prosthetic ankle joint." *Clinical orthopaedics and related research*, vol. 127, pp. 189–196, 1977.

[29] S. Ullman, "Analysis of visual motion by biological and computer systems," *Readings in Computer Vision, chapter Recovering Scene Geometry*, pp. 132–144, 1987.

[30] E. C. Hildreth and C. Koch, "The analysis of visual motion : From computational theory to neuronal mechanisms," *Annual review of neuroscience*, vol. 10, pp. 477–533, 1987.

[31] C. L. Fennema and W. B. Thompson, "Velocity determination in scenes containing several moving objects," *Computer graphics and image processing*, vol. 9, no. 4, pp. 301–315, 1979.

[32] H. v. Helmholtz, "Handbook of physiological optics," vol. 3, 1910 ; 1925 for the English version.

[33] J. J. Gibson, "The perception of the visual world." 1950.

[34] A. Mitiche and J. Aggarwal, *Computer Vision Analysis of Image Motion by Variational Methods.* Springer, 2013.

[35] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis," *Pattern recognition*, vol. 36, no. 3, pp. 585–601, 2003.

[36] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer vision and image understanding*, vol. 104, no. 2, pp. 90–126, 2006.

[37] W. Sunada and S. Dubowsky, "On the dynamic analysis and behavior of industrial robotic manipulators with elastic members," *Journal of Mechanisms, Transmissions, and Automation in Design*, vol. 105, no. 1, pp. 42–51, 1983.

[38] A. Borst and M. Egelhaaf, "Principles of visual motion detection," *Trends in neurosciences*, vol. 12, no. 8, pp. 297–306, 1989.

[39] T. S. Sachs, C. H. Meyer, B. S. Hu, J. Kohli, D. G. Nishimura, and A. Macovski, "Real-time motion detection in spiral mri using navigators," *Magnetic resonance in medicine*, vol. 32, no. 5, pp. 639–645, 1994.

[40] Y. L. Tian and A. Hampapur, "Robust salient motion detection with complex background for real-time video surveillance," in *Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1. Seventh IEEE Workshops on*, vol. 2, Jan 2005, pp. 30–35.

[41] S. M. Smith, "Asset-2 : real-time motion segmentation and shape tracking," in *Computer Vision, 1995. Proceedings., Fifth International Conference on*, Jun 1995, pp. 237–244.

[42] P. Bouthemy and E. Francois, "Motion segmentation and qualitative dynamic scene analysis from an image sequence," *International Journal of Computer Vision*, vol. 10, no. 2, pp. 157–182, 1993.

[43] R. G. Bradski and W. J. Davis, "Motion segmentation and pose recognition with motion history gradients," *Machine Vision and Applications*, vol. 13, no. 3, pp. 174–184, 2002.

[44] D. G. Lowe, "Robust model-based motion tracking through the integration of search and estimation," *International Journal of Computer Vision*, vol. 8, no. 2, pp. 113–122, 1992.

[45] D. Murray and A. Basu, "Motion tracking with an active camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 449–459, May 1994.

[46] T. McInerney and D. Terzopoulos, "A finite element model for 3D shape reconstruction and nonrigid motion tracking," in *Computer Vision, 1993. Proceedings., Fourth International Conference on*, May 1993, pp. 518–523.

[47] L. S. Shapiro, A. Zisserman, and M. Brady, "3D motion recovery via affine epipolar geometry," *International Journal of Computer Vision*, vol. 16, no. 2, pp. 147–182, 1995.

[48] Y. Furukawa and J. Ponce, *Image and Geometry Processing for 3D Cinematography*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2010, ch. Dense 3D Motion Capture from Synchronized Video Streams, pp. 193–211.

[49] D. E. DiFranco, T.-J. Cham, and J. M. Rehg, "Reconstruction of 3D figure motion from 2d correspondences," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, 2001, pp. I–307–I–314 vol.1.

[50] O. D. Faugeras and F. Lustman, "Motion and structure from motion in a piecewise planar environment," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 2, no. 03, pp. 485–508, 1988.

[51] G. Adiv, "Determining three-dimensional motion and structure from optical flow generated by several moving objects," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 7, no. 4, pp. 384–401, 1985.

[52] K. Prazdny, "Motion and structure from optical flow," in *Proceedings of the*

6th international joint conference on Artificial intelligence-Volume 2. Morgan Kaufmann Publishers Inc., 1979, pp. 702–704.

[53] A. Mitiche and H. Sekkati, "Optical flow 3D segmentation and interpretation : A variational method with active curve evolution and level sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1818–1829, Nov. 2006.

[54] H. Sekkati and A. Mitiche, "A variational method for the recovery of dense 3D structure from motion," *Journal of Robotics and Autonomous Systems*, vol. 55, pp. 597–607, 2007.

[55] ——, "Concurrent 3D motion segmentation and 3D interpretation of temporal sequences of monocular images," *IEEE Transactions on Image Processing*, vol. 15, no. 3, pp. 641–653, Mar. 2006.

[56] ——, "Joint optical flow estimation, segmentation, and 3D interpretation with level sets." *Computer Vision and Image Understanding*, vol. 103, no. 2, pp. 89–100, 2006.

[57] S. Srinivasan, "Extracting structure from optical flow using the fast error search technique," *International Journal of Computer Vision*, vol. 37, no. 3, pp. 203–230, 2000.

[58] J. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images : a review," DTIC Document, Tech. Rep., 1988.

[59] S. D. Blostein, L. Zhao, and R. M. Chann, "Three-dimensional trajectory estimation from image position and velocity," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 36, no. 4, pp. 1075–1089, 2000.

[60] U. R. Dhond and J. K. Aggarwal, "Structure from stereo-a review," *IEEE transactions on systems, man, and cybernetics*, vol. 19, no. 6, pp. 1489–1510, 1989.

[61] O. Faugeras, *Three-dimensional computer vision : a geometric viewpoint*. MIT

press, 1993.

[62] M. Han and T. Kanade, "Reconstruction of a scene with multiple linearly moving objects," *International Journal of Computer Vision*, vol. 59, no. 3, pp. 285–300, 2004.

[63] T. S. Huang and A. N. Netravali, "Motion and structure from feature correspondences : A review," *Proceedings of the IEEE*, vol. 82, no. 2, pp. 252–268, 1994.

[64] G. Aubert, R. Deriche, and P. Kornprobst, "Computing optical flow via variational techniques," *SIAM Journal on Applied Mathematics*, vol. 60, pp. 156–182, 1999.

[65] H. C. Longuet-Higgins and K. Prazdny, "The interpretation of a moving retinal image," *Royal Society of London Proceedings Series B*, vol. 208, pp. 385–397, 1980.

[66] A. Mitiche and S. Hadjres, "MDL estimation of a dense map of relative depth and 3D motion from a temporal sequence of images." *Pattern Anal. Appl.*, vol. 6, no. 1, pp. 78–87, 2003.

[67] A. Mitiche, *Computational analysis of visual motion.* Springer Science & Business Media, 2013.

[68] R. Chellappa and A. A. Sawchuk, "Digital image processing and analysis," *Digital image processing and analysis, by Chellappa, Rama.; Sawchuk, Alexander A. Silver Spring, MD : IEEE Computer Society Press; Los Angeles, CA : Order from IEEE Computer Society, c1985.*, vol. 1, 1985.

[69] J. W. Roach, "Determining the movement of objects from a sequence of images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 6, pp. 554–562, 1980.

[70] R. Y. Tsai and T. S. Huang, "Uniqueness and estimation of three-dimensional

motion parameters of rigid objects with curved surfaces," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 1, pp. 13–27, 1984.

[71] B. K. P. Horn and B. G. Schunck, "Determining optical flow." *Artif. Intell.*, vol. 17, no. 1-3, pp. 185–203, 1981.

[72] S. Osher and N. Paragios, *Geometric level set methods in imaging, vision, and graphics*. Springer Science & Business Media, 2003.

[73] G. Aubert and P. Kornprobst, *Mathematical problems in image processing : partial differential equations and the calculus of variations*. Springer Science & Business Media, 2006, vol. 147.

[74] A. Mitiche and I. B. Ayed, *Variational and level set methods in image segmentation*. Springer Science & Business Media, 2010, vol. 5.

[75] W. J. MacLean, A. D. Jepson, and R. C. Frecker, "Recovery of egomotion and segmentation of independent object motion using the em algorithm." in *British Machine Vision Conference, BMVC*, E. R. Hancock, Ed. BMVA Press, 1994, pp. 1–10.

[76] J. Weber and J. Malik, "Rigid body segmentation and shape description from dense optical flow under weak perspective," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 139–143, 1997.

[77] S. Fejes and L. S. Davis, "What can projections of flow fields tell us about visual motion." in *ICCV*, 1998, pp. 979–986.

[78] B. K. P. Horn and E. J. Weldon, "Direct methods for recovering motion," *International Journal of Computer Vision*, vol. 2, no. 1, pp. 51–76, 1988.

[79] A. R. Bruss and B. K. P. Horn, "Passive navigation," *Computer Vision, Graphics, and Image Processing*, vol. 21, no. 1, pp. 3–20, 1983.

[80] B. Shahraray and M. Brown, "Robust depth estimation from optical flow," in *International Conference on Computer Vision, ICCV*, 1988, pp. 641–650.

[81] D. J. Heeger and A. D. Jepson, "Subspace methods for recovering rigid motion I : Algorithm and implementation." *International Journal of Computer Vision*, vol. 7, no. 2, pp. 95–117, 1992.

[82] H. Liu, R. Chellappa, and A. Rosenfeld, "A hierarchical approach for obtaining structure from two-frame optical flow," in *Proceedings of the Workshop on Motion and Video Computing*, ser. MOTION '02. Washington, DC, USA : IEEE Computer Society, 2002, pp. 214–219.

[83] M. A. Taalebinezhaad, "Direct recovery of motion and shape in the general case by fixation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 8, pp. 847–853, aug 1992.

[84] E. De Micheli and F. Giachero, "Motion and structure from one dimensional optical flow," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, Jun 1994, pp. 962–965.

[85] N. Gupta and N. Kanal, "3-D motion estimation from motion field," *Artificial Intelligence,*, vol. 78, pp. 45–86, 1995.

[86] Y. Xiong and S. Shafer, "Dense structure from a dense optical flow," Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-95-10, April 1995.

[87] T. Brodsky, C. Fermuller, and Y. Aloimonos, "Structure from motion : Beyond the epipolar constraint." *International Journal of Computer Vision*, vol. 37, no. 3, pp. 231–258, 2000.

[88] S. Srinivasan, "Extracting structure from optical flow using the fast error search technique," *Int. J. Comput. Vision*, vol. 37, no. 3, pp. 203–230, Jun. 2000.

[89] H. Sekkati and A. Mitiche, "Concurrent 3-D motion segmentation and 3-D interpretation of temporal sequences of monocular images," *IEEE Trans. on Image Processing*, vol. 15, no. 3, pp. 641–653, 2006.

[90] A. Mitiche and J. M. Letang, "Stereokinematic analysis of visual data in active, convergent stereoscopy," *Journal of Robotics and Autonomous Systems*, vol. 705, pp. 43–71, 1998.

[91] A. Mitiche, "On combining stereopsis and kineopsis for space perception," in *First International Conference on Artificial Intelligence Applications*. Denver, CO, 1984, pp. 156–160.

[92] ——, "A computational approach to the fusion of stereopsis and kineopsis," in *Motion Understanding : Robot and Human Vision*, W. N. Martin and e. J. K. Aggarwal, Eds. Kluwer Academic, 1988, pp. 81–99.

[93] J. J. Gibson, "Optical motions and transformations as stimuli for visual perception." *Psychological Review*, vol. 64, no. 5, p. 288, 1957.

[94] H. Wallach and D. O'connell, "The kinetic depth effect." *Journal of experimental psychology*, vol. 45, no. 4, p. 205, 1953.

[95] J. Aloimonos and C. M. Brown, "Direct processing of curvilinear sensor motion from a sequence of perspective images," in *Proc. Workshop on Computer Vision : Representation and Control*, vol. 72, 1984, p. 77.

[96] S. Negahdaripour and B. K. Horn, "Direct passive navigation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 1, pp. 168–176, 1987.

[97] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Readings in Computer Vision : Issues, Problems, Principles, and Paradigms, MA Fischler and O. Firschein, eds*, pp. 61–62, 1987.

[98] X. Zhuang and R. Haralick, "Rigid body motion and the optical flow image," in *The First Conf. on Artificial Intelligence Appl., Denver*, 1986.

[99] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade, "Three-dimensional scene flow," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 475–480, 2005.

[100] J.-P. Pons, R. Keriven, O. Faugeras, and G. Hermosillo, "Variational stereovision and 3D scene flow estimation with statistical similarity measures," in *IEEE International Conference On Computer Vision (ICCV)*, 2003, pp. 597–602.

[101] A. Mitiche, Y. Mathlouthi, and I. Ben Ayed, "Monocular concurrent recovery of structure and motion scene flow," *Front. ICT 2 : 16. doi : 10.3389/fict*, 2015.

[102] Y. Zhang and C. Kambhamettu, "Integrated 3D scene flow and structure recovery from multiview image sequences," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2000, pp. 674–681.

[103] Y. G. Leclerc, "Constructing simple stable descriptions for image partitioning," *International journal of computer vision*, vol. 3, no. 1, pp. 73–102, 1989.

[104] L. A. Vese and C. Le Guyader, *Variational methods in image processing*. CRC Press, 2015.

[105] R. A. Newcombe and A. J. Davison, "Live dense reconstruction with a single moving camera," in *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 1498–1505.

[106] R. Che, X. Xu, R. Nian, B. He, M. Chen, C. Zhang, and A. Lendasse, "Underwater non-rigid 3D shape reconstruction via structure from motion for fish ethology," in *IEEE OCEANS 2016 MTS*, 2016, pp. 1–5.

[107] R. Bowden, T.A. Mitchell, and M. Sarhadi, "Non-linear statistical models for the 3D reconstruction of human pose and motion from monocular image sequences," *Image and Vision Computing*, vol. 18, no. 9, pp. 729–737, 2000.

[108] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, and D. Cremers, "Stereoscopic scene flow computation for 3D motion understanding," *International Journal of Computer Vision*, vol. 95, no. 1, pp. 29–51, 2011.

[109] C. Rabe, T. Müller, A. Wedel, and U. Franke, "Dense, Robust, and Accurate Motion Field Estimation from Stereo Image Sequences in Real-Time," in *Pro-*

ceedings of the 11th European Conference on Computer Vision (ECCV), ser. Lecture Notes in Computer Science, K. Daniilidis, P. Maragos, and N. Paragios, Eds., vol. 4. Springer, September 2010, pp. 582–595.

[110] F. Huguet and F. Devernay, "A variational method for scene flow estimation from stereo sequences," in *IEEE International Conference on Computer Vision (ICCV)*, 2007, pp. 1–7.

[111] A. Tamtaoui and C. Labit, "Constrained disparity and motion estimation for 3Dtv image sequence coding," *Signal Processing : Image Communication*, vol. 199, no. 4, pp. 45–54, 1991.

[112] Y. Altunbasak, A. M. Tekalp, and G. Bozdagi, "Simultaneous motion-disparity estimation and segmentation from stereo." in *ICIP (3)*. IEEE, 1994, pp. 73–77.

[113] I. Patras, N. Alvertos, and G. Tziritas, "Joint disparity and motion field estimation in stereoscopic image sequences," in *IAPR International Conference on Pattern Recognition*. Vienna, Austria, 1996, pp. 359–362.

[114] H. Weiler, A. Mitiche, and A. Mansouri, "Bounday preserving joint estimation of optical flow and disparity in a sequence of stereoscopic images," in *IASTED International Conference on Visualization, Imaging, and Image Processing*, 2003, pp. 102–106.

[115] A. Wedel, C. Rabe, T. Vaudrey, T. Brox, U. Franke, and D. Cremers, "Efficient dense scene flow from sparse or dense stereo data," in *European Conference on Computer Vision (ECCV)*, vol. 1, 2008, pp. 739–751.

[116] T. Basha, Y. Moses, and N. Kiryati, "Multi-view scene flow estimation : A view centered variational approach." *International Journal of Computer Vision*, vol. 101, no. 1, pp. 6–21, 2013.

[117] C. Vogel, K. Schindler, and S. Roth, "Piecewise rigid scene flow," in *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia,*

December 1-8, 2013, 2013, pp. 1377–1384.

[118] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31, 2011.

[119] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2432–2439.

[120] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *Int. J. Comput. Vision*, vol. 92, no. 1, pp. 1–31, mar 2011.

[121] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *European Conference on Computer Vision (ECCV)*, ser. Lecture Notes in Computer Science, vol. 3024. Springer, May 2004, pp. 25–36.

[122] J.-P. Pons, R. Keriven, and O. D. Faugeras, "Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score," *International Journal of Computer Vision*, vol. 72, no. 2, pp. 179–193, 2007.

[123] P. Ciarlet, *Introduction à l'analyse numérique matricielle et à l'optimisation*, ser. Collection Mathématiques appliquées pour la maîtrise. Masson, 1982.

[124] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, 3rd ed., ser. Texts in Applied Mathematics ; 12. New York : Springer, 2002.

[125] M. P. Heinrich, M. Jenkinson, M. Bhushan, T. Matin, F. V. Gleeson, M. Brady, and J. A. Schnabel, "Mind : Modality independent neighbourhood descriptor for multi-modal deformable registration," *Medical Image Analysis*, vol. 16, no. 7, pp. 1423–1435, 2012.

[126] S. Periaswamy and H. Farid, "Medical image registration with partial data,"

Medical image analysis, vol. 10, no. 3, pp. 452–464, 2006.

[127] M. Tappen, W. Freeman, and E. Adelson, "Recovering intrinsic images from a single image," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 9, pp. 1459–1472, Sept 2005.

[128] P. J. Besl and R. Jain, "Invariant surface characteristics for 3D object recognition in range images." *Computer Vision, Graphics, and Image Processing*, vol. 33, no. 1, pp. 33–80, 1986.

[129] D. Terzopoulos, "Regularization of inverse visual problems involving discontinuities," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, no. 4, pp. 413–424, July 1986.

[130] J. Crank , *The mathematics of diffusion.* Clarendon press Oxford, 1975, vol. 2, no. 3.

[131] R. Chartrand, "Numerical differentiation of noisy, nonsmooth data." in *Los Alamos National Laboratory TR*, 2005.

[132] J. Cullum, "Numerical differentiation and regularization," *SIAM Journal on Numerical Analysis*, vol. 8, no. 2, pp. 254–265, 1971.

[133] G. E. Forsythe, M. A. Malcolm, and C. B. Moler, *Computer methods for mathematical computations*, ser. Prentice-Hall series in automatic computation. Englewood Cliffs (N.J.) : Prentice-Hall, 1977.

[134] P. G. Ciarlet, *Introduction a l'analyse numerique matricielle et a l'optimisation.* Masson, 1988.

[135] C. R. Vogel, *Computational methods for inverse problems.* SIAM Frontiers in Applied Mathematics, 2002.

[136] A. Mitiche and J. Aggarwal, *Computer vision analysis of image motion by variational methods.* Springer, 2013.

[137] G. Aubert, R. Deriche, and P. Kornprobst, "Computing optical flow via variational thechniques," *SIAM Journal of Applied Mathematics*, vol. 60, no. 1, pp. 156–182, 1999.

[138] G. Aubert and P. Kornprobst, *Mathematical Problems in Image Processing : Partial Differential Equations and the Calculus of Variations*, ser. Applied mathematical sciences. Springer, 2001, no. vol. 147.

DOI : 10.1007/978-3-319-14249-4_11

Chapitre 2

Direct Estimation of Dense Scene Flow and Depth from a Monocular Sequence

Yosra Mathlouthi, Amar Mitiche, and Ismail Ben Ayed

Advances in Visual Computing, LNCS 8887, pp. 107-117, 2014, Springer.

Résumé : Nous proposons une méthode qui utilise une séquence monoculaire pour l'estimation directe et conjointe du flot de scène dense et de la profondeur relative. C'est un problème qui a été généralement abordé dans la littérature avec des séquences d'images binoculaires ou stéréoscopiques. Le problème est posé sous une forme variationnelle où on optimise une fonctionnelle à deux termes : un terme de données, qui corrèle la vitesse 3D à la profondeur en termes des variations spatiotemporelles, et un terme de régularisation L^2. Basée sur l'expression de la contrainte du gradient du flot optique en termes de la vitesse du flot de scène et de la profondeur, notre formulation est analogue à l'estimation classique du flot optique par l'algorithme de Horn et Schunck, bien qu'elle implique le mouvement 3D et

Chapitre 3

Regularized differentiation for image derivatives

Yosra Mathlouthi, Amar Mitiche, and Ismail Ben Ayed

Accepté à *IET Image processing,* 2016.

Résumé : Dans cette étude, on analyse une méthode de différentiation régularisée pour l'estimation des dérivées d'une image. La formulation minimise une fonctionnelle intégrale qui contient un terme d'adéquation aux données sous forme d'*anti-différentiation* et un terme de lissage sous forme de régularisation. Une fois discrétisées, les conditions d'Euler-Lagrange nécessaires pour minimiser la fonctionnelle objectif produisent un système d'équations linéaire creux et à grande échelle. Ce système peut être résolu d'une manière efficace par les itérations de Jacobi ou de Gauss-Seidel. Nous étudions l'impact de la méthode dans le cadre de deux problèmes importants de vision par ordinateur : l'estimation du flot optique et l'estimation du flot de scène. Les résultats quantitatifs, provenant de l'utilisation des séquences d'images de la base de données *Middlebury* ainsi que d'autres séquences d'images réelles et synthétiques, montrent que notre algorithme

de différentiation régularisée est plus performant que les définitions standards des dérivées par les moyennes de différences finies généralement utilisées dans l'analyse du mouvement. La méthode peut être facilement appliquée dans plusieurs autres problèmes de traitement d'images.

Abstract

This study investigates a regularized differentiation method to estimate image derivatives. The scheme minimizes an integral functional containing an *anti-differentiation* data discrepancy term and a smoothness regularization term. When discretized, the Euler-Lagrange necessary conditions for a minimum of the functional yield a large scale sparse system of linear equations, which can be solved efficiently by Jacobi/Gauss-Seidel iterations. We investigate the impact of the method in the context of two important problems in computer vision : optical flow and scene flow estimation. Quantitative results, using the *Middlebury* dataset and other real and synthetic images, show that our regularized differentiation scheme outperforms standard derivative definitions by smoothed finite differences, which are commonly used in motion analysis. The method can be readily used in various other image analysis problems.

3.1 Introduction

Image derivatives occur in a variety of problems in image processing and computer vision, such as motion estimation and detection [1–4], image registration [5, 6], recovery of intrinsic images from a single image [7], and 3D object recognition [8], among others. Approximation by finite differences of a function derivative is an ill-posed problem as small perturbations of the function values can cause significant changes in the derivative [9, 10]. Therefore, noise in an image can seriously affect the quality

of subsequent image analysis interpretations that use image derivatives. Prior image denoising by filtering does not address the limitations of finite-difference approximations and, therefore, would be a futile, ineffective operation ; see the examples in [10] for the case of a 1D noisy function of a real variable.

To lessen the impact of noise in an image when computing its derivatives, computer vision algorithms have often used locally smoothed finite differences [2–7]. In motion analysis, for instance, most studies use the following formulas, or variants, given by Horn and Shunck for optical flow estimation [3] :

$$\begin{aligned}
I_x(r,c) \approx \tfrac{1}{4}\textstyle\sum_{\Delta r=0}^{1} \quad & \{\ I_0(r+\Delta r, c+1) - I_0(r+\Delta r, c) \\
& +I_1(r+\Delta r, c+1) - I_1(r+\Delta r, c)\ \} \\
I_y(r,c) \approx \tfrac{1}{4}\textstyle\sum_{\Delta c=0}^{1} \quad & \{\ I_0(r+1, c+\Delta c) - I_0(r, c+\Delta c) \\
& +I_1(r+1, c+\Delta c) - I_1(r, c+\Delta c)\ \} \\
I_t(r,c) \approx \tfrac{1}{4}\textstyle\sum_{\Delta r=0}^{1}\sum_{\Delta c=0}^{1} \quad & \{\ I_1(r+\Delta r, c+\Delta c) - I_0(r+\Delta r, c+\Delta c)\ \}, \quad (3.1)
\end{aligned}$$

where I_x, I_y, I_t are the spatiotemporal derivatives of input image I ; $\Delta r, \Delta c$ are displacements with respect to row r and column c, respectively ; I_0 and I_1 are the current and next frames of the input image sequence.

Although used quite often in motion estimation [1], this type of approximation by local smoothing of finite differences is not fitting to general image noise profiles. A more principled smoothing alternative would allow regularization, for instance via minimizing the following functional w.r.t f [9] :

$$\frac{1}{2}\int_{\Omega} \left((f-f_0)^2 + \alpha\|\nabla f\|^2\right) dxdy, \qquad (3.2)$$

where f_0 is some finite-difference approximation of the image derivative, Ω is the image domain, and α is a positive coefficient to weigh the contribution of the smoothness term. This variational formulation improves on the derivative definition but still relies

entirely on finite-difference approximations and, therefore, would not befit general image noise structures.

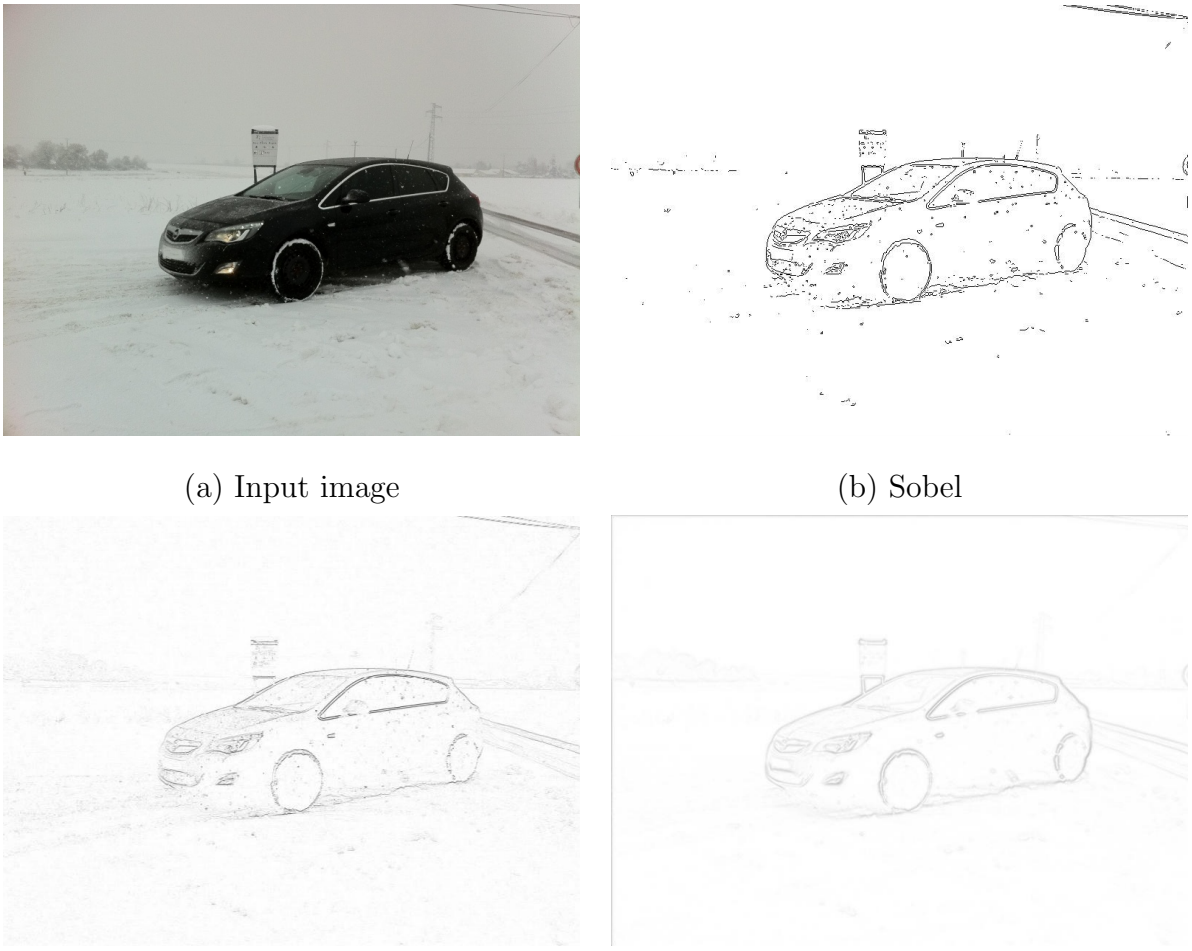
Several studies in computational mathematics have investigated variational formulations where the derivative of a 1D function appears explicitly as the variable to determine, given the function as data and subject to some regularity constraints [10,11]. They typically minimize a sum of terms related to data fidelity and regularity. For instance, in [10], the derivative of a given function is an unknown function, to be determined, which, when integrated, gives back the given function. This *anti-differentiation* definition of the derivative is then used in a classic functional of two terms to minimize. The first is a data fidelity term that penalizes the discrepancy between the observed data and an anti-differentiation operator of the sought derivative function. The second term is a regularization bias toward smooth solutions, e.g., a Tikhonov (L_2) or a total variation (L_1) regularization. The discrete implementation of the ensuing formulation follows a standard numerical scheme [12]. The experiments of [10] in the case of 1D noisy functions show much better performances of regularized differentiation than finite-difference schemes and prior/posterior denoising of input functions ; see Figs. 2 and 3 in [10]. The study in [11] addressed regularized differentiation of 1D functions from a different viewpoint. The problem was to find a smooth approximation of the true derivative y' of a function y from the given inaccurate data $\tilde{y}_i$. This was done by determining an approximation f of y which minimizes an objective functional having a term of discrepancy between f and the given data, and a regularization term to penalize the L_2 norm of the second derivative f''. This regularization brings the minimizer to be a cubic spline, and the derivative was subsequently evaluated on f.

Image derivatives are commonly used in 2D/3D motion analysis. Therefore, the properties of regularized differentiation make it potentially useful since it (a) removes the need for *ad hoc* prior/posterior denoising, (b) avoids noise amplification of common finite-difference methods and (c) controls in a principled way the regularity of

the derivatives. To the best of our knowledge, regularized differentiation has not been investigated in 2D/3D image applications and problems, and related numerical differentiation studies have been stated for real functions of a real variable [10, 13–16].

Fig. 3.1 illustrates the effect of regularization on edge detection, a common visual processing task in which image differentiation is the main component. The example uses a smartphone photograph acquired in low-lighting conditions, and depicts a car in a snowy environment. Fig. 3.1 (b) shows edge detection with the standard Sobel filter [22], which computes image gradient with a locally smoothed finite difference. Fig. 3.1 (c) depicts the edges obtained with the finite difference averaging in [3], and (d) shows the result with our regularized differentiation. The latter removed several spurious and isolated edges, and yielded a solution that reflects better the main object in the image.

In this paper, we formulate image derivative estimation by regularized anti-differentiation, and investigate it in the context of both optical flow (2D) and scene flow (3D) estimation. The scheme computes image spatial derivatives by minimizing an objective functional of two terms : an anti-differentiation data discrepancy term and an L_2 regularization penalty. The data term constrains the derivative function to be close to the image when integrated, and the regularization biases it to be spatially smooth. The Euler-Lagrange equations give the necessary conditions for a minimum of the functional, and the corresponding discretization yields a large-scale sparse system of linear equations, which can be solved efficiently by iterative methods such as Jacobi or Gauss-Seidel. We used the *Middlebury* dataset and other real and synthetic image sequences to evaluate the scheme quantitatively. The results show that regularized differentiation outperforms standard finite difference definitions commonly used in motion analysis. These results justify further investigations to generalize the formulation to preserve motion boundaries. They also justify further investigations of its use in other problems of image analysis, e.g., image registration.



(a) Input image (b) Sobel

(c) Finite-difference smoothing (d) Regularized differentiation

FIGURE 3.1 – An example illustrating the effect of regularizing image differentiation : (a) *Input image. Edge detection by gradient magnitudes using : (b) Sobel filter [22], (c) Standard finite difference smoothing [3] and, (d) Our regularized differentiation* ($\beta = 1$).

3.2 Regularized image differentiation

Let $I : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ be an image function, with n denoting the image dimension. In the following, we will present the case $n = 2$, but our framework extends directly to an arbitrary dimension n. Let I_x and I_y denote the spatial partial derivatives of I in the 2D case. We follow a variational statement of the problem by minimizing a functional containing two terms : (i) an anti-differentiation data discrepancy term and (ii) an L_2 smoothness regularization term. In the following, we detail the computation of partial derivative I_x. Computation of I_y uses the same formulas applied to the image transpose.

We minimize the following functional with respect to a function f, which represents the image spatial derivative we want to determine :

$$\frac{1}{2}\int_{\Omega}\left(\|Af - I\|^2 + \beta\|\nabla f\|^2\right)dxdy, \tag{3.3}$$

where A is the operator of anti-differentiation w.r.t x :

$$Af(x,y) = \int_0^x f(z,y)dz. \tag{3.4}$$

β is a positive constant balancing the contribution of the regularization term, and $\nabla f = (f_x, f_y)$ is the spatial gradient of f.

We obtain the necessary conditions for a minimum of (3.3) by solving the corresponding Euler-Lagrange equations, which we recall in the following for the general form of an integral involving scalar functions of two independent variables [1] :

Let ξ be a functional of the following general form :

$$\xi(w) = \int_B g(x,y,w,w_x,w_y)dxdy, \tag{3.5}$$

where g is a function twice differentiable with respect to its arguments, B is a bounded domain of $\mathbb{R}^2$, $w(x,y)$ is a twice differentiable real function, and w_x and w_y are the

partial derivatives of w. The Euler-Lagrange equation corresponding to functional (3.5) is

$$\frac{\partial g}{\partial w} - \frac{\partial}{\partial x}\left(\frac{\partial g}{\partial w_x}\right) - \frac{\partial}{\partial y}\left(\frac{\partial g}{\partial w_y}\right) = 0. \tag{3.6}$$

Applying (3.6) to $g = \|Af - I\|^2 + \beta\|\nabla f\|^2$ and $w = f$, and after some manipulations, we obtain the Euler-Lagrange equation corresponding to (3.3) :

$$A^*(Af - I) - \beta\nabla^2 f = 0, \tag{3.7}$$

with A^* the adjoint operator of A :

$$A^* f(x, y) = \int_x^l f(z, y)dz. \tag{3.8}$$

The discretization of (3.7) yields a large-scale sparse system of linear equations. We discretize the image domain according to the points of a grid D listed in a one-dimensional array from top to down and left to right. Therefore, the discrete image is a vector of size $N = r \times c$ for an image of size $r \times c$. For $i = 1, ..., N$, let f_i be f evaluated at grid point i and $\mathbf{f} \in \mathbb{R}^N$ the corresponding vector of size N. For notational convenience, symbols of linear operators A and A^* in (3.7) will be reused to denote the corresponding discretization matrices. The definition of the $(N \times N)$ matrix A according to the approximation of the composite trapezoid quadrature rule for integrals, with one-pixel data spacing [17], is as follows :

$$\begin{aligned}
&A(k_r r + i_r, k_c c + 1) = \frac{1}{2},\\
&i_r = 2, ..., r;\ k_r = 0, ..., c - 1;\ k_c = 0, ..., r - 1;\\
&A(k_r r + i_r, k_c c + i_c) = \frac{1}{2},\\
&i_r = 2, ..., r;\ i_c = 2, ..., c;\ k_r = 0, ..., c - 1;\ k_c = 0, ..., r - 1;\\
&A(k_r r + i_r, k_c c + i_c - j_c) = 1,\\
&i_r = 3, ..., r;\ i_c = 3, ..., c;\ j_c = 1, ..., i_c - 2;\ k_r = 0, ..., c - 1;\ k_c = 0, ..., r - 1
\end{aligned}$$

To simplify matrix A and the remainder of the presentation, let us assume that we have a squared image where $r = c = n$. In this case, A becomes :

$$A(kn+i, kn+1) = \tfrac{1}{2},$$
$$i = 2, ..., n;\ k = 0, ..., n-1;$$
$$A(kn+i, kn+i) = \tfrac{1}{2},$$
$$i = 2, ..., n;\ k = 0, ..., n-1;$$
$$A(kn+i, kn+i-j) = 1,$$
$$i = 3, ..., n;\ j = 1, ..., i-2;\ k = 0, ..., n-1.$$

The matrix elements that we omitted in the equations above are all equal to zero. The $kn+1$ row elements (where $k = 0, ..., n-1$) are also equal to zero. These last constraints are defined in order to reflect the integral of (3.4) at the boundary when x is equal to 0. It can be seen that matrix A is block diagonal sparse, where blocks are of size $n \times n$. Similarly, the definition of the $N \times N$ matrix A^* is given by

$$A^*(i, i) = \tfrac{1}{2},$$
$$i \in [1, n^2],\ i \neq kn,\ k = 1, ..., n;$$
$$A^*(kn+i, (k+1)n) = \tfrac{1}{2},$$
$$i = 1, ..., n-1;\ \ k = 0, ..., n-1;$$
$$A^*(kn+i, kn+i+j) = 1,$$
$$i = 1, ..., n-1;\ \ j = 1, ..., n-i-1;\ \ k = 0, ..., n-1.$$

Fig. 3.2 illustrates anti-differentiation matrix A and its adjoint operator A^* for the case $n = 5$.

The discretization of the Laplacian term in (3.7) can be done as $\beta \sum_{j \in \mathcal{N}_i} (f_j - f_i)$, where the constant factor of the approximation is absorbed by parameter β, and $\mathcal{N}_i$ is the set of indices of the neighbors of i. Denoting $n_i = card(\mathcal{N}_i)$, we write the matrix corresponding to this term as follows :

$$L(i, i) = -\beta n_i; \quad i = 1, ..., N$$
$$L(i, j) = \beta; \quad j \in \mathcal{N}_i.$$

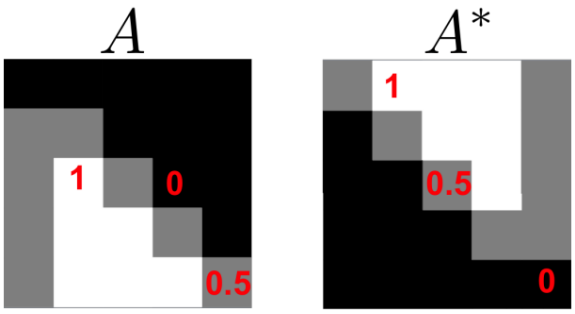


FIGURE 3.2 – *Illustration of anti-differentiation matrix A and its adjoint operator A^* for the case $n = 5$.*

Finally, the system of linear equations to solve is :

$$(A^*A - L)\,\mathbf{f} = A^*I. \tag{3.9}$$

This is a large scale sparse system of linear equations, which can be solved efficiently by iterative methods such as Jacobi or Gauss-Seidel. We choose the Gauss-Seidel method, which yields a better convergence speed compared to Jacobian iterations [18].

3.2.1 Tests on a synthetic example

This example uses the synthetic image depicted in Fig. 3.3 (third row, leftmost image) and a noised version computed by adding a white Gaussian noise, with a signal-to-noise ratio of 0.5db (first row, leftmost image). The image has a pyramidal shape, and the ground-truth derivative is readily calculated. We applied regularized differentiation and the local finite-difference smoothing in (3.1) to each image. Fig. 3.3 depicts the results, which show that, in the noisy case, regularized differentiation computes values closer to the actual derivatives. Table 3.1 lists the mean squared errors between the estimated partial derivative values and the ground truth. Each error is computed as the mean of all the errors over all components of gradients. In the noiseless case, locally averaged finite differences and regularized differentiation yield

similar results, as one would expect. With the noised image, however, regularized differentiation performs much better. The last two lines of Table 3.1 report the results of applying both gradient smoothing (after finite differences) and basic image smoothing (before finite differences), using a Wiener filter based on pixel-neighbourhood statistics. The corresponding results are included in Fig. 3.3. These pre- or post-processing operations improved the results of finite differences, but their errors are still much higher than our regularized differentiation. The smoothness parameter β was determined empirically, and the initial values of the partial derivatives were all set to zero.

TABLE 3.1 – Mean squared errors between the computed and actual values of the partial derivatives

Method	finite differences	reg. differentiation
Pyramid	0.0052	0.0161 ($\beta = 0.1$)
Noised Pyramid (SNR=0.5)	1.0868	0.0409 ($\beta = 5$)
	0.5371 (image smoothing)	-
	0.5231 (gradient smoothing)	-

3.2.2 Optical flow estimation

Optical flow estimation requires the evaluation of image derivatives and the question naturally arises as to how to estimate these from the image data. The purpose in this section is to show the advantage of using regularized differentiation rather than the common locally smoothed finite differences in Eq. (3.1). Of course, our intent here is not to design an optical flow estimation algorithm to beat the state-of-the-art [19]. Rather, we use the benchmark algorithm of Horn and Schunck [1–3] and focus on the impact that the necessary image derivatives estimation can have

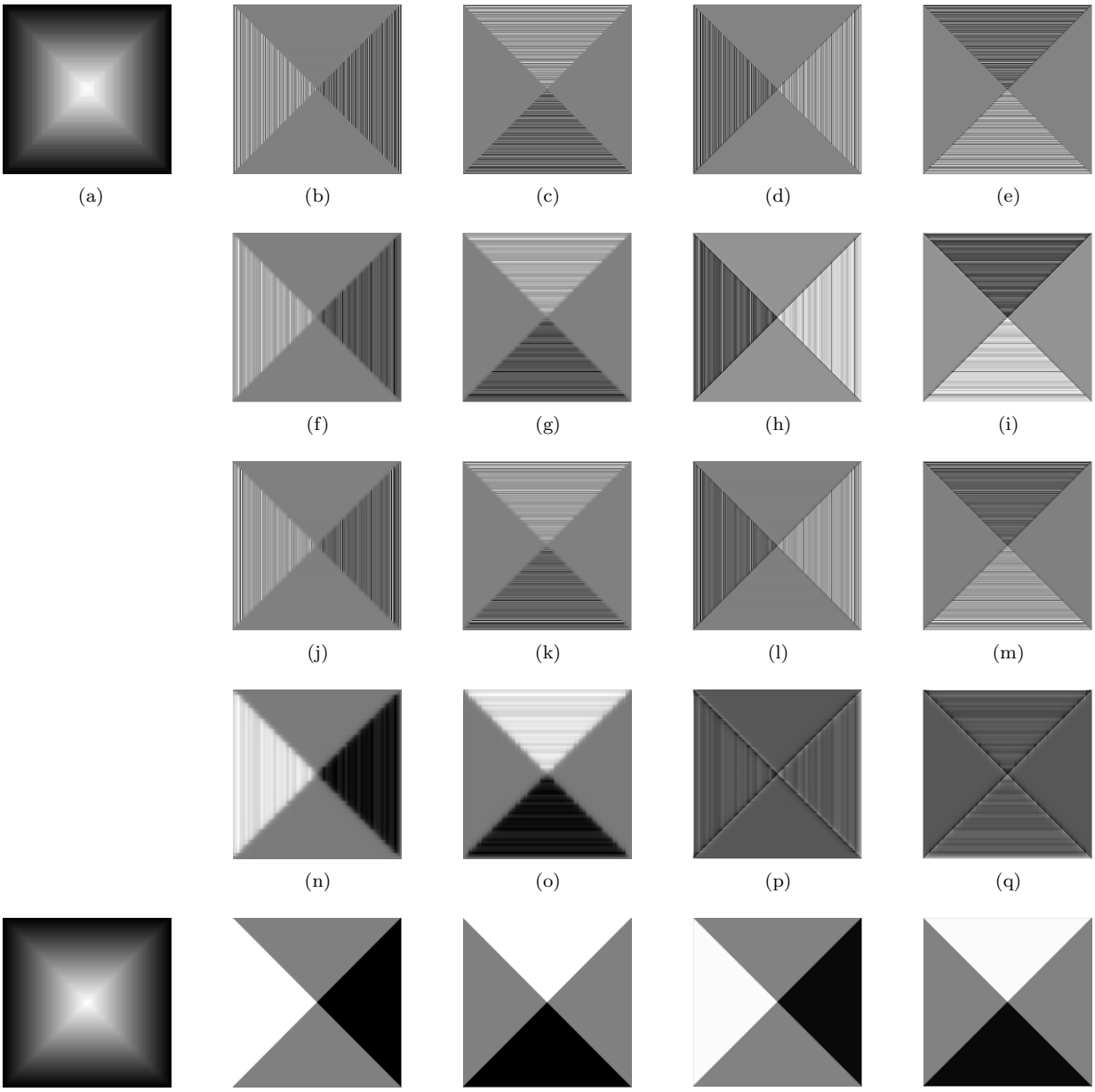


FIGURE 3.3 – *Synthetic examples. First row (the noised version), from left to right :* ***(a)*** *The 2D pyramid image ;* ***(b, c)*** *Partial derivatives I_x and I_y using locally averaged finite differences ;* ***(d, e)*** *The difference between the partial derivatives in (b,c) and the ground truth. Second row (the noised version), from left to right :* ***(f, g)*** *Partial derivatives I_x and I_y using locally averaged finite differences applied to a smoothed version of the input image (Wiener filter) ;* ***(h, i)*** *The difference between the partial derivatives in (f,g) and the ground truth. Third row (the noised version), from left to right :* ***(j, k)*** *Partial derivatives (I_x and I_y) smoothed by Wiener filter ;* ***(l, m)*** *Difference between the partial derivatives in (j, k) and the ground truth. Fourth row (the noised version), from left to right :* ***(n, o)*** *Partial derivatives using regularized differentiation ($\beta = 0.1$) ;* ***(p, q)*** *Difference between the partial derivatives in (n, o) and the ground truth. Fifth row (the case without noise), from left to right :* ***(r)*** *The 2D pyramid image ;* ***(s, t)*** *Partial derivatives using locally averaged finite differences ;* ***(v, w)*** *Partial derivatives using regularized differentiation ($\beta = 5$).*

on the accuracy of the optical flow estimates. For this, given an image sequence $I : (x, y, t) \in \Omega \times \mathbb{R}^+ \rightarrow I(x, y, t) \in \mathbb{R}$, where (x, y) are the spatial coordinates defined on a bounded image domain $\Omega \subset \mathbb{R}^2$ and $t \in \mathbb{R}^+$ is the time coordinate, Horn and Schunck estimate optical flow by minimizing the following functional w.r.t optical flow coordinates (u, v) :

$$\frac{1}{2}\int_{\Omega}(I_x u + I_y v + I_t)^2 dxdy + \frac{\lambda}{2}\int_{\Omega}(\|\nabla u\|^2 + \|\nabla v\|^2)dxdy, \tag{3.10}$$

where I_x, I_y, and I_t are the image spatio-temporal derivatives, and λ is a positive constant that balances the relative contribution of the two terms of the functional. We will not detail this classic functional and its minimization because it has been discussed in numerous studies [1].

With the Horn-Schunck algorithm, we computed optical flow when I_x and I_y are evaluated using regularized differentiation, and also when they are evaluated with the finite-difference smoothing in (3.1), and then compared the results. We did not include I_t in consideration simply because the time axis is sampled only at two points : recall that we are to estimate the derivatives when given two consecutive images. Instead, we computed I_t by finite differences.

We evaluated the results on the *Middlebury* database [19], using the image sequences for which the ground-truth flows are publicly available [1]. Recall that these sequences include (1) real scenes with nonrigid motion, where the dense ground-truth flow is calculated by tracking hidden fluorescent texture, (2) complex realistic synthetic sequences using independent and large motion ranges, realistic texture, and complex occlusions, and (3) modified stereo sequences of real static scenes. We used two standard error measures to evaluate optical flow [19] : average angular error (aae) and endpoint error (epe). Fig. 3.5 plots the means and standard deviations of aae (top) and epe (bottom) over all sequences in the dataset. We further experimen-

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

ted with different values of the weight coefficient β in the regularized differentiation functional. The curves show systematically smaller errors with regularized differentiation (blue curves) than with finite-difference smoothing (red curves), for most of the values of β. These results confirm the practical benefit of regularized differentiation. Fig. 3.4 illustrates this point with a typical example (the *Rubber* sequence) ; compared with locally smoothed finite differences, regularized differentiation gives a flow that is visually more consistent with the ground truth.

3.2.3 Scene flow estimation

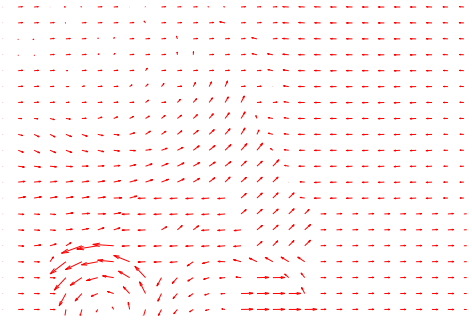
In this section, we consider the problem of estimating dense scene flow and depth from a monocular image sequence [4]. Scene flow is the three-dimensional (3D) velocity field, over the image domain, of the visible environmental surfaces. Given an image sequence $I(x, y, t)$, where, similarly to the notations above, (x, y) are the spatial coordinates and t is the time coordinate, the problem consists of minimizing the following functional w.r.t to the scene flow coordinates U, V, W and depth Z [4] :

$$\begin{aligned} & \tfrac{1}{2} \quad \int_\Omega (fI_xU + fI_yV - (xI_x + yI_y)W + I_tZ + I_tZ_0)^2 dxdy \\ & + \quad \frac{\gamma}{2} \int_\Omega (\|\nabla U\|^2 + \|\nabla V\|^2 + \|\nabla W\|^2 + \|\nabla Z\|^2) dxdy, \end{aligned} \tag{3.11}$$

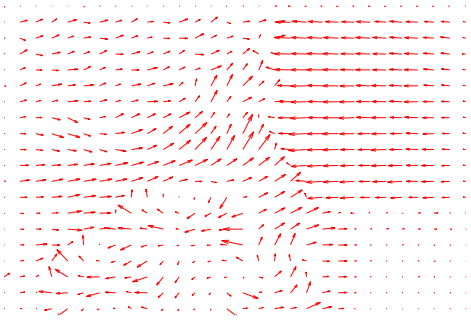
where (U, V, W) is the 3D scene flow, Z is relative depth, f is the focal length of imaging, I_x, I_y and I_t are the image spatiotemporal derivatives and γ is a positive constant, which balances the contribution of the smoothness terms. The discretized Euler-Lagrange conditions for a minimum of (3.11) give a large scale sparse system of linear equations, which can be ordered so as to accommodate an efficient solution by Gauss-Seidel iterations [4].

The formulation in (3.11) involves the image partial derivatives. We compared the scene flow estimation results obtained by computing I_x and I_y via regularized differentiation to those obtained by computing these derivatives with smoothed finite

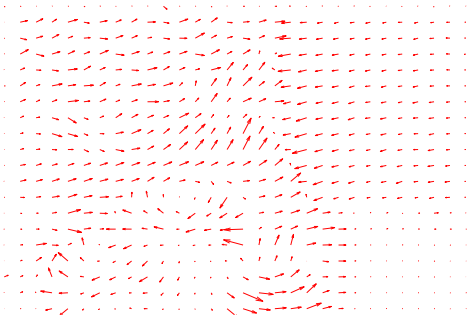
(a) First frame



(b) Ground truth

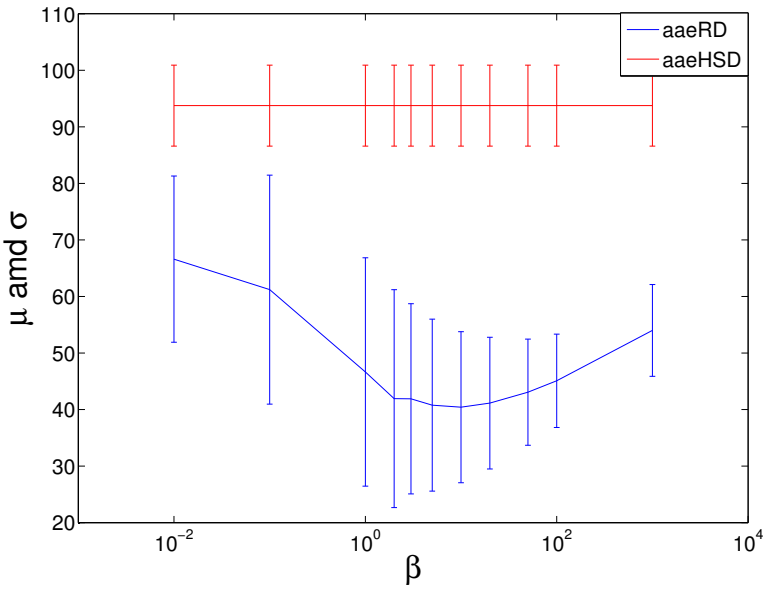


(c) Finite diff. : aae=25.11, epe=0.74

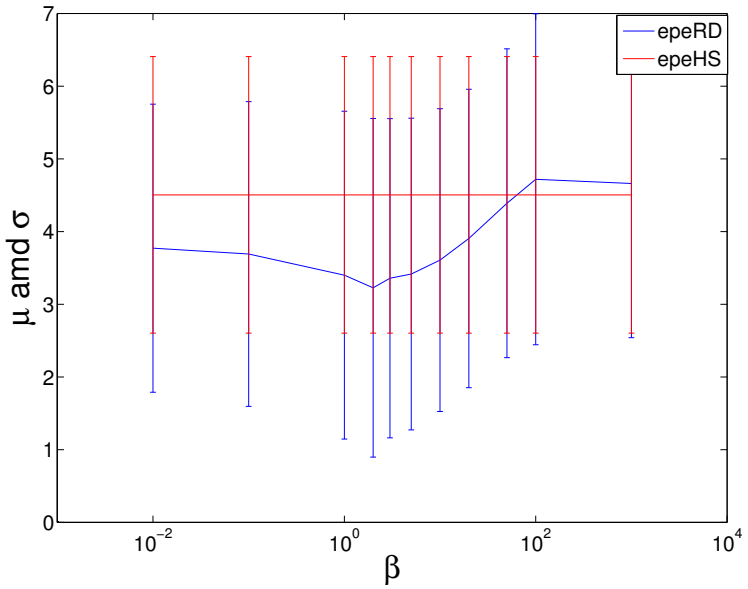


(d) ours : aae=23.02, epe=0.69

FIGURE 3.4 – *Rubber sequence* : (a) first of the two images used. A vector representation of optical flow is shown : (b) ground truth, (c) computed with standard finite differences in (3.1) and, (d) computed with the regularized differentiation scheme ($\beta = 1$).



(a)



(b)

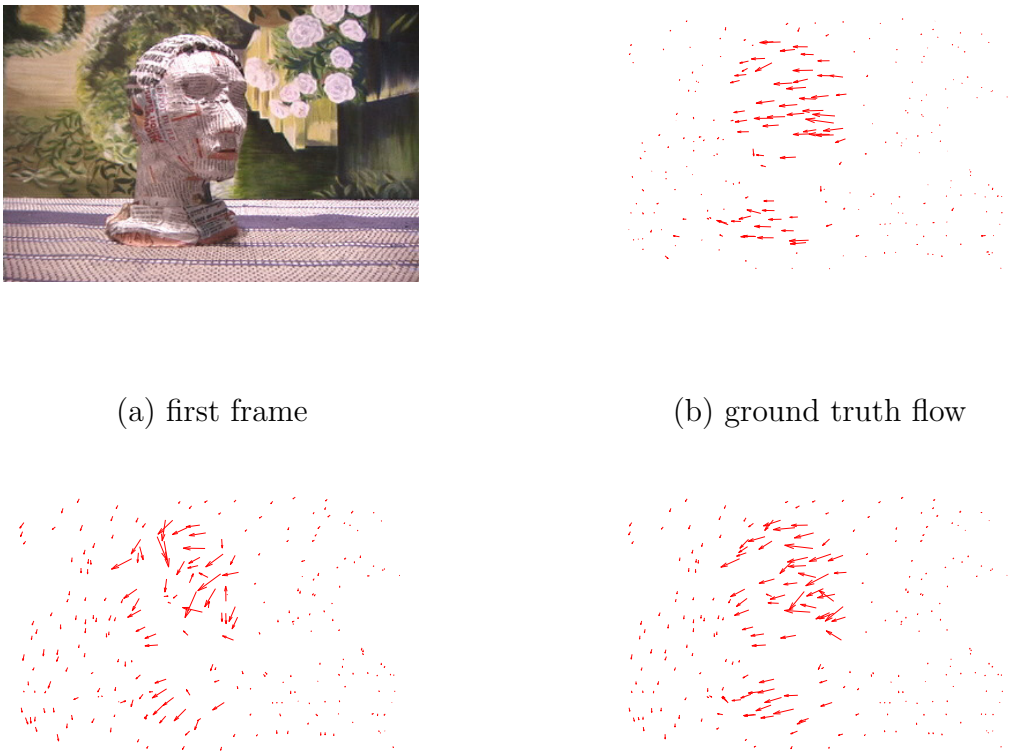
FIGURE 3.5 – Middlebury database : Means and standard deviations of angular error *aae* (top) and endpoint error *epe* (bottom), for different values of the weight coefficient β in the regularized differentiation functional. The length of bars indicates standard deviation.

differences as in Eq. (3.1). We used an experimental dataset of five sequences, each presenting some challenges : (1) the *Marbled block* synthetic image sequence from the database of KOGS/IAKS Laboratory (Germany), where weak spatiotemporal intensity, occluding boundaries, and moving shadows, are present ; (2) the *Cylinder and box* real image sequence with real 3D motion, courtesy of Debrunner and Ahuja [20] ; (3) the *Berber* real image sequence, with weak textures and various depth discontinuities ; (4) the *Pharaohs* real image sequence, also with weak textures and sharp discontinuities, and (5) the *Rock* real image sequence from the CMU/VAS image database, with texture-poor areas, and weak spatiotemporal variations.

The comparison of the scene flow results, when the image derivatives are computed with smoothed finite differences and with regularized differentiation, is done as follows : In each case of computing the image derivatives, we determined the optical flow induced by the recovered scene flow, using the expressions of scene flow in terms of optical flow, and compared it to an optical flow ground truth that we created as follows : In each of the two images used by the scene flow estimation algorithm, we extracted about a hundred key points using the SURF key point detection method [21]. The correspondence between the key points of these two input images then defines the points ground truth optical flow.

Table 3.2 lists the aae and epe errors for the various sequences ; it shows systematically a much better motion accuracy when derivatives are estimated by regularized differentiation. The errors point to the practical benefits of regularized differentiation.

As a typical example of evaluation by visual inspection, Fig. 3.6 shows the results for the *Berber* sequence. The figure displays a vector representation for the scene flow induced optical flow, for the ground truth and for the two cases of computing the image derivatives. One can see clearly that the motion field recovered using regularized differentiation is much more consistent with the ground truth than the one obtained with finite differences averaging.



(a) first frame (b) ground truth flow

(c) smoothed finite differences (d) reg. differentiation

FIGURE 3.6 – A vector representation of scene flow induced optical flow for the Berber figurine movement. First row : (left) the first of the two images used ; (right) ground truth flow. Second row : (left) flow when using smoothed finite differences for image derivatives ; (right) flow when using the regularized differentiation scheme.

TABLE 3.2 – Average angular error (aae) and endpoint error (epe) of optical flow constructed from the estimated scene flow : using the regularized differentiation scheme and averaged finite differences (3.1). Coefficient β was fixed equal to 1 for all experiments.

Sequence	errors	Reg. Differentiation	Smoothing finite diff. (3.1)
Marbled-block	aae	4.14	8.8989
	epe	0.10	0.2001
Cylinder	aae	18.68	59.2658
	epe	0.78	2.0384
Berber	aae	11.61	38.5148
	epe	0.41	1.1136
Pharaohs	aae	20.01	59.1747
	epe	0.62	1.5492
Rock	aae	23.18	90.7150
	epe	0.92	2.4981
Mean	aae	15.52	51.3138
	epe	0.57	1.4799

3.3 Conclusion

This paper advocated a regularized differentiation scheme to use as a general method to estimate image derivatives. Regularized differentiation minimizes an anti-differentiation data discrepancy integral term in conjunction with a smoothness regularization constraint. When discretized, the Euler-Lagrange necessary conditions for a minimum yield a large-scale sparse system of linear equations, which can be solved efficiently by Jacobi/Gauss-Seidel iterations. We evaluated the method in the context of two important problems in computer vision : optical flow and 3D scene flow esti-

mation. The experiments used the *Middlebury* dataset and other real and synthetic images. The results showed that the regularized differentiation outperforms standard finite difference definitions that are commonly used in motion analysis. The method can be extended to boundary preserving estimation by standard formulations such as L_1 and semi-quadratic regularization. The method can also be investigated in several other image analysis problems, e.g., image registration.

3.4 Acknowledgment

This study is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

Bibliographie

[1] Mitiche, A., and Aggarwal, J. K. : "Computer Vision Analysis of Image Motion by Variational Methods" (Springer, 2013)

[2] Sun, D., Roth, S., and Black, M. J. : "Secrets of optical ow estimation and their 220 principles", Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, June 2010, pp. 2432–2439

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

[4] Mitiche, A., Mathlouthi, Y., and Ben Ayed, I. : "Monocular concurrent recovery of structure and motion scene flow", Frontiers in ICT, Computer Image Analysis, 2015, **2**, pp. 1–16

[5] Heinrich, M. P., Jenkinson, M., Bhushan, M., Matin, T., Gleeson, F. V., Brady, M., and Schnabel, J. A. : "Mind : Modality independent neighborhood descriptor for multi-modal deformable registration", Medical Image Analysis, 2012, **16**, (7), pp. 1423–1435

[6] Periaswamy, S., and Farid, H. : "Medical image registration with partial data", Medical image analysis, 2006, **10**, (3), pp. 452–464

[7] Tappen, M., Freeman, W. T., and Adelson, E. H. : "Recovering intrinsic images from a single image", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, **27**, (9), pp. 1459–1472

[8] Besl, P. J., and Jain, R. : "Invariant surface characteristics for 3d object recognition in range images", Computer Vision, Graphics, and Image Processing, 1986, **33**,(1), pp. 33–80

[9] Terzopoulos, D. : "Regularization of inverse visual problems involving discontinuities", IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986, **8**, (4), pp. 413–424

[10] Chartrand, R. : "Numerical differentiation of noisy, non-smooth data", ISRN Applied Mathematics, 2011, Article ID 164564, pp. 1–11

[11] Hanke, M., and Scherzer, O. : "Inverse problems light : numerical differentiation", The American Mathematical Monthly, 2001, **108**, (6), pp. 512–521

[12] Vogel, C. R., "Computational methods for inverse problems" (Frontiers in Applied Mathematics, SIAM, 2002)

[13] Hanke, M., and Scherzer, O. : "Inverse problems light : numerical differentiation", The American Mathematical Monthly, 2001, **108**, (6), pp. 512–521

[14] Khan, I. R., and Ohba, R. : "New finite difference formulas for numerical differentiation", Journal of Computational and Applied Mathematics, 2000, **126**, (1), pp. 269–276

[15] Murio, D., Meja, C. E., and Zhan, S. : "Discrete mollification and automatic numerical differentiation", Computers and Mathematics with Applications, 1998, **35**, (5), pp. 1–16

[16] Ramm, A., and Smirnova, A. : "On stable numerical differentiation", Mathematics of computation, 2001, **70**, (235), pp. 1131–1153

[17] Forsythe, G., Malcolm, M., and Moler, C. : "Computer methods for mathematical computations" (Prentice-Hall, 1977)

[18] Tritsiklis, J. N., "A comparison of Jacobi and Gauss-Seidel parallel iterations", Applied Mathematics Letters, 1989, **2**, (2), pp. 167-170

[19] Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M. J., and Szeliski, R. : "A database and evaluation methodology for optical flow", International Journal of Computer Vision, 2011, **92**, (1), 1-31

[20] Debrunner, C., and Ahuja, N. : "Segmentation and factorization-based motion and structure estimation for long image sequences", IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, **20**, (2), pp. 206–211

[21] Bay, H., Ess, A., Tuytelaars, T., and L. Van Gool, Speeded-up robust features (surf), Computer Vision and Image Understanding, 2008, **110**, (3), pp. 346-359

[22] Gonzalez, R. C., and Woods, R. E. : "Digital Image Processing" (Pearson, 2008)

Chapitre 4

Monocular concurrent recovery of structure and motion scene flow

Amar Mitiche, Yosra Mathlouthi, and Ismail Ben Ayed

Frontiers in ICT, vol. 2, p. 16, 2015.

Résumé : Cet article décrit une méthode variationnelle pour l'estimation conjointe de la structure tridimensionnelle et du mouvement de flot de scène à partir d'une seule séquence d'images. On développe un schéma de base qui minimise une fonctionnelle composée d'un terme de conformité du flot de scène et de la profondeur aux variations spatiotemporelles de la séquence d'images, et des termes de régularisation quadratique pour assurer une solution lisse. Le terme des données est obtenu en réécrivant la vitesse optique dans la contrainte du gradient du flot optique en termes du flot de scène et de la profondeur. Par conséquence, cet énoncé du problème est analogue à la formulation classique de l'estimation du flot optique de Horn and Schunck bien que ceci implique le flot de scène et la profondeur au lieu du mouvement de l'image. La discrétisation des équations d'Euler-Lagrange produit un système d'équations linéaires creux

et à grande échelle dont les inconnues sont les trois coordonnées du flot de scène ainsi que la profondeur. Ces équations peuvent être ordonnées de façon que leur matrice correspondante soit symétrique, définie et positive, ce qui permet une résolution efficace par les itérations de Gauss-Seidel. Les résultats expérimentaux sont présentés afin de vérifier la validité et l'efficacité de la méthode.

Abstract

This paper describes a variational method of joint three-dimensional structure and motion scene flow recovery from a single image sequence. A basic scheme is developed by minimizing a functional with a term of conformity of scene flow and depth to the image sequence spatiotemporal variations, and quadratic smoothness regularization terms. The data term follows by re-writing optical velocity in the optical flow gradient constraint in terms of scene flow and depth. As a result, this problem statement is analogous to the classical Horn and Schunck optical flow formulation except that it involves scene flow and depth rather than image motion. When discretized, the Euler-Lagrange equations give a large scale sparse system of linear equations in the unknowns of the scene flow three coordinates and depth. The equations can be ordered in such a way that its matrix is symmetric positive definite such that they can be solved efficiently by Gauss-Seidel iterations. Experiments are shown to verify the scheme's validity and efficiency.

4.1 Introduction

Scene flow is the field over the image domain of the visible environmental surfaces three-dimensional (3D) velocities. Only the visible surfaces are relevant in the

definition because they alone, not the occluded, contribute to visual information. As a result, the scene flow domain is the image domain : at each image point, scene flow consists of the velocity vector of the corresponding visible environmental surface point. It is the time derivative of the point 3D position.

For a working definition of scene flow, let Ω be the common domain of an image sequence $I(\mathbf{x}, \mathbf{t})$, where $\mathbf{x} = (\mathbf{x}, \mathbf{y})$ designates image position, and t is time. For each point $\mathbf{x} \in \mathbf{\Omega}$, let $\mathbf{P'} = \frac{\mathbf{dP}}{\mathbf{dt}}$ be the velocity vector of the visible environmental point $\mathbf{P} = (X, Y, Z)$ projected on $\mathbf{x}$. Scene flow is the velocity vector field $\mathbf{F} = (\mathbf{U}, \mathbf{V}, \mathbf{W}) = (\frac{\mathbf{dX}}{\mathbf{dt}}, \frac{\mathbf{dY}}{\mathbf{dt}}, \frac{\mathbf{dZ}}{\mathbf{dt}})$ over Ω. It is a function of image position and time : $\mathbf{F} = \mathbf{F}(\mathbf{x}, \mathbf{t})$.

Scene flow is a fundamental dimension of three-dimensional scene analysis for the obvious reason that it describes the motion of real objects in the environment. Moreover, it is related to optical flow via depth [1]. It can be regarded as a three-dimensional analogue of *optical flow* : at each point $\mathbf{x} \in \mathbf{\Omega}$, and each instant of time, scene flow is the velocity vector of the visible environmental point $\mathbf{P}$ which projects on $\mathbf{x}$, whereas optical flow is the velocity vector of the image of $\mathbf{P}$ at $\mathbf{x}$.

From a broad perspective, scene flow can be computed in one of two ways : *parametric* and *non parametric*. Parametric methods use a parametric form of the scene flow coordinates and non parametric methods compute scene flow directly as a vector field without resorting to an intermediate representation.

Investigations of parametric scene flow generally assume that environmental objects are rigid and, therefore, decompose scene flow in terms of 3D translational and rotational parameters [1]. This representation leads to the Longuet-Higgins and Pradzny fundamental equations [2] relating the rigid motion parameters, depth, and optical flow. Depth and the rigid screw motion parameters become the unknown 3D variables to determine, from which scene flow can be recovered a posteriori. In most studies, the Longuet-Higgins and Pradzny equations underlie the recovery of rigid body structure and motion from image sequences, even when not used explicitly.

Current parametric methods can be separated into two broad categories, those which treat the case of a viewing system moving in a static environment and those which allow the viewing system and the environmental objects to move simultaneously and independently. In the first case, the problem is significantly simpler because the single 3D motion to take into account is that of the viewing system [3–15]. Moreover, segmentation of the environment into differently moving objects is not an issue in a static environment, simplifying the problem further.

The simultaneous motion of the viewing system and viewed objects has also been the subject of several studies [16–23]. The non variational methods in [16–18] assume that optical flow is given beforehand and segment the visual field into differently moving rigid objects by grouping processes such as region growing by 3D motion [18], clustering of 3D motion via mixture models [16], and clustering via oriented projections of optical flow [17].

The variational methods in [19–23] use functionals with a data term based on the Longuet-Higgins and Prazdny rigid motion model and a regularization term to account for 3D interpretation discontinuities; they mainly differ in the way these discontinuities are represented.

Non parametric scene flow computation methods seek to recover scene flow at each point of the image domain without recourse to a parametric form of the movements or surfaces in space. Such methods are most relevant when practicable models of scene flow cannot be assumed, as with, for instance, articulated human and animal motion.

Because scene flow is related to depth, via optical flow which they jointly define [1], non parametric scene flow computation has been generally studied in the context of stereoscopy [24–32], although it stands independent of stereoscopy. Here following, we will show that non parametric scene flow can actually be recovered from a single image sequence. We will describe a variational scheme reminiscent of the Horn and Schunck optical flow estimation method. The functional of this formulation has two

terms : a data term which relates 3D velocity to depth via the image sequence spatiotemporal variations, and a classic smoothness regularization term. The data term falls out simply by rewriting the Horn and Schunck optical flow constraint linearly in terms of scene flow and depth. The Euler-Lagrange equations corresponding to the minimization of the objective functional yield, when discretized, a large sparse system of linear equations which can be solved efficiently by Jacobi/Gauss-Seidel iterations. The scheme can be generalized to boundary preserving formulations as in optical flow estimation [33, 34].

The remainder of this paper is organized as follows : Section 2 formulates the problem and develops the objective functional. Section 3 deals with the optimization of the objective functional. It derives the Euler-Lagrange equations and the corresponding discrete system of linear equations in the variables of scene flow and relative depth. It also shows that the matrix of this system is symmetric positive definite, which prescribes a solution by Jacobi/Gauss-Seidel iterations. Section 4 addresses the problem of regularized spatiotemporal derivative computation and Section 5 gives experimental results.

4.2 Formulation

The problem is to recover scene flow and depth from an image sequence $I : (x, y, t) \rightarrow I(x, y, t)$, where (x, y) are the coordinates over the bounded image domain Ω, and $t \in \mathbb{R}^+$ is time. The formulation starts with the Horn and Schunck optical flow gradient constraint [35], which relates the coordinate functions u and v of optical flow to the image sequence spatiotemporal variations :

$$I_x u + I_y v + I_t = 0, \tag{4.1}$$

where I_x, I_y, I_t are the image spatiotemporal partial derivatives. Let $\mathbf{P}$ be a point in space, (X, Y, Z) its 3D coordinates, and (x, y) its image coordinates. The viewing

system model geometry is shown in Fig. 4.1. Derivation with respect to time of the projection equations $x = f\frac{X}{Z}$ and $y = f\frac{Y}{Z}$, where f is the focal length, gives the coordinates u, v of optical velocity as functions of scene flow and depth :

$$u = \frac{dx}{dt} = \frac{fU - xW}{Z}; \quad v = \frac{dy}{dt} = \frac{fV - yW}{Z}, \tag{4.2}$$

where Z designates depth (Fig. 4.1) and $(U, V, W) = (\frac{dX}{dt}, \frac{dY}{dt}, \frac{dZ}{dt})$ is the scene flow at $\mathbf{P}$. Substitution of these optical flow expressions in the gradient constraint (4.1), followed by the multiplication of the left hand side by $Z \neq 0$ gives the following linear constraint relating scene flow and depth to the image spatiotemporal derivatives :

$$fI_xU + fI_yV - (xI_x + yI_y)W + I_tZ = 0. \tag{4.3}$$

There are two important observations to make about this linear equation. First, an obvious observation is that this equation evaluates at each point of the image domain : $\mathbf{fI_x(x)U(x) + fI_y(x)V(x) - (xI_x(x) + yI_y(x))W(x) + I_t(x)Z(x) = 0}$ and, therefore, contains four unknown variables at each point. As with optical flow computation [36], this says that any local interpretation of the variables is ambiguous. Dense interpretation, i.e., global over the image domain, which is of interest to us here, will require additional constraints. In a classic way, we will use a variational statement of the problem where these additional constraints characterize the variables as smooth over the image domain.

The second observation is that the equation, being homogeneous, has a trivial solution, $U = V = W = 0; Z = 0$, in which, of course, we are not interested, and can easily avoid in practice. More importantly, since we are dealing with an actual physical problem, the environmental depth field and scene flow, which gave rise to the image spatiotemporal variations, constitute a solution but so does any scaled version of it : if $(U, V, W), Z$ is a solution, so is $k(U, V, W), kZ$ for some arbitrary real k. This is a limitation inherent to the recovery of 3D structure and motion from 2D image sequences [37].

Theoretically, the scale can be fixed by setting the depth of a particular point, in which case the depth of any other point is relative to it. However, this is hardly an answer, because setting the depth of a point would affect a single equation out of thousands that generally make up the discrete system of equations in practice. Along a different vein, a particular solution can be picked by imposing it a given norm. For instance, we could, say, compute a unit norm solution : $\|(U, V, W, Z)\| = 1$ by solving under this constraint the system of equations of the problem. We will follow instead an effective simple scheme : we will adopt a variational formulation of the problem where the scale of interpretation is fixed by solving for depth Z_r relative to a fronto-parallel plane $\Pi_{Z_0} : Z = Z_0$, for some arbitrary positive depth Z_0. More precisely, this is done by first rewriting Eq. (4.3) as follows :

$$fI_xU + fI_yV - (xI_x + yI_y)W + I_t(Z - Z_0) + I_tZ_0 = 0, \tag{4.4}$$

and then making a change of variable $Z_r \leftarrow Z - Z_0$, which would give :

$$fI_xU + fI_yV - (xI_x + yI_y)W + I_tZ_r + I_tZ_0 = 0 \tag{4.5}$$

For notational simplicity and economy, we can reuse the symbol Z to designate relative depth Z_r, in which case we write Eq. (4.5) as :

$$fI_xU + fI_yV - (xI_x + yI_y)W + I_tZ + I_tZ_0 = 0 \tag{4.6}$$

By rewriting this data equation with respect to reference plane Π_{Z_0} effectively fixes the scale of 3D interpretation because for another reference plane, say $Z_1 = kZ_0$, the equation integrity is maintained by the correspondingly scaled interpretation $k(U, V, W), kZ$ and, inversely, for a scaled solution $k(U, V, W), kZ$ the equation integrity is maintained by a reference plane at kZ_0. The trivial solution, which is now $(0, 0, 0, -Z_0)$, rather than $(0, 0, 0, 0)$, can be avoided in practice simply by initializing away from it in the iterative algorithm we are about to develop. This iterative algorithm, as will be detailed subsequently, consists of Gauss-Seidel iterations which at

each step solve by singular value decomposition local 4×4 systems of linear equations resulting from the objective functional Euler-Lagrange equations.

We can now formulate the problem of joint computation of scene flow and relative depth from a single image sequence as the minimization of the following functional :

$$\begin{aligned}
\mathbf{E}(U,V,W,Z|I) &= \frac{1}{2}\int_\Omega (fI_xU + fI_yV - (xI_x + yI_y)W + I_tZ + I_tZ_0)^2 dxdy \\
&+ \frac{\alpha}{2}\int_\Omega (\|\nabla U\|^2 + \|\nabla V\|^2 + \|\nabla W\|^2) dxdy \\
&+ \frac{\beta}{2}\int_\Omega \|\nabla Z\|^2 dxdy, \qquad (4.7)
\end{aligned}$$

where α and β are positive constants balancing the contributions of the smoothness terms. This functional can be modified to preserve the boundaries of scene flow/depth via discontinuity preserving regularization [33].

4.3 Optimization

The Euler-Lagrange equations corresponding to the objective functional (4.7) are the following coupled partial differential equations :

$$\begin{aligned}
&fI_x(fI_xU + fI_yV + (-xI_x - yI_y)W + I_tZ + I_tZ_0) - \alpha\nabla^2 U = 0 \\
&fI_y(fI_xU + fI_yV + (-xI_x - yI_y)W + I_tZ + I_tZ_0) - \alpha\nabla^2 V = 0 \\
&(-xI_x - yI_y)(fI_xU + fI_yV + (-xI_x - yI_y)W + I_tZ + I_tZ_0) - \alpha\nabla^2 W = 0 \\
&I_t(fI_xU + fI_yV + (-xI_x - yI_y)W + I_tZ + I_tZ_0) - \beta\nabla^2 Z = 0, \qquad (4.8)
\end{aligned}$$

to which we add the Neumann boundary conditions on the solution at the boundary $\partial\Omega$ of Ω :

$$\frac{\partial U}{\partial \mathbf{n}} = 0, \quad \frac{\partial V}{\partial \mathbf{n}} = 0, \quad \frac{\partial W}{\partial \mathbf{n}} = 0, \quad \frac{\partial Z}{\partial \mathbf{n}} = 0, \qquad (4.9)$$

where $\frac{\partial}{\partial \mathbf{n}}$ is the differentiation operator in the direction of the normal $\mathbf{n}$ of $\partial\Omega$.

Let Ω be discretized as a unit-spacing grid D and the grid points indexed by the integers $\{1, 2, ..., N\}$. Pixels are indexed in the lexicographical order, i.e., top-down and left-to-right. $N = n^2$ when the image is $n \times n$. Let $a = fI_x$, $b = fI_y$, $c = -(xI_x + yI_y)$, $d = I_t$. For all grid point $i \in \{1, 2, ..., N\}$, a discrete approximation of the Euler-Lagrange equations (4.8) is :

$$\begin{aligned}
&a_i^2 U_i + a_i b_i V_i + a_i c_i W_i + a_i d_i Z_i + a_i d_i Z_0 - \alpha \sum_{j \in \mathbf{N}_i} (U_j - U_i) = 0 \\
&b_i a_i U_i + b_i^2 V_i + b_i c_i W_i + b_i d_i Z_i + b_i d_i Z_0 - \alpha \sum_{j \in \mathbf{N}_i} (V_j - V_i) = 0 \\
&c_i a_i U_i + c_i b_i V_i + c_i^2 W_i + c_i d_i Z_i + c_i d_i Z_0 - \alpha \sum_{j \in \mathbf{N}_i} (W_j - W_i) = 0 \\
&d_i a_i U_i + d_i b_i V_i + d_i c_i W_i + d_i^2 Z_i + d_i^2 Z_0 - \beta \sum_{j \in \mathbf{N}_i} (Z_j - Z_i) = 0
\end{aligned} \tag{4.10}$$

where $(U_i, V_i, W_i, Z_i) = (U, V, W, Z)_i$ is the scene flow at i ; a_i, b_i, c_i, d_i are the values at i of a, b, c, d, respectively, and $\mathbf{N}_i$ is the set of indices of the neighbors of i. For the 4-neighborhood, $card(\mathbf{N}_i) = 4$ for points interior in D, and $card(\mathbf{N}_i) < 4$ for boundary points. The Laplacian $\nabla^2 Q$, $Q \in \{U, V, W, Z\}$, in the Euler-Lagrange equations has been discretized as $\frac{1}{4} \sum_{j \in \mathbf{N}_i} (Q_j - Q_i)$, where the factor $\frac{1}{4}$ is absorbed by α and β.

Rewriting (4.10), and where $n_i = card(\mathbf{N}_i)$, we have the following system of linear equations, $i \in \{1, ..., N\}$:

$$(S) \quad \begin{cases}
(a_i^2 + \alpha n_i) U_i + a_i b_i V_i + a_i c_i W_i + a_i d_i Z_i - \alpha \sum_{j \in \mathbf{N}_i} U_j &= -a_i d_i Z_0 \\
b_i a_i U_i + (b_i^2 + \alpha n_i) V_i + b_i c_i W_i + b_i d_i Z_i - \alpha \sum_{j \in \mathbf{N}_i} V_j &= -b_i d_i Z_0 \\
c_i a_i U_i + c_i b_i V_i + (c_i^2 + \alpha n_i) W_i + c_i d_i Z_i - \alpha \sum_{j \in \mathbf{N}_i} W_j &= -c_i d_i Z_0 \\
d_i a_i U_i + d_i b_i V_i + d_i c_i W_i + (d_i^2 + \beta n_i) Z_i - \beta \sum_{j \in \mathbf{N}_i} Z_j &= -d_i^2 Z_0
\end{cases}$$

Let $\mathbf{q} = (q_1, ..., q_{4N})^t \in \mathbb{R}^{4N}$ be the vector with coordinates $q_{4i-3} = U_i$, $q_{4i-2} = V_i, q_{4i-1} = W_i, q_{4i} = Z_i$, $i \in \{1, ..., N\}$, and $\mathbf{r} = (r_1, ..., r_{4N})^t \in \mathbf{R}^{4N}$ the vector with

coordinates $r_{4i-3} = -a_i d_i Z_0$, $r_{4i-2} = -b_i d_i Z_0$, $r_{4i-1} = -c_i d_i Z_0$, and $r_{4i} = -d_i^2 Z_0$, $i \in \{1, ..., N\}$. System (S) of linear equations can be written in matrix form as :

$$\mathbf{A}\mathbf{q} = \mathbf{r} \tag{4.11}$$

where $\mathbf{A}$ is the $4N \times 4N$ matrix with elements $\mathbf{A}_{4i-3,4i-3} = a_i^2 + \alpha n_i$; $\mathbf{A}_{4i-2,4i-2} = b_i^2 + \alpha n_i$; $\mathbf{A}_{4i-1,4i-1} = c_i^2 + \alpha n_i$; $\mathbf{A}_{4i,4i} = d_i^2 + \beta n_i$; $\mathbf{A}_{4i-3,4i-2} = \mathbf{A}_{4i-2,4i-3} = a_i b_i$; $\mathbf{A}_{4i-3,4i-1} = \mathbf{A}_{4i-1,4i-3} = a_i c_i$; $\mathbf{A}_{4i-3,4i} = \mathbf{A}_{4i,4i-3} = a_i d_i$; $\mathbf{A}_{4i-2,4i-1} = \mathbf{A}_{4i-1,4i-2} = b_i c_i$; $\mathbf{A}_{4i-2,4i} = \mathbf{A}_{4i,4i-2} = b_i d_i$; $\mathbf{A}_{4i-1,4i} = \mathbf{A}_{4i,4i-1} = c_i d_i$; for all $i \in \{1, ..., N\}$; $\mathbf{A}_{4i-3,4j-3} = \mathbf{A}_{4i-2,4j-2} = \mathbf{A}_{4i-1,4j-1} = -\alpha$ and $\mathbf{A}_{4i,4j} = -\beta$, for all $i, j \in \{1, ..., N\}$ such that $j \in \mathbf{N}_i$, all other elements being equal to zero.

System (S) is a large scale sparse system of linear equations. Such systems are best solved by iterative methods designed for sparse matrices [38, 39]. Here following we prove that matrix $\mathbf{A}$ is symmetric positive definite, which implies an effective solution of Eq. (4.11) by 4×4 block-wise Gauss-Seidel iterations.

One can easily verify that matrix $\mathbf{A}$ is symmetric. Matrix $\mathbf{A}$ is also positive definite. To show this we verify that $\mathbf{q}^t \mathbf{A} \mathbf{q} > 0$ for all $\mathbf{q} \in \mathbf{R}^{4N}, \mathbf{q} \neq \mathbf{0}$. We have :

$$\begin{aligned}
\mathbf{q}^t \mathbf{A} \mathbf{q} &= \sum_{i=1}^{N} \left((a_i^2 + \alpha n_i) U_i + a_i b_i V_i + a_i c_i W_i + a_i d_i Z_i - \alpha \sum_{j \in \mathbf{N}_i} U_j \right) U_i \\
&+ \sum_{i=1}^{N} \left(b_i a_i U_i + (b_i^2 + \alpha n_i) V_i + b_i c_i W_i + b_i d_i Z_i - \alpha \sum_{j \in \mathbf{N}_i} V_j \right) V_i \\
&+ \sum_{i=1}^{N} \left(c_i a_i U_i + c_i b_i V_i + (c_i^2 + \alpha n_i) W_i + c_i d_i Z_i - \alpha \sum_{j \in \mathbf{N}_i} W_j \right) W_i \\
&+ \sum_{i=1}^{N} \left(d_i a_i U_i + d_i b_i V_i + d_i c_i W_i + (d_i^2 + \beta n_i) Z_i - \beta \sum_{j \in \mathbf{N}_i} Z_j \right) Z_i
\end{aligned} \tag{4.12}$$

Following algebraic manipulations, we get :

$$\begin{aligned}
\mathbf{q}^t \mathbf{A} \mathbf{q} &= \sum_{i=1}^{N} (a_i U_i + b_i V_i + c_i W_i + d_i Z_i)^2 + \alpha \sum_{i=1}^{N} \left(n_i (U_i^2 + V_i^2 + W_i^2) \right) + \beta \sum_{i=1}^{N} \left(n_i (Z_i^2) \right) \\
&- \alpha \sum_{i=1}^{N} \left(\sum_{j \in \mathbf{N}_i} U_j U_i + \sum_{j \in \mathbf{N}_i} V_j V_i + \sum_{j \in \mathbf{N}_i} W_j W_i \right) - \beta \sum_{i=1}^{N} \left(\sum_{j \in \mathbf{N}_i} Z_j Z_i \right)
\end{aligned} \tag{4.13}$$

If we distribute the n_i terms U_i of the second row into the corresponding neighborhood sum of the third row, we will get :

$$\sum_{i=1}^{N} n_i U_i^2 - \sum_{i=1}^{N} \sum_{j \in \mathbf{N}_i} U_j U_i = \sum_{i=1}^{N} \sum_{j \in \mathbf{N}_i; j>i} (U_i^2 + U_j^2 - 2U_j U_i) = \sum_{i=1}^{N} \sum_{j \in \mathbf{N}_i; j>i} (U_i - U_j)^2$$

Using similar manipulations for the other variables (V, W, Z), we arrive at the expression we need :

$$\begin{aligned} \mathbf{q}^t \mathbf{A} \mathbf{q} &= \sum_{i=1}^{N} (a_i U_i + b_i V_i + c_i W_i + d_i Z_i)^2 \\ &+ \alpha \sum_{i=1}^{N} \sum_{j \in \mathbf{N}_i; j>i} \left((U_i - U_j)^2 + (V_i - V_j)^2 + (W_i - W_j)^2 \right) \\ &+ \beta \sum_{i=1}^{N} \sum_{j \in \mathbf{N}_i; j>i} \left((Z_i - Z_j)^2 \right) \end{aligned} \tag{4.14}$$

For $\mathbf{q} \neq \mathbf{0}$, we have $\mathbf{q}^t \mathbf{A} \mathbf{q} = 0$ if and only if the terms in both sums on the right-hand side of (4.14) are zero. The second-sum terms are zero if and only if the scene consists of a fronto-parallel plane (plane $Z = Z_0$) under constant translation ($(U_i, V_i, W_i) = \mathbf{T}$). The first-sum terms are zero if and only if all vectors $(a_i, b_i, c_i, d_i)_i = (I_{xi}, I_{yi}, -x_i I_{xi} - y_i I_{yi}, I_{ti})_i$ lie in a hyperplane for all $(x_i, y_i) \in D$. This is possible if and only if the spatiotemporal visual pattern is null, which is an irrelevant case. Therefore, $\mathbf{q}^t \mathbf{A} \mathbf{q} > 0$ for $\mathbf{q} \neq \mathbf{0}$ and $\mathbf{A}$ is positive definite. This means that the point-wise and block-wise Gauss-Seidel and relaxation iterative methods for solving system (4.11) converge [38, 39].

For a 4×4 block division of matrix $\mathbf{A}$, the Gauss-Seidel iterations consist of solving, for each $i \in \{1, ..., N\}$, the following 4×4 linear system of equations, where

k is the iteration number :

$$
\begin{aligned}
(a_i^2 + \alpha n_i)U_i^{k+1} + a_i b_i V_i^{k+1} + a_i c_i W_i^{k+1} + a_i d_i Z_i^{k+1} = \\
-a_i d_i Z_0 + \alpha \left(\sum_{j \in \mathbf{N}_i; j<i} U_j^{k+1} + \sum_{j \in \mathbf{N}_i; j>i} U_j^k \right) \\
b_i a_i U_i^{k+1} + (b_i^2 + \alpha n_i)V_i^{k+1} + b_i c_i W_i^{k+1} + b_i d_i Z_i^{k+1} = \\
-b_i d_i Z_0 + \alpha \left(\sum_{j \in \mathbf{N}_i; j<i} V_j^{k+1} + \sum_{j \in \mathbf{N}_i; j>i} V_j^k \right) \\
c_i a_i U_i^{k+1} + c_i b_i V_i^{k+1} + (c_i^2 + \alpha n_i)W_i^{k+1} + c_i d_i Z_i^{k+1} = \\
-c_i d_i Z_0 + \alpha \left(\sum_{j \in \mathbf{N}_i; j<i} W_j^{k+1} + \sum_{j \in \mathbf{N}_i; j>i} W_j^k \right) \\
d_i a_i U_i^{k+1} + d_i b_i V_i^{k+1} + d_i c_i W_i^{k+1} + (d_i^2 + \beta n_i)Z_i^{k+1} = \\
-d_i^2 Z_0 + \beta \left(\sum_{j \in \mathbf{N}_i; j<i} Z_j^{k+1} + \sum_{j \in \mathbf{N}_i; j>i} Z_j^k \right),
\end{aligned}
$$

which can be done efficiently by the singular value decomposition method [40].

4.4 Estimation of the spatiotemporal derivatives

The purpose is to estimate the spatiotemporal derivatives I_x, I_y, I_t from two consecutive images of a sequence. The estimation of a function derivative from inaccurate data is an ill-posed problem because small changes in the function values can result in arbitrarily large errors in the derivative estimated by finite differences [41]. Therefore, image noise can adversely affect the quality of motion interpretation that uses finite difference image derivatives. In motion analysis, the problem has been generally approached by local averaging of the finite difference derivatives [35]. However, regularized differentiation can be more effective as we show in the following.

4.4.1 Differentiation by averaging finite differences

Following the formulas in the Horn and Schunck paper on optical estimation [35], motion analysis studies have generally used forward first differences to represent derivatives, locally averaged to counter the effect of noise :

$$\begin{aligned}
I_x(r,c) \approx \tfrac{1}{4}\textstyle\sum_{\Delta r=0}^{1} \quad & \{\ I(r+\Delta r, c+1, 0) - I(r+\Delta r, c, 0) \\
& +I(r+\Delta r, c+1, 1) - I(r+\Delta r, c, 1)\ \} \\
I_y(r,c) \approx \tfrac{1}{4}\textstyle\sum_{\Delta c=0}^{1} \quad & \{\ I(r+1, c+\Delta c, 0) - I(r, c+\Delta c, 0) \\
& +I(r+1, c+\Delta c, 1) - I(r, c+\Delta c, 1)\ \} \\
I_t(r,c) \approx \tfrac{1}{4}\textstyle\sum_{\Delta r=0}^{1}\sum_{\Delta c=0}^{1} \quad & \{I(r+\Delta r, c+\Delta c, 1) - I(r+\Delta r, c+\Delta c, 0)\}
\end{aligned} \tag{4.15}$$

where I_0 is the current image and I_1 the next. The spatial derivatives have sometimes been estimated using averages of central differences.

Global averaging of the finite difference approximations can be done using L^2 smoothing of the derivatives finite differences : a derivative estimate g is computed by minimizing :

$$E(g) = \frac{1}{2}\int_\Omega \left((g-g_0)^2 + \gamma\|\nabla g\|^2\right) dxdy, \tag{4.16}$$

where g is a partial derivative function and g_0 its finite difference approximation from I. The corresponding Euler-Lagrange equation $g - g_0 - \gamma\nabla^2 g = 0$ is then discretized to yield a large sparse system of linear of equations.

4.4.2 Regularized differentiation

Although image data smoothing is commonly done in motion analysis, it does not generally solve the derivative estimation ill-posedness, so that prior de-noising of the image independently of differentiation followed by finite difference approximation is not generally effective [42]. A more productive method is to state differentiation within

Tikhonov regularization theory for ill-posed problems [43,44]. In [44] the problem was to find a smooth approximation of the true derivative y' of a function y from given data $\tilde{y}_i$. This was done by determining an approximation f of y which minimizes an objective functional having a term of discrepancy between f and the given data, and a regularization term to penalize the L^2 norm of f''. The derivative was subsequently evaluated on f. The objective functional in [44] was investigated in earlier studies [45, 46] which showed that it is minimized by a natural cubic spline. In [43] , the differentiation process itself was regularized : the formulation sought to determine an approximation u of the true derivative which minimized a functional containing a data fidelity term via a Fredholm integral of anti-differentiation, and penalty term via the L^2 norm of u and u'. More recently, [42] investigated total variation (TV) regularization in conjunction with an anti-differentiation data discrepancy term as in [43]. The discrete implementation of the ensuing problem followed a standard numerical scheme in TV restoration [47].

In the following we will estimate the derivatives I_x and I_y by a variational method which uses an anti-differentiation data discrepancy term as in [42, 43] and an L^2 smoothness regularization. Enforcing smoothness on the derivatives is consistent with the L^2 regularization in the scene flow estimation scheme we have described. For reasons that will become clearer later, the formulation does not apply to I_t given that the time axis is sampled only at two points ; recall that we are to estimate the derivatives from two consecutive images. Instead, I_t can be estimated by regularized forward differences or simply by the Horn and Schunck formulas.

We will describe the method for I_x. The derivative I_y can be treated by the same formulas using the transposed image. Consider I_x at some fixed time t, so that it is viewed as a function of the image spatial coordinates but not of time. For convenience, we will also drop time from the coordinates of I. Let $\Omega = [0, l] \times [0, l]$. The partial

derivative I_x will be computed as the minimizer of the following functional :

$$E(g) = \frac{1}{2}\int_{\Omega} \left(\|Ag - I\|^2 + \lambda \|\nabla g\|^2 \right) dxdy \tag{4.17}$$

where ∇ is the spatial gradient, λ is a positive constant and A is the integral operator of anti-differentiation defined by :

$$Ag(x,y) = \int_0^x g(z,y)dz \tag{4.18}$$

The Euler-Lagrange equation corresponding to (4.17) is :

$$A^*(Ag - I) - \lambda \nabla^2 g = 0 \tag{4.19}$$

where A^* is the adjoint operator of A, defined by :

$$A^*g(x,y) = \int_x^l g(z,y)dz \tag{4.20}$$

Here following is a discretization of Eq. (4.19) leading to a large scale sparse system of linear equations. As before, let the points of the discretization grid D be listed top-down and left to right. The image in this lexicographical order is $I \in \mathbf{R}^N$, where $N = n^2$ for an image of size $n \times n$. Let $g_i, i = 1, ..., N$, be g evaluated at grid point i and $\mathbf{g} \in \mathbf{R}^N$ the corresponding vector. For simplicity, we will use the same symbol to designate the linear operators A and A^* in (4.19) and their corresponding discretization matrix. Using the composite trapezoid quadrature rule for integral approximation (with one-pixel data spacing) [40], the $N \times N$ matrix A is defined by :

$$\begin{aligned}
&A(kn+i, kn+1) = \tfrac{1}{2}; \quad i = 2, ..., n; \;\; k = 0, ..., n-1 \\
&A(kn+i, kn+i) = \tfrac{1}{2}; \quad i = 2, ..., n; \;\; k = 0, ..., n-1 \\
&A(kn+i, kn+i-j) = 1; \quad i = 3, ..., n; \;\; j = 1, ..., i-2, \;\; k = 0, ..., n-1,
\end{aligned}$$

and all of the other elements are zero. The elements of rows $kn+1; \;\; k = 0, ..., n-1$ are zero to reflect the integral in (4.18) when $x = 0$. Matrix A is block diagonal sparse,

with blocks of size $n \times n$. The $N \times N$ matrix A^* is similarly defined :

$$A^*(i,i) = \tfrac{1}{2}; \quad i \in [1, n^2],\ i \neq kn,\ k = 1, ..., n$$
$$A^*(kn+i, (k+1)n) = \tfrac{1}{2}; \quad i = 1, ..., n-1;\ \ k = 0, ..., n-1$$
$$A^*(kn+i, kn+i+j) = 1; \quad i = 1, ..., n-1;\ \ j = 1, ..., n-i-1,\ \ k = 0, ..., n-1,$$

and all the other elements are zero. The elements of rows $kn,\ k = 1, ..., n$ are zero to reflect the integral in (4.20) when $x = l$. Matrix A^* is block diagonal sparse, with blocks of size $n \times n$. The Laplacian term in Eq. (4.19) can be discretized as $\lambda \sum_{j \in \mathbf{N}_i} (g_j - g_i)$, where the factor of the approximation is absorbed by λ, and $\mathbf{N}_i$ is the set of indices of the neighbors of i. The corresponding matrix is defined by :

$$L(i,i) = -\lambda n_i; \quad i = 1, ..., N$$
$$L(i,j) = \lambda; \quad j \in \mathbf{N}_i,$$

where $n_i = card(\mathbf{N}_i)$. The system of linear equations to solve is :

$$(A^*A - L)\,\mathbf{g} = A^*I \tag{4.21}$$

This large scale sparse system of linear equations can be solved efficiently by an iterative method such Gauss-Seidel.

4.4.3 Example

Here following is an example. It uses the noised synthetic pyramidal image of Figure 4.2. The derivatives computed using regularized differentiation and the Horn and Schunck averaging are shown graphically in the figure. The values computed by regularized differentiation are closer to the true values : the mean squared error between true and computed values are 0.0409 for the regularized values and 1.086 for the Horn and Schunck averaging. Derivatives are measured in grey levels ([0 255] range) per pixel. The value of λ is 5.0 and the SNR is 0.5.

4.5 Experimental results

This section presents various experiments on synthetic and real sequences to verify the validity of the method and its implementation. We show the recovered depth using anaglyphs (red/cyan) and color-coded displays, and novel viewpoint images. Color-coded depth is a standard display style. Anaglyphs are a convenient means for the subjective appraisal of the computed object structure. They are constructed from one of the two input images used in the experiment and the recovered depth map. Anaglyphs are best perceived on good-quality photographic paper. When viewed on standard screens, they are generally better perceived with full color resolution. Finally, we also show a novel viewpoint image, i.e., a picture of the reconstructed object as viewed from a viewpoint different from the one of either of the two input images.

For scene flow, we show a vector display of its projection from some viewpoint. Also, and since we have no ground truth of scene flow for the used sequences, we show the optical flow corresponding to it, compared to the optical flow computed directly by the Horn and Schunck algorithm. We provide also a comparison to the optical flow ground truth using three kinds of error : average angular error (aae), standard angular error (stae) and endpoint error (epe). This is a good indirect way to evaluate scene flow computation results because the behavior of the Horn and Schunck method is a generally well understood benchmark.

The formulation parameters were determined empirically. Distances are measured in pixels ; the fronto-parallel plane position Z_0 has been fixed to 6×10^4 pixels. The camera focal length f has been approximated to 600 pixels [20] ; the initial value of scene flow and depth at each point are, respectively, 0 and Z_0. Coefficients α and β are given in the caption of each figure. Regularized differentiation's coefficient λ is fixed to 1 in all the examples.

In general, all of the proof-of-concept examples we show in the following support

the validity of the scheme and its implementation. In the examples discussed below, one can make the following observations/conclusions :

— In all the examples, the scene flow and induced optical flow are consistent with the actual motion of the objects.

— The color-coded depth display is in line with the structure of the objects (i.e., the relative depth of object surfaces).

— The obtained optical flow is in keeping with the output of the well tested/researched benchmark algorithm of Horn and Schunck. It is worth noting here that the velocities we obtained are less noisy than those computed with Horn-and-Schunck algorithm. This can be explained by the fact that our method benefited from the use of (i) 3D information and (ii) better estimates of the image spatiotemporal derivatives via regularized differentiation.

— In all examples, the corresponding anaglyphs offered viewers a strong sense of depth.

A. Synthetic squares sequence : This is a sequence of two consecutive images of two overlapping squares moving against a moving background, to evaluate quantitatively the computed scene flow. This evaluation is done via the image motion that it induces. This induced motion will be compared to the actual image motion and the motion computed by the Horn and Schunck algorithm. The actual image motions are, in pixels : $(-1, -1)$ for the upper square, $(1, 1)$ for the lower, and $(0, -1)$ for the background. Noise has been added independently in the first and second image. Noise values are from a discretized, shifted, truncated Gaussian in the interval between 0 and 100 gray levels, within the overall range 0 to 255 of the image. The first of the two images is shown in the leftmost display of Fig. 4.3 ; the vector-coded ground truth and the computed image motion are displayed in the second and third images, respectively ; the results with the Horn and Schunck method are shown in the rightmost image. In general, such vector displays are meant to reassure that the image motion is

visually consistent with its expected overall appearance. Quantitatively, the average angular error for the image motion induced by the computed scene flow is 15^0 and the average error on the length is 0.4 pixel. The Horn and Schunck algorithm was overwhelmed by the image noise; its average angular error is 42^0 and the average error on the length is 1 pixel. The proposed scheme has performed better than the Horn and Schunck algorithm because it used subsuming higher level 3D information, from which image motion can be recovered point-wise according to model Eq. 4.2, as well as a better estimate of the image spatiotemporal derivatives via regularized differentiation.

B. Marbled block sequence : In this example, we use the *Marbled-block* synthetic sequence from the database of KOGS/IAKS Laboratory, Germany. There are three blocks in this sequence, two of which are moving. The rightmost block moves in depth to the left and the one in the middle moves forward to the left. There are aspects which make 3D interpretation challenging : the blocks have a macro texture of weak spatiotemporal intensity variations within the textons and similar to the texture of the floor. As a result, the occluding boundaries of the blocks are ill defined at places. The blocks also cast shadows which move. Results are shown in Fig. 4.4. The first row depicts (from left to right) : An anaglyph of the structure reconstructed from the method's output and the first frame of the input image sequence; a color-coded display of the recovered depth along with the used color palette [1]; and novel viewpoint images of the two moving blocks. Second row : a view of the scene flow vectors; optical flow corresponding to the estimated scene flow (4.2); the optical flow computed directly by the Horn and Schunck algorithm.

C. Cylinder and boxes sequence : This second example uses a real image sequence (courtesy of Debrunner and Ahuja [48]), shown in Fig. 4.5. This sequence depicts

1. We used the same color palette for all examples, with depth increasing from right (red) to left (purple)

three moving objects : A box moving to the right at an image rate of about 0.30 pixel per frame ; a cylindrical surface rotating about a vertical axis at a velocity of one degree per frame, and moving laterally to the right at an image rate of about 0.15 pixel per frame and, finally, a flat background moving to the right (parallel to the box motion) at approximately 0.15 pixel per frame. In this example, the 3D interpretation and recovery is hard because of its unhelpful 3D motion. Results are displayed in Fig. 4.5 : First row from left to right : An anaglyph of the structure reconstructed from the method's output and the first frame of the input image sequence ; a color-coded display of the recovered depth ; novel viewpoint images the cylindrical surface and the box. Second row : a view of the scene flow vectors ; optical flow corresponding to the estimated scene flow (4.2) ; the optical flow computed by the Horn and Schunck algorithm.

D. Berber sequence : This example uses the *Berber* real sequence. The figurine rotates about a nearly vertical axis and moves forward to the left in a static environment. Fig. 4.6 displays the results : First row from left to right : An anaglyph of the structure reconstructed from the method's output and the first frame of the input image sequence ; a color-coded display of the recovered depth ; novel viewpoint images of the figurine. Second row : a view of the scene flow vectors ; optical flow corresponding to the estimated scene flow (4.2) ; the optical flow computed by the Horn and Schunck algorithm.

E. Pharaohs sequence : This example uses the *Pharaohs* real image sequence. There are two moving figurines in a static environment ; the leftmost translates left and forward ; the rightmost rotates about a nearly vertical axis to the right. Results are shown in Fig. 4.7 : First row from left to right : An anaglyph of the structure reconstructed from the method's output and the first frame of the input image sequence ; a color-coded display of the recovered depth ; novel viewpoint images of the figurines. Second row : a view of the scene flow vectors ; optical flow corresponding

to the estimated scene flow (4.2); optical flow computed by the Horn and Schunck algorithm.

The results for scene flow are shown in Table 4.1 for each of the examples described above.

TABLE 4.1 – Average angular error (aae), standard angular error (stae), and end-point error (epe) for the optical flow corresponding to the estimated scene flow using regularized differentiation (RD) vs. optical flow computed directly by the Horn and Schunck algorithm (HS). Coefficient λ was fixed equal to 1 for all the examples.

	Errors	RD	HS
Marbled-block	aae	4.14	4.09
	stae	8.56	8.73
	epe	0.10	0.08
Cylinder	aae	18.68	15.72
	stae	18.91	18.42
	epe	0.78	0.68
Berber	aae	11.61	10.18
	stae	10.16	10.50
	epe	0.41	0.35
Pharaohs	aae	20.01	27.87
	stae	19.21	27.83
	epe	0.62	0.82

4.6 Conclusion and discussion

The goal of this study was concurrent recovery of scene flow and depth from a monocular image sequence. We developed a variational method which minimizes a functional containing a data term of joint scene flow and depth conformity to the image sequence spatiotemporal variations, and quadratic smoothness regularization terms. The data term follows re-writing optical flow as a function of scene flow and depth in the classical optical flow gradient constraint of Horn and Schunck. As a result, the formulation is analogous to the classical Horn and Schunck optical flow estimation method, except that it involves the variables of scene flow and depth rather than image motion. Monocular processing is a unique feature of this scheme because previous scene flow recovery schemes have used binocular image sequences rather than a single image stream as in this study.

Another characteristic is the occurrence of both depth and scene flow as unknowns in the equations used to state the problem. The variational paradigm fitted naturally with these equations to give a single optimization formulation, free from the intervention of outside processes since all the relevant variables, namely depth and scene flow coordinates, occur simultaneously. As a result, the formulation translates into a tractable algorithm whose behavior can be explained. This algorithm follows the discretization of the objective functional Euler-Lagrange equations, giving a large scale sparse system of linear equations in the unknowns of depth and the three scene flow coordinates. The equations can be ordered in such a way that its matrix is symmetric positive definite such that they can be solved efficiently by Gauss-Seidel iterations.

The focus of this study being on the formulation proper, it was sufficient to use Gauss-Seidel iterations in the proof-of-concepts examples we described in the experimental section. However, one can explore other schemes for more efficient numerical resolution as the literature on large sparse systems of linear equations is quite vast. For

instance, one can investigate [38] classical convergence acceleration of the Gauss-Seidel by successive over-relaxation, or an iterative scheme designed for positive definite systems such as the conjugate gradient algorithm. The sequential subspace correction (SSC) method [49], which would process the minimization in the four independent linear subspaces of depth and scene flow coordinates sequentially can also prove to be quite efficient ; for each subspace, the Gauss-Seidel iterations can be used. The SSC can be parallelized. There is also a rich literature on efficient modern Krylov subspace methods where a matrix need only be specified as a matrix-vector operator [50].

This study used Tikhonov regularization for scene flow and depth. It did not affect the purpose of formulating monocular recovery of these variables. However, quadratic regularization smooths the variables recovered at and in the proximity of their discontinuities, namely sharp changes in depth and motion boundaries. There are several ways of specifying boundary preserving recovery [1]. For instance, one can use the Aubert et al. function, in place of the quadratic function, or simply the L^1 norm. In addition to preserving discontinuities, the L^1 norm can be approximated in practice for faster computation without affecting accuracy in a noticeable way. Both motion and depth discontinuities can also be preserved by concurrent motion computation and segmentation [22].

The examples we gave are for proof of concept only. They show that the formulation is sound, correctly implemented, but has obvious limitations such as boundary blurring interpretation and use of approximate camera parameters. Nevertheless, the results clearly indicate that the method is worthy of further investigation. We are currently extending it to account for motion and depth discontinuities via L^1 regularization. We are also investigating joint motion segmentation and estimation, an extension to scene flow and depth of the scheme in [22]. Experimental validation must be based on a larger database of three-dimensional moving objects of various geometry that would test various difficulties such as motion and depth discontinuities,

motion of large extent, and image noise and resolution in common practical settings. In particular, quantitative validation will require computer graphics generation of appropriate synthetic objects in motion for which ground truth scene flow can be calculated.

Figures

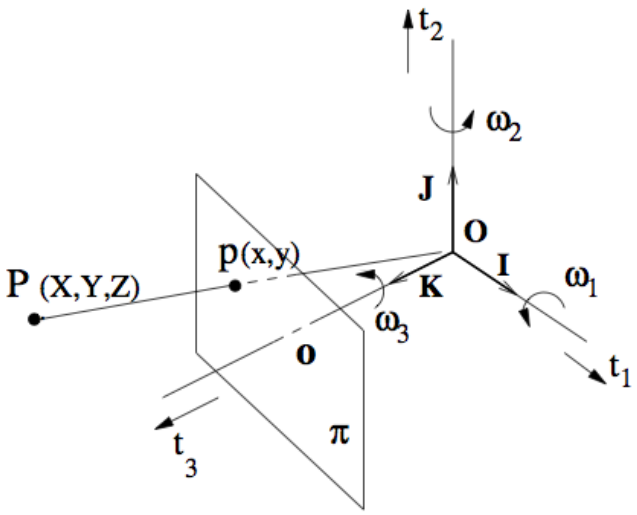


FIGURE 4.1 – The viewing system is symbolized by a Cartesian reference system $(\mathbf{O}; X, Y, Z)$ and central projection through the origin. The $Z-$axis is the depth axis. The image plane π is orthogonal to the Z-axis at distance f, the focal length, from $\mathbf{O}$.

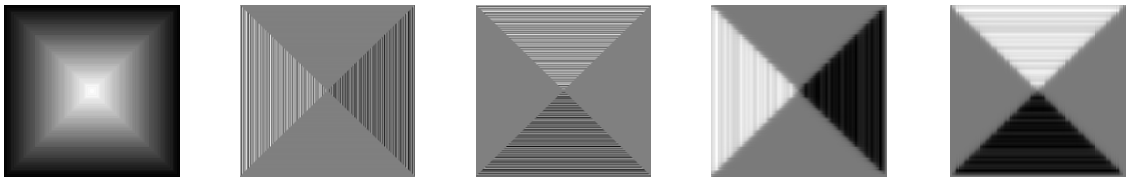


FIGURE 4.2 – From the left to the right : The noised 2D pyramidal image (SNR=0.5) ; the partial derivatives I_x and I_y using Horn and Schunck averaging of forward image differencing ; the partial derivative I_x and I_y using regularized differencing ($\lambda = 5.0$).

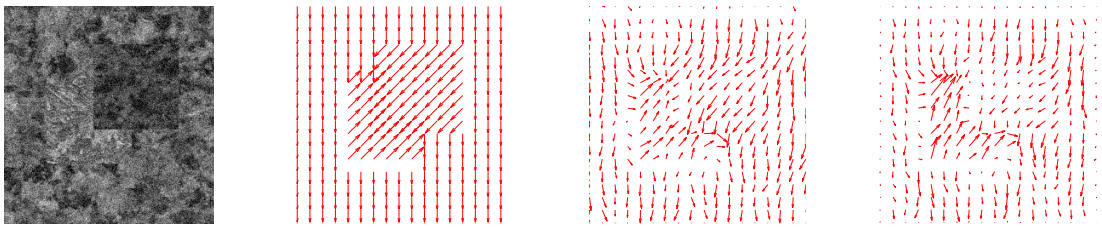


FIGURE 4.3 – *Synthetic squares* sequence. From left to right : The first of the two images ; the vector-coded ground truth ; optical flow corresponding to the estimated scene flow ; optical flow computed directly by the Horn and Schunck method.

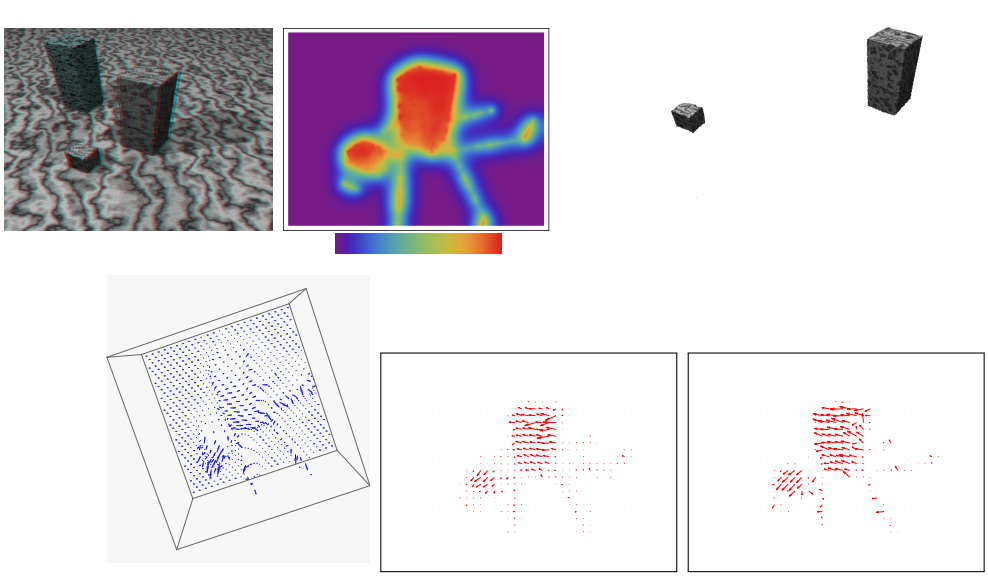


FIGURE 4.4 – *Marbled blocks* sequence results (better perceived when figures are enlarged on screen). Parameters : $\alpha = 6 \times 10^7$ and $\beta = 10^2$. First row from left to right : An anaglyph of the structure reconstructed from the method's output and the first frame of the input image sequence ; a color-coded display of the recovered depth along with the used color palette, with depth increasing from right (red) to left (purple) ; novel viewpoint images of the two moving blocks. Second row : a view of the scene flow vectors ; optical flow corresponding to the estimated scene flow (4.2) ; the optical flow computed directly by the Horn and Schunck algorithm.

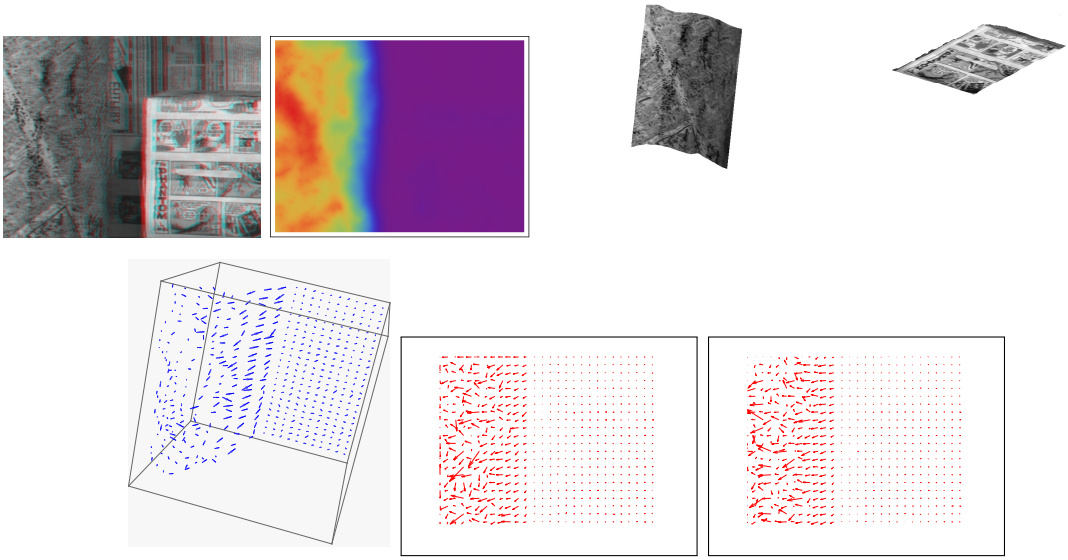


FIGURE 4.5 – *Cylinder and box* sequence results (better perceived when figures are enlarged on screen). Parameters : $\alpha = 6 \times 10^6$ and $\beta = 10^4$.First row from left to right : An anaglyph of the structure reconstructed from the method's output and the first frame of the input image sequence ; a color-coded display of the recovered depth ; novel viewpoint images the cylindrical surface and the box. Second row : a view of the scene flow vectors ; optical flow corresponding to the estimated scene flow (4.2) ; the optical flow computed by the Horn and Schunck algorithm.

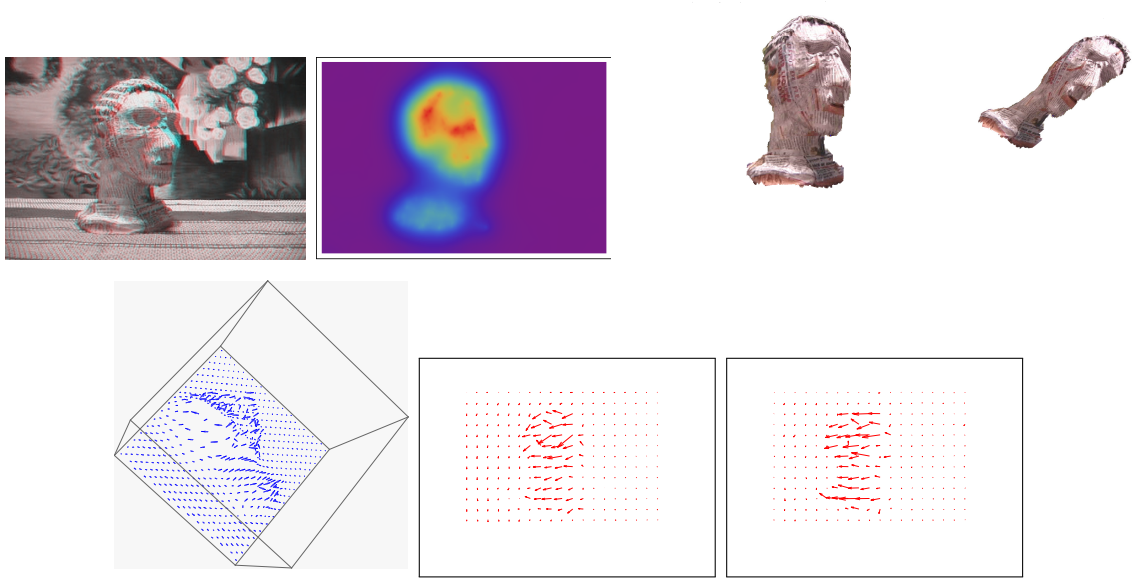


FIGURE 4.6 – *Berber* figurine sequence results (better perceived when figures are enlarged on the screen). Parameters : $\alpha = 6 \times 10^7$; $\beta = 5 \times 10^4$. First row from left to right : An anaglyph of the structure reconstructed from the method's output and the first frame of the input image sequence ; a color-coded display of the recovered depth ; novel viewpoint images of the figurine. Second row : a view of the scene flow vectors ; optical flow corresponding to the estimated scene flow (4.2) ; the optical flow computed by the Horn and Schunck algorithm.

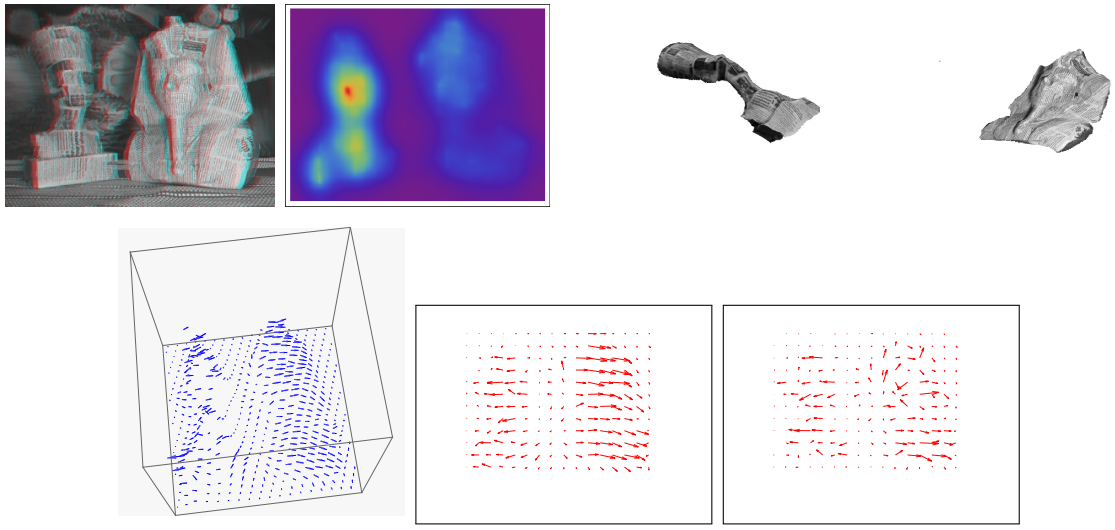


FIGURE 4.7 – *Pharaohs* figurines sequence (better perceived when figures are enlarged on screen). Parameters : $\alpha = 6 \times 10^7$; $\beta = \times 10^2$. First row from left to right : An anaglyph of the structure reconstructed from the method's output and the first frame of the input image sequence ; a color-coded display of the recovered depth ; novel viewpoint images of the figurines. Second row : a view of the scene flow vectors ; optical flow corresponding to the estimated scene flow (4.2) ; optical flow computed by the Horn and Schunck algorithm.

Bibliographie

[1] A. Mitiche and J. Aggarwal, *Computer vision analysis of image motion by variational methods*. Springer, 2013.

[2] H. Longuet-Higgins and K. Prazdny, "The interpretation of a moving retinal image," *Proceedings of the Royal Society of London, B*, vol. 208, pp. 385–397, 1981.

[3] A. R. Bruss and B. K. P. Horn, "Passive navigation," *Computer Vision, Graphics, and Image Processing*, vol. 21, no. 1, pp. 3–20, 1983.

[4] G. Adiv, "Determining three-dimensional motion and structure from optical flow generated by several moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, no. 4, pp. 384–401, 1985.

[5] B. Shahraray and M. Brown, "Robust depth estimation from optical flow," in *International Conference on Computer Vision, ICCV*, 1988, pp. 641–650.

[6] B. Horn and E. Weldon, "Direct methods for recovering motion," *International Journal of Computer Vision*, vol. 2, no. 2, 1988.

[7] D. Heeger and A. Jepson, "Subspace methods for recovering rigid motion I : algorithm and implementation," *International Journal of Computer Vision*, vol. 7, no. 2, 1992.

[8] M. Taalebinezhaad, "Direct recovery of motion and shape in the general case by fixation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 8, 1992.

[9] E. De Micheli and F. Giachero, "Motion and structure from one dimensional optical flow," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, Jun 1994, pp. 962–965.

[10] N. Gupta and N. Kanal, "3-D motion estimation from motion field," *Artificial Intelligence,*, vol. 78, 1995.

[11] Y. Xiong and S. Shafer, "Dense structure from a dense optical flow," Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-95-10, April 1995.

[12] Y. Hung and H. Ho, "A Kalman filter approach to direct depth estimation incorporating surface structure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 6, 1999.

[13] S. Srinivasan, "Extracting structure from optical flow using the fast error search technique," *International Journal of Computer Vision*, vol. 37, no. 3, 2000.

[14] T. Brodsky, C. Fermuller, and Y. Aloimonos, "Structure from motion : Beyond the epipolar constraint," *International Journal of Computer Vision*, vol. 37, no. 3, 2000.

[15] H. Liu, R. Chellappa, and A. Rosenfeld, "A hierarchical approach for obtaining structure from two-frame optical flow," in *Proceedings of the Workshop on Motion and Video Computing*, ser. MOTION '02. Washington, DC, USA : IEEE Computer Society, 2002, pp. 214–219.

[16] W. J. MacLean, A. D. Jepson, and R. C. Frecker, "Recovery of egomotion and segmentation of independent object motion using the em algorithm." in *British Machine Vision Conference, BMVC*, E. R. Hancock, Ed. BMVA Press, 1994, pp. 1–10.

[17] S. Fejes and L. S. Davis, "What can projections of flow fields tell us about visual motion." in *ICCV*, 1998, pp. 979–986.

[18] J. Weber and J. Malik, "Rigid body segmentation and shape description from dense optical flow under weak perspective," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, 1997.

[19] A. Mitiche and S. Hadjres, "Mdl estimation of a dense map of relative depth and 3D motion from a temporal sequence of images," *Pattern Analysis and Applications,*, vol. 6, 2003.

[20] H. Sekkati and A. Mitiche, "A variational method for the recovery of dense 3D structure from motion," *Robotics and Autonomous Systems*, vol. 55, no. 7, 2007.

[21] ——, "Concurrent 3D-motion segmentation and 3D interpretation of temporal sequences of monocular images," *IEEE Transactions on Image Processing*, vol. 15, no. 3, 2006.

[22] A. Mitiche and H. Sekkati, "Optical flow 3D segmentation and interpretation : A variational method with active curve evolution and level sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1818–1829, Nov. 2006.

[23] H. Sekkati and A. Mitiche, "Joint optical flow estimation, segmentation, and 3D interpretation with level sets," *Computer Vision and Image Understanding*, vol. 103, no. 2, 2006.

[24] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade, "Three-dimensional scene flow," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 475–480, 2005.

[25] J.-P. Pons, R. Keriven, O. Faugeras, and G. Hermosillo, "Variational stereovision and 3d scene flow estimation with statistical similarity measures," in *IEEE International Conference On Computer Vision (ICCV)*, 2003, pp. 597–602.

[26] Y. Zhang and C. Kambhamettu, "Integrated 3d scene flow and structure recovery from multiview image sequences," in *IEEE Conference on Computer Vision and*

Pattern Recognition (CVPR), vol. 2, 2000, pp. 674–681.

[27] F. Huguet and F. Devernay, "A variational method for scene flow estimation from stereo sequences," in *IEEE International Conference on Computer Vision (ICCV)*, 2007, pp. 1–7.

[28] A. Wedel, C. Rabe, T. Vaudrey, T. Brox, U. Franke, and D. Cremers, "Efficient dense scene flow from sparse or dense stereo data," in *European Conference on Computer Vision (ECCV)*, vol. 1, 2008, pp. 739–751.

[29] C. Rabe, T. Müller, A. Wedel, and U. Franke, "Dense, Robust, and Accurate Motion Field Estimation from Stereo Image Sequences in Real-Time," in *Proceedings of the 11th European Conference on Computer Vision*, ser. Lecture Notes in Computer Science, K. Daniilidis, P. Maragos, and N. Paragios, Eds., vol. 6314. Springer, September 2010, pp. 582–595.

[30] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, and D. Cremers, "Stereoscopic scene flow computation for 3d motion understanding," *International Journal of Computer Vision*, vol. 95, no. 1, pp. 29–51, 2011.

[31] C. Vogel, K. Schindler, and S. Roth, "Piecewise rigid scene flow," in *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[32] T. Basha, Y. Moses, and N. Kiryati, "Multi-view scene flow estimation : A view centered variational approach." *International Journal of Computer Vision*, vol. 101, no. 1, pp. 6–21, 2013.

[33] R. Deriche, P. Kornprobst, and G. Aubert, *Optical-flow estimation while preserving its discontinuities : A variational approach.* Berlin, Heidelberg : Springer Berlin Heidelberg, 1996, pp. 69–80.

[34] G. Aubert, R. Deriche, and P. Kornprobst, "Computing optical flow via variational thechniques," *SIAM Journal of Applied Mathematics*, vol. 60, no. 1, 1999.

[35] B. Horn and B. Schunk, "Determining optical flow," *Artificial Intelligence*,

vol. 17, no. 17, pp. 185–203, 1981.

[36] E. C. Hildreth, "The measurement of visual motion," Ph.D. dissertation, Cambridge, Mass., London, 1984, th. Ph. D. : Massachusetts Institute of technology : 1983.

[37] S. Ullman, "Computational studies in the interpretation of structure and motion : Summary and extension," MIT, Tech. Rep., 1983.

[38] P. Ciarlet, *Introduction a l'analyse numérique matricielle et a l'optimisation*, 5th ed. Masson, 1994.

[39] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, 3rd ed., ser. Texts in Applied Mathematics ; 12. New York : Springer, 2002.

[40] G. E. Forsythe, M. A. Malcolm, and C. B. Moler, *Computer methods for mathematical computations*, ser. Prentice-Hall series in automatic computation. Englewood Cliffs (N.J.) : Prentice-Hall, 1977.

[41] D. Terzopoulos, "Regularization of inverse visual problems involving discontinuities," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, no. 4, pp. 413–424, July 1986.

[42] R. Chartrand, "Numerical differentiation of noisy, nonsmooth data." in *Los Alamos National Laboratory TR*, 2005.

[43] J. Cullum, "Numerical differentiation and regularization," *SIAM Journal on Numerical Analysis*, vol. 8, no. 2, pp. 254–265, 1971.

[44] M. Hanke and O. Scherzer, "Inverse problems light : numerical differentiation," *The American Mathematical Monthly*, vol. 108, no. 6, pp. 512–521, 2001.

[45] C. H. Reinsch, "Smoothing by spline functions," *Numerische Mathematik*, vol. 10, No.3, pp. 177–183, 1967.

[46] I. J. Schoenberg, "Spline functions and the problem of graduation," *Proceedings of the National Academy of Science*, vol. 54, N0.2, pp. 947–950, 1964.

[47] C. R. Vogel, *Computational methods for inverse problems.* SIAM Frontiers in Applied Mathematics, 2002.

[48] C. Debrunner and N. Ahuja, "Segmentation and factorization-based motion and structure estimation for long image sequences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 2, pp. 206–211, Feb. 1998.

[49] W. Hackbusch, *Iterative solution of large sparse systems of equations.* Springer Science & Business Media, 2012, vol. 95.

[50] V. Simoncini and D. B. Szyld, "Recent computational developments in Krylov subspace methods for linear systems," *Numerical Linear Algebra with Applications*, vol. 14, pp. 1–59, 2007.

DOI : 10.1007/978-3-319-45886-1_29

Chapitre 5

Boundary preserving variational image differentiation

Yosra Mathlouthi, Amar Mitiche, and Ismail Ben Ayed

German Conference on Pattern Recognition (GCPR), LNCS 9796, pp. 355-364, 2016, Springer.

Résumé : Le but de cette étude est l'analyse de la différentiation d'image par une méthode variationnelle qui préserve les frontières. La méthode minimise une fonctionnelle composée d'un terme d'adéquation des données utilisant un opérateur d'anti-différentiation et un terme de régularisation L^1. Pour chaque dérivée partielle de l'image, l'opérateur d'anti-différentiation contraint la dérivée à une fonction qui redonne l'image à une constante additive prés lorsqu'on l'intègre, tandis que le terme de régularisation contraint la dérivée à une fonction lisse partout dans l'image, sauf sur les frontières des objets dans l'image. La discrétisation des équations d'Euler-Lagrange du problème produit un système à grande échelle d'équations non-linéaires. Ce système est creux et presque linéaire, ce qui permet une solution efficace par des approximations linéaires successives. La méthode

Chapitre 6

Monocular, boundary preserving joint recovery of scene flow and depth

Yosra Mathlouthi, Amar Mitiche, and Ismail Ben Ayed

Accepté à *Frontiers in ICT,* 2016.

Résumé : Une méthode d'estimation variationnelle et conjointe du flot de scène et de la profondeur à partir d'une séquence d'images monoculaire, plutôt qu' à partir d'une séquence d'images stéréoscopique obligatoire comme c'est le cas des méthodes dans la littérature a été étudiée dans [1]. Cette méthode utilisait une fonction intégrale avec un terme de conformité du flot de scène et de la profondeur aux variations spatiotemporelles de la séquence d'images et un terme de régularisation L^2 pour obtenir un champ de profondeur et un flot de scène lisses. Le schéma résultant était analogue à la méthode de l'estimation du flot optique de Horn et Schunck, quoique les inconnues étaient la profondeur et le flot de scène au lieu du flot optique. Plusieurs exemples ont été exposés pour

montrer que cette méthode peut récupérer, avec une bonne précision, la profondeur et le mouvement sauf sur leurs frontières à cause de l'utilisation de la régularisation L^2 qui n'identifie pas les discontinuités et lisse partout sans discrimination. La méthode que nous étudions dans cet article généralise la formulation de [1] avec une régularisation L^1 afin de calculer des estimées de profondeur et de flot de scène avec des frontières préservées. Les dérivées d'image, qui apparaissent sous forme de données dans la fonctionnelle, sont aussi calculées à partir de la séquence d'images enregistrée par une méthode variationnelle qui utilise une régularisation L^1 afin de préserver leurs discontinuités. Bien que la régularisation L^1 conduit à des équations d'Euler-Lagrange non-linéaires qui sont nécessaires pour la minimisation de la fonctionnelle, ces équations peuvent être résolues d'une manière efficace. Les avantages de la généralisation, qui se manifestent à des estimations plus précises de la profondeur et du flot de scène, sont mis en évidence dans l'expérimentation qui utilise des images réelles et synthétiques et qui présente et compare les résultats de la régularisation L^1 par rapport à L^2 pour l'estimation de la profondeur et du mouvement, ainsi que les résultats de l'utilisation d'une régularisation L^1 plutôt que L^2 pour le calcul des dérivées de l'image.

Abstract

Variational joint recovery of scene flow and depth from a single image sequence, rather than from a stereo sequence as others required, was investigated in [1] using an integral functional with a term of conformity of scene flow and depth to the image sequence spatiotemporal variations, and L^2 regularization terms for smooth depth field and scene flow. The resulting scheme was analogous to the Horn and Schunck

optical flow estimation method except that the unknowns were depth and scene flow rather than optical flow. Several examples were given to show the basic potency of the method : It was able to recover good depth and motion, except at their boundaries because L^2 regularization is blind to discontinuities which it smooths indiscriminately. The method we study in this paper generalizes to L^1 regularization the formulation of [1] so that it computes boundary preserving estimates of both depth and scene flow. The image derivatives, which appear as data in the functional, are computed from the recorded image sequence also by a variational method which uses L^1 regularization to preserve their discontinuities. Although L^1 regularization yields nonlinear Euler-Lagrange equations for the minimization of the objective functional, these can be solved efficiently. The advantages of the generalization, namely sharper computed depth and three-dimensional motion, are put in evidence in experimentation with real and synthetic images which shows the results of L^1 versus L^2 regularization of depth and motion, as well as the results using L^1 rather than L^2 regularization of image derivatives.

6.1 Introduction

Scene flow is the three-dimensional (3D) motion field of the visible environmental surfaces projected on the image domain : at each image point, scene flow is the 3D velocity of the corresponding environmental surface point. It is the time derivative of 3D position. As such, it is a function of both depth and optical flow [2]. Scene flow computation has been the focus of several recent studies [3–8]. It is a typical inverse problem best stated by variational formulations [3]. Basic variational statements use L^2 (Tikhonov) regularization. This regularization, which imposes smoothness on the solution, yields linear terms in the Euler-Lagrange equations. This is the case with the scheme described in [1], and of which we describe a generalization in this paper. The

scheme improved significantly on others because it needed a single image sequence rather than a stereo stream like other methods. Also, it formulated the problem using an integral functional of two terms : a data fidelity term to constrain the computed scene flow to conform to the image sequence spatiotemporal derivatives, and L^2 regularization terms to constrain the computed scene flow and depth to be smooth, which led to linear Euler-Lagrange equations for the minimization of the objective functional, much like in the Horn and Schunck optical flow formulation [9], except that it involved depth and scene flow rather than optical flow. However, L^2 regularization blurs the computed scene flow and depth boundaries because it imposes their smoothness everywhere. In general, some form of boundary preserving regularization is necessary. This is particularly true when environmental objects move independently relative to the viewing system. The variation of motion and structure in such a case can be sharp and significant at these objects occluding boundaries and, therefore, accuracy would require that they be preserved by the regularization operator.

Boundary preserving recovery of scene flow and depth can be specified in various ways [3]. For example, instead of the L^2 norm, one can use the Aubert et al. function [10, 11], or the L^1 norm. We can also inhibit smoothing across boundaries by joint 3D motion estimation and segmentation [12]. In this follow-up study of our previous investigation [1], we apply the L^1 regularization. There are three basic reasons for this choice : (1) the ability of the L^1 constraint to preserve sharp boundaries while penalizing oscillations. This is well suited for "blocky images" which tend to be smooth inside regions which have sharp significant boundaries, as is, in general, typical of motion fields, (2) in practice, it can be implemented by computationally efficient approximations without affecting the accuracy of results in a noticeable way and, (3) there is a significant body of literature in its support, particularly in image restoration [13].

Recovery of scene flow and depth uses the image sequence spatiotemporal deri-

vatives. The study in [1] computed derivatives by a variational formulation which used an anti-differentiation data fidelity term, and L^2 regularization. The anti-differentiation term expressed the fact that an image derivative is a function which when integrated gives the image. The present study uses the same anti-differentiation data fidelity term, but substitutes L^1 regularization for L^2 in order, first, to be consistent with the use of L^1 regularization of scene flow and depth and, second, to have the evaluation of the derivatives account for their discontinuities.

We conducted several experiments with real and synthetic data to verify the validity and efficiency of the scheme. We show comparative results that put in evidence the improvements one can obtain by using L^1 rather than L^2 regularization of depth and motion, as well as by evaluating the image derivatives with L^1 rather than L^2 regularization.

The remainder of this paper is organized as follows : Section 6.2 develops the objective functional, and Section 6.3 describes its minimization. Section 6.4 shows how the image derivatives are computed. The validation experiments, using synthetic and real image sequences, are described in Section 6.5. Section 6.6 contains a conclusion.

6.2 Formulation

Let $I : (x, y, t) \rightarrow I(x, y, t)$ be an image sequence, where (x, y) are the spatial coordinates on the bounded image domain Ω, and $t \in \mathbf{R}^+$ designates time. Let (X, Y, Z) be the coordinates of a point $\mathbf{P}$ in space and (x, y) the coordinates of its projection. The coordinate system and the imaging geometry are shown in Fig. 6.1. Let U, V, W be the scene flow coordinate functions. The scene flow and depth linear gradient constraint [1], which relates the scene flow coordinates U, V, W and depth Z to the spatiotemporal image is :

$$fI_xU + fI_yV - (xI_x + yI_y)W + I_tZ = 0, \tag{6.1}$$

where I_x, I_y and I_t are the image spatiotemporal derivatives, Z designates depth (Fig. 6.1), and $(U, V, W) = (\frac{dX}{dt}, \frac{dY}{dt}, \frac{dZ}{dt})$ is the scene flow vector at $\mathbf{P}$. This is a homogeneous linear equation in the variables of scene flow and depth. The homogeneity results from the aperture problem. Multiplication of motion and depth (and structure thereof) by a constant (scale) maintains the equation integrity. One can remove this uncertainty of scale by choosing the depth to be *relative* to the frontoparallel plane $Z = Z_0$, for some positive depth Z_0 [1]. Therefore, Eq. (6.1) becomes :

$$fI_xU + fI_yV - (xI_x + yI_y)W + I_t(Z - Z_0) + I_tZ_0 = 0 \tag{6.2}$$

For notational convenience and economy, we will reuse the symbol Z for depth *relative* to the frontoparallel pane $Z = Z_0$, in which case we can write Eq. 6.2 as :

$$fI_xU + fI_yV - (xI_x + yI_y)W + I_tZ + I_tZ_0 = 0 \tag{6.3}$$

In the formulation of [1], scene flow and relative depth resulted from the minimization of the following L^2 smoothness regularization functional :

$$\begin{aligned} \mathbf{E}(U, V, W, Z|I) &= \tfrac{1}{2}\textstyle\int_\Omega (fI_xU + fI_yV - (xI_x + yI_y)W + I_tZ + I_tZ_0)^2 dxdy \\ &+ \tfrac{\alpha}{2}\textstyle\int_\Omega (\|\nabla U\|^2 + \|\nabla V\|^2 + \|\nabla W\|^2) dxdy + \tfrac{\beta}{2}\textstyle\int_\Omega \|\nabla Z\|^2 dxdy, \end{aligned} \tag{6.4}$$

where α and β scale the relative contribution of the terms of the functional. For boundary preservation, we replace the L^2 norm regularization term of each variable by an L^1 norm term, namely : $\int_\Omega \|\nabla Q\| dxdy = \int_\Omega \left(Q_x^2 + Q_y^2\right)^{\frac{1}{2}} dxdy$, where $Q \in \{U, V, W, Z\}$. Therefore, the objective functional is :

$$\begin{aligned} \mathbf{E}(U, V, W, Z|I) &= \tfrac{1}{2}\textstyle\int_\Omega (fI_xU + fI_yV - (xI_x + yI_y)W + I_tZ + I_tZ_0)^2 dxdy \\ &+ \tfrac{\alpha}{2}\textstyle\int_\Omega ((U_x^2 + U_y^2)^{\frac{1}{2}} + (V_x^2 + V_y^2)^{\frac{1}{2}} + (W_x^2 + W_y^2)^{\frac{1}{2}}) dxdy \\ &+ \tfrac{\beta}{2}\textstyle\int_\Omega (Z_x^2 + Z_y^2)^{\frac{1}{2}} dxdy, \end{aligned} \tag{6.5}$$

6.3 Optimization

The Euler-Lagrange equations corresponding to functional (6.5) are :

$$\begin{aligned}
&fI_x(fI_xU + fI_yV + (-xI_x - yI_y)W + I_tZ + I_tZ_0)\\
&-\alpha\frac{\partial}{\partial x}\frac{U_x}{\left(U_x^2+U_y^2\right)^{\frac{1}{2}}} - \alpha\frac{\partial}{\partial y}\frac{U_y}{\left(U_x^2+U_y^2\right)^{\frac{1}{2}}} = 0\\
&fI_y(fI_xU + fI_yV + (-xI_x - yI_y)W + I_tZ + I_tZ_0)\\
&-\alpha\frac{\partial}{\partial x}\frac{V_x}{\left(V_x^2+V_y^2\right)^{\frac{1}{2}}} - \alpha\frac{\partial}{\partial y}\frac{V_y}{\left(V_x^2+V_y^2\right)^{\frac{1}{2}}} = 0\\
&(xI_x + yI_y)(fI_xU + fI_yV - (xI_x + yI_y)W + I_tZ + I_tZ_0)\\
&+\alpha\frac{\partial}{\partial x}\frac{W_x}{\left(W_x^2+W_y^2\right)^{\frac{1}{2}}} - \alpha\frac{\partial}{\partial y}\frac{W_y}{\left(W_x^2+W_y^2\right)^{\frac{1}{2}}} = 0\\
&I_t(fI_xU + fI_yV + (-xI_x - yI_y)W + I_tZ + I_tZ_0)\\
&-\beta\frac{\partial}{\partial x}\frac{Z_x}{\left(Z_x^2+Z_y^2\right)^{\frac{1}{2}}} - \beta\frac{\partial}{\partial y}\frac{Z_y}{\left(Z_x^2+Z_y^2\right)^{\frac{1}{2}}} = 0,
\end{aligned} \tag{6.6}$$

with the Neumann boundary conditions :

$$\frac{\partial U}{\partial \mathbf{n}} = 0, \quad \frac{\partial V}{\partial \mathbf{n}} = 0, \quad \frac{\partial W}{\partial \mathbf{n}} = 0, \quad \frac{\partial Z}{\partial \mathbf{n}} = 0, \tag{6.7}$$

where $\frac{\partial}{\partial \mathbf{n}}$ is the differentiation operator in the direction of the normal $\mathbf{n}$ of the boundary $\partial\Omega$ of Ω.

Were it not for the denominator of the regularization terms in (6.6), the equations would be linear. It is common in numerical analysis to solve such systems of equations using an iterative method where the nonlinear terms are evaluated at the preceding iteration, in which case they are treated as data at the current iteration, and the equations to solve are linear. In our case, we alternate the following two steps at the current k-th iteration.

Step 1 : This step accounts of the non-linearity of the obtained Euler-Lagrange equations, and consists of updating the denominators of the regularization terms in (6.6) at each grid point :

$$g(Q^k) = \frac{1}{\left((Q_x^{k-1})^2 + (Q_y^{k-1})^2 + \epsilon\right)^{\frac{1}{2}}}, Q \in \{U, V, W, Z\} \tag{6.8}$$

where ϵ is a small positive value whose purpose is to remedy the non-differentiability of the Euclidean norm at the origin, without affecting the computational outcome in a noticeable way. Notice that the point-wise updates in (6.8) have a computational complexity that grows *linearly* with respect to N (the image size) : The complexity is N multiplied by the fixed complexity of 4 updates of the form in (6.8).

Step 2 : With pointwise coefficients $g(Q^k)(Q \in \{U, V, W, Z\})$ fixed, this step is an update (iteration k) of variables $\{U, V, W, Z\}$ for solving the following linear system :

$$\begin{aligned}
&fI_x(fI_xU + fI_yV + (-xI_x - yI_y)W + I_tZ + I_tZ_0) - \alpha g(U^k)\nabla^2 U = 0\\
&fI_y(fI_xU + fI_yV + (-xI_x - yI_y)W + I_tZ + I_tZ_0) - \alpha g(V^k)\nabla^2 V = 0\\
&(xI_x + yI_y)(fI_xU + fI_yV - (xI_x + yI_y)W + I_tZ + I_tZ_0) - \alpha g(W^k)\nabla^2 W = 0\\
&I_t(fI_xU + fI_yV + (-xI_x - yI_y)W + I_tZ + I_tZ_0) - \beta g(Z^k)\nabla^2 Z = 0,
\end{aligned} \tag{6.9}$$

The four equations of (6.9) are written for each point of image domain Ω. Let Ω be discretized via a unit-spacing grid, and let the grid points be indexed by the integers $\{1, 2, ..., N\}$. The pixel numbering is according to the lexicographical order, i.e., by scanning the image top-down and left-to-right. If the image is of size $n \times n$ then $N = n^2$. Let $a = fI_x$, $b = fI_y$, $c = -(xI_x + yI_y)$ and $d = I_t$.

For all grid point indices $i \in \{1, 2, ..., N\}$, a discrete approximation of the linear system (6.9) is :

$$(S) \quad \left\{ \begin{array}{rcl}
(a_i^2 + \alpha g_i(U^k)n_i)U_i + a_ib_iV_i + a_ic_iW_i + a_id_iZ_i - \alpha g_i(U^k)\sum_{j\in\mathbf{N}_i} U_j &=& -a_id_iZ_0\\
b_ia_iU_i + (b_i^2 + \alpha g_i(V^k)n_i)V_i + b_ic_iW_i + b_id_iZ_i - \alpha g_i(V^k)\sum_{j\in\mathbf{N}_i} V_j &=& -b_id_iZ_0\\
c_ia_iU_i + c_ib_iV_i + (c_i^2 - \alpha g_i(W^k)n_i)W_i + c_id_iZ_i - \alpha g_i(W^k)\sum_{j\in\mathbf{N}_i} W_j &=& -c_id_iZ_0\\
d_ia_iU_i + d_ib_iV_i + d_ic_iW_i + (d_i^2 + \beta g_i(Z^k)n_i)Z_i - \beta g_i(Z^k)\sum_{j\in\mathbf{N}_i} Z_j &=& -d_i^2Z_0,
\end{array} \right.$$

where $(U_i, V_i, W_i, Z_i) = (U, V, W, Z)_i$ is the scene flow vector at grid point i ; a_i, b_i, c_i, d_i are the values at i of a, b, c, d, respectively, $g_i(Q^k)$, $Q \in \{U, V, W, Z\}$, are the point-

wise updates of (6.8) evaluated at i, and $\mathbf{N}_i$ is the set of indices of the neighbors of i. For the 4-neighborhood, $n_i = card(\mathbf{N}_i) = 4$ for points interior to the discrete image domain, and $n_i < 4$ for boundary image points. Laplacian $\nabla^2 Q$ in the Euler-Lagrange equations ($Q \in \{U, V, W, Z\}$) has been discretized as $\frac{1}{4}\sum_{j\in\mathbf{N}_i}(Q_j - Q_i)$, with α (respectively β) absorbing the factor $1/4$.

Let $\mathbf{q} = (q_1, ..., q_{4N})^t \in \mathbf{R}^{4N}$ be the vector with coordinates $q_{4i-3} = U_i$, $q_{4i-2} = V_i, q_{4i-1} = W_i, q_{4i} = Z_i$, $i \in \{1, ..., N\}$, and $\mathbf{r} = (r_1, ..., r_{4N})^t \in \mathbf{R}^{4N}$ the vector with coordinates $r_{4i-3} = -a_i d_i Z_0$, $r_{4i-2} = -b_i d_i Z_0$, $r_{4i-1} = -c_i d_i Z_0$, and $r_{4i} = -d_i^2 Z_0$, $i \in \{1, ..., N\}$. System (S) of linear equations can be written in a matrix form as :

$$\mathbf{A}\mathbf{q} = \mathbf{r} \tag{6.10}$$

where $\mathbf{A}$ is the $4N \times 4N$ matrix with elements $\mathbf{A}_{4i-3,4i-3} = a_i^2 + \alpha g_i(U^k) n_i$; $\mathbf{A}_{4i-2,4i-2} = b_i^2 + \alpha g_i(V^k) n_i$; $\mathbf{A}_{4i-1,4i-1} = c_i^2 + \alpha g_i(W^k) n_i$; $\mathbf{A}_{4i,4i} = d_i^2 + \beta g_i(Z^k) n_i$; $\mathbf{A}_{4i-3,4i-2} = \mathbf{A}_{4i-2,4i-3} = a_i b_i$; $\mathbf{A}_{4i-3,4i-1} = \mathbf{A}_{4i-1,4i-3} = a_i c_i$; $\mathbf{A}_{4i-3,4i} = \mathbf{A}_{4i,4i-3} = a_i d_i$; $\mathbf{A}_{4i-2,4i-1} = \mathbf{A}_{4i-1,4i-2} = b_i c_i$; $\mathbf{A}_{4i-2,4i} = \mathbf{A}_{4i,4i-2} = b_i d_i$; $\mathbf{A}_{4i-1,4i} = \mathbf{A}_{4i,4i-1} = c_i d_i$; for all $i \in \{1, ..., N\}$, and $\mathbf{A}_{4i-3,4j-3} = -\alpha g_i(U^k)$; $\mathbf{A}_{4i-2,4j-2} = -\alpha g_i(V^k)$; $\mathbf{A}_{4i-1,4j-1} = -\alpha g_i(W^k)$; and $\mathbf{A}_{4i,4j} = -\beta g_i(Z^k)$, for all $i, j \in \{1, ..., N\}$ such that $j \in \mathbf{N}_i$, all other elements being equal to zero. This is a large scale sparse system of linear equations, which can be solved by iterative updates for sparse matrices [14, 15]. It is easy to prove that symmetric matrix $\mathbf{A}$ is positive definite (PD). This means that a fast solution of (6.10) can be obtained by convergent 4×4 block-wise Gauss-Seidel updates. To show that $\mathbf{A}$ is PD, it suffices to perform some algebraic manipulations

to write $\mathbf{q}^t\mathbf{A}\mathbf{q}$ for all $\mathbf{q} \in \mathbf{R}^{4N}, \mathbf{q} \neq \mathbf{0}$, as follows :

$$\begin{aligned}
\mathbf{q}^t\mathbf{A}\mathbf{q} &= \sum_{i=1}^{N}(a_iU_i + b_iV_i + c_iW_i + d_iZ_i)^2 \\
&+ \alpha\sum_{i=1}^{N}\sum_{j\in\mathbf{N}_i;j>i} g_i(U^k)(U_i - U_j)^2 + g_i(V^k)(V_i - V_j)^2 + g_i(W^k)(W_i - W_j)^2 \\
&+ \beta\sum_{i=1}^{N}\sum_{j\in\mathbf{N}_i;j>i} g_i(Z^k)(Z_i - Z_j)^2 > 0 \qquad (6.11)
\end{aligned}$$

The positive definiteness of $\mathbf{A}$ implies that iterative point-wise and block-wise Gauss-Seidel and relaxation updates for system (6.10) converge [14, 15]. For a 4×4 block division of $\mathbf{A}$, the Gauss-Seidel update (iteration k) for each grid point $i \in \{1, ..., N\}$ is :

$$\begin{aligned}
&(a_i^2 + \alpha g_i(U^k)n_i)U_i^{k+1} + a_ib_iV_i^{k+1} + a_ic_iW_i^{k+1} + a_id_iZ_i^{k+1} \\
&\quad = -a_id_iZ_0 + \alpha g_i(U^k)\left(\sum_{j\in\mathbf{N}_i;j<i} U_j^{k+1} + \sum_{j\in\mathbf{N}_i;j>i} U_j^k\right) \\
&b_ia_iU_i^{k+1} + (b_i^2 + \alpha g_i(V^k)n_i)V_i^{k+1} + b_ic_iW_i^{k+1} + b_id_iZ_i^{k+1} \\
&\quad = -b_id_iZ_0 + \alpha g_i(V^k)\left(\sum_{j\in\mathbf{N}_i;j<i} V_j^{k+1} + \sum_{j\in\mathbf{N}_i;j>i} V_j^k\right) \\
&c_ia_iU_i^{k+1} + c_ib_iV_i^{k+1} + (c_i^2 + \alpha g_i(W^k)n_i)W_i^{k+1} + c_id_iZ_i^{k+1} \\
&\quad = -c_id_iZ_0 + \alpha g_i(W^k)\left(\sum_{j\in\mathbf{N}_i;j<i} W_j^{k+1} + \sum_{j\in\mathbf{N}_i;j>i} W_j^k\right) \\
&d_ia_iU_i^{k+1} + d_ib_iV_i^{k+1} + d_ic_iW_i^{k+1} + (d_i^2 + \beta g_i(Z^k)n_i)Z_i^{k+1} \\
&\quad = -d_i^2Z_0 + \beta g_i(Z^k)\left(\sum_{j\in\mathbf{N}_i;j<i} Z_j^{k+1} + \sum_{j\in\mathbf{N}_i;j>i} Z_j^k\right),
\end{aligned} \qquad (6.12)$$

For each point $i \in \{1, ..., N\}$, we solve a 4×4 nonlinear system of equations, which can be done efficiently by a singular value decomposition (SVD) [16]. The computational complexity of this step grows *linearly* with respect to N : The complexity is N multiplied by the fixed complexity of a 4×4 SVD.

6.3.1 Computational load : L^1 vs. L^2

Although L^1 regularization yields nonlinear Euler-Lagrange equations for the minimization of our objective functional, these can be solved efficiently via the two-step scheme we proposed above. The point-wise updates in Eq. (6.8) account for the non-linearity of the L^1 model (Step 1), and are an additional computational load in comparison to L^2 regularization. However, these updates have a complexity that grows *linearly* with respect to N (image size). Therefore, they do not increase the computational time substantially ; see the computational times in Table 6.1. Step 2 has a computational complexity that is similar to the L^2 regularization of [1] : In both cases, we have block-wise (4×4) and relaxation updates for a large scale sparse system of linear equations, with a symmetric positive definite matrix $\mathbf{A}$. The complexity of each iteration is N multiplied by the fixed complexity of a 4×4 singular value decomposition, and the positive definiteness of $\mathbf{A}$ implies that the block-wise updates converge.

Table 6.1 reports the computation (CPU) times for both L^1 and L^2 regularizations, in the case of four different test images of different sizes. Columns 2 and 3 contain the overall CPU times. The third column gives the CPU time for the block-wise SVD updates solving a large scale sparse system. These SVD computations occur in both L^2 and L^1 formulations (Step 2). The last column reports the CPU times for the point-wise updates in Eq. (6.8). These updates appear only in the L^1 formulation (Step 1). The simulations were run on an Intel i7-4500U Processor (4M Cache, up to 3.00 GHz), using MATLAB (2014b version).

6.4 Partial derivatives

In computer vision, image derivatives are often approximated by locally averaged finite differences to lessen the impact of noise [1, 9, 17–20]. However, such fixed-

TABLE 6.1 – Computation times per iteration (in seconds).

Sequences (dimensions in pixels)	L^2(Overall)	L^1(Overall)	Step 2 (L^2/L^1)	Step 1 (L^1)
Berber (240×360)	1,86	2.21	1,86	0.35
Pharaohs (240×320)	1.65	1.96	1.65	0.31
Cylinder (384×500)	4.13	4.90	4.13	0.77
Marbled-block (384×512)	4.23	5.02	4.23	0.79

support low pass filtering does not generally fit the noise profile and can, therefore, be ineffective. A more effective way is to state differentiation as a spatially regularized variational problem, as was done in [1]. The process advocated in [1] looked at the derivative of an image as a function which, when integrated, gives the image function. Consequently, the objective functional contained an anti-differentiation data term which evaluates the conformity of a derivative to the image by constraining the integration of this derivative to produce the image. The functional also contained an L^2 regularization term. In this paper, we investigate a generalization which accounts for derivative discontinuities using an L^1 regularization term because L^2 regularization constrains the image derivatives to be smooth everywhere and, as a result, would adversely blur their boundaries.

6.4.1 L^1 regularized differentiation

The partial derivatives I_x and I_y of an image will be estimated using an anti-differentiation characterization. The method for I_x will be explained in this section. The derivative I_y is computed by the same scheme applied to the transposed image. Since only two images are available along the time axis, the temporal derivative will be estimated using the Horn and Schunck definition of the temporal derivative [9].

Let f designate an approximation of the derivative. Recall that in [1], the derivative was computed by minimizing the following functional with respect to f :

$$E(f) = \frac{1}{2}\int_{\Omega}\left(\|Df - I\|^2 + \gamma\|\nabla f\|^2\right) dxdy, \tag{6.13}$$

where D is the anti-differentiation operator, γ is a positive constant, and ∇f is the spatial gradient of f. The integral operator of anti-differentiation D is defined by :

$$Df(x,y) = \int_0^x f(z,y)dz \tag{6.14}$$

To generalize this formulation to preserve boundaries, we will replace the L^2 regularization term in (6.13) by an L^1 term : $\int_{\Omega}\left(f_x^2 + f_y^2\right)^{\frac{1}{2}} dxdy$. The objective functional becomes :

$$E(f) = \frac{1}{2}\int_{\Omega}\left(\|Df - I\|^2 + \gamma\int_{\Omega}(f_x^2 + f_y^2)^{\frac{1}{2}}\right) dxdy, \tag{6.15}$$

The corresponding Eular-Lagrange equations are :

$$D^*(Df - I) - \gamma\frac{\partial}{\partial x}\frac{f_x}{\left(f_x^2 + f_y^2\right)^{\frac{1}{2}}} - \gamma\frac{\partial}{\partial y}\frac{f_y}{\left(f_x^2 + f_y^2\right)^{\frac{1}{2}}} = 0, \tag{6.16}$$

where D^* is the adjoint operator of D given by, assuming $\Omega = [0,l]\times[0,l]$, $D^*f(x,y) = \int_x^l f(z,y)dz$.

Nonlinearity occurs in the regularization terms of (6.16). As done in the previous section, we can, in practice, and without affecting in any significant way subsequent processing, extend differentiability to the origin by replacing the denominators by $\left(f_x^2 + f_y^2 + \epsilon\right)^{\frac{1}{2}}$, for some small positive ϵ. Also, and as we have done in the previous section, solve (6.16) iteratively, by evaluating, at each iteration k, the nonlinear terms at the preceding iteration $k-1$. More precisely, we have, after initialization, the following equation at iteration k :

$$D^*(Df^k - I) - \frac{\gamma}{\left((f_x^{k-1})^2 + (f_y^{k-1})^2 + \epsilon\right)^{\frac{1}{2}}}\nabla^2 f^k = 0 \tag{6.17}$$

As in [1], discretization of (6.17) yields a large-scale sparse system of linear equations which can be solved by the Gauss-Seidel method.

TABLE 6.2 – Differentiation : L^1 regularization, L^2 regularization and local finite differences (LFD) applied to noised *chessboard* image (SNR = 1) and evaluated using mean squared error (MSE) and standard deviation error (SDE). Top : measurements from the whole image. Bottom : values from 5×5 windows centered on the boundaries. Regularization coefficient $\gamma = 1$.

Methods	L^1	L^2	LFD
Noised *chessboard* image	**MSE=0.06**	MSE=0.14	MSE=0.41
	SDE=0.09	SDE=0.13	SDE=0.31
Noised *chessboard* boundaries	**MSE=0.02**	MSE=0.03	MSE=0.06
	SDE=0.07	SDE=0.09	SDE=0.16

6.4.2 Example

We apply the scheme to compute the derivatives of the noisy (Gaussian white noise, $SNR = 1$) synthetic *chessboard* image of Figure 6.2, which illustrates that L^1 regularized differentiation outperforms both the local finite difference definition and L^2 regularized differentiation. A quantitative evaluation can be done by computing the MSE (Mean Squared Error) and the SDE (Standard Deviation of Error). Table 6.2 lists the results. Derivatives are in gray level ($0-255$ range) per pixel. The top part of the Table gives the measurements as obtained from the whole image. The values listed in the bottom part of the Table come from the vicinity of the image boundaries, using a 5×5 window centered on these. The results confirm visual inspection, i.e., that L^1 regularization outperforms both L^2 regularization and local finite differences.

6.5 Experimental Results

In this section, we expose various experiments showing the application of the described method to synthetic and real image sequences. In addition to displaying various experimental examples, we present a comparative and quantitative analysis that highlights the positive effects of L^1 regularization over the whole image domain, and particularly within motion boundaries. In our experiments, all formulation parameters are determined empirically and distances are measured in pixels. In all the examples, regularized differentiation's coefficient γ has been fixed equal to 1. As approximated in [21], the focal length of the camera is 600 pixels. The position of the fronto-parallel plane Z_0 is fixed to 6×10^4 pixels. The initial values of scene flow and depth at each point are set to, respectively, 0 and Z_0. Coefficients α and β vary from a sequence to another, and are given in the caption of each example.

6.5.1 Examples

Four image sequences with different characteristics served as samples to test the validity of our scheme and its implementation. Our displays include :

— Anaglyphs, which provide a convenient way for subjective appraisal of the recovered object depth. When viewed with chromatic (red−cyan) glasses on good-quality photographic paper or on standard screens, anaglyphs give viewers a strong sense of depth. They are produced from one of the two input images and the computed depth map, and are generally better perceived with full color resolution ;

— Standard displays using color-coded depth so as to highlight image-depth variations ;

— 3D reconstructed objects ;

— 3D scene flow vector fields ; and

— 2D Optical flow fields corresponding to our recovered scene flow. This can serve as an indirect validation of our implementation when the 2D outputs are compared to standard optical-flow methods (ex., the well tested/researched benchmark algorithm of Horn and Schunck).

We provide several figures, each corresponding to an example and organized as follows. The first row includes (from left to right) : (a) an anaglyph of the scene structure reconstructed from our method's output and the first frame of the sequence ; (b) a color-coded display of the recovered depth and the used color palette, with depth increasing from bottom (red) to top (purple) ; and (c) novel viewpoint images of the moving objects in the scene. The second row depicts (from left to right) : (a) a view of the obtained scene flow ; (b) a projected optical flow corresponding to our estimated scene flow ; and (c) optical flow computed directly by the Horn and Schunck algorithm.

The following describes the four image sequences that we used :

— The *Marbled-block* synthetic image sequence (Fig. 6.3) is taken from the database of KOGS/IAKS Laboratory, Germany. This sequence shows three blocks, two of them are moving. The rightmost block is moving backward to the left, whereas the front-most (smallest) block moves forward to the left. Some aspects of this sequence make its 3D interpretation challenging : the blocks and the floor have a similar macro texture with weak spatiotemporal intensity variations within the textons. This makes the occluding boundaries of the blocks ill defined at some places. Also, the source of light position with respect to the blocks causes shadows, which move with the blocks.

— The *Cylinder and box* real sequence (Fig. 6.4), provided in [22], depicts two moving objects, along with a moving background : a cylindrical surface rotating at a velocity of one degree per frame about the vertical axis, and moving laterally at an image rate of about 0.15 pixel per frame toward the right, as well as a box translating at approximately 0.30 pixel per frame toward the right.

Also, the background is translating to the right (parallel to the box motion) at a rate of about 0.15 pixel per frame. Those motions make 3D interpretation and recovery challenging in this example.

— The *Berber* real sequence (Fig.6.5) exhibits a sculpture rotating about the vertical axis, and translating forward to the left in a static environment.

— The *Pharaohs* sequence (Fig. 6.6) shows two moving sculptures in a static environment. The first (leftmost) figurine translates left and forward, whereas the second rotates about a nearly vertical axis to the right.

The examples above support the validity of our scheme and its implementation. The obtained anaglyphs, color-coded depths and 3D object reconstructions are consistent with the actual structures of the scenes. The displays of the estimated 3D scene flow and the corresponding 2D optical flow are consistent with the real motion of the objects in scenes. Also, the optical flow derived from our scene flow estimation is in line with a direct optical-flow computation by the standard Horn-and-Schunck algorithm. We notice that the obtained fields are more regular and present less noise than those computed directly with Horn-and-Schunck algorithm. This is mainly due to the use of 3D information as well as regularized differentiation.

6.5.2 Comparative analysis

We report a comprehensive comparative analysis, which demonstrates the benefits of our L^1 formulations. In particular, we focus our evaluations on the boundaries of motions and objects, so as to illustrate the boundary-preserving effect of our L^1 methods.

We began our comparative analysis by the simple synthetic sequence *Squares*, which includes two images with known motions. This sequence depicts two overlapping squares in opposite motions, along with a moving background. The motions for these three elements are known : A translation of the rightmost square by $(-1, -1)$ pixels

in the downward-left direction, a translation of the leftmost square by $(1, 1)$ pixels in the top-right direction and downward translation of the background by $(0, -1)$ pixels. To better test the performance of evaluated schemes, we added noise independently to the first and second image. Noise values are from a discretized, shifted and truncated Gaussian in the interval between 0 and 100 gray levels, within an overall range of [0, 255]. The first row of Fig. 6.7 displays the first (noised) image (left) and the vector-coded ground truth (right). The second row depicts the optical flows, the first (left) obtained from a projection of the L^2 regularized scene flow [1] and the second from our L^1 regularized scene flow. In both cases, we estimated image derivatives with a finite-difference regularization based on the standard Horn-and-Schunck definition [9]. We refer to these methods as L^2HS (left) and L^1HS (right). In the third row, we repeated the same experiment using regularized image differentiation instead of the finite differences. On the left, we depict the result using L^2 regularization for both scene flow and image-derivative estimations (L^2L^2) whereas, on the right, we show the result for L^1 (L^1L^1). The optical flows resulting from the four methods are consistent with their expected overall appearance. However, we see clearly a difference between the performances of those algorithms.

Visually, the L^1L^1 scheme yielded the closest match to the vector-coded ground truth. To support this quantitatively, we added an evaluation based on two standard error measures for optical flow [23] : average angular error (aae) and endpoint error (epe) ; see Table 6.3. The (L^1L^1) scheme performed better than the other methods.

Fig. 6.8 depicts another example for our comparative analysis. It uses the *Hydrangea* sequence of a real scene from the Middlebury data set [23]. The sequence shows a rotating flower bouquet within a translating background, and the ground-truth flow of the sequence is given [1]. The first row of the figure displays the first of the two input images (left) and the vector-coded ground truth (right). The second row depicts the

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

TABLE 6.3 – Performance of L^2HS, L^1HS, L^2L^2 and L^1L^1 algorithms on the noised Squares image (SNR = 1.12).

L^2HS	L^1HS	L^2L^2	L^1L^1
aae=15.94	aae=12.57	aae=15	**aae=11.95**
epe=0.44	epe=0.41	epe=0.4	**epe=0.36**

TABLE 6.4 – Performance of L^2HS, L^1HS, L^2L^2 and L^1L^1 algorithms on the *Hydrangea* real sequence.

L^2HS	L^1HS	L^2L^2	L^1L^1
aae=21.18	aae=16.72	aae=17.04	**aae=15.96**
epe=2.17	epe=1.78	epe=1.92	**epe=1.54**

optical flows, the first (left) obtained from a projection of the L^2 regularized scene flow [1] and the second from our L^1 regularized scene flow. In both cases, we estimated image derivatives with a finite-difference regularization based on the standard Horn-and-Schunck definition [9]. We refer to these methods as L^2HS (left) and L^1HS (right). In the third row, we repeated the same experiment using regularized image differentiation instead of the finite differences. On the left, we depict the result using L^2 regularization for both scene flow and image-derivative estimations (L^2L^2) whereas, on the right, we show the result for L^1 (L^1L^1). Visually, the L^1L^1 scheme yielded the closest match to the vector-coded ground truth. Table 6.4 supports quantitatively these results. It reports two standard optical flow errors (aae and epe) [23], showing that the L^1L^1 scheme performed better than the other methods.

Figure 6.9 compares visually the results of L^1L^1 (proposed method) and L^2L^2 [1] using the different examples in section 6.5.1. The first column depicts the results of the proposed L^1L^1 method, whereas the second shows the results of [1] (L^2L^2).

The first row shows anaglyphs, the second color-coded depth, and the third 3D object reconstructions. The last two rows displayed both 3D scene flow and projected 2D optical flow fields. We can see that the results are better with L^1L^1 than with L^2L^2 : the 3D parameters are better defined, clearer and sharper, especially on the boundaries ; flow fields are more regular and smooth.

Tables 6.5 and 6.6 report quantitative comparisons of the four algorithms (L^2HS and L^1HS in tab. 6.5 ; L^2L^2 and L^1L^1 in tab 6.6) using three standard error measures [23] : average angular error (aae), standard angular error (stae) and endpoint error (epe). Errors are computed between the motion resulting from scene flow and the optical flow ground truth. As we do not have scene flow ground truth for these real examples, evaluations of the resulting (projected) optical flow is a good indirect way to assess scene flow results. We constructed an optical flow ground-truth using SURF-based [24] detection and correspondences of the key points in the two images. Then, the velocity coordinates of these key points are computed, yielding an optical-flow ground truth. Tables 6.5 and 6.6 confirm that the use of our L^1 regularization improves the results, and that the L^1L^1 formulation outperforms the other methods.

In the following part of our comparative analysis, we will focus on the assessment (both qualitative and quantitative) of motion within objects boundaries, as preserving these is an important feature of our L^1 formulation.

Let us start with a qualitative visual inspection by displaying the images of the gradient of motion for each of the four methods (L^2HS, L^1HS, L^2L^2 and L^1L^1) ; See Fig. 6.10. We note that, with our L^1 regularization, points on the boundaries of motion are brighter (sharper). This is due to the fact that our L^1 formulation preserves sharp boundaries.

To support these results, we added quantitative evaluations based on aae, stae and epe errors : Tab. 6.7 report the results for L^2HS and L^1HS ; Table 6.8 report the results for L^2L^2 and L^1L^1. Errors are computed between the motion projected

TABLE 6.5 – Errors for the L^2HS and L^1HS formulations.

Sequences	errors	L^2HS	L^1HS
Berber	aae	38.51	**22.01**
	stae	43.84	**24.18**
	epe	1.11	**0.68**
Pharaohs	aae	59.17	**43.99**
	stae	41.38	**30.37**
	epe	**1.12**	1.54
Cylinder	aae	59.26	**57.7**
	stae	48.53	**46.82**
	epe	2.03	2.03
Marbled-block	aae	8.89	**7.63**
	stae	22.79	**17.55**
	epe	0.2	**0.15**

from the obtained scene flow and the optical flow ground truth within 7×7 windows centered at a set of key points on motion boundaries (the key points are those for which we have motion ground truth). Tables 6.7 and 6.8 report the results, which clearly indicate that our L^1L^1 formulation outperforms all the other methods.

6.6 Conclusion

This study investigated a boundary-preserving method for joint recovery of scene flow and relative depth from a monocular sequence of images. The scheme built upon the basic formulation of [1]. It minimized a functional composed of the data conformity term of [1], which relates the image sequence spatio-temporal variations to scene flow and depth, and an L^1 regularization term, rather than L^2 as in [1]. Therefore,

TABLE 6.6 – Errors for the L^2L^2 and L^1L^1 formulations.

Sequences	errors	L^2L^2	L^1L^1
Berber	aae	23.27	**11.61**
	stae	14.77	**10.16**
	epe	0.9	**0.41**
Pharaohs	aae	20.01	**14.02**
	stae	19.21	**12.04**
	epe	0.62	**0.36**
Cylinder	aae	18.68	**10.05**
	stae	18.91	**9.55**
	epe	0.78	**0.2**
Marbled-block	aae	**4.14**	4.2
	stae	8.56	**8.33**
	epe	0.1	**0.09**

this afforded a boundary preserving version of the basic formulation. The corresponding nonlinear Euler-Lagrange equations were discretized and solved iteratively by a scheme, which solved at each iteration a large scale sparse system of linear equations in the unknowns of scene flow and depth. The image derivatives were estimated by a variational method with L^1 regularization. This also led to an iterative method of resolution, which consisted of solving a large sparse system of linear equations at each iteration. Experiments show that the scheme is sound and efficient. The examples demonstrated the need to regularize scene flow and depth so as to take into account their boundaries, i.e, sharp spatial transitions of scene flow or depth. The results justify extensive further investigations, particularly concerning quantitative, i.e., ground truth controlled evaluation, motion of large extent, and image noise and resolution in

TABLE 6.7 – Quantitative evaluations on the boundaries of motion (L^2HS and L^1HS).

Sequences	errors	L^2HS	L^1HS
Berber ($Pt = 14$)	aae	51.4	**47.24**
	stae	23.47	**21.72**
	epe	1.53	**1.43**
Pharaohs ($Pt = 29$)	aae	45.49	**32.23**
	stae	28.73	**18.21**
	epe	0.97	**0.69**
Cylinder ($Pt = 15$)	aae	24.78	**23.45**
	stae	13.75	**13.36**
	epe	0.46	**0.44**
Marbled-block ($Pt = 9$)	aae	94.35	**78.81**
	stae	15.33	**7.07**
	epe	2.32	**1.82**

common practical settings.

The choice of a continuous alternating optimization scheme for our problem can be motivated by two important facts. First, we are dealing with continuous variables and, therefor, a continuous (not discrete) Euler-Lagrange regularization approach is a natural choice. Furthermore, our alternating scheme for solving the ensuing non-linear Euler-Lagrange equations has a computational complexity that behaves linearly w.r.t the number of grid points (N). This is important in practice, particularly when dealing with large image sequences. Of course, it would be very interesting to investigate other regularization options for estimating scene flow and depth from a single image sequence, for instance :

TABLE 6.8 – Quantitative evaluations on the boundaries of motion (L^2L^2 and L^1L^1).

Sequences	errors	L^2L^2	L^1L^1
Berber ($Pt = 14$)	aae	26.02	**23.27**
	stae	20.54	**14.77**
	epe	1.01	**0.9**
Pharaohs ($Pt = 29$)	aae	15.45	**14.02**
	stae	**11**	12.04
	epe	0.36	0.36
Cylinder ($Pt = 15$)	aae	11.99	**10.05**
	stae	9.66	**9.55**
	epe	0.23	**0.2**
Marbled-block ($Pt = 9$)	aae	20.44	**11.53**
	stae	11.84	**6.24**
	epe	0.98	**0.62**

— Discrete Markov Random Fields (MRFs) : MRFs were previously investigated for optical flow using sub-modular pairwise potentials [25]. MRF models can benefit from powerful combinatorial optimization techniques such as graph cuts [26]. It is worth noting, however, that adapting discrete MRFs to our continuous setting requires some technical care and is not straightforward.

— Regularization based on nonlocal-mean filtering [27]. Non-local means can preserve edges and textures. They were applied successfully to thin structures in depth-from-defocus problems [27].

— State-of-the-art solvers for non-smooth problems such as the primal-dual algorithm of Chambolle and Pock [28].

Figures

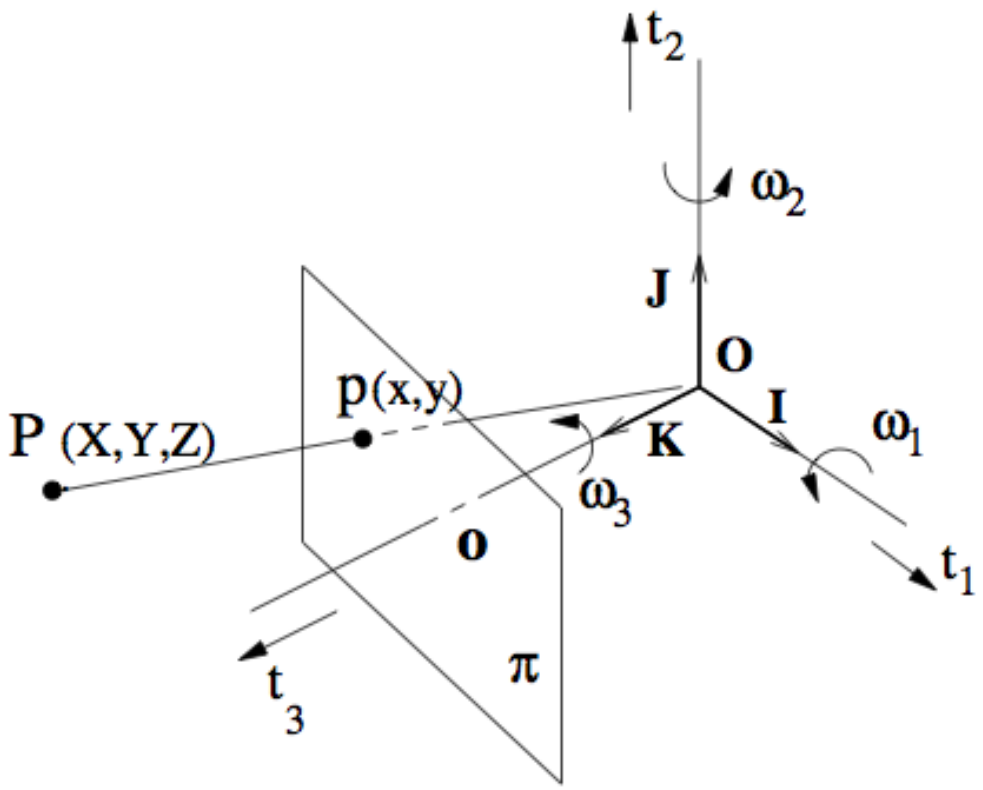


FIGURE 6.1 – The viewing system is represented by a Cartesian coordinate system $(\mathbf{O}; X, Y, Z)$ and central projection through the origin. The $Z-$axis is the depth axis. The image plane π is orthogonal to the Z-axis at distance f, the focal length, from $\mathbf{O}$.

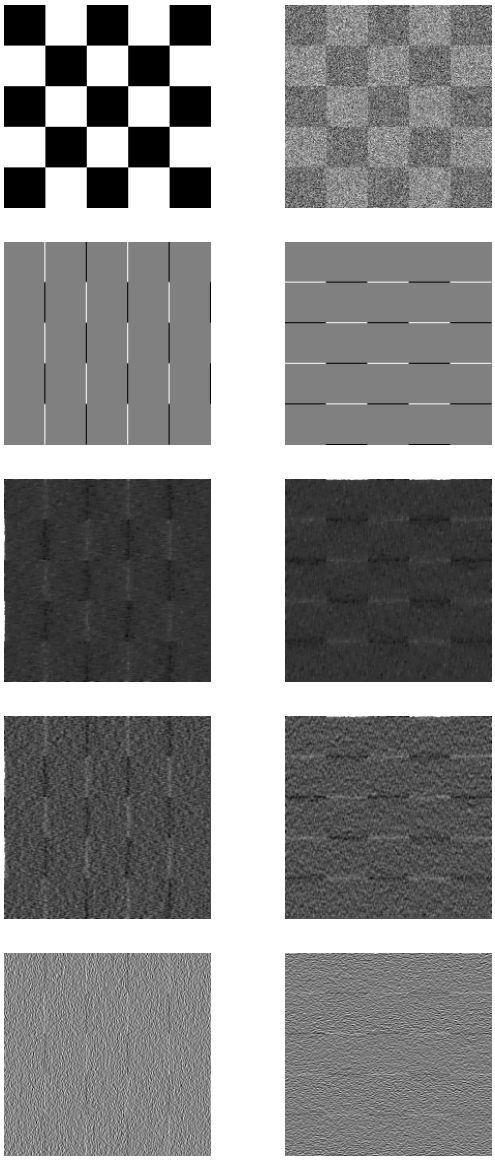


FIGURE 6.2 – *Noised chessboard.* First row, chessboard image on the left and noised chessboard image on the right (SNR = 1). Second row, ground truth of partial derivatives : I_x on the left and I_y on the right. Third row, estimated partial derivatives using TV regularized differentiation with $\gamma = 1$: I_x on the left and I_y on the right. Fourth row, estimated partial derivative using L^2 regularized differentiation with $\gamma = 1$: I_x on the left I_y on the right. Fifth row, estimated partial derivative using forward difference of Horn and Schunck : I_x on the left and I_y on the right.

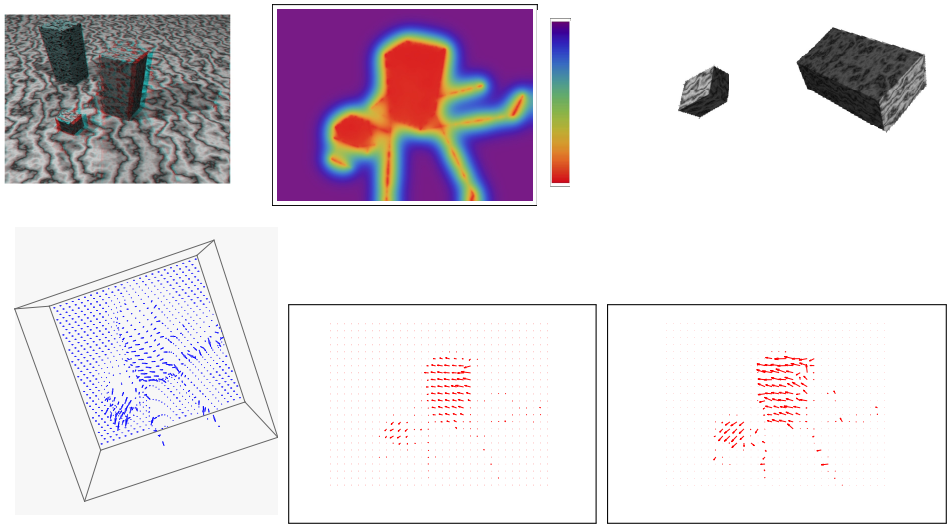


FIGURE 6.3 – *Marbled blocks* sequence results (better perceived when figures are enlarged on screen). Parameters : $\alpha = 6 \times 10^7$ and $\beta = 10^3$. First row from left to right : An anaglyph of the structure reconstructed from the method's output and the first frame of the sequence ; a color-coded display of the recovered depth along with the used colour palette, with depth increasing from bottom (red) to top (purple) ; novel viewpoint images of the two moving blocks. Second row : A view of the scene flow vectors ; optical flow corresponding to the estimated scene flow ; optical flow computed directly by the Horn and Schunck algorithm.

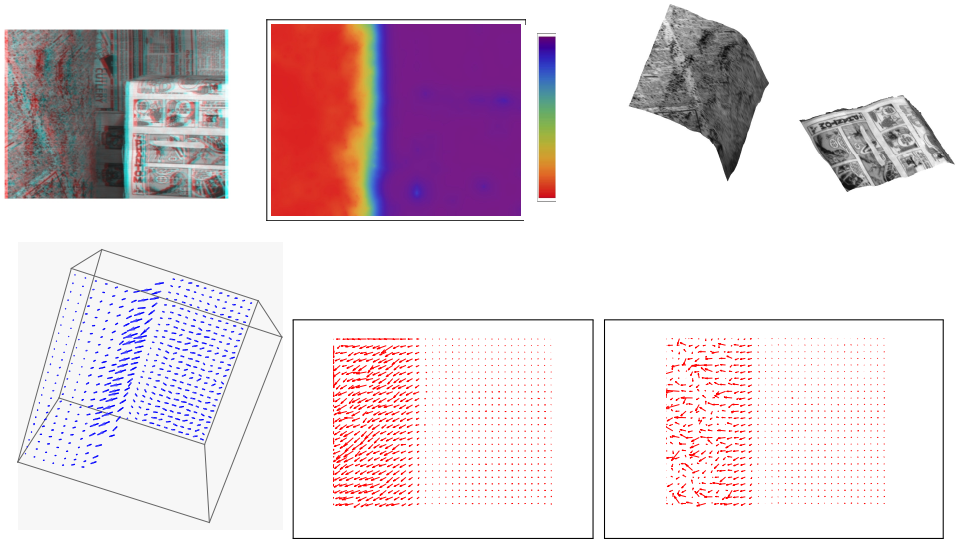


FIGURE 6.4 – *Cylinder and box* sequence results (better perceived when figures are enlarged on screen). Parameters : $\alpha = 6 \times 10^7$ and $\beta = 10^5$. First row from left to right : An anaglyph of the structure reconstructed from the method's output and the first frame of the sequence ; a color-coded display of the recovered depth along with the used color palette, with depth increasing from bottom (red) to top (purple) ; novel viewpoint images of the cylindrical surface and the box. Second row : A view of the scene flow vectors ; optical flow corresponding to the estimated scene flow ; optical flow computed by the Horn and Schunck algorithm.

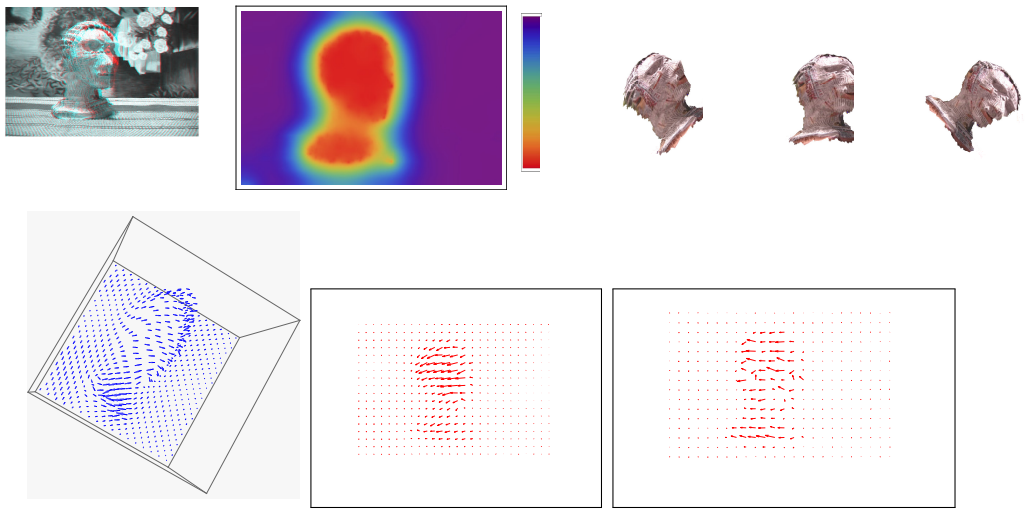


FIGURE 6.5 – *Berber* figurine sequence results (better perceived when figures are enlarged on the screen). Parameters : $\alpha = 6 \times 10^8$; $\beta = 5 \times 10^5$. First row from left to right : An anaglyph of the structure reconstructed from the method's output and the first frame of the input image sequence ; a color-coded display of the recovered depth along with the used color palette, with depth increasing from bottom (red) to top (purple) ; novel viewpoint images of the figurine. Second row : a view of the scene flow vectors ; optical flow corresponding to the estimated scene flow ; optical flow computed by the Horn and Schunck algorithm.

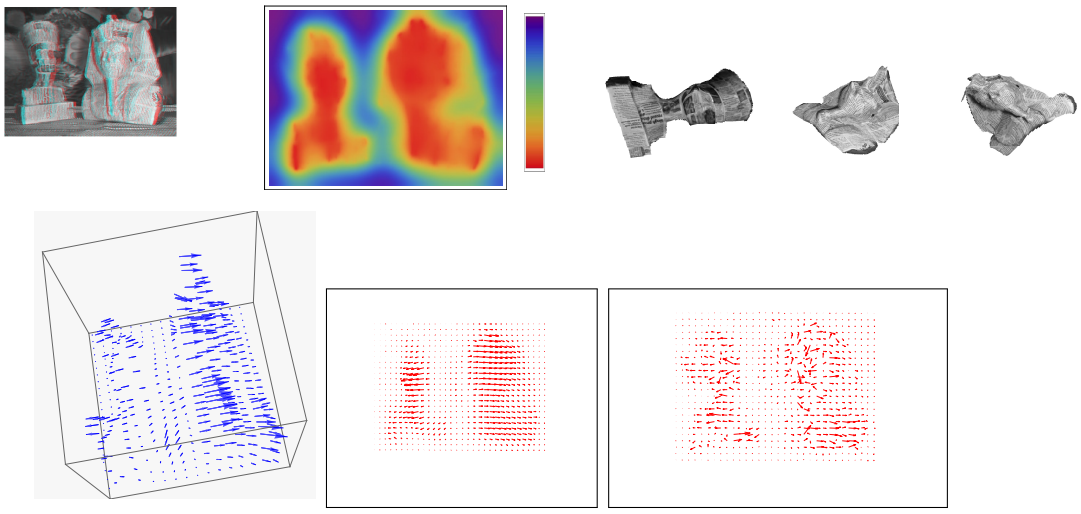


FIGURE 6.6 – *Pharaohs* figurines sequence (better perceived when figures are enlarged on screen). Parameters : $\alpha = 6 \times 10^8$; $\beta = \times 10^2$. First row from left to right : An anaglyph of the structure reconstructed from the method's output and the first frame of the input image sequence ; a colour-coded display of the recovered depth along with the used color palette, with depth increasing from bottom (red) to top (purple) ; novel viewpoint images of the figurines. Second row : A view of the scene flow vectors ; optical flow corresponding to the estimated scene flow ; optical flow computed by the Horn and Schunck algorithm.

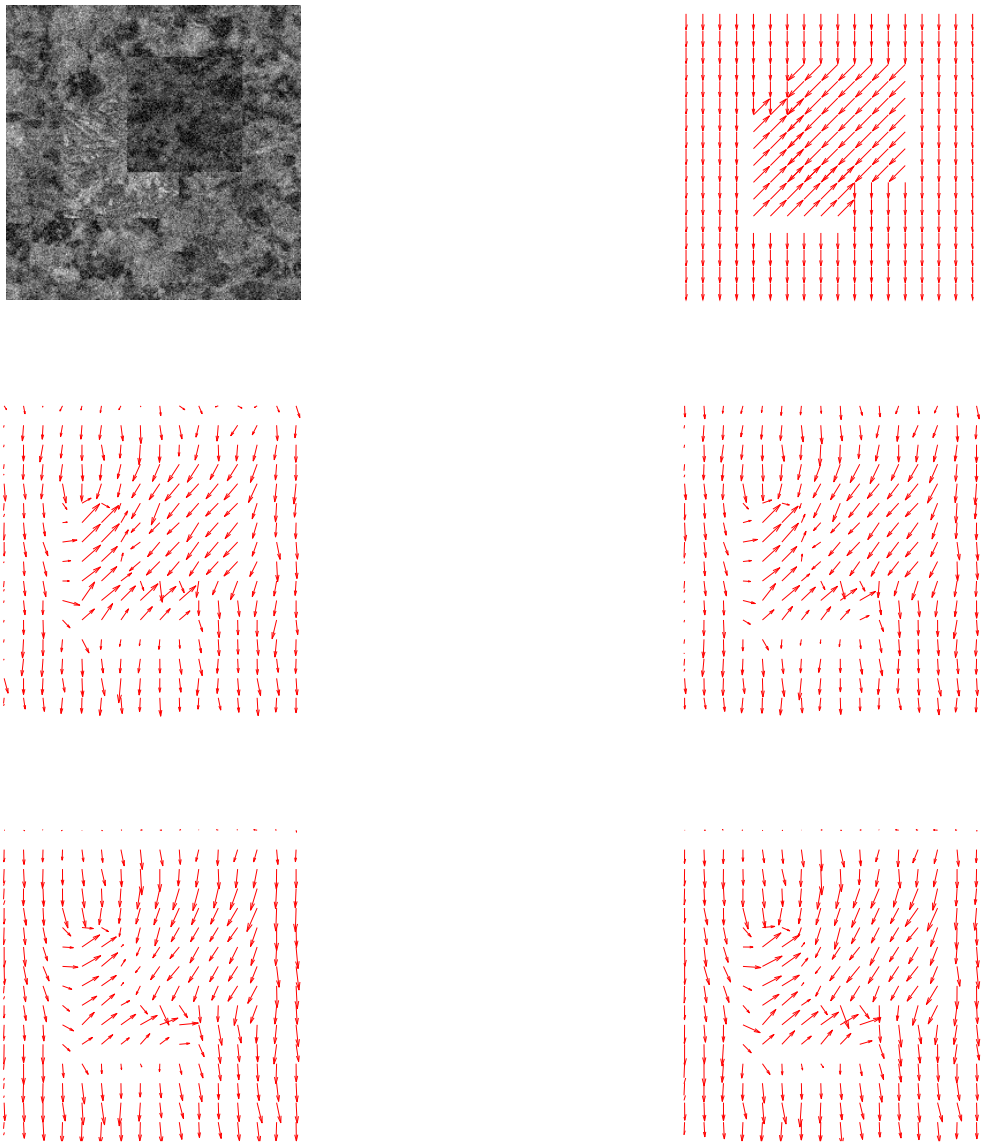
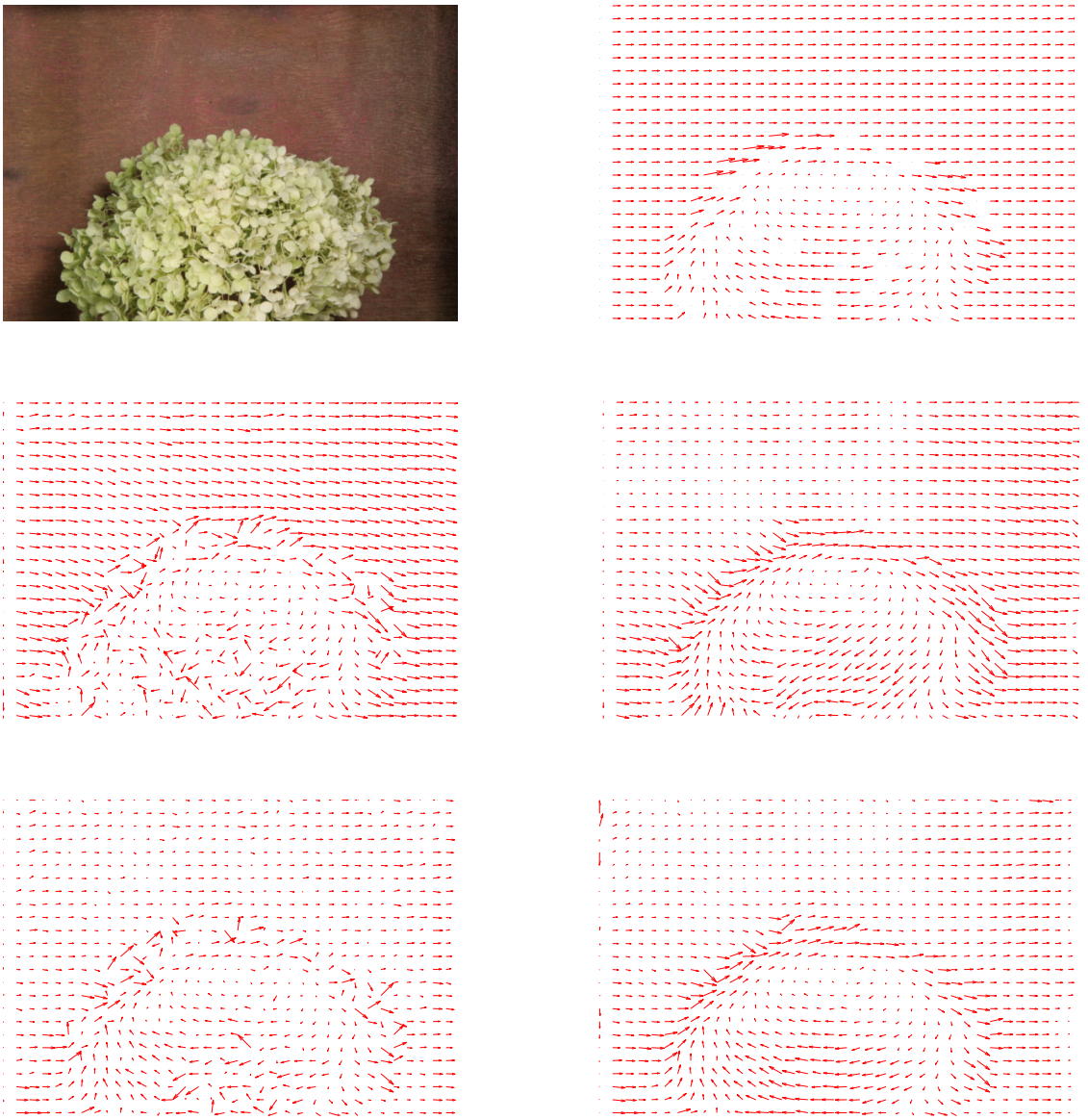


FIGURE 6.7 – *Squares* synthetic sequence results. First row : the first (noised) image of the sequence (left) and the vector-coded ground truth (right). Second row : the optical flow corresponding to L^2HS (left) and L^1HS (right). Third row : the optical flow corresponding to L^2L^2 (left) and L^1L^1 (right).



x

FIGURE 6.8 – *Hydrangea* real sequence results. First row : the first image of the sequence (left) and the vector-coded ground truth (right). Second row : the optical flow corresponding to L^2HS (left) and L^1HS (right). Third row : the optical flow corresponding to L^2L^2 (left) and L^1L^1 (right).

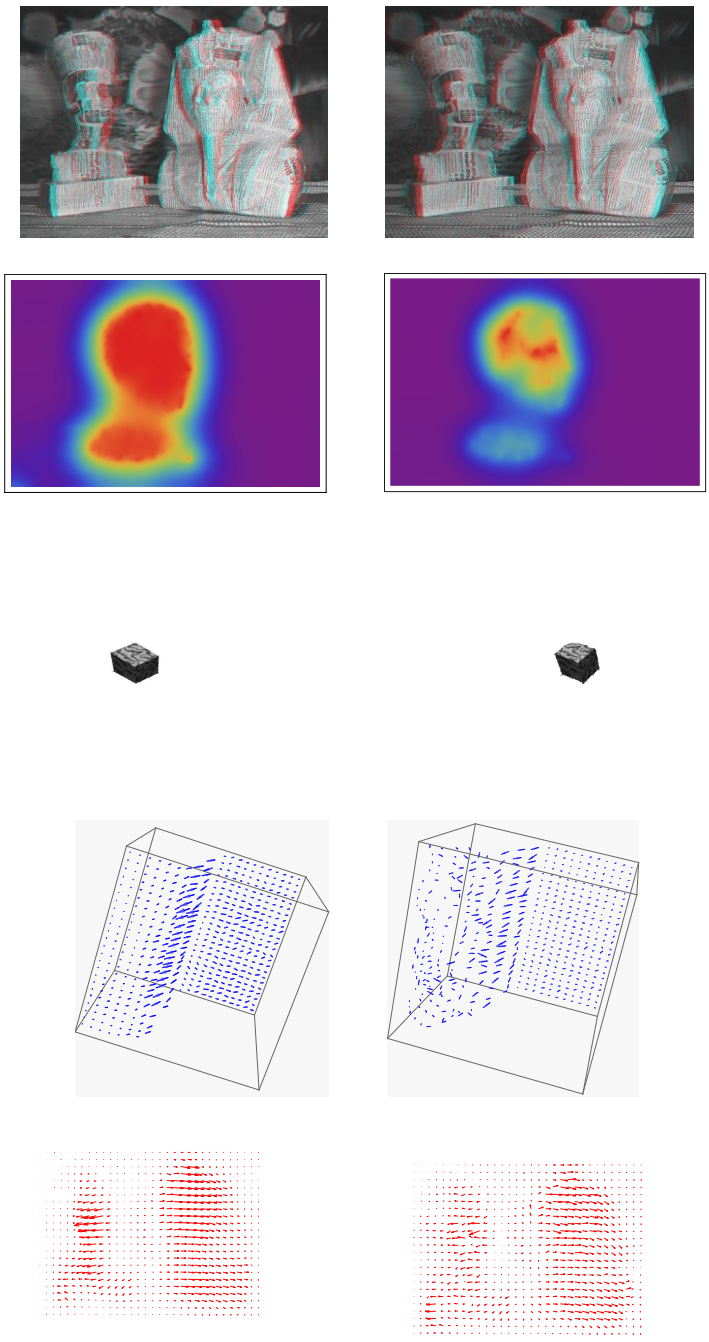


FIGURE 6.9 – Visual inspection of the results of the proposed L^1L^1 method (first column) and the L^2L^2 method in [1] (second column). The first row shows anaglyphs, the second color-coded depth, and the third $3D$ object reconstructions. The last two rows depict 3D scene flow and projected 2D optical flow fields.

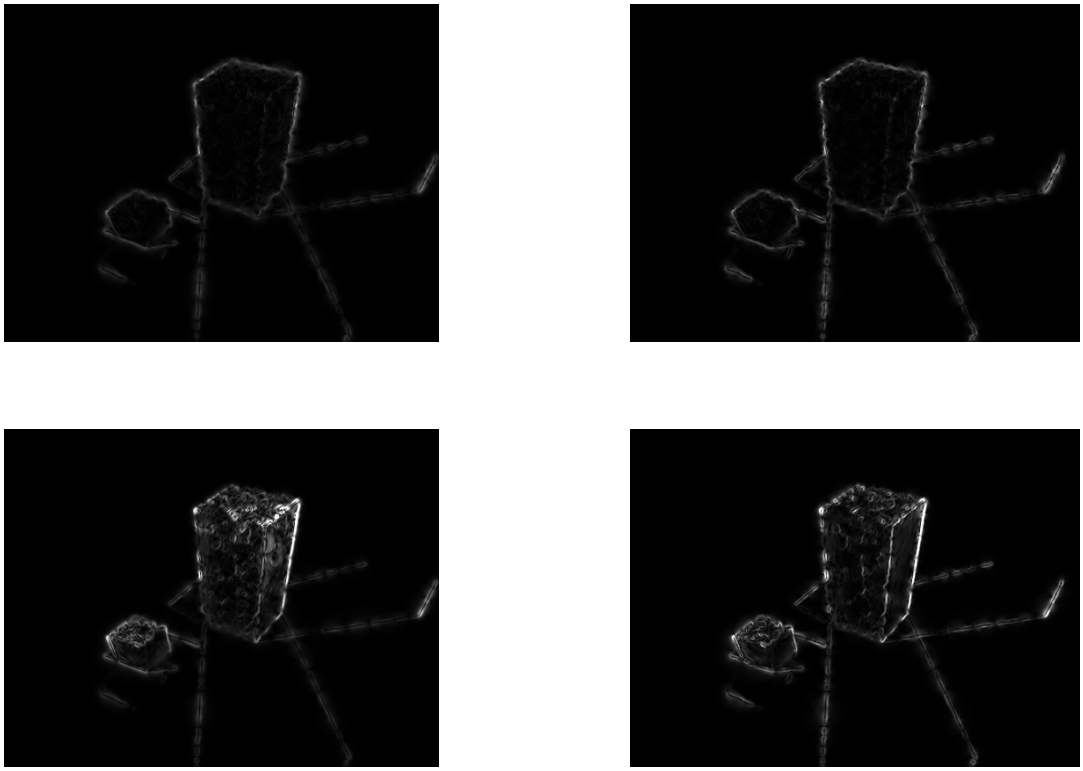


FIGURE 6.10 – Gradients of optical flow for the *Marbled blocks* sequence. First row : L^2HS (left) and L^1HS (right). Second row : L^2L^2 (left) and L^1L^1 (right).

Bibliographie

[1] A. Mitiche, Y. Mathlouthi, and I. Ben Ayed, "Monocular concurrent recovery of structure and motion scene flow," *Front. ICT 2 : 16*, 2015.

[2] H. Longuet-Higgins and K. Prazdny, "The interpretation of a moving retinal image," *Proceedings of the Royal Society of London, B*, vol. 208, pp. 385–397, 1981.

[3] A. Mitiche and J. Aggarwal, *Computer vision analysis of image motion by variational methods*. Springer, 2013.

[4] S. Vedula, P. Rander, R. Collins, and T. Kanade, "Three-dimensional scene flow," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 3, pp. 475–480, 2005.

[5] J.-P. Pons, R. Keriven, and O. Faugeras, "Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score," *International Journal of Computer Vision*, vol. 72, no. 2, pp. 179–193, 2007.

[6] F. Huguet and F. Devernay, "A variational method for scene flow estimation from stereo sequences," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–7.

[7] A. Wedel, C. Rabe, T. Vaudrey, T. Brox, U. Franke, and D. Cremers, *Efficient dense scene flow from sparse or dense stereo data*. Springer, 2008.

[8] C. Vogel, K. Schindler, and S. Roth, "Piecewise rigid scene flow," in *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[9] B. Horn and B. Schunk, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 17, pp. 185–203, 1981.

[10] G. Aubert, R. Deriche, and P. Kornprobst, "Computing optical flow via variational thechniques," *SIAM Journal of Applied Mathematics*, vol. 60, no. 1, pp. 156–182, 1999.

[11] G. Aubert and P. Kornprobst, *Mathematical Problems in Image Processing*. Springer, 2002.

[12] A. Mitiche and H. Sekkati, "Optical flow 3D segmentation and interpretation : A variational method with active curve evolution and level sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1818–1829, Nov. 2006.

[13] C. R. Vogel, *Computational methods for inverse problems*. SIAM Frontiers in Applied Mathematics, 2002.

[14] P. Ciarlet, *Introduction a l'analyse numerique matricielle et a l'optimisation*, 5th ed. Masson, 1994.

[15] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, 3rd ed., ser. Texts in Applied Mathematics ; 12. New York : Springer, 2002.

[16] G. Forsythe, M. Malcolm, and C. Moler, *Computer methods for mathematical computations*. Prentice-Hall, 1977.

[17] S. Marshall, "Depth dependence of ambient noise," *Oceanic Engineering, IEEE Journal of*, vol. 30, no. 2, pp. 275–281, 2005.

[18] M. P. Heinrich, M. Jenkinson, M. Bhushan, T. Matin, F. V. Gleeson, M. Brady, and J. A. Schnabel, "Mind : Modality independent neighbourhood descriptor for

multi-modal deformable registration," *Medical Image Analysis*, vol. 16, no. 7, pp. 1423–1435, 2012.

[19] S. Periaswamy and H. Farid, "Medical image registration with partial data," *Medical image analysis*, vol. 10, no. 3, pp. 452–464, 2006.

[20] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2432–2439.

[21] H. Sekkati and A. Mitiche, "A variational method for the recovery of dense 3D structure from motion," *Journal of Robotics and Autonomous Systems*, vol. 55, pp. 597–607, 2007.

[22] C. Debrunner and N. Ahuja, "Segmentation and factorization-based motion and structure estimation for long image sequences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 206–211, 1998.

[23] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31, 2011.

[24] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[25] V. Lempitsky, S. Roth, and C. Rother, "Fusionflow : Discrete-continuous optimization for optical flow estimation," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[26] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in nd images," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 1. IEEE, 2001, pp. 105–112.

[27] P. Favaro, "Recovering thin structures via nonlocal-means regularization with application to depth from defocus," in *Computer vision and pattern recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 1133–1140.

[28] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011.

Chapitre 7

Conclusion

Les études réalisées au cours de cette thèse ont été principalement consacrées à l'estimation conjointe du flot de scène dense et de la profondeur relative à partir d'une séquence d'images monoculaire. Il s'agit d'une méthode variationnelle, directe et non paramétrique.

Comme première contribution, nous avons développé une méthode linéaire capable de récupérer conjointement le flot de scène et la profondeur relative des objets dans une scène à partir d'une seule séquence d'images. La méthode résulte d'une formulation variationnelle du problème par une fonctionnelle qui contient un terme de conformité de données qui relie la vitesse tridimensionnelle à la profondeur par les variations spatiotemporelles et un terme de régularisation L^2. Le terme de données a été développé en réécrivant la contrainte de flot optique de Horn and Schunck linéairement en fonction de la vitesse du flot de scène et de la profondeur relative.

La deuxième contribution a été l'estimation des dérivées partielles régularisées d'une image. Le processus de dérivation est un problème inverse mal posé. Nous avons proposé une solution variationnelle qui minimise une fonctionnelle composée de deux termes : un terme d'adéquation de l'intégrale des dérivées à l'image et un terme de régularisation par lissage. Le terme des données utilise un opérateur d'anti-

différentiation, ce qui contraint la dérivée à une fonction qui redonne l'image lorsqu'on l'intègre. Le terme de régularisation L^2.

Une amélioration du problème de l'estimation des dérivées partielles de l'image a été abordée comme troisième contribution. Il s'agit d'une version qui préserve les frontières des objets dans l'image. Notre formulation variationnelle a utilisé un terme de régularisation L^1.

La quatrième contribution est une amélioration du problème de l'estimation conjointe du flot de scène et de la profondeur à partir d'une séquence d'images monoculaire. Pour préserver les frontières du mouvement et de la profondeur des objets dans la scène, le terme de lissage quadratique dans notre fonctionnelle est remplacé par une régularisation du type L^1.

Les contributions de cette thèse peuvent être étendues dans plusieurs directions. Les résultats obtenus encouragent à investiguer davantage la méthode proposée et tester ses performances face à d'autres difficultés comme les séquences d'images dont le mouvement est de grande étendue, aussi bien que les séquences d'images avec différents types de bruit, exemple, les squences ultrasons en imagerie médicale. Plusieurs améliorations peuvent être apportées à notre méthode. Par exemple, pour améliorer les performances de la résolution numérique de notre formulation, on peut remplacer les itérations de Gauss-Seidel par une méthode de résolution de systèmes creux et à grande échelle. Ceci inclut plusieurs option comme : (i) la méthode classique d'accélération de convergence appliquée aux itérations de Gauss-Seidel [1], (ii) l'algorithme du gradient conjugué [2], (iii) La méthode séquentielle de correction dans les sous-espaces (sequential subspace correction : SSC) [3] et (iv) Les méthodes modernes basées sur les espaces de Krylov [4].

Aussi, ça serait intéressant d'appliquer d'autres types de régularisation qui permettent de préserver les discontinuités comme : (i) la régularisation par diffusion anisotrope [5], (ii) la régularisation basée sur le débruitage par patchs (non-local

mean filtering) [6], (iii) l'utilisation de la fonction de Aubert et al. [7] dans le terme de régularisation. L'estimation et la segmentation conjointes du mouvement [8–13] pourrait être également une technique qui permet de préserver les discontinuités du mouvement et de la profondeur. En contre partie, une segmentation basée sur le mouvement pourrait bénéficier des informations tridimensionnelles qui découlent de l'estimation du flot de scène et de la profondeur relative pour améliorer les résultats de la segmentation.

Bibliographie

[1] P. Ciarlet, *Introduction a l'analyse numérique matricielle et a l'optimisation*, 5th ed. Masson, 1994.

[2] N. Sundaram, T. Brox, and K. Keutzer, "Dense point trajectories by gpu-accelerated large displacement optical flow," in *European conference on computer vision*. Springer, 2010, pp. 438–451.

[3] W. Hackbusch, *Iterative solution of large sparse systems of equations*. Springer Science & Business Media, 2012, vol. 95.

[4] V. Simoncini and D. B. Szyld, "Recent computational developments in Krylov subspace methods for linear systems," *Numerical Linear Algebra with Applications*, vol. 14, pp. 1–59, 2007.

[5] A. Borst and M. Egelhaaf, "Principles of visual motion detection," *Trends in neurosciences*, vol. 12, no. 8, pp. 297–306, 1989.

[6] P. Favaro, "Recovering thin structures via nonlocal-means regularization with application to depth from defocus," in *Computer vision and pattern recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 1133–1140.

[7] G. Aubert, R. Deriche, and P. Kornprobst, "Computing optical flow via variational thechniques," *SIAM Journal of Applied Mathematics*, vol. 60, no. 1, pp. 156–182, 1999.

[8] W. Sunada and S. Dubowsky, "On the dynamic analysis and behavior of industrial robotic manipulators with elastic members," *Journal of Mechanisms,*

Transmissions, and Automation in Design, vol. 105, no. 1, pp. 42–51, 1983.

[9] T. S. Sachs, C. H. Meyer, B. S. Hu, J. Kohli, D. G. Nishimura, and A. Macovski, "Real-time motion detection in spiral mri using navigators," *Magnetic resonance in medicine*, vol. 32, no. 5, pp. 639–645, 1994.

[10] Y. L. Tian and A. Hampapur, "Robust salient motion detection with complex background for real-time video surveillance," in *Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1. Seventh IEEE Workshops on*, vol. 2, Jan 2005, pp. 30–35.

[11] J. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images : a review," DTIC Document, Tech. Rep., 1988.

[12] S. Srinivasan, "Extracting structure from optical flow using the fast error search technique," *Int. J. Comput. Vision*, vol. 37, no. 3, pp. 203–230, Jun. 2000.

[13] A. Mitiche and H. Sekkati, "Optical flow 3D segmentation and interpretation : A variational method with active curve evolution and level sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1818–1829, Nov. 2006.