Inaugural-Dissertation

zur
Erlangung der Doktorwürde
der
Naturwissenschaftlich–Mathematischen Gesamtfakultät
der
Ruprecht–Karls–Universität
Heidelberg

vorgelegt von
M.Sc. Jing Yuan
aus Wuhan, China

Tag der mündlichen Prüfung:

Convex Variational Approaches to Image Motion Estimation, Denoising and Segmentation

Gutachter: **Prof. Dr. Christoph Schnörr**

To my homeland, China.

To my wife, Dandan Shan and my son, Danry Yuan.

Abstract

Energy minimization and variational methods are widely used in image processing and computer vision, where most energy functions and related constraints can be expressed as, or at least relaxed to, a convex formulation. In this regard, the central role is played by convexity, which not only provides an elegant analytical tool in mathematics but also facilitates the derivation of fast and tractable numerical solvers. In this thesis, four challenging topics of computer vision and image processing are studied by means of modern convex optimization techniques: non-rigid motion decomposition and estimation, TV-L1 image approximation, image segmentation, and multi-class image partition. Some of them are originally modelled in a convex formulation and can be directly solved by convex optimization methods, such as non-rigid flow estimation and non-smooth flow decomposition. The others are first stated as a non-convex model, then studied and solved in a convex relaxation manner, for which their dual models are employed to derive both novel analytical results and fast numerical solvers.

Non-rigid Flow Decomposition and Estimation: The analysis and estimation of non-rigid flow fields are of utmost importance and of academic interest in many areas, such as experimental fluid mechanics, remote sensing, medical imaging, and oceanography. The optical-flow approach constitutes an efficient non-intervening way to estimate such flow fields between image sequences. Standard optical-flow methods, which were developed for mostly rigid or piecewise rigid motion estimation and rely on minimizing the motion-regularized functionals, cannot be well adapted into the context of such non-rigid flows, which often contain highly complex motion patterns and cannot be described or represented by a simple parametric model. In this respect, higher-order div-curl based regularization was introduced in the optical-flow estimation energy to accurately recover complicated flow patterns with different scales. In this thesis, we propose a novel flow decomposition method in order to improve the analysis and numerical treatment of the reduced convex optimization model that incorporates the challenging high-order flow derivatives. For the estimation of physically consistent, e.g. divergence-free

flows, the proposed flow decomposition significantly reduces the computational complexity, and a fast multiplier-based algorithm can be derived using standard convex optimization theory. In addition, we study the new non-smooth high-order div-curl regularizer and show that it leads to a new structure-texture decomposition of the given non-rigid flow field. In particular, we investigate an approach combining flow estimation with such non-smooth decomposition, and demonstrate its advantages for restoring small-scale flow patterns. In order to minimize the resulting highly nonlinear convex energy function, a direct second-order cone programming (SOCP) based method is applied, whose results validate our propositions experimentally.

TV-L1 Image Approximation: We study the convex TV-L1 image approximation model under a new primal-dual perspective. In comparison to the classical TV-L2 image approximation [141], the TV-L1 image approximation method shows superior performance for impulsive denoising and for the scale determination of image patterns. We show that the TV-L1 image approximation results in a novel image decomposition model based on convex cones. Moreover, the TV-L1 image approximation can actually be viewed as a convex relaxation of the associated discrete-constrained image approximation problem. In this regard, it allows to solve the corresponding integer optimization problem globally and exactly. This significantly extends the results recently proposed by Chan et al [42] from simple binary-constrained TV-L1 image approximation to images with discrete grayscales, such as 8-bit gray-scale digital images. Concerning the computational aspects, we build up a new multiplier-based TV-L1 image approximation algorithm based on the proposed dual model, which avoids the non-smoothness of the primal TV-L1 energy function. Experiments indicate that it converges in a very fast, mostly super-linear way.

Image Segmentation: Image segmentation is a fundamental problem of computer vision and image processing, which has been intensively studied. One recent development on this topic is to investigate the binary constrained optimization problem in the spatially continuous setting by means of convex relaxation. This approach, known as continuous min-cut, outperforms level-set formulations in terms of efficiency and reliability, and has no metrication error in contrast to graph-based approaches such as graph-cuts. Despite many similarities between the continuous min-cut problem and the classical min-cut over graphs, its dual model, especially in a flow-maximization form, was rather not well-studied. In addition, a max-flow based algorithm was still missing, in contrast to the associated optimization over graphs: most fast min-cut solvers are designed by the theory of max-flow. In this thesis, we propose a novel continuous max-flow model and prove its duality to the continuous min-cut problem. We also provide a new variational perspective on the connection between 'cuts' and 'flow saturation'. Furthermore, the proposed continuous max-flow model naturally leads to a new max-flow based algorithm which significantly outperforms the state-of-the-art algorithm proposed by Bresson et al [31] in terms of efficiency. We also consider image segmentation with user-supplied constraints, and show that such supervised information can be easily incorporated into our continuous max-flow approach without introducing any additional computational load.

Multi-Class Image Partition: Multi-class image partition with a regularizer based on the minimum total perimeter can be expressed in terms of a Potts model. Typically, the associated energy is formulated on finite graphs and solved using α-expansion, which is often biased by the discrete grid and generates metrication artifacts. Existing convex relaxed formulations of the Potts model in the spatially continuous setting use a total-variation based functional to encode perimeter costs, in order to favor partitionings with smooth boundaries. Such formulations are analogous to the multi-way 'min-cut' problems over graphs. We study the convex relaxed Potts model and propose two convex models, both of which are dual and equivalent to the convex relaxed Potts problem. The first dual model can be smoothed using an entropy-maximization term and boils down to the maximization of a simple smooth and convex energy function, which can be numerically performed in an inexpensive and fast way. The second model amounts to flow maximization in the spatially continuous context, i.e. a continuous max-flow model, which leads to a novel flow perspective of the 'multi-way cut' problem. In addition, the continuous max-flow formulation directly leads to a new and efficient max-flow based algorithm which has significant numerical advantages: the new algorithm properly avoids extra computational load to enforce the pointwise simplex constraints and naturally allows a parallel implementation over different labels. Numerical experiments show substantial superiority of both approaches in terms of quality and efficiency, compared to previous state-of-the-art methods for Potts model based image partition.

Zusammenfassung

Methoden der Energieminimierung und der Variationsrechnung werden auf dem Gebiet der digitalen Bildverarbeitung und des Computersehens vielfach eingesetzt. Dort können die meisten Energiefunktionen und die zugehörigen Nebenbedingungen in konvexer Form ausgedrückt oder zumindest durch Relaxation in eine solche überführt werden. In dieser Hinsicht ist Konvexität von zentraler Bedeutung und stellt nicht nur ein elegantes mathematisches Werkzeug für die Analyse zur Verfügung, sondern unterstützt auch die Entwicklung schneller und leicht handhabbarer numerischer Löser. In dieser Arbeit werden vier herausfordernde Themen des Computersehens und der Bildverarbeitung mittels moderner konvexer Optimierungstechniken untersucht: nicht-starre Zerlegung und Schätzung von Bewegung, TV-L1 Bild-Approximation, Bildsegmentierung sowie Multiklassen-Bildsegmentierung. Manche sind von Haus aus konvex formuliert und können direkt durch konvexe Optimierungstechniken gelöst werden. Darunter fallen z.B. die nicht-starre Schätzung und die nicht-glatte Zerlegung von Bewegungen. Die verbleibenden Probleme werden zunächst nicht-konvex formuliert und dann mit Hilfe konvexer Relaxation untersucht und optimiert. Hierzu werden ihre dualen Modelle verwendet, um sowohl neue theoretische Erkenntnisse als auch schnelle numerische Löser zu erhalten.

Nicht-starre Zerlegung und Schätzung von Bewegung: Die Analyse und Schätzung von nicht-starren Bewegungsfeldern sind in vielen Disziplinen von äußerster Wichtigkeit und von akademischer Bedeutung, z.B. in der experimentelle Strömungsmechanik, Fernerkundung, bildgebenden Verfahren für der Medizin und in der Ozeanographie. Ansätze basierend auf dem optischen Fluss stellen eine effiziente, berührungslose Methode für die Schätzung von Flussfeldern in Bildsequenzen dar. Gewöhnliche auf dem optischen Fluss basierende Ansätze, die für die Schätzung von fast-starrer oder stückweise starrer Bewegung entwickelt wurden und auf der Minimierung von Energiefunktionen, die die Bewegung regularisEren, basieren, können nicht direkt auf nicht-starre Flüsse angepasst werden, da diese oft hoch-komplexe Bewegungsbilder beinhalten und nicht durch einfache parametrische Modelle beschrieben oder repräsentiert werden

können. In diesem Zusammenhang wurde auf div-curl basierende Regularisierer höherer Ordnung für die Energiefunktion der Optischen-Fluss-Schätzung vorgeschlagen, um Bewegungsbilder mit unterschiedlichen Skalen exakt zu rekonstruieren. In dieser Arbeit schlagen wir eine neuartige Methode zur Zerlegung des Flusses vor. Das Ziel ist, die Analyse und die numerische Behandlung des reduzierten konvexen Optimierungsproblem zu verbessern, welches anspruchsvolle Ableitungen höherer Ordnung des Flusses beinhaltet. Für die Schätzung physikalisch konsistenter, beispielsweise divergenzfreier Flüsse reduzieren die vorgeschlagenen Flusszerlegungen die Berechnungskomplexität signifikant und es lässt sich ein schneller Algorithmus basierend auf Lagrange-Multiplikatoren mit Hilfe von existierenden Theorien aus der konvexen Optimierung herleiten. Zusätzlich untersuchen wir den neuen nicht-glatten div-curl Regularisierer höherer Ordnung und zeigen, dass er zu einer neuartigen Zerlegung des gegebenen nicht-starren Flussfeldes in Struktur- und Texturanteil führt. Insbesondere untersuchen wir einen Ansatz, der Flussschätzung mit einer solchen nicht-glatten Zerlegung kombiniert und zeigen seinen Nutzen für die Rekonstruktion von kleinskaligen Strömungsbildern. Um die resultierende hochgradig nicht-lineare konvexe Energiefunktion zu minimieren, wenden wir direkt Second-Order Cone Programming an. Die Resultate bestätigen unseren Vorschlag experimentell.

TV-L1 Bild-Approximation: Wir untersuchen das konvexe, auf TV-L1 basierende Modell zur Bild-Approximation aus einer neuen primal-dualen Sichtweise. Im Vergleich zu der klassischen TV-L2 Bild-Approximation [141] liefert die TV-L1 Bild-Approximation für die Rauschunterdrückung von Impulsrauschen und die Skalen-Bestimmung von Bildmustern hervorragende Ergebnisse. Wir zeigen, dass die TV-L1 Bild-Approximation zu einem neuartigen Zerlegungsmodell für Bilder basierend auf konvexen Kegeln führt. Zudem kann die TV-L1 Bild-Approximation sogar als eine konvexe Relaxation des zugehörigen diskret-wertigen Bild-Approximationsproblems betrachtet werden. In dieser Hinsicht lässt sich das zugehörige ganzzahlige Optimierungsproblem global und exakt lösen. Dies erweitert maßgeblich die kürzlich von Chan et al. [42] veröffentlichten Ergebnisse von der einfachen binärwertigen TV-L1 Bild-Approximation auf Bilder mit diskreter Grauskala, z.B. digitale Bilder mit 8 Bit Kodierung der Intensität. Bezüglich der Berechnung haben wir einen neuen Multiplikator-basierten Algorithmus für TV-L1 Bild-Approximation entwickelt, welcher auf dem vorgeschlagenen dualen Modell basiert und so verhindert, dass die TV-L1 Energiefunktion nicht-glatt ist. Die Experimente zeigen, dass er sehr schnell, meist super-linear, konvergiert.

Bildsegmentierung: Bildsegmentierung ist eine wichtige und ausgiebig erforschte Problemstellung auf dem Gebiet des Computersehens und der Bildverarbeitung. Eine jüngste Entwicklungen auf diesem Gebiet ist, das binärwertige, räumlich kontinuierlich definierte Optimierungsproblem mithilfe konvexer Relaxation zu untersuchen. Dieser Ansatz ist unter dem Namen Min-Cut bekannt, übertrifft Level-Set-Formulierungen in Bezug auf Effizienz und Zuverlässigkeit und hat im Gegensatz zu Graph-basierten Ansätzen, wie z.B. Graph-Cut-Methoden, keine durch die Diskretisierung verursachte Artefakte. Trotz vieler Gemeinsamkeiten zwischen dem kontinuierlichen Min-Cut-Problem und dem klassischen Min-Cut-Problem auf Graphen, war hingegen das zugehörige duale Modell, insbesondere in der Max-Flow-Formulierung, kaum erforscht. Weiterhin existierte bisher kein auf Max-Flow basierender Algorithmus, ganz im Gegensatz zu der zugehörigen Optimierung über Graphen: Die meisten schnellen Min-Cut-Löser werden basierend auf der Theorie des Max-Flows entwickelt. In dieser Arbeit schlagen wir ein neuartiges kontinuierliches Max-Flow Modell vor und zeigen seine Dualität zum kontinuierlichen Min-Cut Problem. Weiterhin betrachten wir die Verbindung zwischen 'Cuts' und 'Flows' aus einer neuen, auf Variationsmethoden basierenden Perspektive. Darüber hinaus führt das vorgeschlagene kontinuierliche Max-Flow Modell ganz natürlich zu einem neuen, auf Max-Flow basierenden Algorithmus, welche den modernen, von Bresson et al. [31] vorgeschlagenen Algorithmus in Bezug auf Effizienz übertrifft. Weiterhin betrachten wir Bildsegmentierung mit Benutzervorgaben und zeigen, dass solche überwachten Informationen sehr einfach und ohne zusätzlichen Rechenaufwand in unseren kontinuierlichen Max-Flow Ansatz miteinbezogen werden können.

Multiklassen-Segmentierung von Bildern: Multiklassen-Bildsegmentierung mit einem Regularisierer basierend auf dem kleinsten Gesamtumfang können mithilfe des Potts-Modells ausgedrückt werden. Gewöhnlich wird die zugehörige Energiefunktion auf endlichen Graphen definiert und mithilfe von α-Expansion gelöst, welche oft zu systematischen Fehlern und Artefakten verursacht durch die Diskretisierung führt. Existierende konvex-relaxierte, räumlich-kontinuierliche Formulierungen des Potts-Modells benutzen ein auf Totaler Variation basierendes Funktional für die Kodierung der Kosten für den Umfang, um Segmentierungen mit glatten Grenzen zu begünstigen. Solche Formulierungen sind analog zu Multi-Way-Min-Cut Problem auf Graphen. Wir untersuchen das konvex-relaxierte Potts-Modell und schlagen zwei konvexe Modelle vor, welche beide dual und äquivalent zu dem konvex-relaxierten Potts-Problem sind. Das erste duale Modell kann mithilfe eines Entropie-Maximierungs-Terms geglättet werden, was darauf hinaus läuft, eine einfache, glatte und konvexe Energiefunktion zu maximieren, was numerisch günstig und schnell ist. Das zweite Modell läuft auf eine räumlich-kontinuierliche Fluss-Maximierung hinaus, also einem kontinuierlichen Max-Flow-Modell, welches zu einer ganz neuen Sichtweise auf das Multi-Way-Cut Problem führt. Zusätzlich führt die kontinuierliche Max-Flow-Formulierung direkt zu einem neuen und effizienten Max-Flow-basierten Algorithmus, welcher bedeutende Vorteile bezüglich der Numerik hat: Der neue Algorithmus benötigt keinen zusätzlichen Rechenaufwand, um die positionsweisen Simplex-Nebenbedingungen zu er-

zwingen und ermöglicht ganz natürlich eine parallele Implementierung bezüglich verschiedener Labels. Numerische Experimente zeigen, dass die beiden Ansätze bis dato aktuellen Algorithmen für Bildsegmentierung basierend auf Potts-Modellen in Bezug auf Qualität und Effizienz substanziell überlegen sind.

Acknowledgements

Doing a PhD is a long and exhausting road which in fact starts already long time before the first word has been written on the first page of the thesis. At all, to accomplish such a distance whose completion is a matter of years is only possible with an adequate support from many people. Therefore, I would like to take the opportunity here to thank all the people who stood by my side during my time as a PhD student, some of them even made this work possible in the first place.

First and foremost, I wish to express my gratitude to Prof. Christoph Schnörr, a gentleman, who led me into the field of variational methods and computer vision. Then, I find his mathematical access to the problems involved to a great extent impressing as well as exciting. His scientific working style and his teaching elegance have shaped me a lot and I cannot thank him enough for all I could learn from him.

I wish to show my great acknowledgements to Prof. Gabriele Steidl, whose serious discussion style and rigorous revisions to my preprints indeed helped me in expressing ideas in a strict mathematician way. Discussions with her actually inspired and motivated many further improvements and followed ideas which form an important part of the contents in this thesis.

Thanks are due to Timo Kohlberger, my colleague at the department and comrade-in-arms in the field of variational motion estimation, for the very pleasant collaboration and the fruitful, oftentimes long discussions which we had over the first four years of my staying in Germany and which frequently brought confidence at the very beginning of my work and also made considerable progress to my work.

I would like to give my sincere thanks to Christian Schellewald and Annette Stahl, who are my colleagues and real pacifisms. Their warmhearted personality and optimism always encouraged me, not only in my studies but also in my living in Germany.

For the delightful time we spend together, I wish to thank all my colleagues accompanying me while being a PhD student: Florian Becker, Martin Bergtholdt, Dirk Breitenreicher, Prof. Daniel Cremers, Christian Gosch, Matthias Heiler, Rezaul Karim, Jörg Kappes, Jens Keuchel, Jan Lellmann, Stefania Petra, Paul Ruhnau, Stefan Schmidt, Thomas Schüle, Andriy Vlasenko, Jennifer Wagner, Stefan Weber. In this context, I also would like to thank Rita Schieker and Evelyn Wilhelm, whose great helps and care were taken to me and my family in such a kind way. I wish to have the chance to see and thank them again personally.

Last but not least, I wish to thank my wife Dandan Shan for her patience and giving me exceptional support such that I could proceed with writing my thesis. Without her encourages in time, I could give up my thesis sometime before its finish. Further, I would like to thank our son Danry for being just the way he is and for always providing me with new strength in his own particular way.

Notations

Throughout this work the following notation will be used, if not explicitly stated otherwise:

$\mathbb{Z}$	Intergers .
$\mathbb{R}$	Real numbers.
$\mathbb{Z}^n$, $\mathbb{R}^n$	The integer or real vectors with n components.
$\Omega \subset \mathbb{R}^2$	An open, bounded and simply-connected domain with Lipschitz-continuous boundary $\partial\Omega$.
$\lvert\Omega\rvert$, $\lvert\partial\Omega\rvert$	The area or perimeter of the given domain Ω.
$\mathrm{div}\,\mathbf{v} := \frac{\partial v_1}{\partial x_1} + \frac{\partial v_2}{\partial x_2}$	The divergence of the 2D vector field $\mathbf{v} := (v_1, v_2)^{\mathsf{T}}$.
$\mathrm{curl}\mathbf{v} := \frac{\partial v_2}{\partial x_1} - \frac{\partial v_1}{\partial x_2}$	The curl of the 2D vector field $\mathbf{v} := (v_1, v_2)^{\mathsf{T}}$.
$\nabla\psi := \left(\frac{\partial\psi}{\partial x_1}, \frac{\partial\psi}{\partial x_2}\right)^{\mathsf{T}}$	The gradient of the scalar field ψ.
$\nabla^{\perp}\phi := \left(\frac{\partial\phi}{\partial x_2}, -\frac{\partial\phi}{\partial x_1}\right)^{\mathsf{T}}$	The perpenticular gradient of the scalar field ϕ.
$\triangle\psi \ := \ \frac{\partial^2\psi}{\partial x_1^2} + \frac{\partial^2\psi}{\partial x_2^2}$	The Laplacian of the scalar field ψ.
$\triangle\mathbf{u} \ := \ (\frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2})\mathbf{u}$	The Laplacian of the 2D vector field $\mathbf{u} := (u_1, u_2)^{\mathsf{T}}$.
A, B, C	Arbitrary sets.
$\mathcal{L} := \{\ell_1, ..., \ell_n\}$	the label set where ℓ_i, $i = 1, ..., n$, denote the n labels.

$u_i(x)$, $i = 1, ..., n$ The n labeling functions correspond to labels ℓ_i, $i = 1, ..., n$ respectively.

$\mathcal{G} := \{\mathcal{V}, \mathcal{E}\}$ A graph including the node set $\mathcal{V}$ and the edge set $\mathcal{E}$, where V and E denote the number of nodes and edges respectively.

$i \in \mathcal{V}$ Index for nodes

$(i, j) \in \mathcal{E}$ An edge linking the two adjacent nodes i and j.

$\mathcal{N}(i)$ $\mathcal{N}(i)$ gives the set of all neighbours of the node $i \in \mathcal{V}$.

$\triangle_+ \in \mathbb{R}^n$ The simplex set: for $\forall u \in \triangle_+$, $u_i \geq 0$ and $\sum_{i=1}^n = 1$.

$h^*(y)$ For the given convex function $h(x)$, $h^*(y)$ gives its conjugate such that

$$h^*(y) := \max_x \langle y, x \rangle - h(x) .$$

$\mathrm{I}_\Omega(x)$ The indicator function of the area Ω is defined by

$$\mathrm{I}_\Omega(x) := \begin{cases} 0 & \text{when } x \notin \Omega \\ 1 & \text{when } x \in \Omega \end{cases} . \tag{0.1}$$

$\mathrm{N}_C(x)$ $\mathrm{N}_C(x)$ gives the normal cone of the convex set C at some point $x \in C$.

Contents

1 Introduction

During these decades, energy minimization and variational methods were developed to be an utmost important tool to mathematically model practical problems of computer vision and image processing and numerically solve them in an efficient way. Successful cases include image denoising and approximation [141, 42, 123, 8, 180], image decomposition [9, 115, 43, 10], optical flow estimation [80, 168, 32, 33, 185], image segmentation/partition [124, 31, 108, 131, 13, 174] etc. In this regard, most problems can be modeled by the minimization of a convex energy function over certain convex constraint, i.e.

$$\min_{u \in C} \; E(u) \tag{1.1}$$

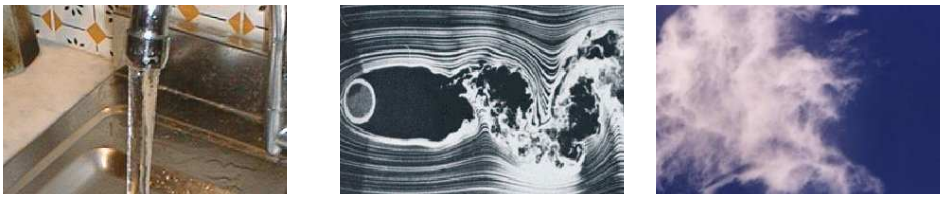
where C is some convex set. Therefore, modern convex optimization techniques can be applied directly to tackle the reduced convex minimization problem (1.1), where convex optimization not only provides an mathematical soundel way to analyze the given problem, but also leads to fast and tractable solvers in numerics.

In this thesis, we study four challenging topics of computer vision and image processing: *non-rigid flow decomposition and estimation, TV-L1 image approximation, image segmentation and multi-class image partition*. We focus on the associated convex optimization methods and convex relaxation approaches to the problems which are not originally formulated in the manner of convex optimization. Especially, we concentrate on analyzing through their respective dual form and constructing efficient algorithms in a primal-dual way. In contrast to the widely-used PDE-descent method which often amounts to inefficient sequential gradient-descent steps and shows a lack of deep insights to the original mathematical formulation, the efficient primal-dual solvers possess an prominent category, e.g. [188, 176, 12, 147, 175] (see also the recent works [38, 60] for a good reference).

In the following part of this chapter, we give a short introduction and review of these topics and summarize our contributions in the last section.

1.1 Non-rigid Flow Decomposition and Estimation

Non-rigid flows can be widely found in many real environments: water streams along a river, clouds drift in the sky, or smoke ascends out of a chimney etc. Movements of these objects are sometimes quite 'simple' and smooth, such as water falling from the faucet (see the left picture of Fig. 1.1), its falling speed field can be modeled by a simple one-dimensional function. But mostly such flows appear in a very 'complex' and irregular way, such as turbulents and blowing clouds (see the last two pictures of Fig. 1.1). It is very hard to decribe or formulate these movement fields by a single parametric model. This is in contrast to motion of rigid bodies which can be easily expressed by piecewise constant or piecewise polynomial functions [161]. In addition, taking a close look to such fluid flows one can find that a non-rigid flow field often contains flow patterns with different scales. For instance, the turbulent flow shown in the second picture of Fig. 1.1 clearly exposes plenty of "circulations" of different sizes behind the cylinder.



Smooth Water Stream **Turbulent Flows** **Coulds**

Figure 1.1: **Typical Non-rigid Flows**

On the other hand, the analysis and accurate estimation of non-rigid fluid flows are very important and useful in many scientific and industrial areas, e.g. estimation of fluid flow fields greatly helps to recover distributions of fluid mechanical features and improves related researches in experimental fluid mechanics [134, 136, 104], environmental sciences like meteorology [187], climatology and oceanography [48, 50].

During recent years, estimating dense non-rigid flow fields based on variational anlysis of image sequences has been developped and received a great attention in the community of computer vision [136, 48, 104, 177, 187]. It provides a both mathematically well-posed and numerically efficient way for the analysis and estimation of non-rigid flows. In parallel, the cross-correlation method also gives one of the state-of-the-art computational methods for fluid motion estimation, e.g. *particle image velocimetry*, and benefits from its robustness against noise and illumination disturbances. Most recent progresses of cross-correlation method is to find appropriate alternatives of the fixed square correlation windows [125, 53, 160], so as to improve the spatial resolution of estimation. For example, Becker et al [18, 17] suggested a bi-level optimization-based method which simultaneously adapts the gaussian weighted window parameters and estimates the motion field. We refer [136] for an excellent overview of the vast number of literature on all aspects of the application of cross-correlation for PIV. Comparing to the cross-correlation method, the variational-based method is superior in its high-accuracy and the ability to incorporate physics-consistent models and conditions [76, 166, 142].

In this thesis, we focus on the variational approach to dense flow estimation, which computes the non-rigid flow field synthesized in the two sequential images $I_{1,2}(x)$ by solving the following energy minimization problem

$$\min_{\mathbf{u}} \; D(I_1(x+\mathbf{u}) - I_2(x)) \, + \, R(\mathbf{u}) \tag{1.2}$$

where the data fidelity term $D(\cdot)$ evaluates the brightness-constant condition and the flow regularization term $R(\mathbf{u})$ smoothes out the computation result. In (1.2), the regularizar $R(\mathbf{u})$ encodes the priori information about the spatial coherence of the flow $\mathbf{u}$ and plays the central role in the variational flow estimation method. It was shown that the widely-used regularizars of optical-flow estimation [32, 33, 168], which implicitly assume simple parametric models of $\mathbf{u}$ like piecewise constant, failed to extract the correct non-rigid flow fields [177, 179]. Even for the most smooth harmonic flow field $\mathbf{u}$, whose divergence and curl fields are both vanishing, such regularizars produce pool results because the incorrect regularization functions usually oversmooth and distort the flow patterns. In contrast, the high-order div-curl regularizar

$$R(\mathbf{u}) \, := \int_\Omega |\nabla \operatorname{div} \mathbf{u}|^2 \; dx + \int_\Omega |\nabla \operatorname{curl} \mathbf{u}|^2 \; dx \tag{1.3}$$

achieved great successes and received much attention [48, 94, 181], which penalizes the variance of divergence and curl fields instead of the flow coordinates directly. In this thesis, we study such high-order div-curl regularized non-rigid flow estimation through flow decomposition, which allows the recovery of physics-consistent imagery flow fields, e.g. solenoidal flows (div-free), in a much simpler way. The proposed flow decomposition schemes also leads to in-depth analyzes and descriptions of non-rigid flow patterns. In addition, it results in estimating the flow field by computing two potential fields, related to divergence and curl respectively. It properly splits the original flow estimation (1.2) into to smaller sub-problems in numerics with respect to the two potentials.

For the highly non-rigid flow fields which are mostly piecewise smooth, e.g. turbulent flows, we investigate the nonsmooth version of (1.3):

$$R(\mathbf{u}) := \int_\Omega \sqrt{|\nabla \operatorname{div} \mathbf{u}|^2 + |\nabla \operatorname{curl} \mathbf{u}|^2}\, dx\,, \tag{1.4}$$

which was first proposed by Yuan et al [181, 179]. We introduce a new structure-texture decomposition of non-rigid flow fields based on the non-smooth div-curl regularizar (1.4), which seperates the given flow into two flow patterns with different scales. This is analougue to the structure-texture image decomposition of the given image f based on total-variation, e.g. [9, 10, 115]:

$$f = v + \operatorname{div} p\,, \tag{1.5}$$

where v scatches the large-scale 'structure' pattern and $\operatorname{div} p$ denotes the small-scale 'textures' described by its G-norm. The proposed structure-texture flow decomposition provides a new variational perspective to analyzing the challenging flow patterns with high non-smoothness and different scales. On the other hand, such nonsmooth structure-texture flow decomposition introduces a complicated convex optimization problem along with nonsmooth convex constraints, therefore a direct convex programming algorithm based on the second-order cone programming (SOCP) is employed to achieve computational results in a high-accuracy.

1.2 TV-L1 Image Approximation

Total-variation based image denoising was first proposed by Rudin et al [141] as the following convex optimization problem

$$\min_u \frac{1}{2} \int_\Omega |f - u|^2 \, dx + \alpha \int_\Omega |\nabla u(x)| \, dx\,. \tag{1.6}$$

Unlike most convex regularizars for image smoothness, the total-variation function is well-known in keeping image edges while smoothing other parts. Since then, the total-variation regularizar was widely used in many tasks of image processing, e.g. image denoising [41, 42, 8, 180], image decomposition [9, 115, 43], image segmentation and partition [42, 31, 174, 132, 13, 108] for which we will go into more details in the relevant parts of this thesis. Along with vast developments of convex optimization techniques in image processing, the minimization of such a total-variation based energy function is continuously of great interests and importance [141, 149, 115, 31, 126, 34, 81, 78].

In this thesis, we investigate the TV-L1 image approximation problem:

$$\min_u \int_\Omega |f - u| \, dx + \alpha \int_\Omega |\nabla u(x)| \, dx\,, \tag{1.7}$$

which achieves much attention since the pioneering works [42, 41] by Chan et al. In fact, the TV-L1 energy function of (1.7) was first introduced and studied by Alliney [2, 1] for discrete one-dimensional signals' denoising, then studied by Nikolova et al [123, 41] for image processing. In comparison to the classical TV-L2 problem (1.6), TV-L1 image approximation outperforms in denoising impulsive noises (see Fig. 1.2), keeping image contrast and determing scales of image patterns [172, 173, 144].

Another interesting property of the TV-L1 model (1.7) was shown by Chan et al [42, 41]: given the binary image $f(x) \in \{0,1\}$, there exists at least one optimum $u^*(x) \in \{0,1\}$ which minimizes (1.7) globally. It follows that for the given the black-white image $f(x)$, the convex TV-L1 formulation (1.7) solves the nonconvex optimization problem:

$$\min_{u(x) \in \{0,1\}} \int_\Omega |f - u| \, dx + \alpha \int_\Omega |\nabla u(x)| \, dx \,, \tag{1.8}$$

globally and exactly! Therefore, (1.7) gives an exact convex relaxation of the binary constrained optimization problem (1.8). Moreover, the rounding scheme introduced by Chan et al actually implies a series of global optimums of the binary-constrained TV-L1 model (1.8) may be discovered.

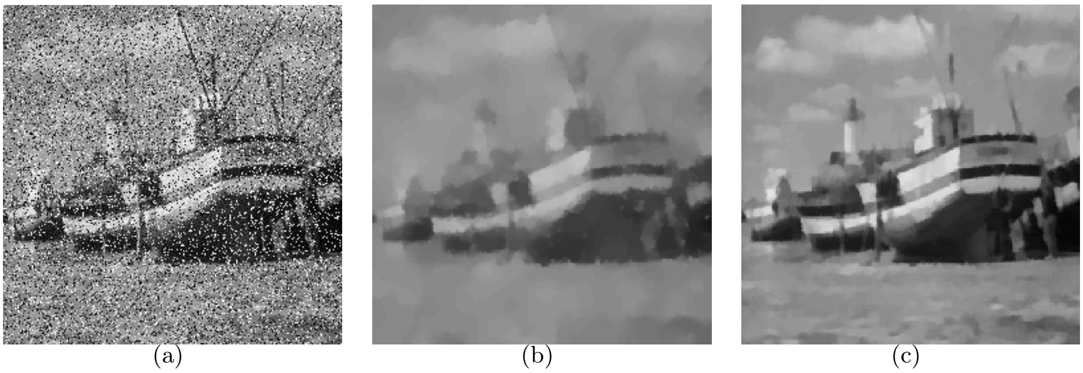


Figure 1.2: **Difference btw. TV-L2** (1.6) **and TV-L1** (1.7)**:** (a) Input noisy image $f(x)$. (b) Computation result $u^*(x)$ by TV-L2 image approximation (1.6). (c) Computation result $u^*(x)$ by TV-L1 image approximation (1.7).

With the help of coarea formula, Chan et al [41, 42] proved that the energy functional $P(u)$ of (1.7) can be equivalently represented in terms of the upper level-sets of the image functions $u(x)$ and $f(x)$, i.e.

$$P(u) = \int_{-\infty}^{+\infty} \left\{ |U^\gamma \triangle F^\gamma| + \alpha \, |\partial U^\gamma| \right\} d\gamma \,, \tag{1.9}$$

where for each γ, U^γ gives the γ-upper level set of the variable $u(x)$ by

$$U^\gamma(x) = \begin{cases} 1 \,, & \text{when } u(x) > \gamma \\ 0 \,, & \text{when } u(x) \leq \gamma \end{cases} \,, \quad x \in \Omega \,, \quad i = 1, \ldots, n \,; \tag{1.10}$$

and so for F^γ to the input image $f(x)$. $|\partial U^\gamma|$ denotes the perimeter of U^γ and $|U^\gamma \triangle F^\gamma|$ gives the area of the symmetric difference of the two level sets U^γ and F^γ.

Yin et al [173] further showed that minimizing the layer-wise energy function (1.9) actually amounts to properly stacking all the optimal U^γs, each of which solves (1.8) for the corresponding binary indicator function F^γ. In other words, solving (1.7) can be reduced to optimizing a sequence of binary constrained problems as (1.8). Since $U^{\gamma_1} \subset U^{\gamma_2}$ when $\gamma_1 \geq \gamma_2$, the process recovers the optimum $u^*(x)$ of (1.7) by properly arranging all the associated level sets U^γ, $\gamma \in (-\infty, +\infty)$. The same result was also discovered by Darbon et al. [51, 52] in the image graph setting, where the anisotropic total-variation function was considered in consistent to the given grid graph and resulted in a graph cut problem which can be solved by the introduced fast graph optimization algorithm. Goldfarb and Yin [71] also developped an efficient pre-flow based min-cut approach to such anisotropic total-variation regularized L1 image approximation. Another similar and interesting work of the spatially continuous image labeling along with n linearly ordered labels, i.e. layered level sets, was recently addressed by Bae et al [12], which simulated Ishikawa's graph-cut method [87] and proposed a fast continuous max-flow approach.

However, as stated in [173], such approach means both bad and good news for processing gray-scale images in practice: on the one hand, the total number of gray values is finite, i.e. $u(x) \in \{0, \ldots, 255\}$, hence only a finite number of optimization problems as (1.8) should be considered; on the other hand, solving (1.8) for each layer F^γ is not trivial; and in order to globally tackle (1.7), one has to examine a large number of obtained level-sets to restrict its search legally. This makes such computation method by direct addressing multiple layered level-sets impratical to a real image processing task especially with a large number of different gray values, e.g. CT or MRI medical imaging which often contains over thousands of gray values.

In addition, the PDE-descent method is often used to numerically approximate the global optimum of (1.7) [42, 41, 173, 55], which smoothes the total-variation term by $\sqrt{\partial_x u^2 + \partial_y u^2 + \epsilon^2}$. Actually as mentioned in [56], even if ϵ takes a small enough value, the coarea formula is no longer satisfied, new gray levels appear and the indicator functions are blurred.

Motivated by the above observations, we introduce and study the primal and dual perspective of the TV-L1 model (1.7). We show the it results in a new image decomposition model, in contrast to the image decomposition [9, 115] by the TV-L2 model (1.6). In addition, we prove the exactness of (1.7) as the convex relaxation model of the discrete-constrained optimization problem:

$$\min_{u(x) \in \{f_1, \ldots, f_n\}} \int_\Omega |f - u| \, dx + \alpha \int_\Omega |\nabla u(x)| \, dx \,, \tag{1.11}$$

given $f(x) \in \{f_1, \ldots, f_n\}$ with the order $f_1 < \ldots < f_n$, i.e. (1.7) gives the global and exact optimum of (1.11). Clearly, the assumption that $u(x)$ and $f(x)$ takes the discrete gray-scale values f_i, $i = 1, \ldots, n$, with an ascent order such that $f_1 < \ldots < f_n$, properly models the 8-bit gray-scale images in digital image processing and CT or MRI medical imaging which contains over thousands of discrete gray-scales. It follows that once the input $f(x)$ is discretely valued in a finite set with linear orders, which is often the case for digital gray-scale images, one can find the global optimum $u^*(x)$ to (1.11), which has the same discrete values' set, by simply solving the convex optimization problem (1.7). That is one can simply solve the challenging nonconvex discrete-constrained optimization problem (1.11), exactly and globally, by a much simpler convex optimization problem (1.7), which significantly reduce the related computation complexity and load.

Moreover, we also derive a new fast algorithm based on the proposed dual model of (1.7) through the standard multiplier-based convex optimization theories [20, 21, 138].

1.3 Image Segmentation

Image segmentation is a central topics of image processing, which has been intensively studied through different theories during these decades. In this thesis, we regard image segmentation as a low-level vision problem and model it in the form of energy minimization through Markov Random Fields (MRF) (see [130, 110] for a good reference), such that the task of image segmentation tries to assign the value 0 or 1 to each image pixel subject to certain optimal criterion, where 0 means 'background' and 1 'forground'. That is image segmentation seperates the given image domain Ω into two regions, namely background and foreground (see images (a) and (b) of Fig. 1.3). The topics of image segmentation with multiple partitions is discussed indepently in another part of this thesis.

A convenient way to mathematically formulate such binary image segmentation problem is to compute the min-cuts over graphs, where, for each image grid, it segments grid nodes through the minimization of the total data cost subject to a smooth segmentation boundary. Its reduced discrete energy function is encoded over an appropriately designed graph such that searching for a min-cut over this graph just amounts to the minimization of the energy function. The

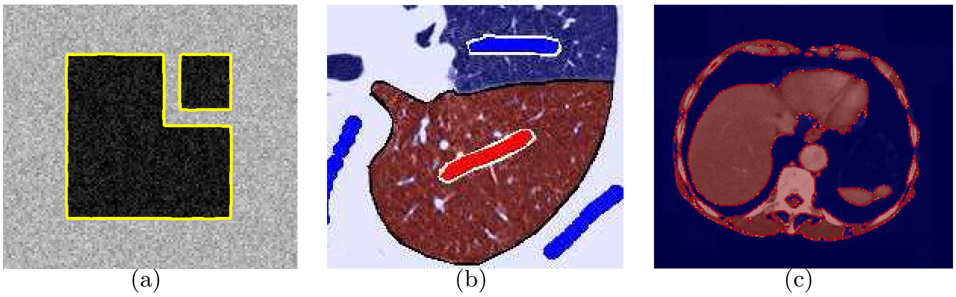

(a) (b) (c)

Figure 1.3: **Examples of binary image segmentation:** (a) shows the two segmentation regions (outside or inside the yellow contour) [31]. (b) shows the two regions of lobe (marked by red and blue) [28]. (c) shows the two segments of a liver (marked by blue and red) [176].

main advantage for expressing image segmentation in the form of min-cuts is that the min-cut problem can be efficiently computed by means of the maximization of corresponding flows along the graph through the classical duality of min-cut and max-flow [64, 46]. In practice, most fast min-cut algorithms are designed in the manner of recovering the maximum flow over the graph network, e.g. Edmonds-Karp algorithm [57] in $O(VE^2)$, push-relabel algorithm [69] in $O(V^2E)$!

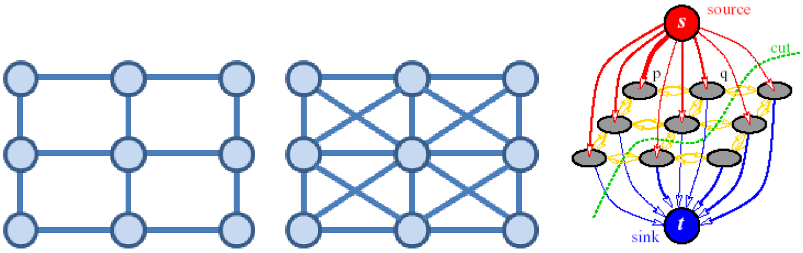


Figure 1.4: **Graphs and graph-cuts:** 1st and 2nd images show the graphs constructed from image grids: 4-connected and 8-connected respectively. 3rd image shows the minimum cut of the graph, where the nodes of the graph are partitioned into two subsets belonging to the source s and the sink t respectively.

There have been a vast amount of researches on the graph-cut based image segmentation during the past ten years, e.g. [28, 30, 140]. One main drawback of such graph-based approaches is visible grid bias generated in segmentation results, where the interaction potential penalizes some spatial directions more than others and this leads to visible artifacts in computational results (see images (a) and (c) of Fig. 1.5). Reducing such metrication errors can be done by considering more neighboring nodes with a proper distance measure [29, 96] (see images (b) and (d) of Fig. 1.5 with comparisons to images (a) and (c)) or high-order interaction potentials [95, 88]. However, this either results in a heavy memory load and high computation cost or amounts to a more complex algorithmic scheme, e.g. QPBO [23, 98].

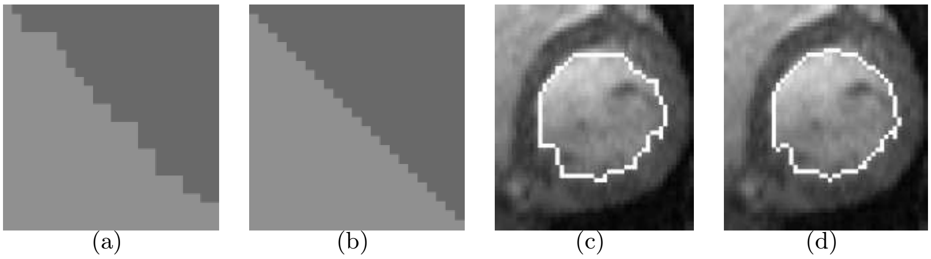


Figure 1.5: **Metric effects by min-cuts [29]:** (a) and (b) show the restoration results computed by the graph with 4-connected neighbours and the graph with 8-connected neighbours respectively. (c)and (d) show segmentation results computed by the graph with 4-connected neighbours and the graph with 26-connected neighbours respectively. Clearly, applying a large neighbour node system allows to reduce inaccuracy from metrification errors.

In contrast, formulating and minimizing the energy functional over the labeling function under a spatially continuous setting can properly avoid such metric biases and carries the solution with a subpixel accuracy at the same time. In this regard, classical approaches, e.g. level sets [127, 40, 112, 109] and phase fields [90, 19], give most direct and cheap implementations. Unfortunately, both methods suffer from their highly non-convex energy function: computation often gets trapped on the local optimum and the quality of results highly depends on the initial condition. Recent studies [42, 124] showed that expressing the spatially continuous min-cut problem in the form of convex relaxation leads to both the global and exact solution and the fast and reliable solvers in numerics [31, 72, 174]. G. Strang [150, 151] was the first to study max-flow and min-cut problems over a continuous image domain. Related studies include [158, 4], where Appleton et al proposed an edge-based continuous minimal surface approach to segmenting 2-D and 3-D objects.

In [42, 124], Chan et al considered binary image segmentation with the spatially continuous notations:

$$\min_S \int_{\Omega \backslash S} C_s(x)\, dx + \int_S C_t(x)\, dx + \alpha\, |\partial S| \;, \tag{1.12}$$

then relaxed the indicator function $u(x) \in \{0,1\}$ of S to $u(x) \in [0,1]$, Chan et al proved that the binary-constrained nonconvex formulation (1.12) can be globally solved by the convex minimization problem

$$\min_{u(x)\in[0,1]} \int_\Omega (1-u(x))C_s(x)\,dx + \int_\Omega u(x)C_t(x)\,dx + \alpha \int_\Omega |\nabla u(x)|\,dx\,. \tag{1.13}$$

More specifically, threshholding the optimum $u^*(x) \in [0,1]$ of (1.13) by any value $t \in (0,1]$ leads to a sequence of global binary optima to (1.12). Consequently, the optimum of the convex optimization problem (1.13) gives rise to a set of global binary solvers to the original nonconvex segmentation problem (1.12), not just one which is the case for graph-cuts! In this sense, (1.13) is also named as the *continuous min-cut model*.

However, in contrast to the duality between discrete max-flow and min-cut models [64] where fast min-cut algorithms are designed in the max-flow fashion [46], the corresponding max-flow model over a continuous image domain, as the dual of (1.13), is rather lost in recent studies, except Yuan et al [174, 176]. On the other hand, to tackle the constraints $u(x) \in [0,1]$ at each image pixel in research so far, previous algorithms were designed to explicitly force the primal variable $u(x)$ to the feasible set at every iteration, either by projections or by adding forcing terms [124, 31, 72]. This is also in contrast to the classical min-cut and max-flow scheme, where the min-cut problem can be completely solved by just pushing flows as much as possible without modifying any label during computation. It helps to design most efficient min-cut algorithms, e.g. the Ford-Fulkerson algorithm [46], push-relabel algorithm [69], Dinitz blocking flow algorithm [54] etc. These facts motivate our studies of the max-flow approach to binary image segmentation in the spatiall continuous setting (1.13).

Moreover, we will also investigate the binary image segmentation problem (1.13) subject to priori user inputs, where some areas are marked as 'background' or 'foreground' interatively by the user (see image (b) of Fig. 1.3). Such priori information introduces hard constraints to the labeling function $u(x)$, basically the overlap constraints. It also gives the clue to the image models of 'background' and 'foreground', which help to build up more accurate data terms [140, 153]. We will see that the proposed continuous max-flow model can easily adapt these supervised information without increasing any computational and memory load.

1.4 Multi-Class Image Partition

The multi-class image partition problem, or multi-labeling problem, is one of important mathematical models in image processing and computer vision, which tries to assign a label l_i, $i = 1, ..., n$, from the finite label set $L := \{l_1, ..., l_n\}$ to each pixel of the given image domain Ω subject to some optimal criterion. Such kind of problems appear extensively in the areas of image processing and computer vision. They are often formulated as the minimization of an energy function $E(u)$, $u(x) \in L$ for $\forall x \in \Omega$, which mathematically encodes the imposed optimality considerations for the imaging and vision task. The posteriori estimation is regarded as better as the energy is lower. Considering the mapping of $u : \Omega \to L$ is single-valued, the image domain Ω is therefore partitioned into n regions by the labeling function $u(x)$:

$$\Omega = \cup_{i=1}^{n} \Omega_i ; \quad \Omega_k \cap \Omega_l = \emptyset , \quad \forall k \neq l$$

where

$$u(\Omega_i) = l_i , \quad i = 1, ..., n .$$

Looking for such an optimal labeling function with respect to some energy functional is an important mathematical strategy to model a wide range of applications, e.g. image segmentation [30, 140, 158] where each non-overlapped segment directly corresponds to one label $l_i \in L$, $i = 1...n$, image denoising [100, 163, 153] where each label is associated with one discrete gray-scale or color value, stereo reconstruction [99, 100, 143] where each label gives a discrete disparity value or a piece of geometrical model [116].

In this thesis, we focus on the multi-class image partitioning problem which minimizes the total perimeter of the one-label (constant) regions and does not favor any particular order of the labels, namely Potts model [135]. More specially, for the given n labels $\{l_1, \dots, l_n\}$, one tries to partition the given image domain Ω into n disjoint subdomains $\{\Omega_i\}_{i=1}^{n}$ and assign each subdomain Ω_i a label l_i by solving

$$\min_{\{\Omega_i\}_{i=1}^{n}} \sum_{i=1}^{n} \int_{\Omega_i} \rho(l_i, x) \, dx + \alpha \sum_{i=1}^{n} \left| \partial \Omega_i \right| , \tag{1.14}$$

subject to

$$\cup_{i=1}^{n} \Omega_i = \Omega , \, \Omega_j \cap \Omega_k = \emptyset , \, \forall j \neq k , \tag{1.15}$$

where $|\partial \Omega_i|$ measures the perimeter of each disjoint subdomain Ω_i, $i = 1, \dots, n$, and the function $\rho(l_i, x), i = 1, \dots, n$, evaluates the performance of each label assignment l_i at the specified position x.

In a discrete graph setting, Potts model (1.14) corresponds to a practically important special case of a Markov Random Field (MRF) defined over a graph [110], where a typical MRF energy sums unary potentials defined over graph nodes and pairwise potentials defined over graph edges. When pixels can take only one of 2 labels, the resulting binary energy function can be efficiently and globally minimized by graph cuts [73], provided that the pairwise potentials are submodular [100]. However, for more than two labels typical MRF optimization problems are NP hard, so is Potts model. In particular, Potts model corresponds to a multi-terminal graph cut problem where only provably good approximate solutions are guaranteed, for example, via α-expansion or $\alpha - \beta$ swap [30], max-product loopy belief propagation [66, 65, 63], LP relaxations [101, 102, 167]; see [153] for a good review. Another drawback of such discrete setting is that the results are often biased by the discrete grid causing metrication errors, see sec. 1.3 and Fig. 1.5. Such visual artifacts can be largely reduced by either adding more neighbour nodes [29, 96] or applying high-order cliques [95, 88]. However, extra computation and memory load are introduced.

Parallel to these developments, variational methods have been developed for solving the same Potts model (1.14) in a spatially continuous setting where the bounded continuous image domain is considered. In this regard, level set introduces the most direct and natural way to formulate the piecewise constant labeling function and its related computation provides a feasible way to resolve the optimal partitions with a subgrid accuracy, see e.g. [127, 40, 164] and its variant of the piecewise constant level set method (PCLSM) [111, 112]. To this end, the phase-field method [90, 19] provides another direct and cheap implementations. Unfortunately, both approaches typically employ nonconvex functions, for which the numerical solvers often get stuck in a local minima and their results depend on the initial value.

During recent years, convex relaxation approaches were introduced, e.g. [133, 37, 13, 131, 108, 175, 106, 107] etc, to explore such multi-class Potts model (1.14) in the spatially continuous setting over the minimization of a convex energy function, so-called *convex relaxed Potts model*:

$$\min_{u \in S} \sum_{i=1}^{n} \int_{\Omega} u_i(x) \, \rho(l_i, x) \, dx \, + \, \alpha \sum_{i=1}^{n} \int_{\Omega} |\nabla u_i| \, dx \tag{1.16}$$

where S is the convex constrained set of the labeling functions $u(x) := (u_1(x), \ldots, u_n(x))$:

$$S \, = \, \{ u(x) \, | \, (u_1(x), \ldots, u_n(x)) \, \in \, \triangle_+ \, , \; \forall x \in \Omega \, \} \, ,$$

$\triangle_+$ is the simplex set, i.e.

$$\text{for } \forall x \in \Omega \, , \quad \sum_{i=1}^{n} u_i(x) \, = \, 1 \, ; \quad u_i(x) \in [0, 1] \, , \quad i = 1 \ldots n \, .$$

In (1.16), assigning each pixel by an unique label is encoded by the convex pixelwise simplex constraint and the minimization of the total perimeter is formulated by the sum of total-variation functions over labelings. Comparing to level set methods, great numerical advantages can be achieved, e.g. fast and reliable algorithms can be easily build up by standard convex optimization theories [20, 21, 45]. Since a strict mathematical proof of the exactness of such a convex relaxation approach to the nonconvex Potts model is still open [37], its approximation result can only be accepted as suboptimal. One may claim the convex relaxation method gives the solution which is closer to the exact global minimum than the local minima by the level set formulation. Practical experiments confirmed this.

For a short review of recent studies on the convex relaxed Potts model (1.16), Zach et al [184] introduced an alternating optimization approach to solve (1.16) in a numerically splitting way:

$$\min_{u,v\in S} \sum_{i=1}^{n} \int_{\Omega} v_i(x)\,\rho(l_i,x)\,dx \;+\frac{1}{2\theta}\left\|u-v\right\|^2 \;+\;\alpha\sum_{i=1}^{n}\int_{\Omega}\left|\nabla u_i\right|\,dx\,.$$

Obviously, when θ takes a value small enough, the above convex optimization problem properly approximates the convex relaxed Potts model (1.16). Within each iteration, two substeps are taken to tackle the total-variation term and explore the pointwise simplex constraint S respectively. A Douglas-Rachford splitting algorithm was proposed, by Lellmann et al [108], to solve a quite similar problem as (1.16), where a variant of the total-variation term is considered:

$$\int_{\Omega}\sqrt{\left|\nabla u_1(x)\right|^2+\ldots+\left|\nabla u_n(x)\right|^2}\,dx\,.$$

As in [184], the proposed splitting procedure involves an outer loop with two substeps, where the first substep solves a total-variation minimization problem iteratively until convergence, while the second substep projects the current solution to the pixelwise simplex constraint set S. In [106], the authors introduced a weighted variant of the classical total variation for vector fields in order to encode non-trivial interclass distances for multi-class labeling and the Nestorov based algorithm was applied to approximate the labeling problem within a suboptimality bound. In [37, 131, 107], the authors introduced another convex relaxation based on a multi-layered configuration, which was shown to be tighter. A more complex constraint on the dual variable p is given to avoid multiple countings. In addition, either a PDE-based primal-dual scheme [37, 131] was applied to achieve the minimum or a multiple-constrained Douglas-Rachford splitting approach [107] was presented to achieve global convergence and avoid inner iterative loops along with enforcements of the exact constraints.

In comparison to previous studies [184, 108, 37, 131, 107], we are interested in discussing the convex relaxed Potts model (1.16) over its dual formulations through convex optimization theories. We show this does provide a completely new way to elegantly analyze (1.16) and design efficient algorithms.

1.5 Contributions

A Short Overview of Convex Optimization Approaches

We study some specified topics of computer vision and image processing, based on modern theories of convex analysis and optimization. Some topics of this thesis can be formulated as a smooth unconstrained convex minimization problem, e.g. the nonrigid flow estimation problem, which can be essentially computed by the gradient descent method. The other problems of this thesis are reduced to be a nonsmooth convex optimization problem mostly along with complex constraints, e.g. nonsmooth flow decomposition and convex relaxed Potts model. The simple gradient descent method does not work for these cases. Therefore, either the direct convex programming method, like SOCP, or the duality-based method is considered. To this end, direct convex programming provides an easy way to handle many general convex optimization problems, e.g. the nonsmooth flow estimation and decomposition problems in this thesis. But it suffers from its high computation load and unefficiency. In this regard, duality based methods are employed to design cheap and fast approaches in the topics of TV-L1 image approximation, continuous min-cuts and convex relaxed Potts model, where dual models play as the central role.

We summarize our contributions to each topics of this thesis as follows:

Contributions to Non-rigid Flow Decomposition and Estimation

For the studies of non-rigid flow decomposition and estimation:

1. We propose a flow decomposition based approach to the high-order div-curl regularized non-rigid flow estimation, where the variational flow estimation problem is reformulated by the optimization over two potential fields associated to divergence and curl respectively. Then a space-decomposition algorithm is proposed to explore two subproblems, each of which has much less unknowns, at each iteration.

2. A mimetic finite difference method is introduced to build up accurate and reliable discretization schemes of flow decomposition and estimation. Especially, the nonsmooth functions, e.g. the L1 norm and total-variation function, are also well defined over the proposed discretization method. Such discretization scheme is applied in all the topics of this thesis. Experiments show its outstanding performance in implementing fast and stable algorithms in numerics.

3. We introduce and study the new nonsmooth div-curl regularizar and show its resulting structure-texture decomposition of non-rigid flows. In addition, the novel flow estimation joint with the proposed nonsmooth flow decomposition is also studied. We use the second-order cone programming to compute the reduced highly nonsmooth convex minimization problems. Experiments show the proposed nonsmooth flow decomposition is superior in achieving high accuracy of flow estimation and keeping small-scale flow patterns.

The results of this topics were first published in [177, 181]. Their journal version versions were published in [178, 179].

Contributions to TV-L1 Image Approximation

To the study of TV-L1 image approximation:

1. We derive new equivalent convex formulations to TV-L1 image approximation (1.7) in terms of primal and dual, which builds up a new analytical framework and results in a new variational perspective of (1.7).
2. By the proposed equivalent formulations, we prove that the TV-L1 formulation (1.7) introduces an exact convex relaxed model for its non-convex model (1.11). This extends the Chan et al's results [42] to the more general cases with discrete gray-scales. The same theoretical results can be naturally extended to image inpainting, which is also new.
3. In term of numerics, we propose the rounding scheme and show that the discrete-constrained optimization problem (1.11) can be exactly discovered by solving the convex optimization problem (1.7) which is much simpler than solving (1.11) directly. Besides its simplicity, it also largely reduces the computational and memory cost. Comparing to graph-cut based approaches, e.g. [51, 87], such reduction is especially significant when the total number of gray values is large, e.g. medical images are often recorded by over thousands of gray levels.
4. We introduce a new and efficient multiplier-based algorithm which explores the equivalent primal-dual formualtion through two simple projection substeps, instead of tackling the highly nonsmooth TV-L1 energy functional directly. Its reliability and efficiency can be verified by standard optimization theories and various experiments.

In parallel to our multiplier-based method, several other dual formulations and algorithmic schemes were proposed recently in the literature, see [59, 169, 155, 186, 147, 146]. In contrast to [59, 169, 155, 186], we apply the proposed equivalent primal-dual and dual formulations as a comprehensive approach, which seamlessly associate variational convex analyses with the proposed algorithm to (1.7), not just derive the algorithmic scheme on its own. In addition, the primal-dual algorithm we have proposed in this study is different from [59, 169, 155, 186]. In our algorithm, the variable u is treated as the multiplier to a linear equality constraint.

A journal version of this study was published in [183] and its extended version appeared as a technical report [182].

Contributions to Image Segmentaion

For the studies of image segmentation, we propose and study new continuous max-flow formulations which is equivalent to the continuous min-cut model (1.13) in the sense of primal and dual. This is in analogy with the graph based max-flow and min-cut. We summarize our main contributions in this topics as follows:

1. We propose the novel continuous max-flow models, which provide a new equivalent convex models of the respective continuous min-cut problem, with or without supervision constraints, in terms of dual.
2. We revisit and give explanations of fundamental conceptions used in graph cuts, which connect 'saturated' / 'unsaturated' flows with 'cuts', through a new variational perspective. This also leads to a new variational viewpoint to understand the classical max-flow/min-cut algorithms. With helps of the proposed continuous max-flow formulation, we prove that the nonconvex image segmentation problems, with or without supervision constraints, can be solved exactly and globally in a convex relaxation manner.
3. For the continuous min-cut model under supervised constraints, the proposed continuous max-flow formulation encodes such user-input constraints implicitly and does not require change flow capacities artificially. Meanwhile, the new supervised max-flow share the same complexity as the unsupervised one.
4. New and fast max-flow based algorithms are proposed, which splits the optimization problem into simple subproblems over independent flow variables, where the labeling function $u(x)$ works as a multiplier and is simply updated at each iteration. Their global convergence can be easily validated by standard convex optimization theories. Experiments show our continuous max-flow algorithms outperforms over the previous continuous min-cut methods, e.g. [31], in terms of efficiency and graph based methods in terms of accurarcy and minor metrification errors.

A shorter conference version of this study appeared in [174]. Its extended version was presented in the technical report [176]. Another technical report [12] studied the same type of the overlap constraint by a variant flow-maximization scheme.

Contributions to Multi-Class Image Partition

To the study of multi-class image partition, our contributions can be summarized as follows:

1. We discuss the convex relaxed Potts model under a primal-dual perspective. Two novel dual models are proposed, both of which equally reformulate the convex relaxed problem in terms of the dual variables only and thereby avoids the complex side constraint of the primal labeling functions. This contrasts with previous approaches which tackle the constrained optimization problem over the labeling function directly.

2. The first dual formulation also provides a new thresholding scheme for recovering binary primal solutions of the convex relaxed Potts problem which are globally optimal to the relaxed Potts model in terms of energy, along with an analytical tool to derive sufficient conditions about when this is possible, i.e. for when the relaxation is exact. In order to deal with non-smoothness of the resulting dual energy functional, a smoothed version of the convex dual model is introduced. Such a smoothing scheme for the dual model has some interesting connections to the formulations of maximum entropy clustering algorithms. A new algorithm derived from the smoothed dual formulation is shown to be more efficient than the state of art works; and it can be easily implemented. Experiments demonstrate the algorithm for the smoothed model may yield better binary approximations to the original non-convex problem with lower energy in connection with the new thresholding scheme. Numerical results are also compared extensively with the well-known algorithms alpha expansion and alpha-beta swap from discrete optimization, which show that our algorithm can produce results of equal or lower energy than these approaches.

3. The second dual model can be interpreted in terms of maximization over flows, hence called the continuous max-flow method. Variational analysis of the proposed continuous max-flow formulation leads to a new perspective of the corresponding 'cuts' or convex relaxed Potts problem. Therefore, close connections between flow and cut can be presented in a new variational way. The proposed continuous max-flow model leads to a new multiplier-based max-flow algorithm. To this end, we propose the fast linearized implementation of the max-flow based algorithm. Its great advantages over previous works in

numerics can be clearly presented in the following folds: it avoids pointwise projections onto the simplex constraint within each outer loop; in comparison to the first dual model and its related smoothed version, the continuous max-flow algorithm exactly solves the convex relaxed Potts model without any smoothing procedure; it is globally optimized based on an efficient and reliable multiplier-based max-flow algorithm, in contrast to the PDE-descent method whose convergence may suffer from uncareful stepsizes resulting in suboptimum. Experiments show a faster convergence rate, about 4 times, than previous methods.

The study of the smoothed duality based method was accepted and will appear in [14]. Its technical report was published in [13]. The continuous max-flow model to the convex relaxed Potts model was published in [175].

1.6 Organization

This thesis is organized by the studied topics. Each chapter is used for the discussions of one topics:

1. In Chapter 2, we study the problems of non-rigid flow decomposition and estimation: in Sec. 2.1, we first review the mimetic finite-difference method over the 2-D square image grid at Sec. 2.1.1 and introduce the classical flow decompositions and the equivalent flow representation in their associated discretized forms at Sec. 2.1.2; the new extended flow decompositions are presented in Sec. 2.1.4; the proposed flow decomposition based estimation method is discussed in details at Sec. 2.2; finally, we study the new nonsmooth flow decomposition and estimation in Sec. 2.3.
2. In Chapter 3, the TV-L1 image approximation and the associated discrete-constrained TV-L1 image approximation are discussed: we introduce the equivalent convex models to TV-L1 image approximation at Sec. 3.1; and show its exactness of the convex relaxation to the discrete-constrained image approximation in Sec. 3.2; in Sec. 3.3, the new multiplier based TV-L1 algorithm is proposed based on convex optimization theories; we show related experiment results of TV-L1 image approximation in Sec.3.4.
3. Chapter 4 is for the study of the continuous min-cut approach to image segmentation: in Sec. 4.1, we shortly review the classical min-cut and max-flow theory over graphs; then we revisit previous works of continuous min-cut in Sec. 4.2; the new continuous max-flow theory and its variational analysis are introduced in Sec. 4.3 and Sec. 4.4 to incorporate

supervision constraints; the continuous max-flow based algorithms are proposed in Sec. 4.5; experiment results are presented in Sec. 4.6.

4. In Chapter 5, the multi-class image partition problem is explored by the convex relaxation approach: we review the convex relaxed Potts model and related works in Sec. 5.1; then we discuss the first dual model based approach in Sec. 5.2; we investigate the second dual model based approach, i.e. the continuous max-flow method, in Sec. 5.3.

5. Finally, conclusions and future topics are stated in Chapter 6. Details of discretization and some proofs are given in Chapter 7.

2 Non-rigid Flow Decomposition and Estimation

In this chapter, we discuss variational decomposition and estimation of highly non-rigid flow fields and consider high-order div-curl regularized flow approximation problems, which are formulated in the form of convex optimization. We formulate the studied high-order div-curl regularizars as follows:

$$R(\mathbf{u}) \ := \ \alpha\Big(\int_\Omega |\nabla \operatorname{div} \mathbf{u}|^2 \ dx + \int_\Omega |\nabla \operatorname{curl} \mathbf{u}|^2 \ dx \Big) \tag{2.1a}$$

and

$$R(\mathbf{u}) \ := \ \alpha\Big(\int_\Omega |\nabla \operatorname{div} \mathbf{u}| \ dx + \int_\Omega |\nabla \operatorname{curl} \mathbf{u}| \ dx \Big) \, . \tag{2.1b}$$

Higher-order regularization is necessary in order to accurately recover important flow structure like vortices, and to incorporate key physical properties such as vanishing divergence. For proper discretization, we apply the finite mimetic difference method, which provides a compatible discretization scheme in numerics, i.e. preserves the identities fulfilled by the continuous differential operators, and helps to set up reliable numerical computation related to high differential operators.

We start by introducing our discretization scheme, i.e. the mimetic finite difference method, which defines scalar and vector fields on primal and dual grids together with appropriate norms. Then we introduce the associated discrete first-order operators and show how integral identities such as the Gauss integral identity and the Helmholtz decomposition carry over to the discrete situation. We also define discrete TV and G norms with respect to the primal and dual grid. Based on these definitions and results, we first deal with flow decomposition by the TV-G norm model. To this end, we have to introduce the decomposition of a non-rigid flow field into components with constant divergence and curl and variable components which can be further decomposed into a "structural" part and a "textural" part. In fact, these parts comprise flow patterns at different scales. Then we consider the high-order flow estimation taskand show its outperformance over the other flow estimation approaches in preserving coherent flow patterns. Finally, we study the flow estimation approach joint with the decomposition model in order to solve both tasks simultaneously. For challenging tasks of tackling the associated convex optimization problems, both the variational method and the direct convex programming method, e.g. second-order cone programming (SOCP), are applied.

2.1 Discrete Representations and Decompositions of Non-rigid Flows

Now we discuss the discretization of 2-D non-rigid flow fields. As the key step in dealing with the numerical problems encountered in this thesis, discretization should be taken in such a way which helps to avoid information losses of the original problem and its structure. In this regard, we introduce the compatible discretization method, which was proposed and investigated in various publications for years and well-known to yield numerically stable, accurate and physically consistent approximations.

We list the following statements for a short review of compatible discretization, in [25, 24, 26, 103], Bossavit et al. demonstrated connections between stable finite elements for the Maxwell's equations and Whitney forms. In parallel, the covolume methods have been introduced in discrete fields of finite volume methods for a long time, see [120, 121, 122, 119]. In [85, 83, 82, 148], the support differential operators and mimetic methods combined the Stokes theorem with variational Green's identities were applied to derive compatible schemes in a finite-difference way, which is called the *mimetic finite difference method*. Meanwhile, mimetic discretizations were interpreted and discussed by algebraic topology, first by Hyman et al in [84] and more recently by Mattiussi [114], Schwalm et al. [145]. We refer [22, 6, 5] for more detailed explainations.

In the following parts of this section, we focus on and introduce the mimetic finite-difference method to design discrete consistent integral and differential operators which follow strictly the vector calculus and theorems in discrete settings. It follows two main advantages: first, an overall discrete interpretation of operations and calculus is presented; second, the key functions of differential forms and algebraic topology are revealed and accepted to build up concise formalism which encodes the structure of problems and exposes their local and global invariants in the discrete setting.

2.1.1 Mimetic Finite Difference Method

Discretization

The *mimetic finite difference method* was first introduced by Hyman and Shashkov [148, 83, 82, 148]. Its derived defintions of integral and differential operators will be widely used in all studies of this thesis, since the mimetic finite difference method is useful for us to keep important properties and structures of the original continuous optimization problems after discretization.

In this work, we focus on the 2-D rectanglar image domain Ω and only consider regular grids on Ω with unit square cells, where both the side-length h and the area of each cell h^2 are 1, i.e. $h = 1$. The superscript h is added to denote the discretized counterparts. For examples, Ω^h denotes the whole discretized image area with its discretized boundary $\partial\Omega^h$, and $\mathbf{u}^h$ denotes the discretized version of the flow field $\mathbf{u}$. We mostly stick to the notations without the superscript h for easy reading, when the definitions are clear in the context.

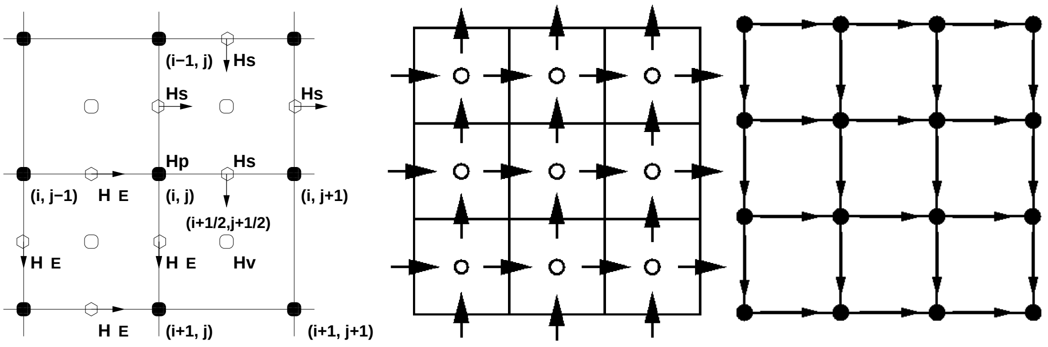


Figure 2.1: Illustrations of discrete scalar fields, H_V and H_P, and discrete vector fields, H_E and H_S, on a 2-D rectangular grid.

By this, let the grid consist of $m \times n$ vertices which are denoted by $P_{i,j}$, $i = 1, \ldots, n$ $j = 1, \ldots, m$, at each discrete vertex (i, j), see the filled rounds of Fig. 2.1. For each square area bounded by four stencils: $P_{i,j}$, $P_{i+1,j}$, $P_{i,j+1}$ and $P_{i+1,j+1}$, we call it the *cell* $\Omega_{i+1/2,j+1/2}$, see the empty rounds of Fig. 2.1 at its lower-right corner, where $(i + 1/2, j + 1/2)$ denotes the coordinate of the center of each cell. Let $L_{i,j+1/2}$ be the edge between vertices $P_{i,j}$ and $P_{(i,j+1)}$, see the empty hexagons of Fig. 2.1, where $(i, j + 1/2)$ gives the coordinate of the center of this edge. To simplify notations, we often index cell $\Omega_{\alpha,\beta}$ and side $L_{\alpha,\beta}$ directly by the corresponding coordinates (α, β).

Based on these discrete elements, the *mimetic finite-difference method* define four types of discrete 2-D fields over this discrete 2-D grid: two for scalars and two for vectors, see also the two figures at the righthand of Fig. 2.1:

H_V: the space of *scalar fields* defined on cells: the value of the scalar field is given at the center of each cell (see the empty circles of the first and second graphs in Fig. 2.1);

H_P: the space of *scalar fields* defined on vertices: the value of the scalar field is given at each vertex (see the filled circles of the first and of third graphs in Fig. 2.1);

H_E: the space of *vector fields* defined tangential to each edge: the value of the vector field is given at the center of each edge of the cell and parallel to its hosting edge (see the edges of the first and third graphs in Fig. 2.1);

H_S: the space of *vector fields* defined normal to each edge: the value of the vector field is given at the center of each edge of the cell and normal to the hosting edge (see the edges of the first and second graphs in Fig. 2.1).

Furthermore, we also denote with H_P^o, H_S^o, H_E^o the subspaces of inner scalar and inner vector fields, obtained by restricting the spaces H_P, H_S, H_E, respectively, with zero boundary values. The subspace H_V^o defines the space of all the scalar fields whose values vanish at the extended boundary.

Inner products and Norms

Thanks to the unit edge-length and area of each cell, i.e. $|L_{\alpha,\beta}| = 1$ and $|\Omega_{\alpha,\beta}| = 1$, the linear spaces H_P and H_V are equipped with their respective inner product

$$\langle f, g\rangle_{H_V} = \sum_{\Omega_{\alpha,\beta}\in\Omega} f_{\alpha,\beta} g_{\alpha,\beta} \ , \quad \langle f, g\rangle_{H_P} = \sum_{P_{i,j}\in\Omega} f_{i,j} g_{i,j} \ ,$$

and the corresponding L_2 norms,

$$\|g\|_{H_V}^2 = \langle g, g\rangle_{H_V} = \sum_{\Omega_{\alpha,\beta}\in\Omega} g_{\alpha,\beta}^2 \ , \quad \|g\|_{H_P} = \langle g, g\rangle_{H_P} = \sum_{P_{i,j}\in\Omega} g_{i,j}^2 \ .$$

The inner products on H_S and H_E are defined as follows: Let the indices D, T, R, L refer to the four edges around the cell $\Omega_{\alpha,\beta}$. At each cell $\Omega_{\alpha,\beta}$, we define $\mathbf{u} \in H_S$ as

$$\mathbf{u}_{(\alpha,\beta)} := \frac{1}{\sqrt{2}} (\mathbf{u}_D, \mathbf{u}_T, \mathbf{u}_R, \mathbf{u}_L)_{\alpha,\beta}^{\intercal} \ .$$

By the mimetic finite-difference method [83, 82], the inner product of two discrete vector fields $\mathbf{u}, \mathbf{v} \in H_S$ is given by:

$$\langle \mathbf{u}, \mathbf{v} \rangle_{H_S} := \sum_{\Omega_{\alpha,\beta} \in \Omega} \langle \mathbf{u}_{(\alpha,\beta)}, \mathbf{v}_{(\alpha,\beta)} \rangle = \sum_{\Omega_{\alpha,\beta} \in \Omega} \frac{1}{2} (\mathbf{u}_D \mathbf{v}_D + \mathbf{u}_T \mathbf{v}_T + \mathbf{u}_R \mathbf{v}_R + \mathbf{u}_L \mathbf{v}_L)_{\alpha,\beta} \ ,$$

and the corresponding L_2 norm of $\mathbf{u}$ given by: $\|\mathbf{u}\|^2_{H_S} := \langle \mathbf{u}, \mathbf{u} \rangle_{H_S}$.

Definitions of L_1 and L_∞ Norm

The L_1 norm of the scalar field $g \in H_V$ is defined by

$$\|g\|_{L_1, H_V} \ := \ \sum_{\Omega_{\alpha,\beta} \in \Omega} |g|_{\alpha,\beta} \ ,$$

where $|g|_{\alpha,\beta}$ is the absolute value of the scalar field g at the cell $\Omega_{\alpha,\beta}$.

The L_1 norm of a discrete vector field $\mathbf{u} \in H_S$ is given as follows

$$\|\mathbf{u}\|_{L_1, H_S} := \sum_{\Omega_{\alpha,\beta} \in \Omega} \sqrt{\langle \mathbf{u}_{(\alpha,\beta)}, \mathbf{u}_{(\alpha,\beta)} \rangle} = \frac{1}{\sqrt{2}} \sum_{\Omega_{\alpha,\beta} \in \Omega} \sqrt{(\mathbf{u}_D^2 + \mathbf{u}_T^2 + \mathbf{u}_R^2 + \mathbf{u}_L^2)_{\alpha,\beta}} \ . \qquad (2.2)$$

Observe that

$$\frac{1}{\sqrt{2}} \sqrt{(\mathbf{u}_D^2 + \mathbf{u}_T^2 + \mathbf{u}_R^2 + \mathbf{u}_L^2)_{\alpha,\beta}} \ = \ \max_{\mathbf{p}_{(\alpha,\beta)}} \ (\mathbf{u}_D \mathbf{p}_D + \mathbf{u}_T \mathbf{p}_T + \mathbf{u}_R \mathbf{p}_R + \mathbf{u}_L \mathbf{p}_L)_{\alpha,\beta}$$

where

$$\frac{1}{\sqrt{2}} \sqrt{(\mathbf{p}_D^2 + \mathbf{p}_T^2 + \mathbf{p}_R^2 + \mathbf{p}_L^2)_{\alpha,\beta}} \leq 1 \ , \qquad (2.3)$$

therefore the L_1 norm of the vector field $\mathbf{u} \in H_S$ can be expressed as

$$\|\mathbf{u}\|_{L_1, H_S} \ = \ \max_{\mathbf{p} \in H_S} \ \langle \mathbf{u}, \mathbf{p} \rangle_{H_S} \ , \quad \|\mathbf{p}\|_{L_\infty, H_S} \leq 1$$

and the infty norm $\|\mathbf{p}\|_{L_\infty, H_S} \leq 1$ is equally given by the cell-wise constraint (2.3).

The inner product, L_2 norm and L_1 norm of the discrete vector field $\mathbf{u} \in H_E$ can also be defined in the same manner.

Primal and Dual Differential Operators

The mimetic finite-difference method defines the first-order differential operators over primal and dual grids [83, 82], where two versions of these operators are given in terms of primal and

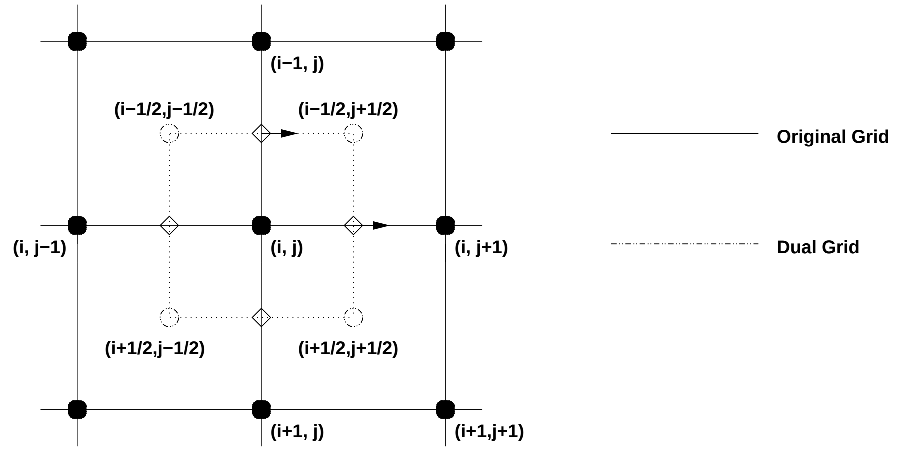


Figure 2.2: Primal Grid (normal lines) and Dual Grid (dashed lines)

dual, so as to preserve important identities of vector calculus in the discrete sense and construct conservative numerical schemes.

In addition to the original 2-D image grid of Ω, so-called *primal grid*, a dual grid is given such that it has a half-size shift to the primal grid, see Fig. 2.2. Data can be equivalently expressed over both the primal grid and the dual one: for example, given one vector field $\mathbf{u} \in H_S$ on the primal grid, we can also take $\mathbf{u}$ as the vector field defined in H_E over the dual grid without any loss of information about $\mathbf{u}$. With this, the support differential operators over the dual grid are provided.

Now we draft definitions of the four first-order differential operators, div, curl, ∇ and $\nabla^{\perp}$, over the 2-D square primal and dual grid, respectively, as follows.

1. **Primal First-Order Differential Operators:**
   - **Primal divergence operator div^h:** For $\forall \mathbf{u} \in H_S$, let the indices D, T, R, L refer to the four edges around the cell $\Omega_{\alpha,\beta}$, we have:

$$(\mathrm{div}^h\,\mathbf{u})_{\alpha,\beta} \;=\; \big((\mathbf{u}_D - \mathbf{u}_T) + (\mathbf{u}_R - \mathbf{u}_L)\big)_{\alpha,\beta}\,; \tag{2.4}$$

     which gives the discrete scalar field $\mathrm{div}^h\,\mathbf{u} \in H_V$.
   - **Primal gradient operator ∇^h:** For $\forall v \in H_P$, let $P_{i,j}$ and $P_{k,l}$ be the two vertices of the edge $L_{\alpha,\beta}$ where $k \geq i$ and $l \geq j$, we have:

$$(\nabla^h v)_{L_{\alpha,\beta}} \;=\; v_{k,l} - v_{i,j}\,, \tag{2.5}$$

     and along the edge direction, which gives the discrete vector field $\nabla^h v \in H_E$.

- **Primal curl operator** curl^h**:** For $\forall \mathbf{u} \in H_E$, let the indices D, T, R, L refer to the four edges around the cell $\Omega_{\alpha,\beta}$, we have:

$$(\mathrm{curl}^h\, \mathbf{u})_{\alpha,\beta} \;=\; \big((\mathbf{u}_T - \mathbf{u}_D) + (\mathbf{u}_R - \mathbf{u}_L)\big)_{\alpha,\beta}\,; \tag{2.6}$$

which gives the discrete scalar field $\mathrm{curl}^h\, \mathbf{u} \in H_V$.

- **Primal operator** $\nabla^{\perp,h}$**:** For $\forall v \in H_P$, let $P_{i,j}$ and $P_{i,j+1}$ be the two vertices of the horizontal edge $L_{\alpha,\beta}$, we have

$$(\nabla^{\perp,h} v)_{L_{\alpha,\beta}} \;=\; v_{i,j+1} - v_{i,j}\,, \tag{2.7}$$

which is perpenticular to the edge direction; and let $P_{i,j}$ and $P_{i+1,j}$ be the two vertices of the vertical edge $L_{\alpha,\beta}$, we have

$$(\nabla^{\perp,h} v)_{L_{\alpha,\beta}} \;=\; v_{i,j} - v_{i+1,j}\,, \tag{2.8}$$

which is perpenticular to the edge direction; thus the discrete vector field $\nabla^{\perp,h} v \in H_S$.

2. **Dual First-Order Differential Operators:**

- **Dual divergence operator** $\overline{\mathrm{div}^h}$**:** For $\forall \mathbf{u} \in H_E$, we regard $\mathbf{u}$ equally as the dual vector field $(\overline{\mathbf{u}} := \mathbf{u}) \in H_S$ on the dual grid, we have:

$$(\overline{\mathrm{div}^h}\mathbf{u})_{i,j} \;:=\; \mathrm{div}^h\, \overline{\mathbf{u}}_{\alpha,\beta} \;=\; \big((\overline{\mathbf{u}}_D - \overline{\mathbf{u}}_T) + (\overline{\mathbf{u}}_R - \overline{\mathbf{u}}_L)\big)_{\alpha,\beta}\,; \tag{2.9}$$

where the indices D, T, R, L refer to the four edges around the cell $\Omega_{\alpha,\beta}$ of the dual grid, which gives the discrete scalar field $\overline{\mathrm{div}^h}\mathbf{u} \in H_P$.

- **Dual gradient operator** $\overline{\nabla^h}$**:** For $\forall v \in H_V$, we regard v equally as the dual scalar field $(\overline{v} := v) \in H_P$, we have:

$$(\overline{\nabla^h} v)_{L_{\alpha,\beta}} \;:=\; \nabla^h \overline{v}_{\overline{L}_{\alpha,\beta}} \;=\; \overline{v}_{k,l} - \overline{v}_{i,j}\,, \tag{2.10}$$

where $P_{i,j}$ and $P_{k,l}$ be the two vertices of the edge $\overline{L}_{\alpha,\beta}$ where $k \geq i$ and $l \geq j$, and the vector element $(\overline{\nabla^h} v)_{L_{\alpha,\beta}}$ along the edge direction of its dual grid. This gives the discrete vector field $\overline{\nabla^h} v \in H_S$.

- **Dual curl operator** $\overline{\mathrm{curl}^h}$**:** For $\forall \mathbf{u} \in H_S$, we regard $\mathbf{u}$ equally as the dual vector field $(\overline{\mathbf{u}} := \mathbf{u}) \in H_E$ on the dual grid, we have:

$$(\overline{\mathrm{curl}^h}\mathbf{u})_{i,j} \;:=\; \mathrm{curl}^h\, \overline{\mathbf{u}}_{\alpha,\beta} \;=\; \big((\overline{\mathbf{u}}_T - \overline{\mathbf{u}}_D) + (\overline{\mathbf{u}}_R - \overline{\mathbf{u}}_L)\big)_{\alpha,\beta}\,; \tag{2.11}$$

where the indices D, T, R, L refer to the four edges around the cell $\Omega_{\alpha,\beta}$ along its dual grid, which gives the discrete scalar field $\overline{\mathrm{curl}^h}\mathbf{u} \in H_P$.

- **Dual operator $\overline{\nabla^{\perp,h}}$:** For $\forall v \in H_V$, we regard v equally as the dual scalar field $(\overline{v} := v) \in H_P$ on the dual grid, we have:

$$(\overline{\nabla^{\perp,h}} v)_{L_{\alpha,\beta}} := (\nabla^{\perp,h}\overline{v})_{\overline{L}_{\alpha,\beta}} = \overline{v}_{i,j+1} - \overline{v}_{i,j} \, , \tag{2.12}$$

  where $P_{i,j}$ and $P_{i,j+1}$ be the two vertices of the horizontal edge $\overline{L}_{\alpha,\beta}$ along its dual grid, and the vector element $(\overline{\nabla^{\perp,h}} v)_{L_{\alpha,\beta}}$ is perpenticular to the edge direction;

$$(\overline{\nabla^{\perp,h}} v)_{L_{\alpha,\beta}} := (\nabla^{\perp,h}\overline{v})_{\overline{L}_{\alpha,\beta}} = v_{i,j} - v_{i+1,j} \, , \tag{2.13}$$

  where $P_{i,j}$ and $P_{i,j+1}$ be the two vertices of the vertical edge $\overline{L}_{\alpha,\beta}$ along its dual grid, and the vector element $(\overline{\nabla^{\perp,h}} v)_{L_{\alpha,\beta}}$ is perpenticular to the edge direction; thus the discrete vector field $\overline{\nabla^{\perp,h}} v \in H_E$.

Finally, for the discretization of the boundary condition $\mathbf{n} \cdot \mathbf{u}|_{\partial\Omega}$, we introduce the boundary operator

$$\mathbb{B}_{\mathrm{n}}^{h} : H_S \to \partial H_S := H_S \backslash H_S^o \, .$$

It restricts the vector field to the vectors at the grid's boundary which are directed outside the boundary. Thus for any vector field $\mathbf{u} \in H_S$, the operator $\mathbb{B}_{\mathrm{n}}^{h}$ is just

$$\mathbb{B}_{\mathrm{n}}^{h}\mathbf{u} := \begin{cases} \mathbf{u}_{\tilde{L}_{\alpha,\beta}} & \text{at } \partial\Omega \\ 0 & \text{otherwise} \end{cases} ,$$

where $\tilde{L}_{\alpha,\beta}$ are edges locating at the grid's boundary $\partial\Omega$.

Properties of Mimetic Finite Difference Method

It has been shown [83] that using the operators defined above, elementary properties of continuous fields in terms of div, curl, ∇ and $\nabla^{\perp}$ carry over to the discrete case.

For example, if the curl of a vector field $\mathbf{u}$ is zero, $\mathrm{curl}^h \mathbf{u} \equiv 0$, then the vector field can be expressed as the gradient of a scalar field v, $\mathbf{u} = \nabla^h v$; or, if the divergence of a vector field $\mathbf{u}$ is a zero, $\mathrm{div}^h \mathbf{u} \equiv 0$, then it should be the curl of another vector field, i.e. in 2-D case $\mathbf{u} = \nabla^{\perp,h} v$ (only consider the 2D case in this thesis).

Or the other way round, the curl of the gradient of a scalar field v is always zero, i.e. $\mathrm{curl}^h \nabla^h v \equiv 0$; the divergence of the perpenticular gradient of a scalar field v is also zero, i.e. $\mathrm{div}^h \nabla^{\perp,h} v \equiv 0$.

Moreover, Green's theorem

$$\int_{\Omega} g \cdot \operatorname{div} \mathbf{u} \, dx + \int_{\Omega} \nabla g \cdot \mathbf{u} \, dx \; = \; \int_{\partial\Omega} g \mathbf{u}_n ds \tag{2.14}$$

is still kept in discrete setting such that

$$\left\langle g, \operatorname{div}^h \mathbf{u} \right\rangle_{H_V} + \left\langle \overline{\nabla^h} g, \mathbf{u} \right\rangle_{H_S} \; = \; \sum_{L_{\alpha,\beta} \in \partial\Omega} (g_{\alpha,\beta} \, \mathbf{u}_n)_{\alpha,\beta} \,, \tag{2.15}$$

for the discrete vector field $\mathbf{u} \in H_S$ and the discrete scalar field $g \in H_V$.

Similarly, Gauss theorem

$$\int_{\Omega} \operatorname{div} \mathbf{u} \, dx \; = \; \int_{\partial\Omega} \mathbf{u}_n ds \tag{2.16}$$

can be equally rewitten in the discrete case such that

$$\sum_{\Omega_{\alpha,\beta} \in \Omega} \operatorname{div}^h \mathbf{u} \; = \; \sum_{L_{\alpha,\beta} \in \partial\Omega} (\mathbf{u}_n)_{\alpha,\beta} \,. \tag{2.17}$$

Using the definitions above, we rewrite this equation more concisely as

$$\mathbf{1}^{\mathsf{T}}_{\dim H_V} \operatorname{div}^h \mathbf{u} = \mathbf{1}^{\mathsf{T}}_{\dim \partial H_S} \mathbb{B}^h_{\mathrm{n}} \mathbf{u} \,, \tag{2.18}$$

when we formulate all discrete scalar and vector fields by one-dimensional vetcors, $\mathbf{1}_n$ denotes the one-vector with n elements and $\dim H$ gives the dimension of the linear space H.

Most importantly as shown in (2.15) above, the additional dual operators resolve the incompatibilities of domains and ranges of the primal operators, for example, ∇^h and div^h cannot be regarded as mutually adjoint operators, whereas ∇^h, $\overline{\operatorname{div}^h}$ and $\overline{\nabla^h}$, div^h do. Thus they are used to build *compound second order* differential operators in a more strict sense. For instance, given a scalar field $v \in H_V$, its gradient field $\overline{\nabla^h} v$ is provided in H_S as above definitions of dual operators, then it is easy to find that the vector field $\overline{\nabla^h} v$ is curl-free, i.e. $\overline{\operatorname{curl}^h} \overline{\nabla^h} v = 0$, and its divergence builds up the second-order differential operator, e.g. the Laplacian operator

$$\triangle_D \; = \; \operatorname{div}^h \overline{\nabla^h} \,.$$

2.1.2 Discrete Classical Decompositions

Discrete Orthogonal Decomposition

By means of the defined mimetic finite difference notations in the previous section, we consider the discrete orthogonal decomposition which corresponds to the classical Helmholtz decomposition [67], i.e. any 2-D vector field $\mathbf{u}(x) \in (L_2(\Omega))^2$ given on a bounded, simply-connected

Lipschitz 2-D domain Ω can be uniquely represented by the sum of two orthogonal flow fields such that

$$\mathbf{u} \;=\; \nabla\psi + \nabla^{\perp}\phi \tag{2.19}$$

where $\partial_{\mathrm{n}}\psi = \mathbf{u}\cdot\mathbf{n}|_{\partial\Omega}$ and $\phi_{\partial\Omega} = 0$. $\nabla\psi$ and $\nabla^{\perp}\phi$ are the curl-free and div-free flow fields respectively. Similarly, we give the discrete orthogonal decomposition of the given discrete field $\mathbf{u} \in H_S$ over a 2-D rectangle grid, in terms of div-free and curl-free components, by the following proposition which was proposed and proved by Hyman and Shashkov [86]:

Theorem 1 (Discrete Orthogonal Decomposition). *For any 2D vector field $\mathbf{u} \in H_S$, it can represented in terms of $\psi \in H_{V+\partial V}$ and $\phi \in H_P$:*

$$\mathbf{u} = \overline{\nabla^h}\psi + \nabla^{\perp,h}\phi, \qquad \textit{where } \mathbb{B}^h_{\mathrm{n}}\mathbf{u} = \mathbb{B}^h_{\mathrm{n}}\overline{\nabla^h}\psi\,,\;\; \phi_{\partial\Omega} = 0\,. \tag{2.20}$$

The decomposition (2.20) *is unique up to a constant of ψ and orthogonal such that*

$$\left\langle \overline{\nabla^h}\psi, \nabla^{\perp,h}\phi \right\rangle_{H_S} = 0\;, \quad \forall\mathbf{u} \in H_S \tag{2.21}$$

We can directly compute the two discrete potential fields ψ and ϕ by solving the following linear equations with the respective Neumann and Dirichlet boundary conditions:

$$\triangle_D\psi = \mathrm{div}^h\,\mathbf{u}\;, \quad \mathbb{B}^h_{\mathrm{n}}\overline{\nabla^h}\psi = \mathbb{B}^h_{\mathrm{n}}\mathbf{u}\;, \tag{2.22}$$

$$\triangle_C\phi = \overline{\mathrm{curl}^h}\mathbf{u}\;, \quad \phi_{\partial\Omega} = 0\;, \tag{2.23}$$

where the discrete Laplacians are defined by

$$\triangle_D := \mathrm{div}^h\,\overline{\nabla^h}\;, \qquad \triangle_C := \overline{\mathrm{curl}^h}\nabla^{\perp,h}\;, \tag{2.24}$$

and the additional constraint $\mathbf{1}^{\mathsf{T}}_{\mathrm{dim}H_V}\,\psi = 0$ (continuous case: $\int_\Omega \psi dv = 0$) is used to eliminate the arbitrary constant in (2.20) and (2.22).

The orthogonality between the two discrete components $\overline{\nabla^h}\psi$ and $\nabla^{\perp,h}\phi$ at the righthand of (2.20) comes naturally from the primal and dual operators of the mimetic finite difference scheme [83, 82]:

$$\mathrm{div}^h\,\nabla^{\perp,h} \equiv 0\;, \qquad \overline{\mathrm{curl}^h}\,\overline{\nabla^h} \equiv 0\;.$$

Therefore, we have

$$\left\langle \overline{\nabla^h}\psi, \nabla^{\perp,h}\phi \right\rangle_{H_S} = \left\langle \overline{\mathrm{curl}^h}\,\overline{\nabla^h}\psi, \phi \right\rangle_{H_P} \equiv 0\;. \tag{2.25}$$

Defining two finite-dimentional subspaces corresponding to discrete curl-free and div-free vector fields respectively:

$$S_{ir} := \left\{ \mathbf{u} \in H_S \,|\, \mathbf{u} = \overline{\nabla^h}\psi \right\} \,, \tag{2.26}$$

$$S_{sol} := \left\{ \mathbf{u} \in H_S \,|\, \mathbf{u} = \nabla^{\perp,h}\phi,\, \phi_{\partial\Omega} = 0 \right\} \,, \tag{2.27}$$

Theorem 1 simply asserts:

Theorem 2. *For the discrete function space H_S, the direct sum holds:*

$$H_S = S_{ir} \oplus S_{sol} \,. \tag{2.28}$$

In summary, Theorem 1 provides an exact orthogonal decomposition of the finite-dimensional space of vector fields $\mathbf{u} \in H_S$. Furthermore, as detailed below, the decomposition allows to estimate the two potentials $\psi \in H_{V+\partial V}$ and $\phi \in H_P$, which amount to curl-free and div-free vector fields respectively, from a image sequence in a direct way. Using variational optical flow approaches, the estimation can be done in parallel by applying subspace correction methods. Alternatively, we may first estimate the motion field $\mathbf{u}$, and then compute ψ and ϕ in a subsequent step by solving the Neumann and Dirichlet problems (2.22) and (2.23).

Discrete Hodge Decomposition

Given the discrete vector field $\mathbf{u} \in H_S$, its discrete Hodge decomposition can be expressed as the following proposition, which corresponds to the classical Hodge decomposition[1].

Theorem 3 (Discrete Hodge Decomposition). *On the 2-D rectangular grid, any discrete vector field $\mathbf{u} \in H_S$ can be represented by*

$$\mathbf{u} = \overline{\nabla^h}\psi + \nabla^{\perp,h}\phi + \mathbf{h}, \quad \textit{where } \psi_{\partial\Omega} = 0 \,,\; \phi_{\partial\Omega} = 0 \,, \tag{2.30}$$

and $\mathbf{h} \in H_S$ is harmonic such that $\operatorname{div}^h \mathbf{h} = 0$ and $\overline{\operatorname{curl}^h}\mathbf{h} = 0$.

Moreover, the three elements given at the right side of (2.30) are orthogonal to each other:

$$\left\langle \mathbf{h}, \overline{\nabla^h}\psi \right\rangle_{H_S} = < \mathbf{h}, \nabla^{\perp,h}\phi >_{H_S} = < \overline{\nabla^h}\psi, \nabla^{\perp,h}\phi >_{H_S} = 0 \,. \tag{2.31}$$

[1] On a bounded, simply-connected and Lipschitz 2-D domain Ω, any vector field $\mathbf{u} \in (L_2(\Omega))^2$, whose divergence and curl exist and are integrable, can be represented in terms of two potential fields ψ, ϕ together with one harmonic vector field $\mathbf{h}$, i.e. $\operatorname{div}\mathbf{h} = 0$ and $\operatorname{curl}\mathbf{h} = 0$, such that

$$\mathbf{u} = \nabla\psi + \nabla^{\perp}\phi + \mathbf{h}, \quad \psi_{\partial\Omega} = 0 \;,\; \phi_{\partial\Omega} = 0 \;. \tag{2.29}$$

In addition, the three components given at the rightside of (2.29) are orthogonal to each other.

We can simply derive the discrete Hodge decomposition of $\mathbf{u} \in H_S$ from its orthogonal decomposition:

$$\mathbf{u} = \overline{\nabla^h}\tilde{\psi} + \nabla^{\perp,h}\phi, \quad \text{where } \mathbb{B}^h_{\mathrm{n}}\mathbf{u} = \mathbb{B}^h_{\mathrm{n}}\overline{\nabla^h}\tilde{\psi}\,,\ \phi_{\partial\Omega} = 0\,. \tag{2.32}$$

Then we seperate the flow component $\overline{\nabla^h}\tilde{\psi}$ into two parts:

$$\overline{\nabla^h}\tilde{\psi} \;=\; \overline{\nabla^h}\psi + \mathbf{h}$$

through the following two linear equations:

$$\triangle_D\psi \;=\; \triangle_D\tilde{\psi}\,(:= \operatorname{div}^h \mathbf{u})\,; \quad \psi_{\partial\Omega} = 0$$

and

$$\operatorname{div}^h \mathbf{h} \;=\; 0\,, \quad \overline{\operatorname{curl}^h}\mathbf{h} \;=\; 0\,; \quad \mathbb{B}^h_{\mathrm{n}}\mathbf{h} \;=\; \mathbb{B}^h_{\mathrm{n}}\overline{\nabla^h}(\tilde{\psi} - \psi)$$

where $\triangle_D$ is the discrete Laplacian operator given by the compound second-order differential operator $\operatorname{div}^h \overline{\nabla^h}$. The solvablities of the above linear equations can be varified by means of consistences of matrices.

Let

$$S_{ir,0} := \left\{\mathbf{u} \in H_S \,|\, \mathbf{u} = \overline{\nabla^h}\psi,\ \psi_{\partial\Omega} = 0\right\}\,, \tag{2.33}$$

$$S_{sol} := \left\{\mathbf{u} \in H_S \,|\, \mathbf{u} = \nabla^{\perp,h}\phi,\ \phi_{\partial\Omega} = 0\right\}\,, \tag{2.34}$$

$$S_{hm} := \left\{\mathbf{u} \in H_S \,|\, \operatorname{div}^h \mathbf{u} = 0,\ \overline{\operatorname{curl}^h}\mathbf{u} = 0\right\}\,, \tag{2.35}$$

be the subspaces of discrete vector fields which are div-free, curl-free and harmonic respectively.

Likewise, discrete Hodge decomposition proposed in theorem 3 also leads to the seperation of the finite-dimensional function space H_S:

Theorem 4. *For the finite-dimensional function space H_S of discrete vector fields, the direct sum holds:*

$$H_S \;=\; S_{ir,0} \oplus S_{sol} \oplus S_{hm}\,. \tag{2.36}$$

2.1.3 Equivalent Flow Representations

Now we consider the equivalent representations of the vector field $\mathbf{u} \in H_S$ with the notations of mimetic finite-difference introduced above.

Consider Gauss' theorem (2.17) and (2.18) for any vector field $\mathbf{u} \in H_S$. We say that $\rho \in H_V$ and $\nu \in \partial H_S$ fulfill the *compatibility condition* if

$$\langle \mathbf{1}_{\dim H_V}, \rho\rangle_{H_V} \;=\; \mathbf{1}^{\mathsf{T}}_{\dim\partial H_S}\,\nu\,, \tag{2.37}$$

where $\mathbf{1}_n$ is a $n-$dimensional vector whose elements are all 1 and $\dim \mathrm{H}$ gives the dimension of the finite-dimentional space H.

In what follows, we will make use of another equivalent representation, besides $\mathbf{u} \in H_S$. To this end, we consider the operetor $A : H_S \to H_V \oplus H_P^o \oplus \partial H_S$ given by

$$A := \begin{pmatrix} \mathrm{div}^h \\ \overline{\mathrm{curl}^h} \\ \mathbb{B}_\mathrm{n}^h \end{pmatrix} \in \mathbb{R}^{\mathrm{dim} H_S+1, \mathrm{dim} H_S} \ , \tag{2.38}$$

where the $\overline{\mathrm{curl}^h}$ operator is naturally extended to the whole space H_S. The operator A has full rank $\mathrm{dim} H_S$. Moreover, we see by (2.18) that $(\rho, \omega, \nu)^\intercal$ is in the image of A if and only if ρ and ν fulfill the compatibility condition (2.37). In this case, the representation of $\mathbf{u} \in H_S$ in terms of $(\rho, \omega, \nu)^\intercal$ is given by $u = A^\dagger(\rho, \omega, \nu)^\intercal$, where $A^\dagger = (A^\intercal A)^{-1} A^\intercal$ denotes the pseudoinverse of A.

Proposition 5. *There is a one–to–one correspondence between the spaces H_S and*

$$V_S := \{(\rho, \omega, \nu)^\intercal : \mathbf{1}^\intercal_{\mathrm{dim} H_V} \rho = \mathbf{1}^\intercal_{\mathrm{dim} \partial H_S} \nu\} \ , \tag{2.39}$$

where $\mathbf{u} \in H_S$, $\rho = \mathrm{div}^h u$, $\omega = \overline{\mathrm{curl}^h} u$, $\nu = \mathbb{B}_\mathrm{n}^h u$, and

$$\mathbf{u} = A^\dagger(\rho, \omega, \nu)^\intercal \tag{2.40}$$

Remark. In practice, we do not compute $\mathbf{u} = A^\dagger(\rho, \omega, \nu)^\intercal$ which is ill-conditioned. Rather, we solve both the Neumann problem (2.22) and the Dirichlet problem (2.23), and insert the solutions into (2.20).

By means of the equivalent representation (2.39) of $\mathbf{u} \in H_S$, we can write the discrete orthogonal decomposition (2.20) and Hodge decomposition (2.30) in a simplified manner:

$$\big((\mathbf{u} \in H_S) := (\rho, \omega, \nu)^\intercal\big) \ = \ (\rho, 0, \nu)^\intercal \oplus (0, \omega, 0)^\intercal$$

for the discrete orthogonal decomposition, where $(\rho, 0, \nu)^\intercal$ represents the curl-free vector field (the vanishing curl element), $(0, \omega, 0)^\intercal$ represents the div-free vector fieled (the vanishing curl element) and the operator $\oplus$ represents the orthogonality of the two flow elements; and

$$\big((\mathbf{u} \in H_S) := (\rho, \omega, \nu)^\intercal\big) \ = \ (\rho, 0, \nu_1)^\intercal \oplus (0, \omega, 0)^\intercal \oplus (0, 0, \nu_2)^\intercal$$

for discrete Hodge decomposition, where $\nu = \nu_1 + \nu_2$ and $(0, 0, \nu_2)^\intercal$ gives a harmonic vector field.

2.1.4 Discrete Extended Decomposition

Let C_ρ and C_ω denote the constants proportional to the mean of the divergence and the curl of $\mathbf{u} \in H_S$, that is

$$C_\rho \,=\, \mathbf{1}^\intercal_{\dim H_V}\, \rho/(m\,n)\,, \quad C_\omega \,=\, \mathbf{1}^\intercal_{\dim H_P^o}\, \omega/\big((m-1)\,(n-1)\big) \tag{2.41}$$

where m and n are the horizontal and vertical dimensions of the image grid.

Using these averaged quantities, we can further decompose the flow $\mathbf{u} \in H_S$, represented by $(\rho, \omega, \nu)^\intercal \in V_S$ such that

$$(\rho, \omega, \nu)^\intercal \,=\, (C_\rho, C_\omega, \nu)^\intercal \,+\, (\rho^o, 0, 0)^\intercal \,\oplus\, (0, \omega^o, 0)^\intercal \tag{2.42}$$

where $\mathbf{1}^\intercal_{\dim H_V}\, \rho^o \,=\, 0$ and $\mathbf{1}^\intercal_{\dim H_P^o}\, \omega^o \,=\, 0$ are compatible to the boundary conditions in the sense of (2.39).

Accordingly, we define the components $\mathbf{u}^c \,=\, A^\dagger(C_\rho, C_\omega, \nu)^\intercal$, $\mathbf{u}^o_d \,=\, A^\dagger(\rho^o, 0, 0)^\intercal$ and $\mathbf{u}^o_c \,=\, A^\dagger(0, \omega^o, 0)^\intercal$, and have the following flow decomposition with respect to (2.42):

$$(\mathbf{u} \in H_S) \,=\, \mathbf{u}^c \,+\, \mathbf{u}^o_d \,\oplus\, \mathbf{u}^o_c\,.$$

Vector $\mathbf{u}^c$ or $(C_\rho, C_\omega, \nu)^\intercal$ represents the basic pattern of the non-rigid flow $\mathbf{u} \in H_S$ and its boundary distribution, while $\mathbf{u}^o_d$ and $\mathbf{u}^o_c$, $(\rho^o, 0, 0)^\intercal$ and $(0, \omega^o, 0)^\intercal$ equivalently, are related to the oscillating flow patterns that are curl-free and divergence-free, respectively. Due to nonvanishing spatial averages of C_ρ and C_ω, the component $\mathbf{u}^c$ determines the global structure of the flow field, justifying the term basic flow pattern. It is easy to verify that orthogonality between the components $\mathbf{u}^o_d$ and $\mathbf{u}^o_c$ is preserved such that

$$\langle \mathbf{u}^o_d, \mathbf{u}^o_c \rangle_{H_S} \,=\, 0\,,$$

while the basic flow pattern $\mathbf{u}^c$ and $\mathbf{u}^o_d$, $\mathbf{u}^o_c$ are not orthogonal.

We summarize these properties, thereby extending the discrete orthogonal decomposition Theorem (1)

Proposition 6 (Discrete Extended Decomposition). *Given any 2-D vector field $\mathbf{u} \in H_S$ and $\mathbb{B}^h_{\mathrm{n}}\mathbf{u} \neq 0$, the decomposition (2.42) of $\mathbf{u}$ admits the representation in terms of potential functions ψ^c, ϕ^c, ψ^o and ϕ^o*

$$\mathbf{u} \,=\, (\overline{\nabla^h}\psi^c + \nabla^{\perp,h}\phi^c) + \overline{\nabla^h}\psi^o + \nabla^{\perp,h}\phi^o\,, \tag{2.43}$$

subject to the boundary conditions:

$$\mathbb{B}^h_{\mathrm{n}}\overline{\nabla^h}\psi^c = \mathbb{B}^h_{\mathrm{n}}\mathbf{u}\,,\ \mathbb{B}^h_{\mathrm{n}}\overline{\nabla^h}\psi^o = 0\,,\ \phi^c_{\partial\Omega} = \phi^o_{\partial\Omega} = 0$$

where $\triangle_D\psi^c$ *and* $\triangle_C\phi^c$ *are constants. The discrete flow fields* $\overline{\nabla^h}\psi^o$ *and* $\nabla^{\perp,h}\phi^o$ *are in* H_S *whose diveregence and curl fields are mean-value vanishing respectively.*

This representation is unique up to two constants of ψ^c *and* ψ^o*, respectively. Moreover, the orthogonality relation*

$$\left\langle\overline{\nabla^h}\psi^o, \nabla^{\perp,h}\phi^o\right\rangle_{H_S} = 0 \tag{2.44}$$

holds.

While the components of the decomposition (2.42) and (2.43) are easy to interpret, a single orthogonality relation (2.44) only holds. To improve the latter situation, we consider the alternative decomposition

$$(\rho,\omega,\nu)^\top = (C_\rho,0,\nu)^\top + (\rho^o,0,0)^\top + (0,\omega(:= C_\omega+\omega^o),0)^\top\,. \tag{2.45}$$

Let $\mathbf{u}^c_d = A^\dagger(C_\rho,0,\nu)^\top$, $\mathbf{u}^o_d = A^\dagger(\rho^o,0,0)^\top$ and $\mathbf{u}_c = A^\dagger(0,\omega,0)^\top$. Then the vector field $\mathbf{u}\in H_S$ can be given by $\mathbf{u} = \mathbf{u}^c_d + \mathbf{u}^o_d + \mathbf{u}_c$. As will be shown below, this decomposition provides the basis for representing any vector field in H_S, under additional conditions to be specified, by three mutually orthogonal components. We first summarize the properties of (2.45):

Proposition 7. *Given any 2-D vector field* $\mathbf{u}\in H_S$ *and* $\mathbb{B}^h_{\mathrm{n}}\mathbf{u}\neq 0$*, the decomposition* (2.45) *of* $\mathbf{u}$ *admits the representation in terms of potential functions* ψ^c*,* ψ^o *and* ϕ

$$\mathbf{u} = \overline{\nabla^h}\psi^c + \overline{\nabla^h}\psi^o + \nabla^{\perp,h}\phi\,, \tag{2.46}$$

subject to the boundary conditions:

$$\mathbb{B}^h_{\mathrm{n}}\overline{\nabla^h}\psi^c = \mathbb{B}^h_{\mathrm{n}}\mathbf{u}\,,\ \mathbb{B}^h_{\mathrm{n}}\overline{\nabla^h}\psi^o = 0\,,\ \phi_{\partial\Omega} = 0$$

where $\triangle_D\psi^c$ *is constant.*

This representation is unique up to two constants of ψ^c *and* ψ^o*, respectively. Moreover, the orthogonality conditions*

$$\left\langle\overline{\nabla^h}\psi^c, \nabla^{\perp,h}\phi\right\rangle_{H_S} = 0\,, \quad \left\langle\overline{\nabla^h}\psi^o, \nabla^{\perp,h}\phi\right\rangle_{H_S} = 0 \tag{2.47}$$

holds.

It remains to work out conditions under which the flow components $\overline{\nabla^h}\psi^c$ and $\overline{\nabla^h}\psi^o$ are orthogonal, too. By Green's theorem (2.15), we have

$$\left\langle \overline{\nabla^h}\psi^c, \overline{\nabla^h}\psi^o \right\rangle_{H_S} = -\left\langle \psi^o, \triangle_D \psi^c \right\rangle_{H_V} + \langle \psi^o, \nu \rangle_{\partial\Omega} \, .$$

Taking into account the compatibility condition such that $\mathbf{1}^{\mathsf{T}}_{\mathrm{dim} H_V} \triangle_D \psi^c = \mathbf{1}^{\mathsf{T}}_{\mathrm{dim}\partial H_S} \nu$, we observe that the right hand side is invariant with respect to an arbitrary additive constant C of ψ^o:

$$\begin{aligned} & -\left\langle \psi^o + C, \triangle_D \psi^c \right\rangle_{H_V} + \langle \psi^o + C, \nu \rangle_{\partial\Omega} \\ = & -\left\langle \psi^o, \triangle_D \psi^c \right\rangle_{H_V} + \langle \psi^o, \nu \rangle_{\partial\Omega} + C(-\mathbf{1}^{\mathsf{T}}_{\mathrm{dim} H_V} \triangle_D \psi^c + \mathbf{1}^{\mathsf{T}}_{\mathrm{dim}\partial H_S} \nu) \\ = & -\left\langle \psi^o, \triangle_D \psi^c \right\rangle_{H_V} + \langle \psi^o, \nu \rangle_{\partial\Omega} \, . \end{aligned}$$

Hence, fixing this constant by setting $\langle \psi^o, \nu \rangle_{\partial\Omega} = 0$, we obtain

$$-\left\langle \psi^o, \triangle_D \psi^c \right\rangle_{H_V} + \langle \psi^o, \nu \rangle_{\partial\Omega} = -C_\rho \big(\mathbf{1}^{\mathsf{T}}_{\mathrm{dim} H_V} \psi^o \big) \, ,$$

given that $\triangle_D \psi^c = C_\rho$ is constant. It follows that $\left\langle \overline{\nabla^h}\psi^c, \overline{\nabla^h}\psi^o \right\rangle_{H_S} = 0$ if $C_\rho = 0$, which means that the total divergence of flow $\mathbf{u}$ is zero and that the flow entering and leaving the domain Ω is balanced.

The above procedures can be summarized as

Proposition 8. *Given any 2-D vector field $\mathbf{u} \in H_S$ with $\mathbb{B}^h_{\mathrm{n}} \mathbf{u} \neq 0$ and that the balanced boundary flow condition*

$$\mathbf{1}^{T}_{\mathrm{dim}\partial H_S} \mathbb{B}^h_{\mathrm{n}} \mathbf{u} = 0$$

holds. Then $\mathbf{u} \in H_S$ can be represented in terms of potential functions ψ^c, ψ^o and ϕ

$$\mathbf{u} = \overline{\nabla^h}\psi^c + \overline{\nabla^h}\psi^o + \nabla^{\perp,h}\phi \, , \tag{2.48}$$

subject to the boundary conditions:

$$\mathbb{B}^h_{\mathrm{n}} \overline{\nabla^h}\psi^c = \mathbb{B}^h_{\mathrm{n}} \mathbf{u} \, , \; \mathbb{B}^h_{\mathrm{n}} \overline{\nabla^h}\psi^o = 0 \, , \; \phi_{\partial\Omega} = 0$$

where $\triangle_D \psi^c \equiv 0$.

This representation is unique up to two constants of ψ^c and ψ^o, respectively. Moreover, all the three flow components $\overline{\nabla^h}\psi^c$, $\overline{\nabla^h}\psi^o$ and $\nabla^{\perp,h}\phi$ are mutually orthogonal to each other.

Note that the flow part $\overline{\nabla^h}\psi^c$ turns out to be a harmonic flow, since it is both divergence and curl free.

Now we define further subspaces in addition to (2.26) and (2.27):

- $H_{S,\bar{o}} \subset H_S$: the subspace of vector fields with $\mathbf{1}^{\mathsf{T}}_{\mathrm{dim}\partial H_S} \mathbb{B}^h_{\mathrm{n}} \mathbf{u} \, = \, 0$;
- $S_{ir,o} \subset S_{ir}$: the subspace of irrotational vector fields with zero boundary flow;
- $S_{ir,C} \subset S_{ir}$: the subspace of irrotational vector fields with constant divergence;
- $S_{sol,C} \subset S_{sol}$: the subspace of div-free vector fields with constant curl;
- $S_{\mathrm{div},0}$: the subspace of vector fields with vanishing divergence.

Based on these definitions, we summarize consequences of Prop. 7 - 8:

Corollary 9. *The linear space H_S of 2D vector fields admits the following decompositions:*

$$H_S = (S_{ir,C} + S_{ir,o}) \oplus S_{sol} \, ; \tag{2.49}$$

$$H_{S,\bar{o}} = S_{\mathrm{div},0} \oplus S_{ir,o} \oplus S_{sol} \, . \tag{2.50}$$

2.2 Non-rigid Flow Estimation

In this part, we consider the variational method to non-rigid flow estimation through two sequential images. Such technique has been extensively investigated after the pioneering work of Horn and Schunck [80]. For the applied flow regularizars, we especially focus on the high-order div-curl penalty term (2.1a).

2.2.1 Introduction

Given the image sequence $\{g(x,y,t) : t \in [0,T]\}$ with a time parameter t, a common assumption is that intensities of the images given at two sequential time spots t_1 and t_2, $t_2 > t_1$ are preserved over time:

$$g_1(x+u_1, y+u_2) \; = \; g_2(x,y) \, , \tag{2.51}$$

where $g_i(x,y) = g(x,y,t_i)$, $i = 1,2$, and the displacement or flow field $\mathbf{u}(x,y) = (u_1(x,y), u_2(x,y))^{\top}$ denotes the instantaneous displacements of image elements during time $t_2 - t_1$.

Then the linearized version of (2.51), by the firs-order expansion of $g_1(x+u_1, y+u_2)$ for *small enough displacements* of $u_1(x,y)$ and $u_2(x,y)$, yields the *brightness-contant constraint*

$$g_2(x,y) - g_1(x+u_1, y+u_2) \; \simeq \; (g_t := g_2 - g_1) - \nabla g_1 \cdot \mathbf{u} \; = \; 0 \, . \tag{2.52}$$

Instead of (2.52), variational optical-flow methods consider the least squares function $\|g_t - \nabla g \cdot \mathbf{u}\|^2$ to force the linear equality (2.52) approximately. Other generalizations of the brightness-constant constraint (2.52) exist, such as the local Lucas-Kanade estimator [32, 33] or L_1 penalization term to (2.52), but are out of scope of studies in this thesis.

Obviously, (2.52) can not be uniquely approached because at each location $x \in \Omega$ and time t it consists of solving a single scalar equation for two scalar unknowns and leads to an underdetermined linear equation. This is known as the *aperture problem*. To overcome it, additional requirements have to be imposed. Hence, the variational optical-flow method applies a priori information about the flow field $\mathbf{u}$ as the regularization term to construct a theoretically well-posed energy minimization problem, mostly convex, and single out the unique optimal vector field $\mathbf{u}^*$, which results in the following flow estimation:

$$\min_{\mathbf{u}} \|g_t - \nabla g \cdot \mathbf{u}\|^2 + R(\mathbf{u}) \,. \tag{2.53}$$

In 1981, Horn and Schunck [80] first introduced such variational formulation (2.53) for the optical-flow estimation joint with a quadratic flow smoothness term, i.e. the *Horn–Schunck model*

$$\min_{\mathbf{u}} \|g_t - \nabla g \cdot \mathbf{u}\|^2 + \alpha \int_{\Omega} \left(|\nabla u_1|^2 + |\nabla u_2|^2\right) \mathrm{d}x \,. \tag{2.54}$$

Obviously, the Horn–Schunck model suffers from the same drawbacks as the linear image filtering. The solution of (2.54) often creates very blurry optic flow fields where the blur appears also across important flow discontinuities or boundaries. One way to overcome this limitation consists again of using other nonsmooth function as the regularizer, e.g.,

$$\min_{\mathbf{u}} \|g_t - \nabla g \cdot \mathbf{u}\|^2 + \alpha \int_{\Omega} (|\nabla u_1|^2 + |\nabla u_2|^2)^{\frac{1}{2}} \, \mathrm{d}x \,, \tag{2.55}$$

as counterpart to the total-variation image denoising, cf. [7, 74, 77].

In this thesis, we focus on the estimation of non-rigid motion fields with complex spatial scales, which poses great challenges for analysis and estimation. Directly using derivatives of coordinates u_1 and u_2 in the regularizer like (2.54) and (2.55) often yields oversmoothness and distortion of flow patterns in its computation result, where the nonsmooth total-variation like regularizar used in (2.55) makes results even more worse. Rather, the high-order div-curl regularizar (2.1a) which employs the smoothness of variances of divergence and curl was found to be significantly successful for the recovery of non-rigid vector fields (see Fig. 2.3 for the motivation). The regularization function (2.1a) was first introduced in vector interpolation by splines [3]. After that, it was applied for optical flow estimation by Suter [152], then Gupta et al [75] for cadiac motions. In [48, 49], Memin et al explicitly applied it to build up a multi-scale estimation scheme of fluid flows. In this work, we study the non-rigid flow estimation approach based on flow decomposition (2.20) discussed in the previous section. Introduce the orthogonal flow decomposition (2.19) of $\mathbf{u}$, i.e. $\mathbf{u} = \nabla\psi + \nabla^{\perp}\phi$, to the high-order div-curl regularizar (2.1a) and apply it to (2.53), we, therefore, obtain the following energy minimization model:

$$\min_{\psi,\phi} \|g_t - \nabla g \cdot (\nabla\psi + \nabla^{\perp}\phi)\|^2 + \alpha \int_{\Omega} \big(|\nabla\triangle\psi|^2 \, \mathrm{d}x + |\nabla\triangle\phi|^2\big) \, \mathrm{d}x \tag{2.56}$$

A direct advantage for (2.56) in numerics is that it replaces the original estimation of the vector field $\mathbf{u}$ by the approximation of two potential functions ψ and ϕ, each of which poses a much smaller estimation problem (nearly half of the original problem in terms of dimensions). At the same time, flow decompositions also provide a new and deep perspective to analyze non-rigid flow estimation problem in theory. In addition, (2.56) can be extended to estimate physically incompressible flows in a straight way such that

$$\min_{\phi} \|g_t - \nabla g \cdot \nabla^{\perp}\phi\|^2 + |\nabla\triangle\phi|^2\big) \, \mathrm{d}x \,. \tag{2.57}$$

(2.57) greatly simplifies estimation of such non-rigid flows incorporating the div-free constraint: it properly avoid forcing the div-free constraint and gives the flow field exactly in the solenoidal subspace; moreover, the unknown variables of (2.57) are clearly only half of direct estimation of unknown $\mathbf{u}$!

2.2.2 Flow Decomposition Based Non-rigid Flow Estimation

Now we consider the high-order div-curl regularizar (2.1a). By the orthogonal decomposition (2.19), it gives

$$\alpha \int_{\Omega} \big(|\nabla \operatorname{div} \mathbf{u}|^2 \, \mathrm{d}x + |\nabla \operatorname{curl} \mathbf{u}|^2\big) \, \mathrm{d}x \;=\; \alpha \int_{\Omega} \big(|\nabla\triangle\psi|^2 \, \mathrm{d}x + |\nabla\triangle\phi|^2\big) \, \mathrm{d}x \,. \tag{2.58}$$

In view of the *extended flow decomposition* (Prop. 6), (2.1a) measures the variation of flow patterns in terms of variances of divergence and curl, but *does not penalize* the basic components $(C_\rho, C_\omega, \nu)^{\mathsf{T}}$.

As discussed in sec.2.1.1, both standard finite differences or finite elements discretization lead to finite-dimensional representations which do not satisfy (2.20) and (2.21) exactly. As a result, penalizing one component may affect the other component too. Therefore, we adopt the mimetic finite-difference framework of sec.2.1.2 which leads to the following discretization

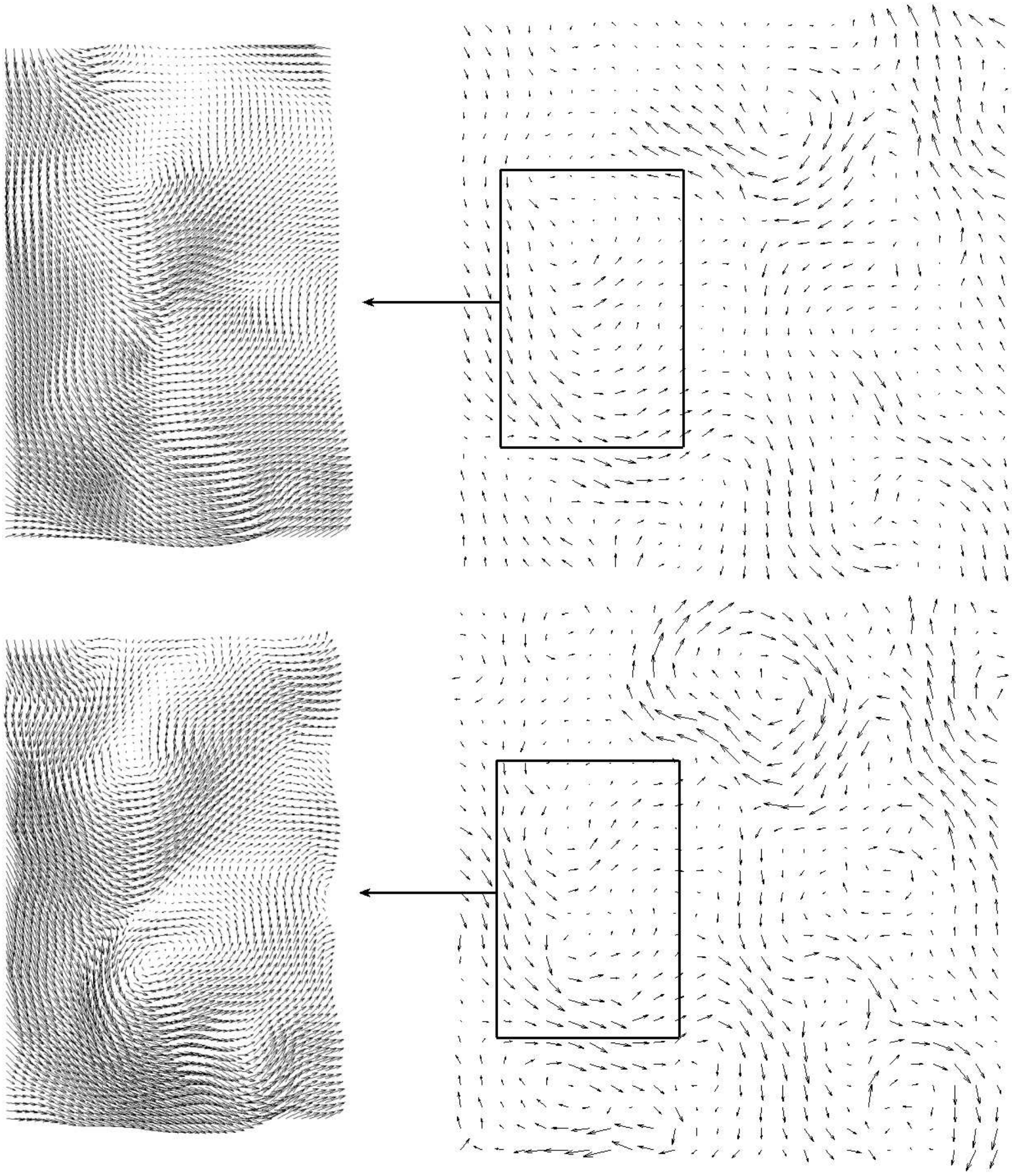


Figure 2.3: Estimation of a typical solenoidal flow field u. Top: Restored flow based on the Horn–Schunck model (2.54). Bottom: Restored flow based on the second-order model (2.56). Vortex structures are better recovered by the div-curl approach (bottom) introduced in this section.

of (2.58):

$$\alpha \int_{\Omega} \big(|\nabla \triangle \psi|^2 \,\mathrm{d}x + |\nabla \triangle \phi|^2\big) \,\mathrm{d}x = \alpha\big(\Big\| \overline{\nabla^h} \operatorname{div}^h u \Big\|_{H_S}^2 + \Big\| \nabla^h \overline{\operatorname{curl}^h} u \Big\|_{H_E}^2 \big) \tag{2.59}$$

$$= \alpha\big(\Big\| \overline{\nabla^h} \triangle_D \psi \Big\|_{H_S}^2 + \lambda_2 \Big\| \nabla^h \triangle_C \phi \Big\|_{H_E}^2 \big) \,. \tag{2.60}$$

Therefore, for the non-rigid flow estimation (2.56), inserting (2.60) boils down to the following optimization problem

$$\min_{\psi,\phi} \quad \|g_t - \overline{\nabla^h} g \cdot (\overline{\nabla^h} \psi + \nabla^{\perp,h}\phi)\|_{H_V}^2 + \alpha\big(\|\overline{\nabla^h}\triangle_D \psi\|_{H_S}^2 + \|\nabla^h \triangle_C \phi\|_{H_E}^2\big) \tag{2.61}$$

subject to the linear constraints

$$\mathbf{1}_{\dim H_{V+\partial V}}^{\mathsf{T}} \psi = 0 \,, \quad \phi_{\partial\Omega} = 0 \,. \tag{2.62}$$

Note that the first constraint fixes the free constant mentioned in the Orthogonal Decomposition Theorem 1. Furthermore, the vector fields in (2.61) are elements in two orthogonal subspaces (2.28), and thus may be determined parallelly by subspace correction methods [170, 44, 157, 154] in a fast way.

Estimation of Incompressible Flows

An important special case, particularly in applications of experimental fluid dynamics, concerns the estimation of incompressible flows, i.e. div-free. In this case, the decomposition represented by (2.48) reduces to :

$$\mathbf{u} = \overline{\nabla^h} \psi^c + \nabla^{\perp,h}\phi \tag{2.63}$$

where $\overline{\nabla^h}\psi^c$ gives the harmonic component which only depends on the boundary flow $\mathbb{B}_{\mathrm{n}}^h \mathbf{u}$:

$$\triangle_D \psi^c = 0 \,, \quad \mathbb{B}_{\mathrm{n}}^h \overline{\nabla^h} \psi^c = \mathbb{B}_{\mathrm{n}}^h \mathbf{u}. \tag{2.64}$$

Thus, in order to estimate solenoidal flows, we consider, instead of (2.61), the minimization problem:

$$\min_{\psi^c,\phi} F_{sol}(\psi^c,\phi) = \Big\| g_t - \overline{\nabla^h} g \cdot (\overline{\nabla^h}\psi^c + \nabla^{\perp,h}\phi) \Big\|_{H_V}^2 + \alpha \Big\| \nabla^h \triangle_C \phi \Big\|_{H_E}^2 \tag{2.65}$$

subject to the constraints:

$$\triangle_D \psi^c = 0 \,, \quad \mathbf{1}_{\dim H_{V+\partial V}}^{\mathsf{T}} \psi^c = 0 \,, \quad \phi_{\partial\Omega} = 0 \tag{2.66}$$

Note that the vector fields of (2.65) are elements of orthogonal subspaces of harmonic and solenoidal, and thus may be determined in parallel by subspace correction methods.

2.2.3 Well-Posedness and Stability

In this section, we analyse well-posedness of the variational problem (2.61). To this end, we state the conditions under which the respective functionals are strictly convex to the variable $\mathbf{u} \in H_S$.

In order to compactly state the various conditions for well-posedness, we complement the list of subspaces defined at the end of section 2.1.4. To this end, we denote the linear operator $G := (\overline{\nabla^h} g \cdot\ )$ and use the notation $N(A)$ for the null-space of a linear operator A:

- $S_{ir,C} \cap S_{sol,C}$ gives subspace of H_S with both constant div and curl, which gives
$$S_{C,\mathrm{div}} \cap S_{C,\mathrm{curl}} \;=\; N(\overline{\nabla^h}\,\mathrm{div}^h) \,\cap\, N(\nabla^h \overline{\mathrm{curl}^h});$$
- $S_{G0} \subset H_S$: the subspace of vector fields
$$S_{G0} = \{\mathbf{u} \mid G\mathbf{u} \;=\; 0\}.$$

Well-Posedness of General Flows

In terms of $\mathbf{u} \in H_S$, the variational approach (2.61) for estimating general non-rigid flows amounts to the unconstrained convex minimization problem:

$$\min_{\mathbf{u} \in H_S} \quad \|\partial_t g - G\mathbf{u}\|^2_{H_V} + \alpha\Big(\left\|\overline{\nabla^h}\,\mathrm{div}^h u\right\|^2_{H_S} + \left\|\nabla^h \overline{\mathrm{curl}^h} u\right\|^2_{H_E} \Big)\,. \tag{2.67}$$

As a consequence, the following is immediate:

Proposition 10. *Problem* (2.61) *is well-posed, i.e. strictly convex, if and only if*

$$S_{G0} \cap (S_{ir,C} \cap S_{sol,C}) = \{0\}$$

As the subspace $S_{ir,C} \cap S_{sol,C}$ is fixed with the problem dimension, this condition mostly requires a sufficiently high spatial variation of the grayvalue images g to obtain well-posedness. A counter-example is given by any image g with $\triangle_C g \;=\; \mathrm{curl}^h\, \overline{\nabla^h}^{\perp} g \;=\; C$, because for the vector field $\overline{\nabla^h}^{\perp} g \in H_E$ the product with $\overline{\nabla^h} g \,\cdot\, \overline{\nabla^h}^{\perp} g \equiv 0$ vanishes at each grid cell.

Well-Posedness on Solenoidal Flows

The variational approach (2.65) for estimating divergence-free flows amounts to a convex quadratic minimization problem with linear equality constraints. Expressing the restriction

$\mathbf{u}S_{sol}$ through the constraint $\operatorname{div}^h \mathbf{u} = 0$, we reformulate (2.65):

$$\begin{aligned} \min_{\mathbf{u}\in H_S} \quad & \|\partial_t g - G\mathbf{u}\|^2_{H_V} + \alpha \left\|\nabla^h \overline{\operatorname{curl}^h} \mathbf{u}\right\|^2_{H_E} \\ \text{s.t.} \quad & \operatorname{div}^h \mathbf{u} = 0 \end{aligned} \tag{2.68}$$

As a consequence, its condition for well-posedness reads:

Proposition 11. *Problems* (2.65) *and* (2.68) *are well-posed if and only if*

$$S_{G0} \cap S_{sol} \; = \; \{0\} \, .$$

Stabilities

It is well-known that existence of a unique solution, as established in the previous section, does not say much about *numerical* stability. Indeed, inspection of the second-order regularizer (2.58) reveals a particular sensivity of $\mathbf{u}$ with respect to the image data and suggests using a corresponding regularizer.

To motivate this additional term, we rewrite the estimation functional using the representation $(\rho, \omega, u_n)^\top$ (cf. Prop. 5):

$$\min_{\rho,\omega,u_n} \quad \left\|\partial_t g - GA^\dagger(\rho,\omega,u_n)^\top\right\|^2 + \alpha(\|\nabla\rho\|^2 + \|\nabla\omega\|^2) \, . \tag{2.69}$$

We consider the extended decomposition due to Prop. 6 and (2.43). Considering (2.42), the variance of divergence and curl field related to the two components $(\rho^o, 0, 0)^\top$ and $(0, \omega^o, 0)^\top$ can be penalized and constrained by the respective regularizer terms of (2.1a). However, for the last part $(C_\rho, C_\omega, \nu)^\top$ which has constant divergence and curl, both regularization terms are not effective. This part strongly depends on the normal flow at the boundary ν. In fact, the flow $(C_\rho, C_\omega, \nu)^\top$ is only weakly constrained by the data term, that is the gradient field of image data g at the boundary whose estimate is particularly noisy and unreliable.

Therefore, in practice, it turns out to be useful to reduce this sensivity of $\mathbf{u}$ by including an additional regularizer which constrains the boundary values such that

$$\int_{\partial\Omega} (\partial_n u)^2 \, dl \tag{2.70}$$

Obviously, this constraint term favors continuity of vector field $\mathbf{u}(x)$ between the boundary $\partial\Omega$ and the interior $\mathring{\Omega}$. By virtue of the orthogonal decomposition, (2.70) can be directly expressed in terms of the potential field ψ

$$\int_{\partial\Omega} |\partial_n(\nabla\psi)|^2 \, dl \, . \tag{2.71}$$

Therefore, its discrete version reads

$$\left\| \mathbb{P}\, \overline{\nabla^h} \psi \right\|_{bc}^2 \tag{2.72}$$

where the respective matrix $\mathbb{P}$ represents the operator ∂_n in (2.71), and $\|\cdot\|_{bc}$ is the norm evaluating elements along the boundary $\partial\Omega$.

2.2.4 Space Decomposition Algorithms and Multi-level Implementation

So as to solve the flow estimation problem (2.61), we apply the method of function space decomposition to restore the two potential fields $\phi(x)$ and $\psi(x)$ directly. Space decomposition provides a nice framework for analysing domain decomposition and multigrid methods [170, 171]. The essence is to decompose the solution space into a sum of subspaces and then solve the original optimization problem sequentially or in parallel in each subspace. Extensions to the common convex optimization problems were presented in [156, 157], and convergence rates are analyzed in [44, 154].

We describe the space decomposition method and its application to our approach in sections 2.2.4 and 2.2.4. Subsequently, we detail in sections 2.2.4 a multi-level representation of flow fields adapted to the orthogonal decomposition.

Iterative Subspace Corrections

Suppose that for a general convex optimization problem

$$\min_{\mathbf{u} \in V} F(\mathbf{u}) \,, \tag{2.73}$$

the solution function space V can be decomposed into a sum of subspaces

$$V = V_1 + V_2 + \ldots + V_m \,, \tag{2.74}$$

not necessarily orthogonal to each other. For any $\mathbf{u} \in V$, there exist $\mathbf{u}_i \in V_i$, such that $\mathbf{u} = \sum_{i=1}^m \mathbf{u}_i$. Conversely, if $\mathbf{u}_i \in V_i$, then $\sum_{i=1}^m \mathbf{u}_i \in V$. Note that in general the sum is not the direct sum, and the decomposition of $\mathbf{u}$ is also not unique.

There are two versions of iterative algorithms, the Parallel Subspace Corrections (PSC) and the Successive Subspace Corrections (SSC). In each step, PSC and SSC compute the next iterate in V through searching each subspaces V_i, $i = 1, \ldots, m$, in parallel or sequentially, respectively.

We see that either PSC or SSC is an algorithm also called alternating direction method. With suitable assumptions about the objective function $F(\mathbf{u})$ and the decomposition scheme of the function space V, both PSC and SSC converge.

In this thesis, we focus on SSC which often converges faster than PSC in practice and state it in Alg. 1

Algorithm 1 Successive Subspace Corrections

- Step 1. Choose $\mathbf{u}_i^0 \in V_i$.
- Step 2. For the n-th iteration, compute $\hat{\mathbf{u}}_i^{n+1} \in V_i$ sequentially for $i = 1, \dots, m$, by minimization:

$$\begin{aligned} &F\big(\sum_{1 \le k < i} \mathbf{u}_k^{n+1} + \hat{\mathbf{u}}_i^{n+1} + \sum_{i < k \le m} \mathbf{u}_k^n\big) \\ \le &F\big(\sum_{1 \le k < i} \mathbf{u}_k^{n+1} + \mathbf{v}_i + \sum_{i < k \le m} \mathbf{u}_k^n\big), \quad \forall \mathbf{v}_i \in V_i. \end{aligned} \tag{2.75}$$

  Choose $\mathbf{u}_i^{n+1} \in V_i\,,\ i = 1, \dots, m$, such that

$$\left\| \mathbf{u}_i^{n+1} - \hat{\mathbf{u}}_i^{n+1} \right\|_V \le \epsilon_0 \left\| \mathbf{u}_i^n - \hat{\mathbf{u}}_i^{n+1} \right\|_V,\ 0 \le \epsilon_0 \le 1\,. \tag{2.76}$$

- Step 3. Go to the next iteration.

In practice, we choose $\mathbf{u}_i^{n+1} = (1 - \epsilon_0)\hat{\mathbf{u}}_i^{n+1} + \epsilon_0 \mathbf{u}_i^n$, $\epsilon_0 \in [0.5, 0.75]$, for the SSC algorithm 1.

Application to Flow Estimation

Based on algorithm 1, the estimation of general flows amounts to solving the two subproblems including the boundary stability term (2.72) with a penalty parameter $\gamma > 0$,

$$\begin{aligned} \min_{\psi} \widetilde{F}(\psi, \bar{\phi}) \ &= \left\| \partial_t g - \overline{\nabla^h} g \cdot (\overline{\nabla^h} \psi + \nabla^{\perp,h} \bar{\phi}) \right\|_{H_V}^2 + \\ &\alpha \left\| \overline{\nabla^h} \triangle_D \psi \right\|_{H_S}^2 + \gamma \left\| \mathbb{P}\, \overline{\nabla^h} \psi \right\|_{bc}^2, \end{aligned} \tag{2.77}$$

and

$$\min_{\phi} \widetilde{F}(\bar{\psi}, \phi) \ = \ \left\| \partial_t g - \overline{\nabla^h} g \cdot (\overline{\nabla^h} \bar{\psi} + \nabla^{\perp,h} \phi) \right\|_{H_V}^2 + \alpha \left\| \nabla^h \triangle_C \phi \right\|_{H_E}^2, \tag{2.78}$$

and subject to the linear constraint (2.62) for ψ, whereas the constraint for ϕ is directly encoded by the discretization. Here $\bar{\psi}$ and $\bar{\phi}$ in (2.77) and (2.78) are fixed variables at each iteration. To this end, each subproblem is an convex quadratic problem to which the preconditioned conjugate gradient iteration was applied. The $\psi-$step includes a simple projection due to the linear constraint in (2.62).

Concerning estimation of divergence-free flows, approach (2.65) together with (2.66) requires as part of algorithm 1 to solve a linearly constrained quadratic problem in the subspace of harmonic flows. To this end, the Augmented Lagrangian Method [21, 20] is applied. The corresponding augmented Lagrangian function for the $\psi-$subproblem reads:

$$\begin{aligned} L_{\bar{\phi}}(\psi^c, r) = & \left\| \partial_t g - \overline{\nabla^h} g \cdot (\overline{\nabla^h} \psi^c + \nabla^{\perp,h} \bar{\phi}) \right\|^2 \\ & + \langle r, \triangle_D \psi^c \rangle + \frac{c}{2} \left\| \triangle_D \psi^c \right\|^2_{H_V} + \gamma \left\| \mathbb{P} \, \overline{\nabla^h} \psi^c \right\|^2_{bc} \end{aligned} \tag{2.79}$$

with $\bar{\phi}$ being fixed at each iteration step. Direct incorporation into the augmented Langrangian iteration of the remaining linear equality $\sum_{H_{V+\partial V}} \psi = 0$ in (2.66) would destroy the sparsity of the matrix of the penalty term and increase computation load. This, in turn, affects the efficiency of the sparse solver. Instead, we simply remove the average from iterates $(\psi^c)^n$ as a simple post-processing step. In practice, the augmented Langrangian iteration converged in less than 10 iterations to achieve a high accuracy.

Multi-level Implementation

In most cases where large displacements of $\mathbf{u}$ exist between the image sequence, the linearized approximation (2.52) of the brightness-constant condition (2.51) is no longer valid. In order to alleviate the local minima problem and to capture such large displacements, the standard multi-level procedure using a sequence of linearizations is applied such that

$$F^l(\mathbf{u}^l) := \left\| \partial_t g^l - \overline{\nabla^h} g^l \cdot \mathbf{u}^l \right\|^2_{H_V} + R(\mathbf{u}^l) \; , \tag{2.80}$$

where $\{g_1^l, g_2^l\}_{l=0,1,\dots,m}$ denote linear scale-space representations of the given image pair, $\partial_t g^l = g_2^l(x) - g_1^l(x + \tilde{\mathbf{u}}^l(x))$ and $\tilde{\mathbf{u}}^l$ is the approximation from the coarser levels.

In this regard, two image pyramids $\{g_i^l\}_{l=1,\dots,m}$, $i = 1, 2$, are constructed. $l = 0$ denotes the original image, and $l = m$ denotes the coarsest level. At the level l, given potential fields $\tilde{\psi}^l$, $\tilde{\phi}^l$ and the velocity field $\tilde{\mathbf{u}}^l = \overline{\nabla^h} \tilde{\psi}^l + \nabla^{\perp,h} \tilde{\phi}^l$, image g_1^l is warped to $\tilde{g}_1^l = g_1^l(x + \tilde{\mathbf{u}}^l)$. The image flow between the two images $\tilde{g}_1^l$ and g_2^l is assumed to be small enough to allow for accurate linearization:

$$\partial_t g^l = g_2^l - \tilde{g}_1^l \tag{2.81}$$

$$\Delta_{\mathbf{u}}^l = \overline{\nabla^h} g_1^l \cdot (\mathbf{u}^l - \tilde{\mathbf{u}}^l) \tag{2.82}$$

$$\Delta_{\psi,\phi}^l = \overline{\nabla^h} g_1^l \cdot (\overline{\nabla^h} (\psi^l - \tilde{\psi}^l) + \nabla^{\perp,h} (\phi^l - \tilde{\phi}^l)). \tag{2.83}$$

The residual motion field $\mathbf{u}^l$, in terms of $\overline{\nabla^h}\psi^l + \nabla^{\perp,h}\phi^l$, is estimated by solving the problem

$$\begin{aligned}\min_{\psi^l,\phi^l} F(\psi^l,\phi^l) &= \left\|\Delta^l_{\psi,\phi} + \partial_t g^l\right\|^2_{H_V} + \alpha\left\|\overline{\nabla^h}\triangle_D\psi^l\right\|^2_{H_S} \\ &\quad + \alpha\left\|\nabla^h\triangle_C\phi^l\right\|^2_{H_E} + \gamma\left\|\mathbb{P}\,\overline{\nabla^h}\psi^l\right\|^2_{bc} \qquad (2.84)\end{aligned}$$

The minimizer ψ^l, ϕ^l and $\mathbf{u}^l$ are postprocessed, by extention, to yield the initialization $\tilde{\psi}^{l-1}$, $\tilde{\phi}^{l-1}$ and $\tilde{\mathbf{u}}^{l-1}$ of the next finer level $l-1$. The whole process is started at the coarsest level m with $\tilde{\psi}^m = 0$, $\tilde{\phi}^m = 0$ and $\tilde{\mathbf{u}}^m = 0$.

2.2.5 Experiments

In this section, we validate the proposed flow decomposition based approach with numerical experiments. Some of the vector fields depicted in corresponding figures are scaled by a factor 2 or color-coded for better visibility.

In pactice, evaluating non-rigid flows by computing the average angular and norm error, respectively, induced by the inner product of the space $(L^2(\Omega))^2 = L^2(\Omega) \times L^2(\Omega)$ [16], appeared to us too insensitive to the important flow structures. Therefore, we suggest error measures that also take into account divergence and curl of flow structures:

$$e_{norm} := \frac{\langle \mathbf{u},\mathbf{u}\rangle_{DC}}{N}, \quad e_{ang} := \arccos\frac{\langle \mathbf{u},\mathbf{v}\rangle_{DC} + 1}{\sqrt{\langle \mathbf{u},\mathbf{u}\rangle_{DC} + 1}\sqrt{\langle \mathbf{v},\mathbf{v}\rangle_{DC} + 1}} \qquad (2.85)$$

where we adopt the average angular and norm error measures but use the inner products of the space $H(\mathrm{div};\Omega) \cap H(\mathrm{curl};\Omega)$ (see, e.g., [67]):

$$\langle \mathbf{u},\mathbf{v}\rangle_{DC} := \langle \mathbf{u},\mathbf{v}\rangle_{H_S} + \left\langle \mathrm{div}^h\,\mathbf{u}, \mathrm{div}^h\,\mathbf{v}\right\rangle_{H_V} + \left\langle \overline{\mathrm{curl}^h}\mathbf{u}, \overline{\mathrm{curl}^h}\mathbf{v}\right\rangle_{H_P}. \qquad (2.86)$$

Numerical Stability

The additional boundary regularizar (2.71) is essential for numerical stability. To demonstrate this, we warped a computer-generated grayvalue function with the ground-truth flow field whose color-coded image is shown in the left panel of Figure 2.4, and examined the numerical stability for the resulting variational flow estimation problem (2.61).

Omitting the boundary term by setting for the corresponding weight parameter $\gamma = 0$ leads to a numerically instable problem. The value $\gamma = 1e-6$ results in an ill-conditioned problem and very slow convergence of the numerical iteration. Moreover, the resulting vector field shown in the middle of Figure 2.4 oscillates at the boundary. The value $\gamma = 0.1$ yields a well-conditioned system that converged after 7 iterations and results in an accurate result (Fig. 2.4, right panel).

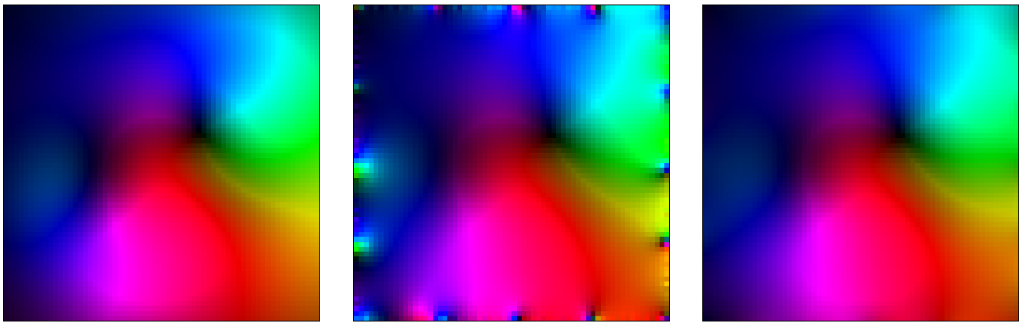


Figure 2.4: Influence of the boundary regularizer (2.71). **Left:** a ground-truth non-rigid velocity field shown color-coded for better visibility. **Middle:** the velocity field estimated with $\lambda_3 = 1e{-}6$ for the boundary regularizer results in an ill-conditoned problem. The estimated motion field may oscillate near the boundary. **Right:** the velocity field estimated with $\lambda_3 = 0.1$.

Ground Truth Experiment

Figure 2.5 shows a particle image sequence provided by the European project FLUID [35] through direct numerical simulation (DNS) [134] of an incompressible turbulent flow field.

The traditional Horn-Schunck method (2.54) was used for comparison. We tuned the related penalty parameters by hand for the Horn-Schunck approach, $\alpha = 0.1$, and for the flow decomposition based approach (2.65), $\alpha = 0.05$ and $\gamma = 0.05$ where γ weights the boundary term (2.72).

Figure 2.5 shows the experimental results. The corresponding errors for the approach (2.56), $e_{norm} = 1.49e-2\,,\ e_{ang} = 6.94°$, are smaller than for the approach of Horn and Schunck (2.54): $e_{norm} = 3.70e-1\,,\ e_{ang} = 31.19°$. By (2.86), these error measures include flow *derivatives*, divergence and curl, as opposed to common measures used in the literature. It can be clearly observed in Figure 2.5 that the flow decomposition based approach recovers the curl field more accurately. Furthermore, even if the image sequence data correspond to incompressible flows, ignoring the corresponding constraint leads to significant flow estimation errors of the Horn and Schunck method (mid-bottom panel).

Estimating Real Solenoidal Flows

Figure 2.6 shows a real world 2D turbulent flow which has been captured in laboratory. This experimental flow has been generated between two thin glass plates [89]. It is visualized through a passive scalar (a mixture of fluorescein and water) that is transported by the flow $\mathbf{u}$. A

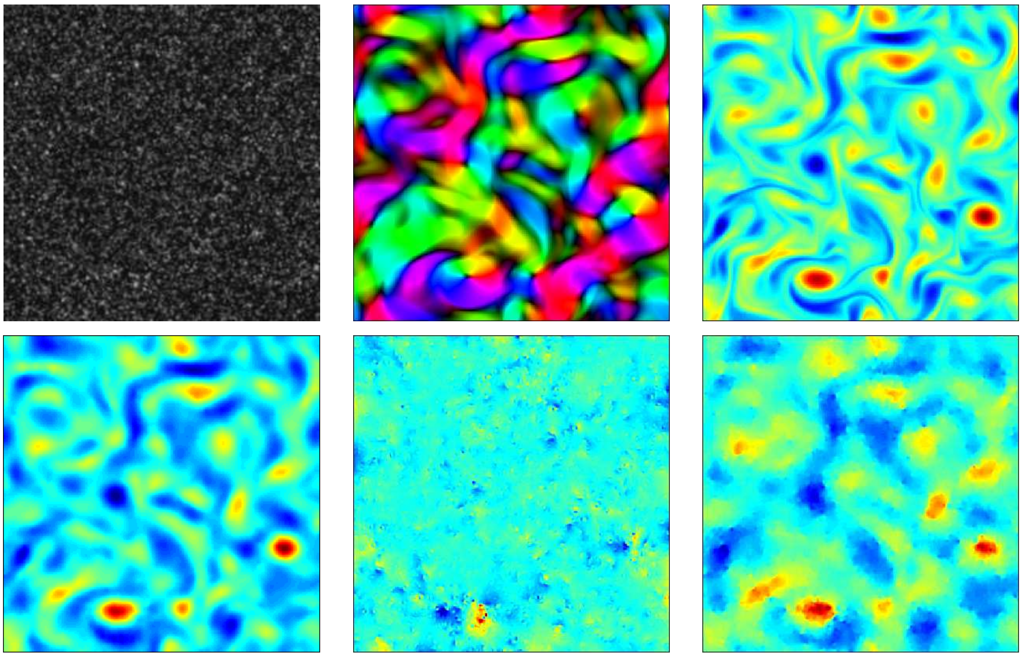


Figure 2.5: **Top left:** the first image of a particle image pair. **Top middle and right:** the ground-truth DNS-simulated divergence-free flow and its curl field. **Bottom left:** the curl field of the estimated flow, which is totally divergence-free, computed by the potential-based approach (2.65). **Bottom middle:** the divergence field, which is also the divergence-error, of the flow resulting from the Horn-Schunck method (2.54). **Bottom right:** the curl field of the flow estimated using the Horn-Schunck approach (2.54). It is apparent that the recovered curl patterns by our approach are much more accurate. The flow error measures reported in the text confirm this quantitatively.

diffusion of the passive scaler can also be observed along time. The measurement of a slight non-vanishing divergence for this 2D flow corresponds to this diffusion effect. Rather than taking into account this effect through developing a specific data term, we focus in this paper on imposing vanishing divergence as a constraint, along with the higher order regularization.

Figure 2.6 shows the result of estimating the div-free flow for the real image sequence based on the multi-level framework, see section 2.2.4 for details. The comparison with first-order regularization (Horn-Schunck approach) in Figure 2.7 clearly reveals the superiority of the potential-based approach (2.65) regarding the reconstruction of vortex structures. Furthermore, the physically plausible constraint of vanishing divergence is satisfied accurately.

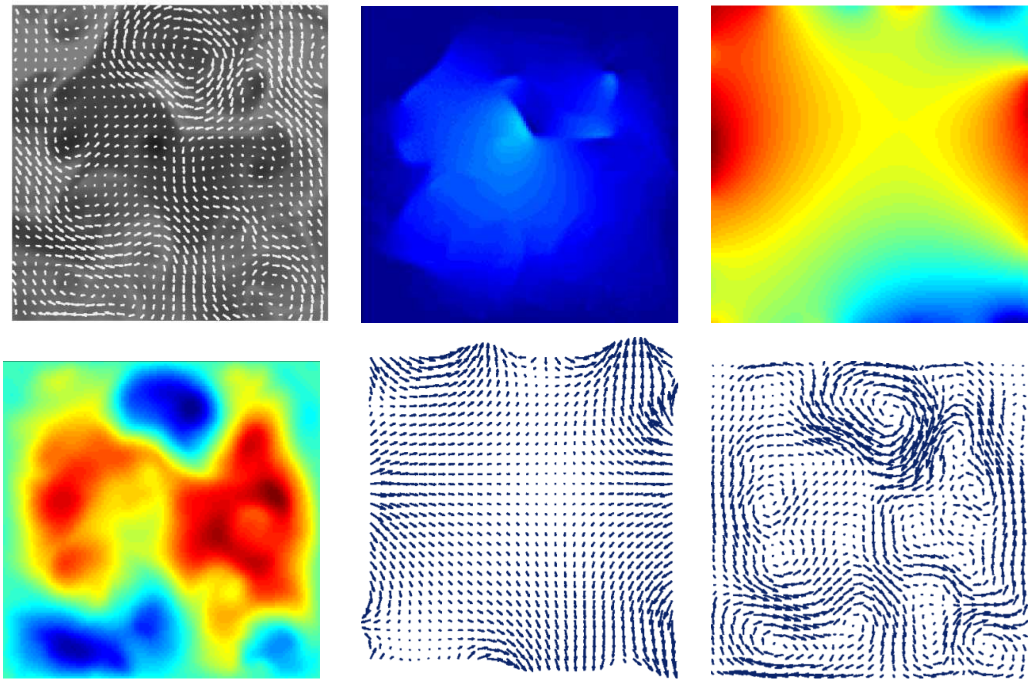


Figure 2.6: **Top left:** the first frame g_1 of a real image sequence together with the estimated solenoidal flow. **Top middle:** the divergence field of the flow is less than $3 * 10^{-12}$. **Top right:** the potential field $\psi(\Omega)$ related to the harmonic flow. **Bottom left:** the potential field $\phi(\Omega)$. **Bottom middle:** the first component of flow: the harmonic flow $\overline{\nabla^h}\psi$. **Bottom right:** the second component of flow $\nabla^{\perp,h}\phi$. A close-up view for comparison with standard regularization is depicted in Figure 2.7.

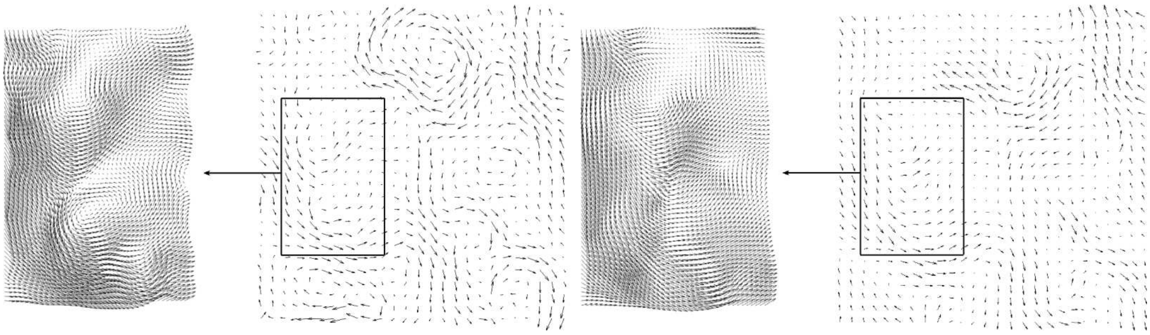


Figure 2.7: **Left:** the restored div-free flow $\mathbf{u}(\Omega)$. **Right:** the restored flow $\mathbf{u}_{hs}(\Omega)$ by the Horn-Schunck approach (2.54). This result clearly shows that vortex structures are better recovered by the potential fields-based approach. Furthermore, the magnitude of the divergence is below 10^{-11}, nearly zero, throughout the image plane.

Estimating General Non-Rigid Real Flows

Figures 2.8 and 2.9 show general non-rigid flows estimated for two different real image sequences, by the multi-level scheme (2.84). The images show convective cloud cells at high altitudes. These clouds undergo strong upward motions until they reach the top of the atmosphere, the tropaupose, and then start to decline slowly. These cells are responsible for violent showers and generate locally very strong winds. Their apparent motions correspond to highly divergent motions whose supports roughly correlate with cloud boundaries. Nevertheless, such a divergent motion does not necessarily have exactly the same shape as the cloud cell. The recovered motion fields show that we have been able to estimate blobs of diverging motions which correspond to the apparent motion of these type of cloud systems.

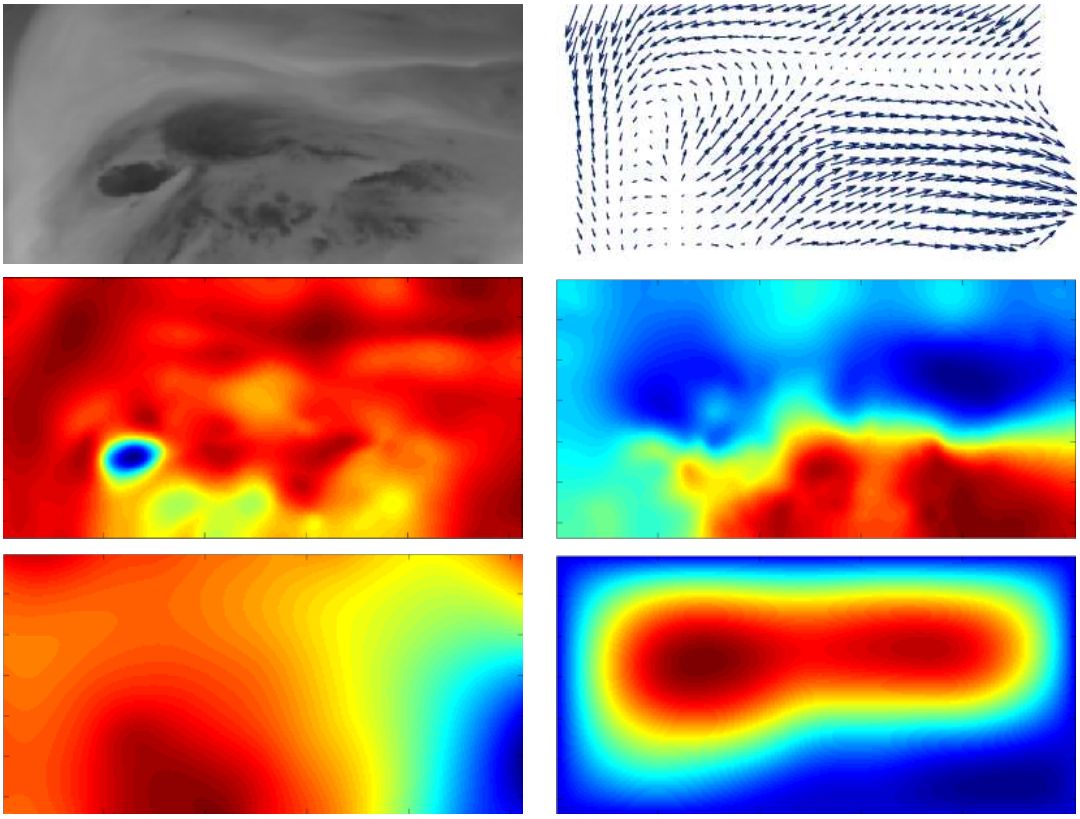


Figure 2.8: **Top:** the first image g_1 with the restored flow field $\mathbf{u}$. **Middle left:** the divergence field of $\mathbf{u}$, i.e. $\triangle_D\psi$. **Middle right:** the curl field of $\mathbf{u}$, i.e. $\triangle_C\phi$. **Bottom left:** the potential field ψ. **Bottom right:** the potential field ϕ. The divergence field reveals a "source"(blue blob) corresponding to a convective cloud cell at high altitude (see text).

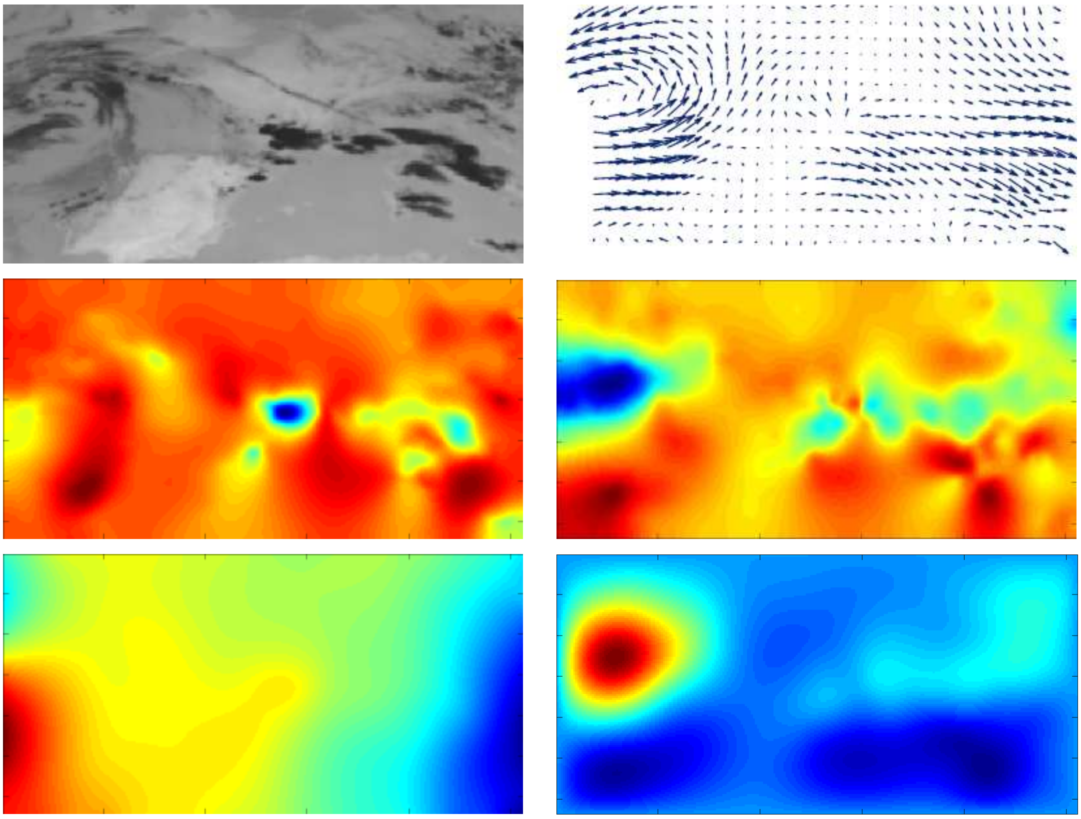


Figure 2.9: **Top:** the first image g_1 with the restored flow field $\mathbf{u}$. **Middle left:** the divergence field of $\mathbf{u}$, i.e. $\triangle_D\psi$. **Middle right:** the curl field of $\mathbf{u}$, i.e. $\triangle_C\phi$. **Bottom left:** the potential field $\psi(\Omega)$. **Bottom right:** the potential field $\phi(\Omega)$. As in the previous figure, the potential functions provide a useful representation of qualitative properties of the flow.

Application to Particle Image Velocimetry (PIV)

Figure 2.10 shows the result of the multi-level potential-based approach (2.84) applied to the PIV image sequence from a flow around two cylinders. The divergence field and curl field inside the area of two cylinders are zero since the apparent motion vanishes there. Note that the two potential fields $\psi(\Omega)$ and $\phi(\Omega)$ are not zero in these domains, but rather the sum of $\nabla\psi$ and $\nabla^{\perp}\phi$ is.

Finally, Figure 2.11 shows the results computed from a PIV image pair of a liquid freezing experiment, recorded by Tomasz A. Kowalewski (http://www.ippt.gov.pl/ tkowale/). Again, both the divergence and the curl field reveal the basic patterns of the underlying non-rigid motion.

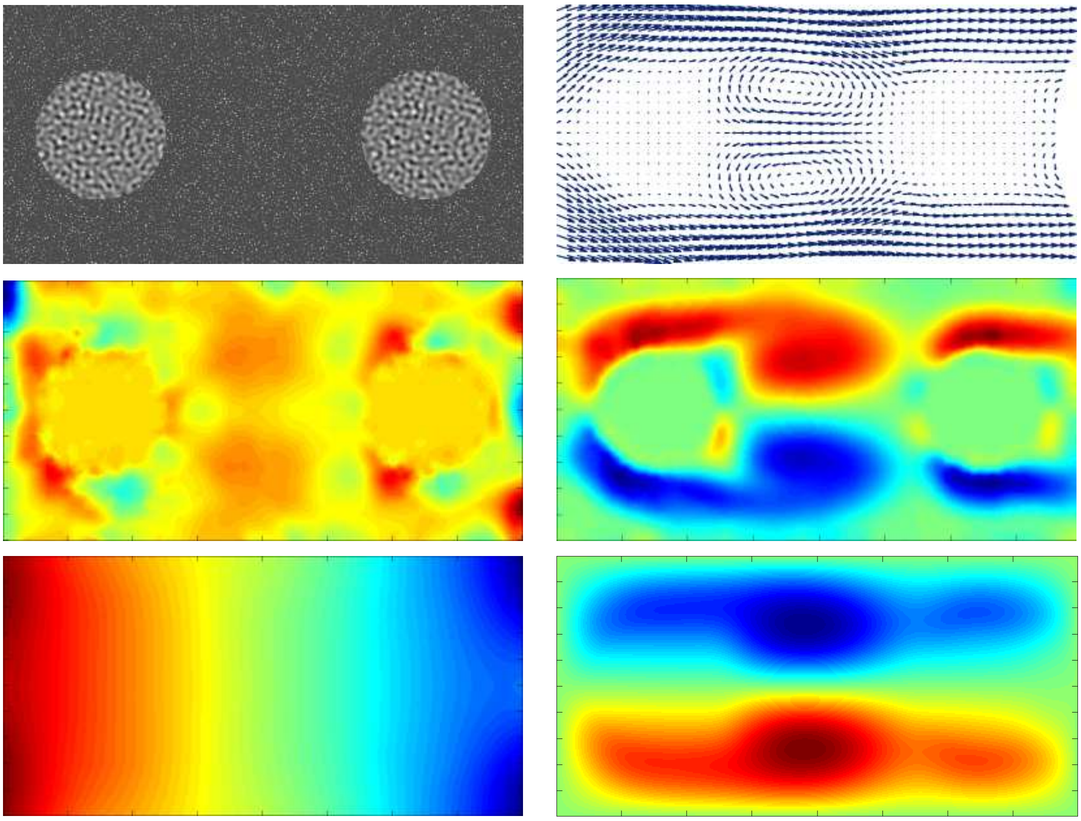


Figure 2.10: **Top:** the first image g_1 with the restored flow field $\mathbf{u}$. **Middle left:** the divergence field of $\mathbf{u}$, i.e. $\triangle_D \psi$. **Middle right:** the curl field of $\mathbf{u}$, i.e. $\triangle_C \phi$. **Bottom left:** the potential field $\psi(\Omega)$. **Bottom right:** the potential field $\phi(\Omega)$. The two potential fields ψ and ϕ are not zero at the area of two cylinders even if the flow they represent disappears in these domains. The divergence and curl fields provide clear flow information around the two cylinders.

Comparisons of Incompressible Flow Estimation

An extensive evaluation and comparison have been made based on the large data set of FLUID poject (http://fluid.irisa.fr/) [76]. Both particle and for scalar synthetic image sequences, generated from direct numerical simulations (DNS) of two-dimensional turbulence, have been used. Comparing to the correlation technique of Lavision and Corpetti et al's second-order method [47], the proposed approach with the mimetic finite-difference implementation yields an enlarged dynamic range with accurate measurements at small and large scales. This behaviour is displayed in Fig. 2.12 showing the better estimated spectrum and the lowest spectrum of the error obtained by the proposed flow decomposition approach. This higher accuracy is also

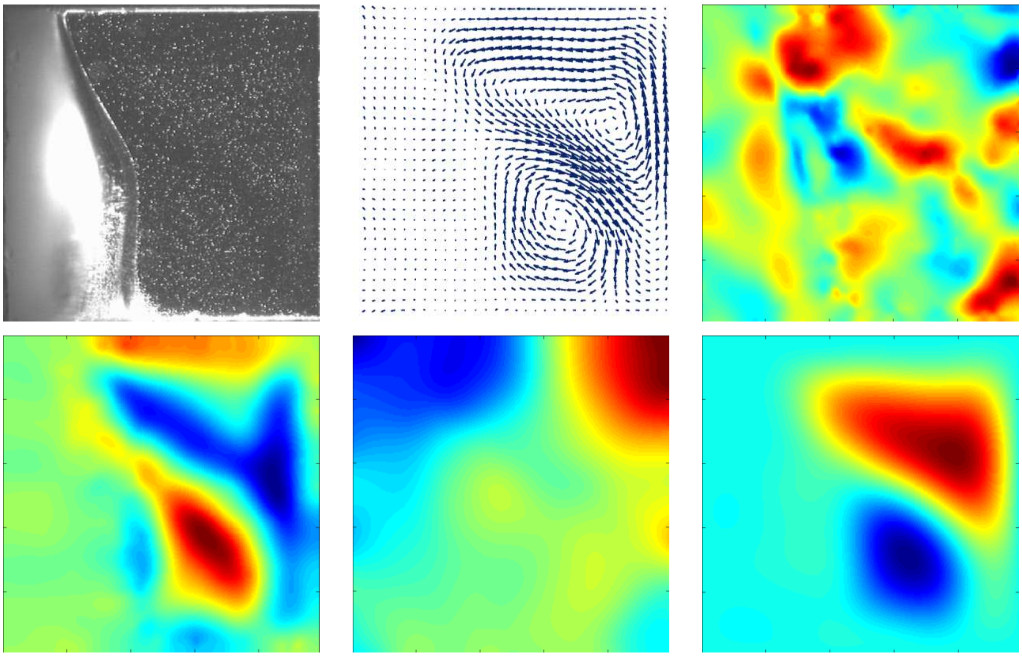


Figure 2.11: **Top left and middle:** the first image g_1 with the restored flow field $\mathbf{u}$. **Top right:** the divergence field of $\mathbf{u}$, i.e. $\triangle_D \psi$. **Bottom left:** the curl field of $\mathbf{u}$, i.e. $\triangle_C \phi$. **Bottom middle:** the potential field $\psi(\Omega)$. **Bottom right:** the potential field $\phi(\Omega)$. The potential functions provide a useful representation of qualitative properties of the flow.

observed in Fig. 2.13 with vorticity maps and vector fields. With scalar image sequences, the differences between this approach and the others is more pronounced, especially at large scales, where as expected the correlation technique completely failed (see Fig. 2.14).

2.3 Nonsmooth Flow Decomposition and Estimation

2.3.1 Introduction

In this section, we investigate a novel class of variational flow decomposition and estimation schemes by combining nonsmooth higher-order flow regularization, which adapt recent techniques developed for nonsmooth image decomposition to processing of non-rigid flows. As a result, we obtain variational approaches that not only allow for estimating fluid flow from image sequences but simultaneously yield a decomposition of the flow into coherent spatio-temporal flow patterns and small-scale structures.

In the following, we briefly describe the respective basic ideas in a continuous setting. In

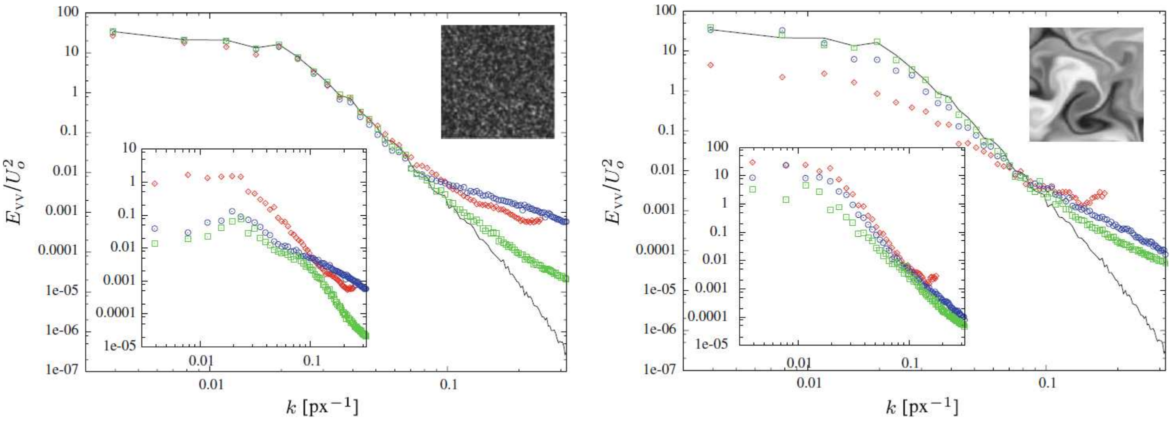


Figure 2.12: Spectrum of the vertical velocity component in a twodimensional turbulent flow. Top synthetic particle image sequence; Bottom synthetic scalar image sequence. Black line DNS reference; Red symbols correlation approach; Blue symbols Corpetti et al's approach [47]; Green symbols the proposed approach. Spectra of the error for the same data are shown in inset. Results in this figure come from [76].

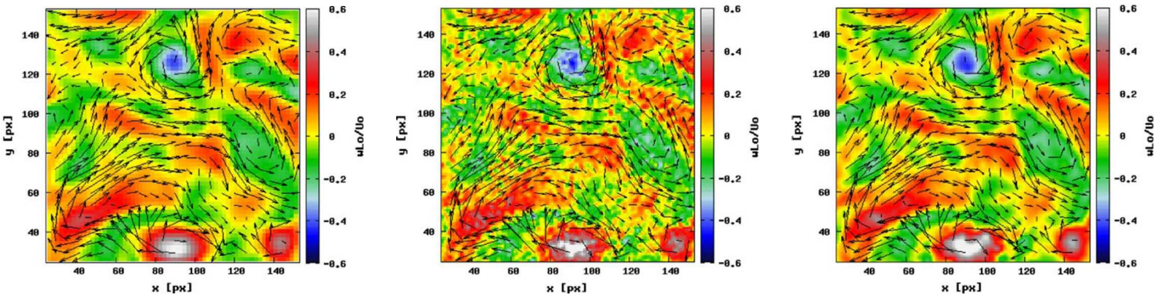


Figure 2.13: Vorticity maps and vector fields in a two-dimensional turbulent flow obtained with a synthetic particle image sequence. From left to bottom, correlation approach, Corpetti et al's approach [47] and the proposed approach. Results in this figure come from [76].

the remainder of this section, we will derive and investigate discrete approaches using the mimetic finite difference method introduced in the previous sections, which preserves the integral identities fulfilled by the corresponding continuous integral operators.

In *image denoising*, one is typically interested in removing noise without destroying important structures such as edges. This goal cannot be achieved with linear filters, e.g., by

$$\min_u \frac{1}{2}\|u-f\|_\Omega^2 + \alpha \int_\Omega |\nabla u|^2 \, \mathrm{d}x \tag{2.87}$$

for a given noisy image $f(x)$ in $\Omega \subset \mathbb{R}^2$. The regularizer incorporates the quadratic function. Via the Euler–Lagrange equation this variational approach can be related to a linear diffusion

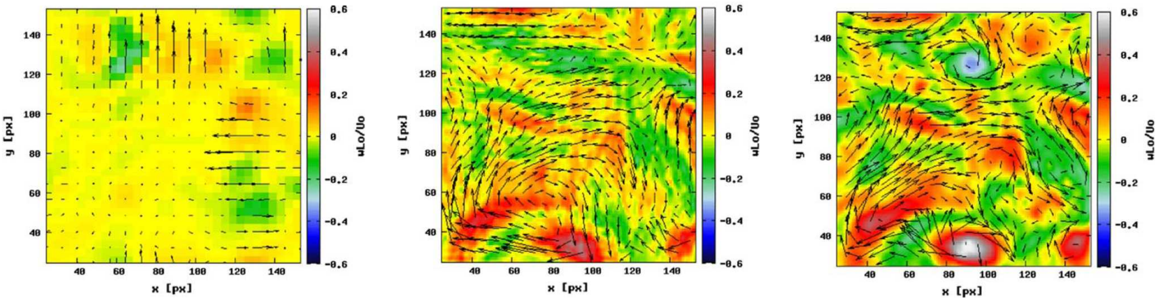


Figure 2.14: Vorticity maps and vector fields in a two-dimensional turbulent flow obtained with a synthetic scalar image sequence. From left to bottom, correlation approach, Corpetti et al's approach [47] and the proposed approach. Results in this figure come from [76].

equation. As a consequence, the optimal solution u^* smooths the given image f in a completely homogeneous way and therefore blurs semantically important signal structures. To overcome this drawback a variety of nonlinear methods have been proposed. One of the frequently applied approaches replaces the function in the regularization term by the total-variation term and thus penalizes larger deviations of $|\nabla u|$ not as hard as the quadratic function:

$$\min_u \frac{1}{2} \|u - f\|_\Omega^2 + \alpha \int_\Omega |\nabla u| \, \mathrm{d}x \,, \tag{2.88}$$

which gives the well-known TV-L2 image approximation problem and was first considered by Rudin, Osher, and Fatemi [141]. In contrast to the *linear approach* (2.87) we will refer to this method as the TV-L2 approach. Based on the dual TV norm, the so-called G norm [115], this denoising model was enlarged for the decomposition of given images f into a structural (cartoon) part u^s and a textural part u^t as

$$\min_{u^s, u^t} \|f - (u^s + u^t)\|_\Omega^2 + \alpha \int_\Omega |\nabla u^s| \, \mathrm{d}x \,, \quad \text{s.t. } \|u^t\|_{\mathrm{G}} \leq \delta.$$

For a more sophisticated treatment of the TV and G norms, we refer to [61, 115]. Meanwhile there exist various numerical realizations of (2.89), e.g. [9, 128, 165]. In Sec. 3.1.3, we also give a comparative study of such image decomposition and TV-L1 based image decomposition.

In this part, we are interested in the decomposition of vector fields rather than scalar images. Specifically, we want to deal with optical flow fields arising, e.g., in experimental fluid dynamics. Inspired by the total-variation regularized image decompisition (2.89), we propose the nonsmooth second-order div-curl regularizar (2.1b) involving penalization terms that are additionally suited to preserve jumps of the divergence and the curl of a flow field, respectively, by utilizing the TV-norm. As in image decomposition we will apply this special term to our variational approach to the decomposition of motion vector fields into physically relevant components at different scales by using a discrete equivalent of the G norm. Moreover, we will study the feasibility of an extension to the *simultaneous* estimation and decomposition of optical flows.

2.3.2 Nonsmooth Flow Decomposition

Now we consider the decomposition of a given non-rigid flow field $u \in H_S$ in a meaningful way. To this end, we have to compute some basic decomposition of u first. We apply Prop. 5 and consider $A\,\mathbf{u} = (\rho, \omega, \nu)^\top$. Let C_ρ and C_ω be the mean values of the discrete divergence and curl fields given by (2.41), which are the discrete versions of $|\Omega|^{-1}\int_\Omega \operatorname{div}\,\mathbf{u}\,dx$ and $|\Omega|^{-1}\int_\Omega \operatorname{curl}\mathbf{u}\,dx$.

In view of (2.42) and the *discrete extended decomposition* of Prop. 6, we can decompose the given flow field, represented by $(\rho, \omega, \nu)^\top \in V_S$, as

$$(\rho, \omega, \nu) \;=\; (C_\rho, C_\omega, \nu) + (\rho^o, \omega^o, 0), \tag{2.89}$$

where $\mathbf{1}^\top_{\dim H_V}\rho^o = \mathbf{1}^\top_{\dim H_P^o}\omega^o = 0$. Obviously, both summands are in V_S again, so that

$$\mathbf{u} \;=\; \mathbf{u}^c + \mathbf{u}_o$$

is the corresponding basic decomposition of $\mathbf{u} \in H_S$, where

$$\mathbf{u}^c = A^\dagger(C_\rho, C_\omega, \nu)^\top, \quad \mathbf{u}^o = A^\dagger(\rho^o, \omega^{,}0)^\top\,.$$

We call vector $\mathbf{u}^c$ (resp., (C_ρ, C_ω, ν)) the basic pattern of the nonrigid flow and its boundary behavior, while $\mathbf{u}^o$ (resp., $(\rho^o, \omega^{,}0)$) is related to the variable (oscillating) flow pattern.

We are interested in further decomposing the intrinsic flow variation $\mathbf{u}^o$ into a structural part $\mathbf{u}^s$ and a texture part $\mathbf{u}^t$, i.e.,

$$\mathbf{u}^o \;=\; \mathbf{u}^s + \mathbf{u}^t\,.$$

This can be done in two ways. The *first approach* uses Prop. 5. Given $\mathbf{u} \in H_S$, we compute $(\rho, \omega, \nu)^\top = A\mathbf{u}$ and then $\rho^o = \rho - C_\rho$ and $\omega^o = \omega - C_\omega$. Next we decompose $\rho^o = \rho^s + \rho^t$ and $\omega^o = \omega^s + \omega^t$ by minimizing separately

$$\begin{aligned}
&\min_{\rho^s,\rho^t} \quad \|\rho^o - \rho^s - \rho^t\|_2^2 \;+\; \alpha_d\|\rho^s\|_{\overline{\mathrm{TV}}} \;+\; \mu_d\|\rho^t\|_{\overline{\mathrm{G}}}\,,\\
&\min_{\rho^s,\rho^t} \quad \|\omega^o - \omega^s - \omega^t\|_2^2 \;+\; \alpha_c\|\omega^s\|_{\mathrm{TV}} \;+\; \mu_c\|\omega^t\|_{\mathrm{G}}
\end{aligned}$$

where the related TV and G norms are defined in the Appendix 7.1. Clearly, this is reduced to two image (scalar) decompositions which can be computed as proposed in [9].

By (7.3), we see that ρ^t equals to $\div\mathbf{p}$ where $\mathbf{p} \in H_S^o$ such that $\mathbf{1}^\mathsf{T} \operatorname{div}^h \mathbf{p} = 0$, we have that $(\rho^t, \omega^t, 0)^\mathsf{T} \in V_S$ and further by (2.89) that $(\rho^s, \omega^s, 0)^\mathsf{T} \in V_S$. Thus we can finally compute u^s and u^t by

$$u^s = A^\dagger(\rho^s, \omega^s, 0)^\mathsf{T}, \quad u^t = A^\dagger(\rho^t, \omega^t, 0)^\mathsf{T}.$$

In this section, we prefer the *second approach* that computes the components of $\mathbf{u}$ directly. This variational approach extends Meyer's model for the decomposition of scalar functions to the *simultaneous* decomposition of vector fields into basic and variable (structure and texture) flow patterns. Moreover, it also fits into our flow estimation-decomposition model in the next part.

For $\mathbf{u} \in H_S$, we propose to find $\mathbf{u}^c \in H_S$ and $\mathbf{u}^s$, $\mathbf{u}^t \in H_S^o$ by minimizing

$$\begin{aligned} \min_{\mathbf{u}^c, \mathbf{u}^s, \mathbf{u}^t} \quad & \alpha_d \left\| \operatorname{div}^h \mathbf{u}^s \right\|_{\overline{\mathrm{TV}}} + \alpha_c \left\| \overline{\operatorname{curl}^h} \mathbf{u}^s \right\|_{\mathrm{TV}} \\ \text{s.t.} \quad & \| \operatorname{div}^h \, \mathbf{u}^t \|_{\overline{\mathrm{G}}} \le \delta_d \,, \quad \| \overline{\operatorname{curl}^h} \, \mathbf{u}^t \|_{\mathrm{G}} \le \delta_c \,, \\ & \mathbf{u}^c + \mathbf{u}^s + \mathbf{u}^t \; = \; \mathbf{u} \,, \\ & \overline{\nabla^h} \operatorname{div}^h \mathbf{u}^c = 0 \,, \quad \nabla^h \overline{\operatorname{curl}^h} \mathbf{u}^c = 0 \,, \\ & \mathbf{1}^\mathsf{T}_{\dim H_P^o} \overline{\operatorname{curl}^h} \mathbf{u}^s = 0 \,. \end{aligned} \tag{2.90}$$

(2.90) minimizes the sum of total-variations of divergence and curl fields of $\mathbf{u}^s$ subject to $\operatorname{div} \mathbf{u}^t$ and $\operatorname{curl} \mathbf{u}^t$with bounded $G-$norms.

Concerning the constraints of (2.90), we note the following: to obtain the desired decomposition we have to ensure that $\operatorname{div}^h \mathbf{u}^c$ and $\overline{\operatorname{curl}^h} \mathbf{u}^c$ are constant and that $\mathbf{1}^\mathsf{T}_{\dim H_V} \operatorname{div}^h (\mathbf{u}^s + \mathbf{u}^t) = 0$, $\mathbf{1}^\mathsf{T}_{\dim H_P^o} \overline{\operatorname{curl}^h} (\mathbf{u}^s + \mathbf{u}^t) = 0$ which makes $\operatorname{div}^h \mathbf{u}^c$ and $\overline{\operatorname{curl}^h} \mathbf{u}^c$ give the mean values of related div and curl fields. The first two conditions are fulfilled by the fourth and fifth constraint. The third condition is fulfilled by the mimetic Gaussian integral identity and since $\mathbf{u}^s$, $\mathbf{u}^t \in H_S^o$. The last condition follows by the last constraint and since $\overline{\operatorname{curl}^h} \mathbf{u}^t \in \mathcal{R}(\overline{\operatorname{div}^h})$ and $\mathbf{1}^\mathsf{T} \overline{\operatorname{div}^h} \mathbf{u}^t = \mathbf{0}$.

2.3.3 Flow Estimation Joint with Flow Decomposition

In contrast to non-rigid flow estimation by the smooth div-curl regularizar $(2.1\mathrm{a})$ which is discussed in sec 2.2, we study the nonsmooth second-order div-curl regularizar $(2.1\mathrm{b})$ based flow estimation which can be formulated by

$$\min_{\mathbf{u}} \ \|g_t - \nabla g \cdot \mathbf{u}\|^2 + \alpha_d \int_\Omega |\nabla \mathrm{div}\, \mathbf{u}|\, dx + \alpha_c \int_\Omega |\nabla \mathrm{curl}\, \mathbf{u}|\, dx + \gamma \int_{\partial\Omega} (\partial_{\mathrm{n}} u)^2 \,\mathrm{d}s, \tag{2.91}$$

By means of mimetic finite difference, the estimation problem (2.91) becomes

$$\min_{\mathbf{u}\in H_S} \ \|g_t - \overline{\nabla^h} g \cdot \mathbf{u}\|_{H_V}^2 + \alpha_d \left\|\mathrm{div}^h\, \mathbf{u}\right\|_{\overline{\mathrm{TV}}} + \alpha_c \left\|\overline{\mathrm{curl}^h}\mathbf{u}\right\|_{\mathrm{TV}} + \gamma \left\|\mathbb{B}_n^h \mathbf{u}\right\|_{\partial H_S}^2 , \tag{2.92}$$

and by applying the orthogonal decomposition,

$$\min_{\psi,\phi} \ \|g_t - \overline{\nabla^h} g \cdot (\overline{\nabla^h}\psi + \nabla^{\perp,h}\phi)\|_{H_V}^2 + \alpha_d \,\|\triangle_D\, \psi\|_{\overline{\mathrm{TV}}} + \alpha_c \,\|\triangle_C\, \phi\|_{\mathrm{TV}} + \gamma \left\|\mathbb{B}_n^h \overline{G}\psi\right\|_{\partial H_S}^2 \tag{2.93}$$

where $\mathbf{1}_{\dim H_{V+\partial V}}^{\mathsf{T}} \psi = 0$. In case of solenoidal flows the functional further reduces as in the linear case.

In the following part of this section, we combine optical flow estimation (2.92) with structure-texture flow decomposition introduced in the previous section: for the given image sequence, we want to compute the component $\mathbf{u}^c \in H_S$ with constant divergence and curl, the large scale pattern $\mathbf{u}^s \in H_S^o$ of divergence and curl with bounded TV norms, and the small scale pattern $\mathbf{u}^t \in H_S^o$ of divergence and curl with bounded G norms.

To this end, we introduce the fitting functional

$$F(\mathbf{u}^c, \mathbf{u}^s, \mathbf{u}^t) = \|g_t - \overline{\nabla^h} g \cdot (\mathbf{u}^c + \mathbf{u}^s + \mathbf{u}^t)\|_2^2 \,. \tag{2.94}$$

Then, with respect to the previous section, one can consider minimizing

$$\begin{aligned} \min_{\mathbf{u}^c,\mathbf{u}^s,\mathbf{u}^t} \quad & F(\mathbf{u}^c, \mathbf{u}^s, \mathbf{u}^t) + \alpha_d \|\,\mathrm{div}^h\, \mathbf{u}^s\|_{\overline{\mathrm{TV}}} + \alpha_c \|\overline{\mathrm{curl}^h}\mathbf{u}^s\|_{\mathrm{TV}} + \gamma \|\mathbb{B}_n^h \mathbf{u}\|_{\partial H_S}^2 \\ \text{s.t.} \quad & \|\,\mathrm{div}^h\ \mathbf{u}^t\|_{\overline{\mathrm{G}}} \le \delta_d \,, \quad \|\overline{\mathrm{curl}^h}\, \mathbf{u}^t\|_{\mathrm{G}} \le \delta_c, \\ & \overline{\nabla^h}\, \mathrm{div}^h\, \mathbf{u}^c = 0 \,, \quad \nabla^h \overline{\mathrm{curl}^h}\mathbf{u}^c = 0 \,, \\ & \mathbf{1}_{\dim H_P^o}^{\mathsf{T}} \overline{\mathrm{curl}^h}\mathbf{u}^s = 0 \,. \end{aligned}$$

Unfortunately, this approach is not well posed. For the image areas where $\nabla g = 0$, the data term disappears, and the local constraints on the G norms will lead to unbounded solutions.

Therefore, we propose to replace the G norm by the L_2 norm to both $\mathrm{div}^h\, \mathbf{u}^t$ and $\overline{\mathrm{curl}^h}\mathbf{u}^t$, which leads to

$$\begin{aligned} \min_{\mathbf{u}^c,\mathbf{u}^s,\mathbf{u}^t} \quad & F(\mathbf{u}^c, \mathbf{u}^s, \mathbf{u}^t) + \alpha_d \|\,\mathrm{div}^h\, \mathbf{u}^s\|_{\overline{\mathrm{TV}}} + \alpha_c \|\overline{\mathrm{curl}^h}\mathbf{u}^s\|_{\mathrm{TV}} \\ & + \mu_d \|\,\mathrm{div}^h\ \mathbf{u}^t\|_{H_V}^2 + \mu_c \|\overline{\mathrm{curl}^h}\ \mathbf{u}^t\|_{H_P^o}^2 + \gamma \|\mathbb{B}_n^h \mathbf{u}\|_{\partial H_S}^2 \\ \text{s.t.} \quad & \overline{\nabla^h}\, \mathrm{div}^h\, \mathbf{u}_{\mathrm{const}} = 0, \quad \nabla^h \overline{\mathrm{curl}^h}\mathbf{u}_{\mathrm{const}} = 0, \\ & \mathbf{1}_{\dim H_P^o}^{\mathsf{T}} (\overline{\mathrm{curl}^h}\mathbf{u}^s + \overline{\mathrm{curl}^h}\mathbf{u}^t) = 0 \,. \end{aligned} \tag{2.95}$$

Our experiments show that this approach works well, although the superiority of the G norm over the L_2 norm in capturing (scalar) oscillating patterns was experimentally shown in [9].

Finally, we are interested in div-free flows $\mathbf{u} \in H_S$. Coupling the extended decomposition (2.43) and its representation $\mathbf{u} \longleftrightarrow (0, \omega, \nu)^{\mathsf{T}} \in V_S$, we have the following decomposition type:

$$(0, \omega, \nu)^{\mathsf{T}} \;=\; (0, C_\omega, \nu)^{\mathsf{T}} + (0, \omega^o, 0)^{\mathsf{T}} \,.$$

Therefore, we have

$$\mathbf{u} \;=\; \mathbf{u}^c + \mathbf{u}^s + \mathbf{u}^t \;=\; (\overline{\nabla^h}\psi + \nabla^{\perp,h}\phi^c) + \nabla^{\perp,h}\phi^s + \nabla^{\perp,h}\phi^t \tag{2.96}$$

where $\omega^o = \omega^s + \omega^t$, ϕ^s and ϕ^t are the potential fields associated with ω^s and ω^t respectively. Indeed, given some scalar field $\tilde{\phi} \in H_P$ which extends $\phi^c \in H_P^o$ with a non-zero boundary, we can properly represent the div-free vector field $\overline{\nabla^h}\psi + \nabla^{\perp,h}\phi^c$ by $\nabla^{\perp,h}\tilde{\phi}$. Then, (2.96) can be rewritten as

$$\mathbf{u} \;=\; \nabla^{\perp,h}(\tilde{\phi} + \phi^s + \phi^t) \,, \qquad \tilde{\phi} \in H_P \,, \;\; \phi^s \,, \; \phi^t \in H_P^o \,.$$

Then the fitting term (2.94) reads

$$F(\tilde{\phi}, \phi^s, \phi^t) \;=\; \|g_t - \overline{\nabla^h} g \cdot \nabla^{\perp,h}(\tilde{\phi} + \phi^s + \phi^t)\|_{H_V}^2 \,,$$

and (2.95) can be reformulated as

$$\begin{aligned} \min_{\tilde{\phi},\phi^s,\phi^t} \quad & F(\tilde{\phi}, \varphi^s, \varphi^t) + \alpha_c \|\triangle_C \phi^s\|_{\mathrm{TV}} + \mu_c \|\triangle_C \phi^t\|_{H_P^o}^2 + \gamma \|\mathbb{B}_n^h \nabla^{\perp,h} \tilde{\phi}\|_{\partial H_S}^2 \\ \text{s.t.} \quad & \nabla^h \,\overline{\mathrm{curl}^h}\, \nabla^{\perp,h}\tilde{\phi} = 0 \,, \quad \mathbf{1}_{\mathrm{dim} H_P^o}^{\mathsf{T}} \triangle_C(\phi^s + \phi^t) = 0 \,, \quad \mathbf{1}_{\mathrm{dim} H_P}^{\mathsf{T}} \tilde{\phi} = 0 \,, \end{aligned} \tag{2.97}$$

where the last constraint appears since $\tilde{\phi}$ is determined only up to an additive constant from the kernel of $\nabla^{\perp,h}$.

2.3.4 Second-Order Cone Programming (SOCP)

Our computational approach to flow estimation via (2.90) and to simultaneous flow estimation and decomposition via (2.97) is based on SOCP [113]. This amounts to minimizing a linear objective function subject to the constraints that several affine functions of the variables have to lie in a *second-order cone* $\mathcal{L}^{n+1} \subset \mathbb{R}^{n+1}$ defined by the convex set

$$\mathcal{L}^{n+1} = \left\{ \begin{pmatrix} x \\ t \end{pmatrix} = (x_1, \ldots, x_n, t)^\top \,:\, \|x\|_2 \le t \right\} .$$

With this notation, the general form of an SOCP is given by

$$\inf_{x\in\mathbb{R}^n} \; f^\top x \,, \quad \text{s.t.} \begin{pmatrix} A_i x + b_i \\ c_i^T x + d_i \end{pmatrix} \in \mathcal{L}^{n+1} \,, \quad i = 1, \dots, r. \tag{2.98}$$

Problem (2.98) is a convex program for which efficient large-scale solvers are available [117]. In connection with TV-based image decomposition the application of SOCPs was suggested in [70].

We rewrite the variational approach (2.90) in the form of SOCP

$$\begin{aligned}
\min_{\mathbf{u}^c,\mathbf{u}^s,\mathbf{u}^t} \quad & \lambda_d \mathbf{1}^\mathsf{T}_{\mathrm{dim} H_V}\, v + \lambda_c \mathbf{1}^\mathsf{T}_{\mathrm{dim} H_P^o}\, w, \\
\text{s.t.} \quad & u\mathbf{u}^c + \mathbf{u}^s + \mathbf{u}^t = \mathbf{u} \,, \quad \overline{\nabla^h} \, \mathrm{div}^h \, \mathbf{u}^c = 0 \,, \;\; \nabla^h \overline{\mathrm{curl}^h} \mathbf{u}^c = 0 \,, \\
& \mathbf{1}^\mathsf{T}_{\mathrm{dim} H_P^o} \overline{\mathrm{curl}^h} \mathbf{u}^s = 0 \,, \quad \mathrm{div}^h \, \mathbf{u}^t = \mathrm{div}^h \, \mathbf{p}_d \,, \qquad \overline{\mathrm{curl}^h} \mathbf{u}^t = \mathrm{div}^h \, \mathbf{p}_c, \\
& \begin{pmatrix} (\overline{\nabla^h} \, \mathrm{div}^h \, \mathbf{u}^s)_{\Omega_{\alpha,\beta}} \\ v_{\Omega_{\alpha,\beta}} \end{pmatrix} \in \mathcal{L}^5 \,, \qquad \begin{pmatrix} (\overline{\nabla^h} \overline{\mathrm{curl}^h} \mathbf{u}^s)_{P_{i,j}} \\ w_{P_{i,j}} \end{pmatrix} \in \mathcal{L}^5, \\
& \begin{pmatrix} (\mathbf{p}_d)_{\Omega_{\alpha,\beta}} \\ \delta_d \end{pmatrix} \in \mathcal{L}^5 \,, \qquad \begin{pmatrix} (\mathbf{p}_c)_{P_{i,j}} \\ \delta_c \end{pmatrix} \in \mathcal{L}^5
\end{aligned}$$

where $v \in H_V$ and $w \in H_P^o$ are two scalar fields, $\Omega_{\alpha,\beta}$ gives the cell of image grid and the last two types of cone definitions follows from (2.3).

In order to incorporate the quadratic terms of the variational approaches to optical flow estimation, we use the following rotated version of the standard cone:

$$\mathcal{R}^{n+2} := \left\{ \left(\boldsymbol{x}, x_{n+1}, x_{n+2}\right)^\top \in \mathbb{R}^{n+2} : \frac{1}{2} \, \|\boldsymbol{x}\|_2^2 \le x_{n+1} x_{n+2}, \; x_{n+1}, x_{n+2} \ge 0 \right\} .$$

Fixing $x_{n+2} = 1/2$, we have $\|\boldsymbol{x}\|_2^2 \le x_{n+1}$. Now we can rewrite (2.97) as follows:

$$\begin{aligned}
\min_{\phi^c,\phi^s,\phi^t} \quad & v + \mu_c \, t + \lambda_c \mathbf{1}^\mathsf{T}_{\mathrm{dim} H_P^o} w, \\
\text{s.t.} \quad & \nabla^h \overline{\mathrm{curl}^h} \nabla^{\perp,h} \phi^c = 0 \,, \quad \mathbf{1}^\mathsf{T}_{\mathrm{dim} H_P} \phi^c = 0 \,, \quad \mathbf{1}^\mathsf{T}_{\mathrm{dim} H_P^o} \triangle_C (\phi^s + \phi^t) = 0 \,, \\
& \begin{pmatrix} (\overline{\nabla^h} \triangle_C \phi^s)_{\Omega_{\alpha,\beta}} \\ w_{\Omega_{\alpha,\beta}} \end{pmatrix} \in \mathcal{L}^5 \,, \quad \begin{pmatrix} \triangle_C \phi^t \\ t \\ 1/2 \end{pmatrix} \in \mathcal{R}^{\mathrm{dim} H_P + 2} \,, \\
& \begin{pmatrix} \overline{\nabla^h} g \cdot \nabla^{\perp,h} (\phi^c + \phi^s + \phi^t) + g_t \\ v \\ 1/2 \end{pmatrix} \in \mathcal{R}^{\mathrm{dim} H_V + 2} \,.
\end{aligned}$$

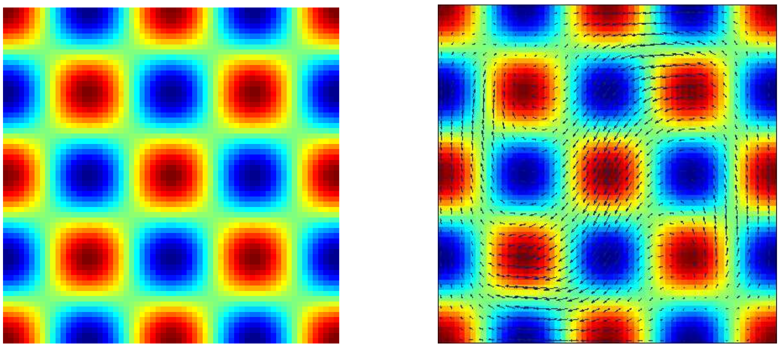


Figure 2.15: Frame g (left) and its warping with the groundtruth flow (right) for flow estimation by the linear approach and the TV approach.

2.3.5 Numerical Experiments

In this section we verify our approaches by numerical examples. The programs were written in MATLAB and used the software package SeDuMi for SOCP (see http://sedumi.mcmaster.ca/ for more information). Unfortunately, we do not have an automatized choice of parameter values. Parameter values were chosen by hand following two general rules: (i) choose the weights as small as possible in order to not smooth out turbulent motion, (ii) for TV terms, smaller values than in the linear case (e.g., (21)) are appropriate, because these regularizers return larger values than their linear counterparts (assuming that image data are scaled to the range $[0, 1]$). Rule (ii) leads to parameter values of the order $\propto 10^{-4}$. Thanks to the mimetic discretization, this suffices for numerical stability.

Flow estimation

We start by comparing flow estimations obtained by the linear approach (2.61) and the TV approach (2.93). We consider the artificial example in Figure 2.15 with the groundtruth on top of Figure 2.16. Figure 2.16 (middle) was obtained by solving the linear systems of equations resulting from (2.61) with parameters $\lambda_d = 0.06$, $\lambda_c = 0.048$. As the boundary parameter we have used $\gamma = 0.04$. The result shows the typical blurring effects at the edges. If we decrease the parameter values, the rectangular shape of div and curl becomes easier to see, but the artifacts always visible in Figure 2.16 (middle) increase too. The bottom of Figure 2.16 contains the solution of (2.92) with the parameters $\lambda_d = 10^{-6}$, $\lambda_c = 10^{-6}$, and $\gamma = 10^{-5}$ by SOCP and 17 iterations. As expected for this example, the TV approach gives very good results by preserving discontinuities of the flow derivatives (div, curl).

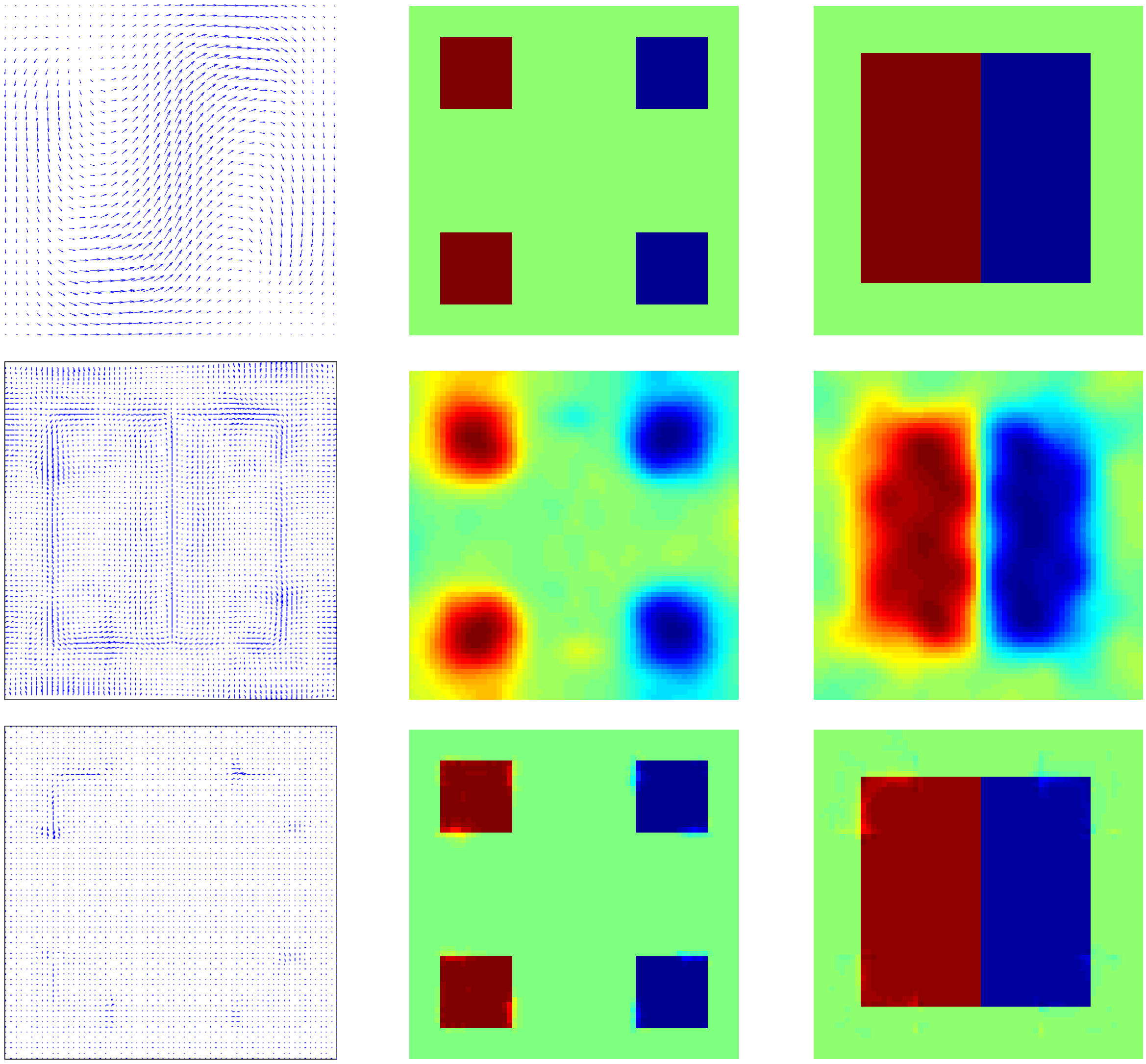


Figure 2.16: Top: The groundtruth flow u, its div and curl (left to right). Middle: Linear reconstruction by (2.61), difference between $\mathbf{u}$ and its estimation, reconstructed div and curl (left to right). Bottom: TV reconstruction by (2.92), difference between $\mathbf{u}$ and its estimation, reconstructed div and curl (left to right).

Flow decomposition

Figure 2.17 shows a turbulent flow field u as groundtruth along with its divergence ρ and curl ω. Applying the variational method (2.90) with $\lambda_d = 0.2\,,\ \lambda_c = 0.5$, and $\delta_d = 0.05\,,\ \delta_c = 0.1$, we obtain the decompositions depicted in Figures 2.18 and 2.19. The structural and textural components recovered the interesting motion patterns at different scales, which are not easily visible in the flow $\mathbf{u}$ itself.

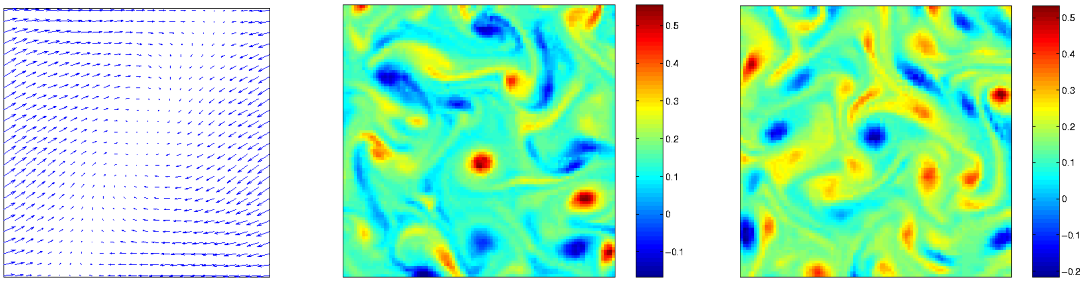


Figure 2.17: Groundtruth data to be decomposed: flow field $\mathbf{u}$ (left), its divergence field ρ (middle), and its curl field ω (right).

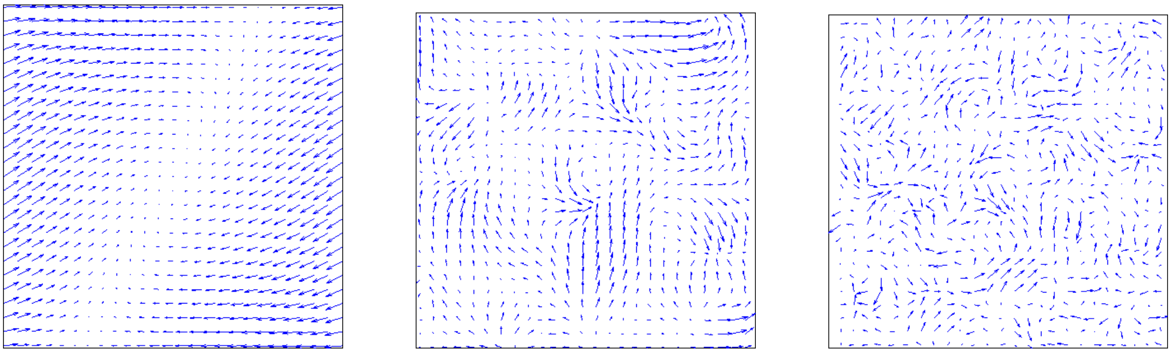


Figure 2.18: The components of the flow $\mathbf{u}$ from Figure 2.17: $\mathbf{u}^c$ (left), $\mathbf{u}^s$ (middle), and $\mathbf{u}^t$ (right). The vectors of $\mathbf{u}^s, \mathbf{u}^t$ are scaled up for better visibility. Note that despite $|\mathbf{u}| \approx |\mathbf{u}^c|$, the structural and texture parts $\mathbf{u}^s$ and $\mathbf{u}^t$ are recovered well.

Flow estimation and decomposition

In this section we will validate the flow estimation-decomposition model (2.97). First we create a divergence-free groundtruth flow field $\mathbf{u}$ by superimposing a dominant laminar flow with some turbulent vortex structures; see Figure 2.20. Using this flow, an artificial image sequence $\{g_1\,,\,g_2\}$ was created.

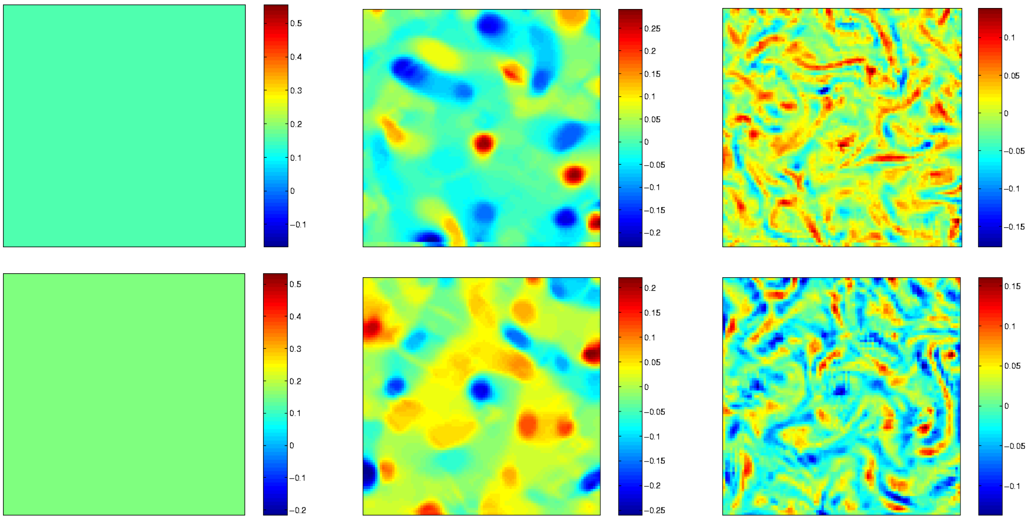


Figure 2.19: Decomposition of $\mathbf{u}$ from Figure 2.17 with the approach (2.90). Top: ρ^c, ρ^s, ρ^t (left to right). Bottom: $\omega^c, \omega^s, \omega^t$ (left to right). The structure and texture components reveal turbulent flow patterns at different scales, which are not easily visible in the flow $\mathbf{u}$ itself.

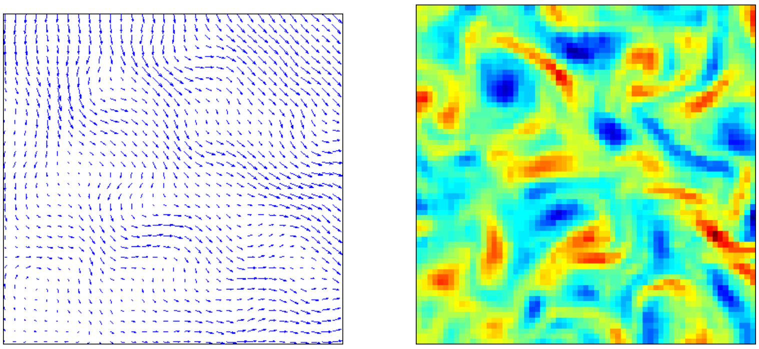


Figure 2.20: Groundtruth data $\mathbf{u}$ (left) and its curl (right) to be estimated from a corresponding artificially created image sequence. Here $\mathbf{u}$ is a superposition of a laminar flow and turbulent vortices.

Figures 2.21 and 2.22 show the decomposition-based optical flow estimates, where we have used $\lambda_c = 6 \times 10^{-5}$ and $\mu_c = 3 \times 10^{-4}$. The boundary parameter was chosen slightly smaller than λ_d. The $\mathbf{u}^c$ component nicely recovers the laminar flow, whereas the structural and textural components reveal the turbulent curl field. Finally, Figure 2.22 gives a close-up view of a section of Figure 2.21.

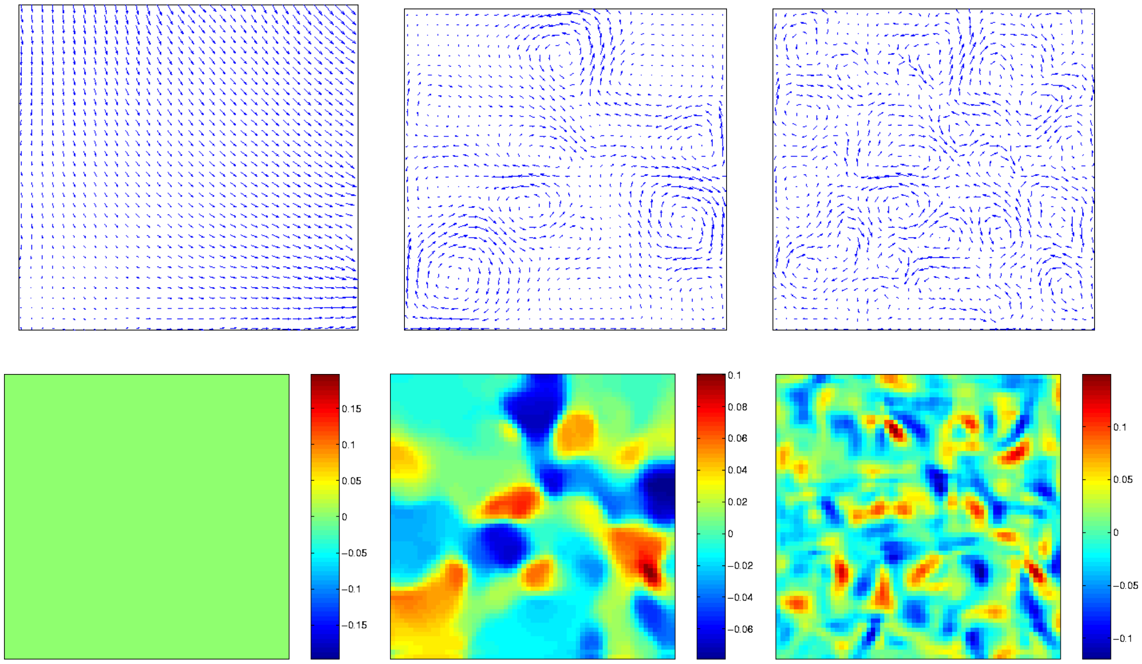


Figure 2.21: Estimated and decomposed flow corresponding to Figure 2.20 using the $TV - L_2$ approach (2.97). Top: $\mathbf{u}^c$, $\mathbf{u}^s$, and $\mathbf{u}^t$ (left to right). Bottom: ω^c, ω^s, and ω^t (left to right).

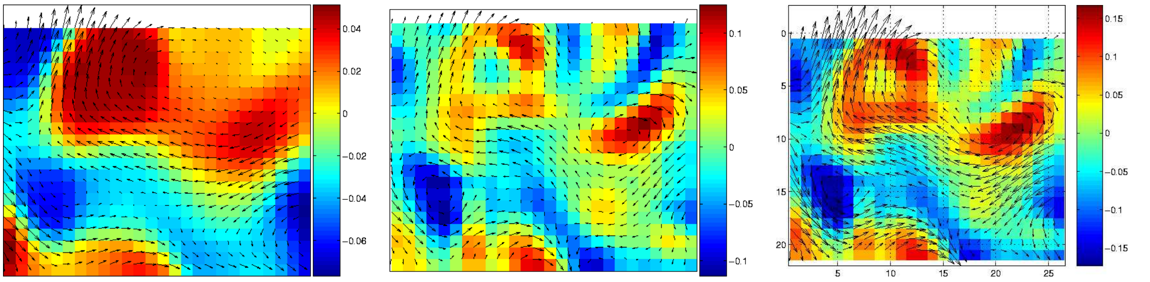


Figure 2.22: Close-up view of a section of Figure 2.21. From left to right: $\omega^s, \omega^t, \omega^s + \omega^t$ with the corresponding flows as overlays.

A real-world example

Figure 2.23 (top-left) shows a sample image of the experimental evaluation of the spreading of a low-diffusivity dye in a two-dimensional turbulent flow, forced at a large scale. The passive scalar is a mixture of fluorescein and water. The divergence of the corresponding flow vanishes. For more details about the experimental setup, we refer to [89].

Figure 2.23 shows the components of the flow and curl field estimated in terms of ϕ^c, ϕ^s, ϕ^t by minimizing (2.97) (parameter values: $\lambda_c = 10^{-4}$, $\mu_c = 3 \times 10^{-4}$, $\gamma = 5 \times 10^{-4}$). This

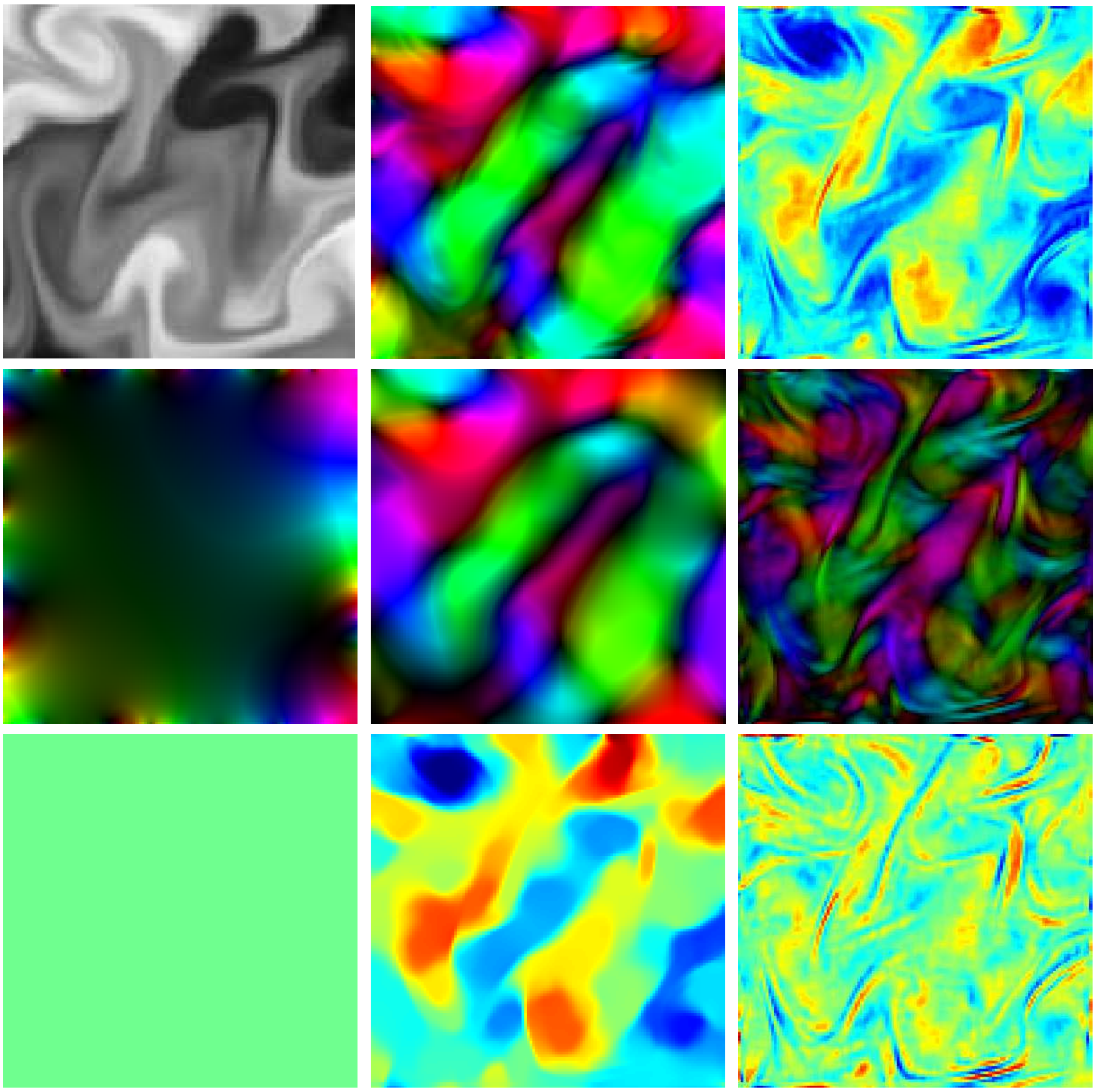


Figure 2.23: Top-left: frame of a real image sequence depicting the mixture of fluorescein and water [89]. Top-center: the turbulent solenoidal flow estimated by minimizing (2.97). Flow vectors are color-coded for better visibility (color $\simeq$ direction, magnitude $\simeq$ brightness). Top-right: the curl field comprising large-scale patterns immersed in turbulent oscillations. Middle, from left to right: $\mathbf{u}^c, \mathbf{u}^s, \mathbf{u}^t$. Note that $\mathbf{u}^c$ contains the (nonvanishing) boundary values, and that $\mathbf{u}^s$ and $\mathbf{u}^t$ separate coherent motion patterns and turbulent fluctuations, respectively. Bottom, from left to right: the curl fields ω^c, ω^s, and ω^t.

result clearly demonstrates how the convex constrained optimization approach allows for estimating complex flows while simultaneously separating large-scale coherent motion patterns from turbulent fluctuations.

3 TV-L1 Image Approximation

In this chapter, we study the TV-L1 image approximation approach (1.7) based on modern convex optimization theories. We propose new equivalent convex models of TV-L1 image approximation and show that TV-L1 image approximation introduces a new image decomposition model based on convex cones which is in comparison to the TV-L2 reduced image decomposition. We also prove that the convex TV-L1 image approximation model actually gives an exact and global optimum to the corresponding discrete-constrained TV-L1 image approximation problem (1.11), i.e. (1.7) works as the exact convex relaxation model to (1.11). Finally, we propose the new multiplier-based TV-L1 algorithm based on its equivalent dual model, which presents an efficient and reliable numerical scheme to (1.7) and can be easily implemented.

We list the formulations of TV-L1 image approximation (1.7) and its discrete-constrained version (1.11) in this chapter again to ease reading:

$$\min_u \, \big\{ P(u) := \int_\Omega |f - u| \, dx + \alpha \int_\Omega |\nabla u(x)| \, dx \big\} \,, \tag{3.1}$$

and

$$\min_{u(x) \in \{f_1, \ldots, f_n\}} \int_\Omega |f - u| \, dx + \alpha \int_\Omega |\nabla u(x)| \, dx \,, \tag{3.2}$$

given $f(x) \in \{f_1, \ldots, f_n\}$ and $f_1 \ldots f_n$ are linearly ordered such that $f_1 < \ldots < f_n$.

3.1 Equivalent Formulations

We call TV-L1 image approximation (3.1) the *primal model* in this chapter, as comparison to other convex models introduced in this part.

3.1.1 Equivalent Primal-Dual Model

With the help of conjugates [137], the data term of (3.1) can be equally expressed by

$$\alpha \int_\Omega |f - u| \, = \, \max_{q \in S} \, \langle q, f - u \rangle \ , \quad S \, := \, \{ q \, | \, |q(x)| \leq 1 \, , \ \forall x \in \Omega \, \} \, . \tag{3.3}$$

Moreover, it is well known that the total-variation term of (3.1) can also be reformulated [68] as follows

$$\alpha \int_\Omega |\nabla u| \; dx \, = \, \max_{p \in C_\alpha} \, \langle \operatorname{div} p, u \rangle \ , \quad C_\alpha \, := \, \{ p \, | \, p \in C_c^1(\Omega, \mathbb{R}^2) \, , \ |p(x)| \leq \alpha \, , \ \forall x \in \Omega \, \} \, . \tag{3.4}$$

In view of (3.3) and (3.4), after some rearrangements, the TV-L1 approximation formulation (1.7) can be equally rewritten as

$$\max_{q \in S} \max_{p \in C_\alpha} \min_u \ \big\{ E(u; q, p) \, := \, \langle q, f \rangle + \langle \operatorname{div} p - q, u \rangle \, \big\} \, , \tag{3.5}$$

which is called the equivalent *primal-dual model* to the primal model (3.1).

3.1.2 Equivalent Dual Model

Observe that u is unconstrained, minimizing (3.5) over u, therefore, leads to the linear equality

$$\operatorname{div} p \, = \, q \, , \tag{3.6}$$

and the constrained maximization problem

$$\max_{q \in S} \max_{p \in C_\alpha} \ \big\{ D(q, p) \, := \, \langle q, f \rangle \, \big\} \, , \quad \text{s.t.} \quad \operatorname{div} p = q \, . \tag{3.7}$$

Likewise, we call (3.7) the equivalent *dual model* to (3.1).

Let the C_{in} be the intersection set of S and $\operatorname{div} C_\alpha$, denoted by $C_{in} \, := \, S \cap \operatorname{div} C_\alpha$. The dual formulation (3.7) actually can be equally written by

$$\max_{q \in C_{in}} \ \langle q, f \rangle \ . \tag{3.8}$$

The above formulation (3.8) gives a simpler equivalent expression for TV-L1 image approximation and illustrates the function of the dual variable q. However, our aim is to compute the optimal image function $u(x)$ which works as the multiplier function of the linear equality constraint (3.6) of the its dual model (3.7). To this end, we stick to the dual formulation (3.7).

3.1.3 Optimization and Variational Facts

Minimax Theorem

For the primal-dual formulation (3.5), the conditions of the minimax theorem (see e.g., [58, 62]) are all satisfied. That is: the constraints of dual variables p and q are convex and the energy function is linear to both u and (p, q), hence convex l.s.c. for fixed u and concave u.s.c. for fixed p and q. This follows that there exists at least one saddle point, s ee [58, 62]. As a consequence, the min and max operators of the primal-dual model (3.5) can be interchanged, i.e.

$$\max_{q \in S} \max_{p \in C_\alpha} \big\{ \min_u \, E(u; q, p) \big\} \, = \, \min_u \big\{ \max_{q \in S_\alpha} \max_{p \in C_1} \, E(u; q, p) \big\} \, . \tag{3.9}$$

It is easy to see that the optimization of the primal-dual model (3.5) over the dual variables q and p react on the primal formulation (3.1) of TV-L1 image approximation, i.e. the right hand side of (3.9):

$$P(u) \, = \, E(u; q^*, p^*) \, = \, \max_{q \in S_\alpha} \max_{p \in C_1} \, E(u; q, p) \, .$$

Likewise, the dual model (3.7) can be achieved by optimizing the image function $u(x)$ in (3.5), i.e. the left hand side of (3.9):

$$D(q, p) \, = \, E(u^*, q, p) \, = \, \min_u \, E(u; q, p) \, . \tag{3.10}$$

TV-L1 Image Decomposition and Geometrical Explanation

By the definitions of conjugate (3.3) and (3.4), it easy to see that given the optimal saddle point (q^*, p^*, u^*) to the primal-dual optimization formulation (3.5), we clearly have

$$f - u^* \, \in \, \partial S(q^*) \, , \quad u^* \in \partial(\operatorname{div} C_\alpha)(p^*) \tag{3.11}$$

where S and $\operatorname{div} C_\alpha$ denote the indicator functions to their associated convex sets given by (3.3) and (3.4). The subgradients ∂S and $\partial(\operatorname{div} C_\alpha)$ in (3.11) give rise to the normal cones [79], denoted by $N_S(q^*)$ and $N_{\operatorname{div} C_\alpha}(p^*)$ respectively, to their corresponding convex set at the given point.

In view of (3.11), it is easy to see

Proposition 12. *Given the optimal saddle point (q^*, p^*, u^*) to the primal-dual optimization formulation (3.5), $q^*, p^*, u^*)$ gives rise to the decomposition of the input image $f(x)$ such that*

$$f \, = \, u^* + v^* \, , \quad \textit{where } v^* \, \in \, N_S(q^*) \, , \; u^* \in N_{\operatorname{div} C_\alpha}(p^*) \, . \tag{3.12}$$

Proof directly follows from the fact $v^* = f - u^*$.

Now we consider the geometrical explanation of the proposed TV-L1 image decomposition by Prop. 12: observe the maximization problem (3.8) which is equivalent to the dual model, we simply have

$$f \in N_{C_{in}}(q^*) \tag{3.13}$$

where $N_{C_{in}}$ is the normal cone of the convex set C_{in} at q^*. Then by (3.12), we have the geometrical relationship of three components in TV-L1 image decomposition (3.12) as

$$(f \in N_{S \cap \operatorname{div} C_\alpha}(q^*)) = u^* + v^* \,, \quad \text{where } v^* \in N_S(q^*) \,, \; u^* \in N_{\operatorname{div} C_\alpha}(p^*) \,. \tag{3.14}$$

Hence, the studied TV-L1 image approximation boilds down to the image decomposition (3.14) based on the three normal cones $N_{S \cap \operatorname{div} C_\alpha}$, N_S and $N_{\operatorname{div} C_\alpha}$ at some specified point q^* and p^*. (3.14) is so called the *normal cone based image decomposition* of the image $f(x)$.

This is in contrast to the classical TV-L2 image approximation (1.6), which gives the following image decomposition [9, 115]:

$$f = u^* + \operatorname{div} p^*$$

where the dual variable p^* gives the projection of the input image $f(x)$ to the convex set $\operatorname{div} C_\alpha$, i.e.

$$p^* := \arg \min_{p \in C_\alpha} \|f - \operatorname{div} p\|^2 \,.$$

This defines a *projection based image decomposition* of the image $f(x)$.

3.2 Global and Exact Optimums of Discrete-Constrained TV-L1 Model

In this section, we study the nonconvex optimization problem (3.2) and show that the TV-L1 formulation (3.1), which gives an exact convex relaxed model of (3.2), i.e. solves the nonconvex minimization problem (3.2) globally and exactly through the proposed rounding scheme. We state our results and proof by several propositions.

Proposition 13 (Extremum Principle). *Given the image function $f(x) \in \{f_1, \ldots, f_n\}$, $\forall x \in \Omega$, along with the order $f_1 < \ldots < f_n$, each optimum $u^*(x)$ of (3.1) suffices $f_1 \leq u^*(x) \leq f_n$.*

The same results which state any optimum $u^*(x)$ should suffice $u^*(x) \in [f_{\min}, f_{\max}]$, i.e. $u^*(x) \in [f_1, f_n]$ considering the ascent ordering $f_1 < \ldots < f_n$ in this work, can also be found in other works, e.g. [41] where $f(x) \in \{0, 1\}$ or [56] where $f(x) \in [f_{\min}, f_{\max}]$. We also provide the proof as follows to ease reading.

Proof. Let u^* be the minimum of (1.7). Due to the convexity of (1.7), u^* is simply accepted as the global minimum. We first prove that $u^*(x) \leq f_n$ for $\forall x \in \Omega$.

If $u^*(x) > f_n$ at some area $\tilde{\Omega} \subset \Omega$, then we define the function u' which just threshholds the value $u^*(x)$ to be not larger than f_n, i.e.

$$u'(x) \,=\, \begin{cases} f_n & \text{at } x \in \tilde{\Omega} \\ u^*(x) & \text{at } x \in \Omega \backslash \tilde{\Omega} \end{cases} .$$

Obviously, in view of $f(x) \leq f_n$ and $u^*(x) > f_n$ for $\forall x \in \tilde{\Omega}$, we have

$$\begin{aligned} \int_\Omega |u^* - f| \, dx \,&=\, \int_{\Omega\backslash\tilde{\Omega}} |u^* - f| \, dx \,+\, \Big\{ \int_{\tilde{\Omega}} |f_n - f| \, dx \,+\, \int_{\tilde{\Omega}} |u^* - f_n| \, dx \Big\} \\ &=\, \int_\Omega \big|u' - f\big| \, dx \,+\, \int_{\tilde{\Omega}} |f_n - f| \, dx \,. \end{aligned}$$

It follows that

$$\int_\Omega \big|f - u'\big| \, dx \,<\, \int_\Omega |f - u^*| \, dx \,. \tag{3.15}$$

By the coarea formula of the total variation term:

$$\mathrm{TV}(u) \,=\, \int_{-\infty}^{+\infty} L_\gamma(u) \, d\gamma \,,$$

where $L_\gamma(u)$ is the length of the $\gamma-$upper level set of u, it follows that

$$\mathrm{TV}(u') \,<\, \mathrm{TV}(u^*) \,, \tag{3.16}$$

because the f_n-upper level set of u' is threshholded to vanish.

Observe (3.15) and (3.16), we must have

$$\int_\Omega \big|f - u'\big| \, dx \,+\, \alpha \mathrm{TV}(u') \,<\, \int_\Omega |f - u^*| \, dx \,+\, \alpha \mathrm{TV}(u^*) \,.$$

This is in contradiction to the fact that u^* is the global minimum of (1.7).

Likewise, we can also prove $u^*(x) \geq f_1$ $x \in \Omega$ in the same way. In consequence, we prove that each minimum $u^*(x)$ of (1.7) suffices $u^*(x) \in [f_1, f_n]$. □

Proposition 14. *Given a bounded scalar function $f_1 \leq u(x) \leq f_n \ \forall x \in \Omega$, if p^* maximizes the integral $\int_\Omega u \operatorname{div} p \, dx$ over the convex set C_α, i.e.*

$$\int_\Omega |\nabla u| \, dx \ = \ \int_\Omega u \operatorname{div} p^* \, dx \, ,$$

then in view of (1.10)*, for every $\gamma-$upper level set $U^\gamma(x)$ of $u(x)$ with $\gamma \in [f_1, f_n)$, p^* also maximizes the integral $\int_\Omega U^\gamma \operatorname{div} p \, dx$ over the convex set C_α and*

$$\int_\Omega U^\gamma \operatorname{div} p^* \, dx \ = \ \alpha \, |\partial U^\gamma| \ ,$$

which is the perimeter of the level set $U^\gamma(x)$.

Proof. Denote the interval $\Gamma = [f_1, f_n]$. The coarea formula gives

$$\int_\Omega |\nabla u| \, dx = \int_\Gamma \int_\Omega |\nabla U^\gamma| \, dx \, d\gamma. \tag{3.17}$$

By applying this formula we can deduce

$$\int_\Omega u \ \operatorname{div} p^* \, dx = \int_\Omega |\nabla u| \, dx = \int_\Gamma \int_\Omega |\nabla U^\gamma| \, dx \, d\gamma = \int_\Gamma \left(\max_{p \in C_\alpha} \int_\Omega U^\gamma \operatorname{div} p \, dx \right) d\gamma. \tag{3.18}$$

By the fact that $u(x) = \int_{f_1}^{u(x)} \, d\gamma = \int_\Gamma U^\gamma(x) d\gamma$ for any $x \in \Omega$, we have

$$\int_\Omega u \ \operatorname{div} p^* \, dx = \int_\Omega \left(\int_\Gamma U^\gamma(x) d\gamma \right) \operatorname{div} p^*(x) \, dx = \int_\Gamma \int_\Omega U^\gamma \operatorname{div} p^* \, dx \, d\gamma. \tag{3.19}$$

Therefore, combining (3.18) and (3.19):

$$\int_\Gamma \int_\Omega U^\gamma \operatorname{div} p^* \, dx d\gamma \ = \ \int_\Gamma \left(\max_{p \in C_\alpha} \int_\Omega U^\gamma \operatorname{div} p \, dx \right) d\gamma. \tag{3.20}$$

This equality (3.20) together with the fact that for any $\gamma \in [f_1, f_n)$

$$\int_\Omega U^\gamma \operatorname{div} p^* \, dx \leq \max_{p \in C_\alpha} \int_\Omega U^\gamma \operatorname{div} p \, dx \, . \tag{3.21}$$

Then it follows that

$$\int_\Omega U^\gamma \operatorname{div} p^* \, dx = \max_{p \in C_\alpha} \int_\Omega U^\gamma \operatorname{div} p \, dx$$

for almost every $\gamma \in [f_1, f_n)$. Clearly, the perimeter of the level set U^γ is given by

$$\alpha \, |\partial U^\gamma| = \int_\Omega |\nabla U^\gamma| \, dx = \max_{p \in C_\alpha} \int_\Omega U^\gamma \operatorname{div} p \, dx \, .$$

□

Corollary 15. *Given a bounded scalar function $f_1 \leq u(x) \leq f_n$ $\forall x \in \Omega$ and $n-1$ different values γ_i, $i = 1, \dots, n-1$, such that $f_1 \leq \gamma_1 < \dots < \gamma_{n-1} \leq f_n$, if p^* maximizes the integral $\int_\Omega u \operatorname{div} p \, dx$ over the convex set C_α, then for the image function*

$$u^\gamma(x) \,=\, \sum_{i=1}^{n-1} (f_{i+1} - f_i) \, U^{\gamma_i}(x) \,,$$

p^* also maximizes the integral $\int_\Omega u^\gamma \operatorname{div} p \, dx$ over the convex set C_α, i.e. we have

$$\alpha \int_\Omega |\nabla u^\gamma| \, dx \,=\, \int_\Omega u^\gamma \operatorname{div} p^* \, dx \,.$$

Proof. By virtue of Prop. 14, p^* also maximize the integral

$$\int_\Omega U^{\gamma_i} \operatorname{div} p \, dx,$$

over the convex set C_α for each γ_i, $i = 1, \dots, n-1$.

Then it follows that for the piecewise constant image function

$$u^\gamma(x) \,=\, \sum_{i=1}^{n-1} (f_{i+1} - f_i) \, U^{\gamma_i}(x) \,,$$

p^* also maximizes the integral

$$\int_\Omega u^\gamma \operatorname{div} p^+ \, dx \,=\, \sum_{i=1}^{n-1} \big\{ (f_{i+1} - f_i) \int_\Omega U^{\gamma_i} \operatorname{div} p^* \, dx \big\} \,,$$

over the convex set $p \in C_\alpha$, because $f_1 < \dots < f_n$ is ordered such that

$$f_{i+1} - f_i \,>\, 0 \,, \quad i = 1, \dots, n-1 \,.$$

Therefore, by Prop. 14, we have

$$\alpha \int_\Omega |\nabla u^\gamma| \, dx \,=\, \int_\Omega u^\gamma \operatorname{div} p^* \, dx \,.$$

□

With helps of the above propositions, we then achieve our final result:

Proposition 16. *Given the image function $f(x) \in \{f_1, \dots, f_n\}$, where $f_1 < \dots < f_n$ and the boundary of each concerning upper level set $F^{f_i}(x)$, $i = 1, \dots, n$, is regular, then for any given $n-1$ values γ_i, $i = 1 \dots n-1$, such that*

$$f_1 < \gamma_1 < f_2 < \dots < \gamma_{n-1} < f_n \,, \tag{3.22}$$

we define the image function $u^\gamma(x)$ by the $n-1$ upper level sets (1.10) of any computed optimum $u^*(x)$ of (3.1):

$$u^\gamma(x) \;=\; f_1 + \sum_{i=1}^{n-1} (f_{i+1} - f_i)\, U^{\gamma_i}(x) \,. \tag{3.23}$$

Then $u^\gamma(x) \in \{f_1, \dots, f_n\}$ and $u^\gamma(x)$ gives an exact global optimum of (3.2).

Proof. Let $(q^*, p^*, u^*,)$ be the optimal primal-dual pair of (3.5). Hence q^* maximizes the integral $\int_\Omega q(f-u)\,dx$ over the convex set S and p^* maximizes the integral $\int_\Omega u \operatorname{div} p\, dx$ over the convex set C_α.

$u^\gamma(x) \in \{f_1, \dots, f_n\}$ as (3.23) can be rearranged as

$$u^\gamma(x) \;=\; f_1\,(1 - U^{\gamma_1}(x)) + \sum_{i=2}^{n-1} f_i\,(U^{\gamma_{i-1}}(x) - U^{\gamma_i}(x)) + f_n\, U^{\gamma_{n-1}}(x) \,.$$

Now we prove u^γ is also a global optimum of (1.7). It can be shown by considering the following facts:

By Coro. 15, p^* also maximizes the integral $\int_\Omega u^\gamma \operatorname{div} p\, dx$ over the convex set C_α and

$$\alpha \int_\Omega |\nabla u^\gamma|\, dx \;=\; \langle u^\gamma, \operatorname{div} p^* \rangle \;. \tag{3.24}$$

On the other hand, we can prove

$$\int_\Omega |f - u^\gamma|\; dx \;=\; \langle q^*, f - u^\gamma \rangle \;. \tag{3.25}$$

The optimal dual variable $q^*(x)$ actually gives the sign of $f(x) - u^*(x)$ at each $x \in \Omega$, when $f(x) \neq u^*(x)$; when $f(x) = u^*(x)$, $q^*(x)$ can take any value in $[-1, 1]$. Now we assume $u^*(x) \in [f_k, f_{k+1}]$ for the position $x \in \Omega$, then in view of (1.10) and (3.23), we have

$$u^*(x) \in [f_k, \gamma_k] \implies u^\gamma(x) = f_k \,,$$

and

$$u^*(x) \in (\gamma_k, f_{k+1}] \implies u^\gamma(x) = f_{k+1} \,.$$

Since $f(x) \in \{f_1, \dots, f_n\}$ takes discrete values, we can analyze $q^*(x)$ in two cases: $f(x) \le f_k$ and $f(x) \ge f_{k+1}$.

- When $f(x) \le f_k$, in view of $u^*(x) \in [f_k, f_{k+1}]$, we have $q^*(x) = -1$ for $u^*(x) > f_k$ or $q^*(x) \ge -1$ for $u^*(x) = f(x)$ in order to maximize $q(x) \cdot (f(x) - u^*(x))$ over $q(x) \in [-1, 1]$. Then in both cases, $q^*(x)$ also maximizes the product $q(x) \cdot (f(x) - f_k)$, or $q(x) \cdot (f(x) - f_{k+1})$, over $q(x) \in [-1, 1]$. Hence $q^*(x)$ maximizes $q(x) \cdot (f(x) - u^\gamma(x))$ over $q(x) \in [-1, 1]$.
- When $f(x) \ge f_{k+1}$, in view of $u^*(x) \le f_{k+1}$, we have $q^*(x) = 1$ for $u^*(x) < f_{k+1}$ or $q^*(x) \le 1$ for $u^*(x) = f(x)$ in order to maximize $q(x) \cdot (f(x) - u^*(x))$ over $q(x) \in [-1, 1]$. In both cases, $q^*(x)$ also maximizes the product $q(x) \cdot (f(x) - f_k)$ or $q(x) \cdot (f(x) - f_{k+1})$, over $q(x) \in [-1, 1]$. Hence $q^*(x)$ maximizes $q(x) \cdot (f(x) - u^\gamma(x))$ over $q(x) \in [-1, 1]$.

Therefore, we have q^* maximizes the integral $\langle q, f - u^\gamma \rangle$ over the convex set $S1$. Then the fact (3.25) is proved.

By virtue of (3.24), (3.25) and the dual model (3.7), we have

$$P(u^\gamma) \;=\; E(u^\gamma, p^*, q^*) \;=\; \langle q^*, f \rangle + \langle u^\gamma, \operatorname{div} p^* - q^* \rangle \;=\; \langle q^*, f \rangle \;=\; P(u^*)\,.$$

Then it follows that u^γ is also a global minimum of (3.1) as u^* is the global minimum of (3.1) and both u^γ and u^* give the same energy. Since (3.1) is just the relaxed version of (3.2), $u^\gamma(x) \in \{f_1, \dots, f_n\}$ solves (3.2) exactly and globally. □

The proposed rounding scheme (3.23) actually gives

$$u^\gamma(x) \;=\; \begin{cases} f_1\,, & \text{when } u^*(x) < \gamma_1 \\ f_i\,, & \text{when } \gamma_{i-1} \le u^*(x) < \gamma_i,\ i = 2, \dots, n-1 \\ f_n\,, & \text{when } u^*(x) \ge \gamma_{n-1} \end{cases} .$$

In the following experiment part, we adopt the above scheme to compute the rounding results.

3.3 Multiplier-Based TV-L1 Algorithm

In this thesis, we build up the algorithm through the equivalent primal-dual model (3.5). Clearly, the primal variable u works as the multiplier in (3.5) for the linear equality $\operatorname{div} p - q = 0$. The energy function of (3.5) just gives the corresponding Lagrangian function to the dual formulation (3.7). Upon these observations, we define its augmented Lagrangian function as

$$L_c(q,p,u) \;=\; \langle q, f\rangle + \langle \operatorname{div} p - q, u\rangle - \frac{c}{2}\left\|\operatorname{div} p - q\right\|^2$$

where $c > 0$.

Thereafter, the classical augmented Lagrangian algorithm [138, 20] can be applied, which gives a splitting optimization framework over each dual variables q and p respectively, by exploring projections to their corresponding convex constrained sets. In this regard, we call Alg. 2 the *multiplier-based algorithm*. It explores two simple projection sub-steps: (3.27) and (3.28) at each iteration, which properly avoids tackling the nonsmooth terms in (3.1) in a direct way. The projection in (3.27) is easy and cheap to compute. For projection (3.28), we can use one or a few steps of the iterative algorithm in [39]. In fact, we take just one step of the following projected-gradient decent to approximately solve (3.28):

$$p^{k+1} \;=\; \operatorname{Proj}_{\|p\|_\infty \le \alpha}\big(p^k + \tau\nabla(\operatorname{div} p^k - (q^{k+1} + u^k/c))\big) \tag{3.26}$$

where $\tau \le 0.25$ gives a positive step-size and its optimal value depends on the discretization scheme. Its detailed implementation can be found in Appendix 7.2. The optimal value of the step-size τ for the mimetic finite-difference based discretization in this thesis is around 0.16. Such one-step inner iteration of updating p^k does propose a super fast numerical algorithm to TV-L1 image approximation, mostly with a superlinear convergence rate!

3.4 Experiments

The experiments for TV-L1 image approximation are designed in two parts: we evaluate both the theoretical results and efficiency of the proposed algorithm in terms of iterations in the first part; experiments of practical impulsive denoising are performed in the second part. In all experiments, convergence is evaluated by:

$$\mathrm{err}^{\mathrm{k}} \;:=\; c\,\frac{\|\operatorname{div} p^k - q^k\|}{|\Omega|}\,,$$

which just equals to $\int_\Omega \left|u^{k+1} - u^k\right| dx / |\Omega|$, through (3.29), i.e. the avarage absolute difference between u^{k+1} and u^k per image pixel. All the codes are developed on Matlab.

To evaluate the performance of rounded results in the following experiments, we take the energy difference associated to the computated optimum u^* and the rounded result u^γ which

Algorithm 2 Multiplier-Based Algorithm

```
• Set the starting values: $q^0$, $p^0$ and $u^0$, and let $k = 1$;
• Start the $k-$th iteration which includes two successive sub-steps:
    1. Optimize $q^{k+1}$ by fixing $p^k$ and $u^k$:
           $q^{k+1} := \arg \max_{\|q\|_\infty \le 1} L_c(q, p^k, u^k)$
           $\quad = \arg \max_{\|q\|_\infty \le 1} \langle q, f \rangle - \frac{c}{2} \left\| q - (\operatorname{div} p^k - u^k/c) \right\|^2 ,$
       which is approximated by the projection
           $q^{k+1} = \mathrm{Proj}_{\|q\|_\infty \le 1}(f/c + (\operatorname{div} p^k - u^k/c)) ;$    (3.27)
    2. Optimize $p^{k+1}$ by fixing $q^{k+1}$ and $u^k$:
           $p^{k+1} := \arg \min_{p \in C^\lambda} \frac{1}{2} \left\| \operatorname{div} p - (q^{k+1} + u^k/c) \right\|^2 ,$    (3.28)
       which is the projection of $(q^{k+1} + u^k/c)$ to the convex set $\operatorname{div} C_\alpha$.
• Update $u^{k+1}$ by
           $u^{k+1} = u^k + c\,(q^{k+1} - \operatorname{div} p^{k+1}) ;$    (3.29)
  and let $k = k + 1$, repeat untill convergence.
```

is measured by the ratio:

$$\text{ratio} = \left| P(u^*) - P(u^\gamma) \right| / P(u^*) \,.$$

For the comparisons to other state of art methods, the Peak Signal to Noise Ratio (PSNR) between the ground truth and the outputs, i.e.

$$\mathrm{PSNR}(u, v) = 10\log_{10} \frac{255^2}{\frac{1}{MN} \sum_{i,j} (u_{i,j} - v_{i,j})^2}$$

is measured, where $u_{i,j}$ and $v_{i,j}$ denote the pixel values of initial ground truth images and denoised images respectively.

3.4.1 Validation and Convergence

Synthetic Image

Given the synthetic image $f(x) \in \{0, 0.5, 1\}$ (see figure (b) of Fig. 3.1), which is colorized by red: 0, green: 0.5, blue: 1, we take it for the validation of Prop. 16. We set the penalty parameter $\alpha = 1$ and the augmented parameter $c = 2$. In fact, the projection-gradient step (3.26) is by setting the step-size $\tau = 0.16$. The experiment shows a fast convergence rate (see figure (a)): we run the algorithm for 3000 iterations and converges at $\mathrm{err} \simeq 1 \times 10^{-8}$.

In this experiment, two rounding schemes are taken: $\{\gamma_1 = 0.25, \gamma_2 = 0.75\}$ and $\{\gamma_1 = 0.35, \gamma_2 = 0.65\}$. For the computed result u^*, it gives the energy $P(u^*) = 2938.7$. The two corresponding rounded results produce the energy $P(u^\gamma) = 2938.7$ and 2938.7, i.e. both rounding schemes give the same energy as the convex relaxed energy $P(u^*)$! Both energy ratios are zero.

Gray Value Images

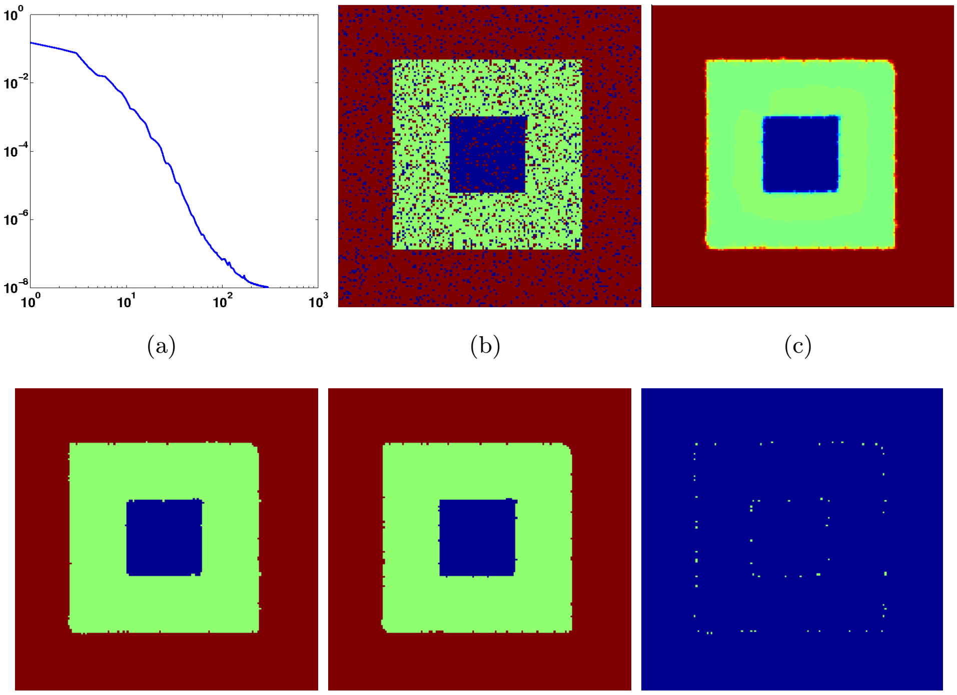
For the given gray-value images $f(x)$ of the experiments, 256 gray-scale levels are naturally encoded, i.e. $f(x) \in \{0, \dots, 255\}$.

The experiment results given in Fig. 3.2 show the denoising of the penguin image (see figure (a) of Fig. 3.2), which is downloaded from the middleburry data set: http://vision.middlebury.edu/MRF. The rounding scheme is simply taken by $\gamma = \{0.5, 1.5, \dots, 254.5\}$, i.e. it just gives the nearest integer. For the following experiments where $\alpha = 1.3$, 1, 0.5, Alg. 2 converges to a very low error shown as Fig. 3.2 (see figure (e) of Fig. 3.2): 4×10^{-11} (red line, for $\alpha = 1.3$), 5×10^{-9} (blue line, for $\alpha = 1$), 7×10^{-8} (green line, for $\alpha = 0.5$).

The energy differences associated to the computated optimum u^* and the rounded result u^γ for the three experiments are nearly zero in numerics.

The images processed in the experiments, shown in Fig. 3.3, are downloaded from the Berkeley segmentation dataset and benchmark. For all the experiments, we set $\alpha = 0.5$ and the experiment results show the ratios of energy differences are nearly zero!

Concerning the efficiency, all the experiments are made on a Linux desktop with AMD Athlon 64x2 5200+ and 3 Gb memory. For the computation of images (about 350×500 pixels) given in Fig. 3.3, we set the convergence error bound to be less than 10^{-4} and computation mostly converges within 38 iterations (about 3 sec.).



(a) (b) (c)

(d) (e) (f)

Figure 3.1: (a) convergence rate (300 iterations); (b) the input image $f(x)$ colorized by red: 0, green: 0.5, blue: 1; (c) the computed image $u^*(x)$ where $\alpha = 1$; (d) the image u^γ rounded by $\{\gamma_1 = 0.25, \gamma_2 = 0.75\}$; (e) the image u^γ rounded by $\{\gamma_1 = 0.35, \gamma_2 = 0.65\}$; (f) the difference between two rounded results.

3.4.2 Applications and Comparisons

In this section, we apply the propsoed algorithm to some real applications: impulsive image denoising and image inpainting. In addition, we will also show comparisons to the method proposed recently by [169].

Impulsive Denoising

For restoration of real images corrupted by impulsive (Salt and Pepper) noises, we first make the experiment shown by Fig. 3.4, where a Dragonfly image (which has thin and elongated details) is taken for image denoising: see the image without noise (figure (b)) and the noisy image (figure (a)) where the Salt and Pepper noise with level 5% has been added. For different choice of α which trades off the balance of keeping image details and extracting small-scale

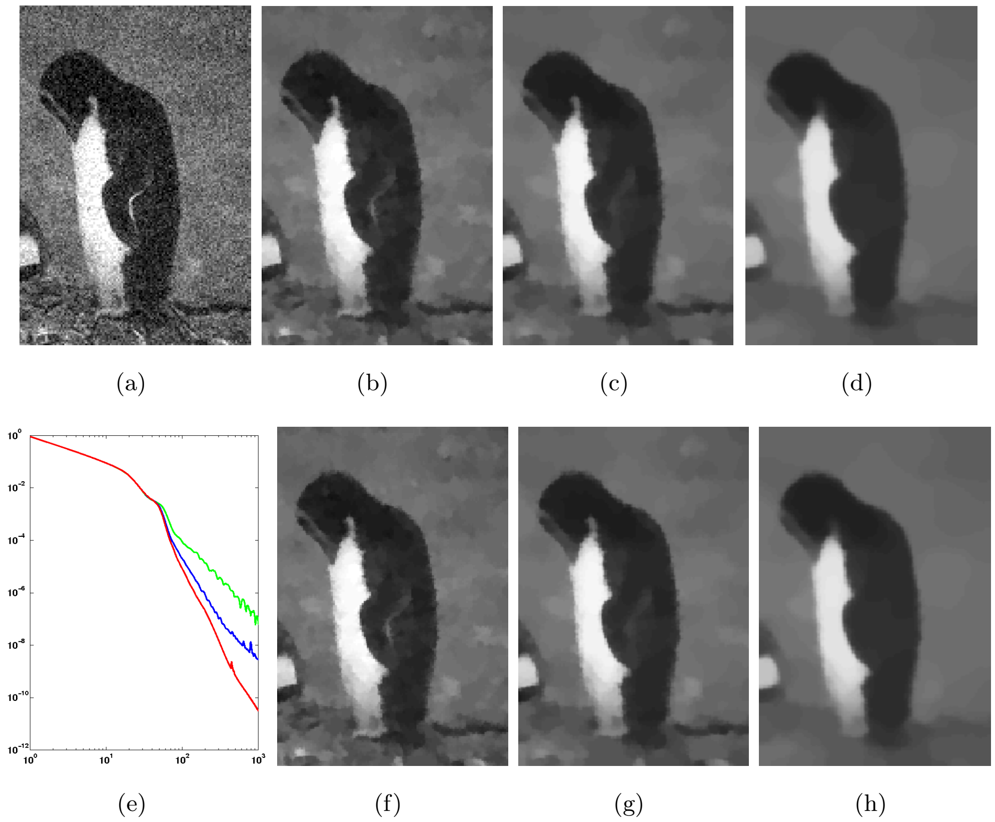


Figure 3.2: (a) the input image $f(x)$; figure (b) - (d) show the computation results when $\alpha = 1.3, 1, 0.5$ respectively; (b) plot of convergences (1000 iterations): red line: $\alpha = 1.3$, blue line: $\alpha = 1$ and green line: $\alpha = 0.5$; figure (f) - (h) show the rounding results when $\alpha = 1.3, 1, 0.5$ respectively.

structures, e.g. noises. We achieve restoration images with slight differences as shown in figure (c)-(e) of Fig. 3.4. Visually, the best result is computed by setting $\alpha = 2$. The difference between the input image $f(x)$ and the restored image, given by figure (f) of Fig. 3.4, also demostrates that detail losts of the image is very small.

In addition, we verify the performance of our method by several experiments with comparisons to the algorithmic scheme proposed in [169]. The algorithm proposed in [169] involve a substep where Chambolle-projections need to be performed iteratively. The proposed algorithm in this thesis only need one projected-gradient descent step, which greatly improve the efficiency in practice. Experiments show our algorithm is about 4 times faster than the algorithm in [169] and the computation results are even better than [169].

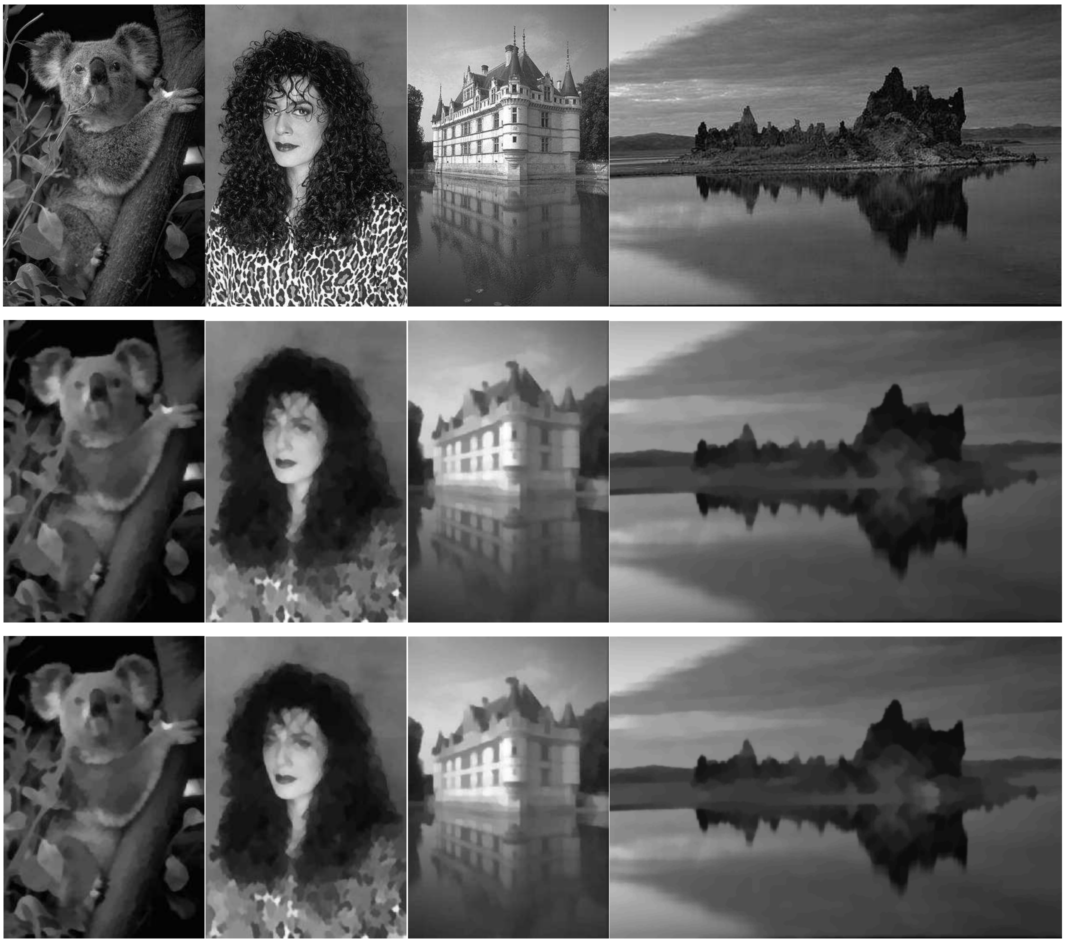


Figure 3.3: Four input images are shown in **the first row**; the computed images $u^*(x)$ are given in **the 2nd row** respectively; the rounded images $u^*(x)$ are shown in **the 3rd row** respectively. In all experiments, we set $\alpha = 0.5$.

We compare our restoration results with [169], with impulsive noise levels ranging from 10% to 50%. As the comparison results given by Tab. 3.1, the restored images computed by our proposed algorithm are better than [169] for experiments with low noise level; for the cases of high noise level, our method still keeps higher PSNR values, i.e. more image details.

For the experiment of Fig. 3.6, we try high noise levels ranging from 50% to 80%. Results show that our approach still get reasonable resutls, as shown in Fig. 3.6.

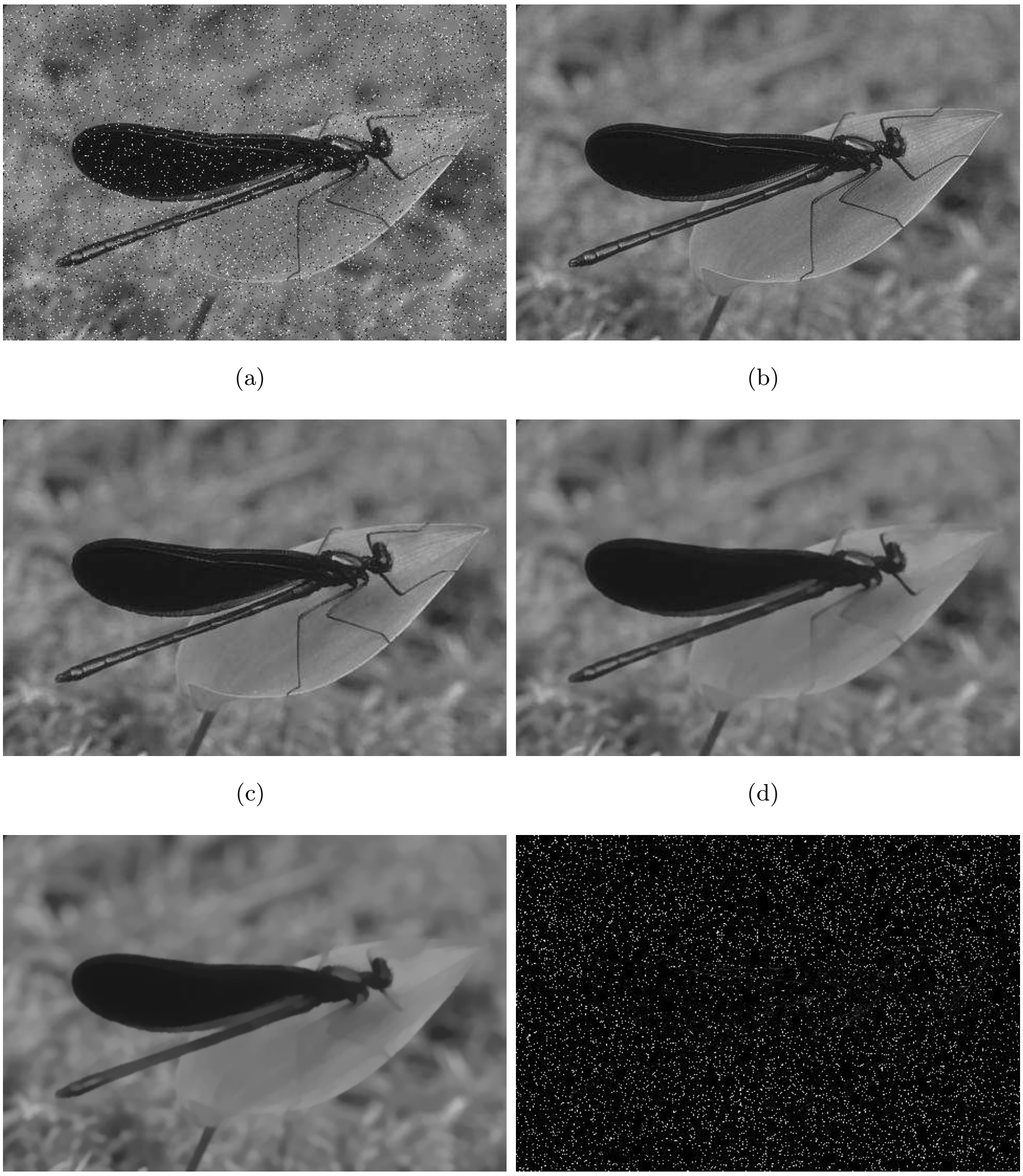


Figure 3.4: (a)noisy image noise level 5%, (b)ground truth, (c)restorated image with $\alpha = 2$, (d)restored image with $\alpha = 1.0$, (e)restored image with $\alpha = 0.5$, (f)image difference between (a) and (c).

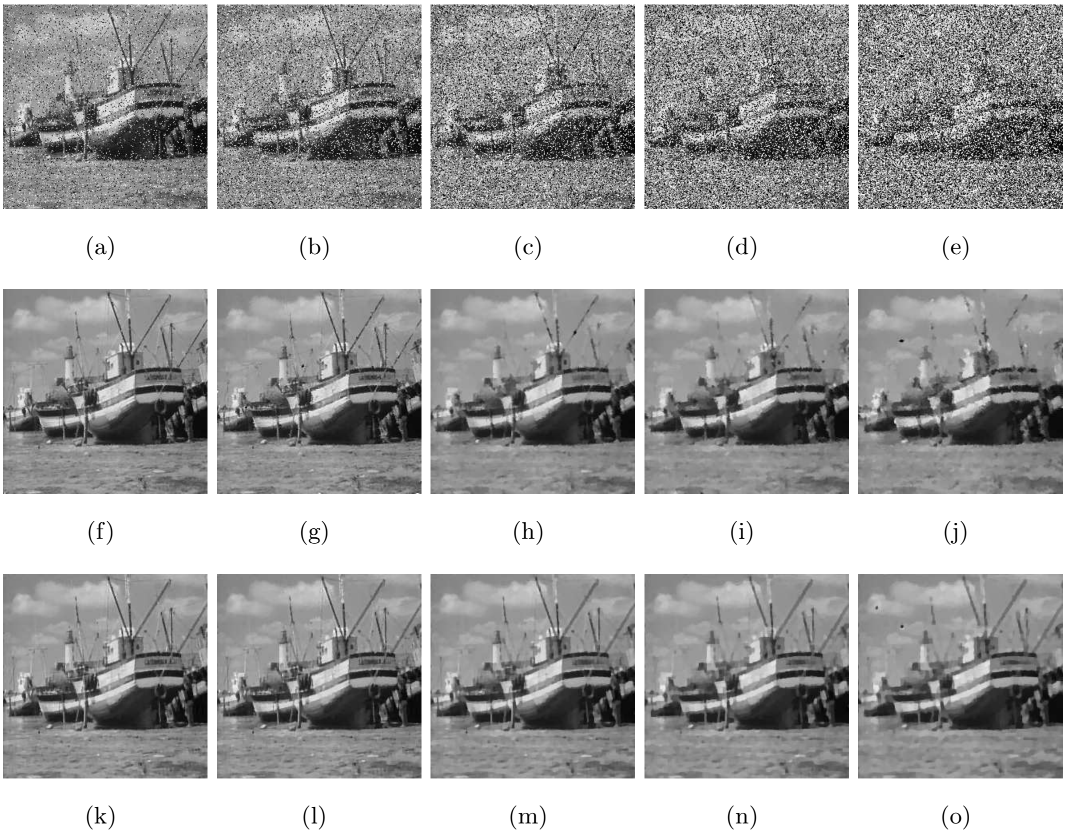


Figure 3.5: Boat denoising result with noise level from 10% to 50%, (256×256). (a)-(e)noisy image with noise level from 10% to 50% respectively,(f)-(j)denoising results by our algorithm with $\lambda = 0.7$, (k)-(o)denoising results by [169].

Noise Level	Our Approach	ALM of [169]
10%	$39.61dB$	$35.04dB$
20%	$37.56dB$	$34.71dB$
30%	$35.65dB$	$33.90dB$
40%	$34.26dB$	$33.35dB$
50%	$33.58dB$	$32.83dB$

Table 3.1: Comparison results by PSNR for the experiment (Fig. 3.5)

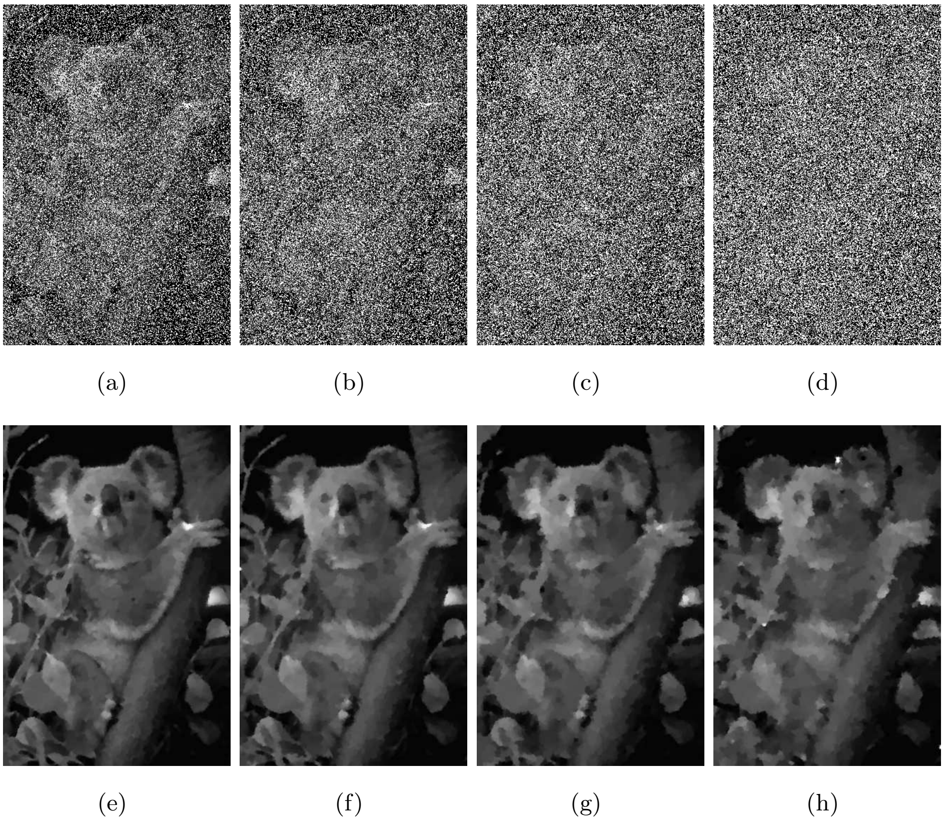


Figure 3.6: Image denoising with noise levels from 50% to 80%: (a) noisy image with noise level 50%,(b)noisy image with noise level 60%,(c)noisy image with noise level 70%, (d)noisy image with noise level 80%, (e)restored image for 50% noise level, (f)restored image for 60% noise level, (g) restored image for 70% noise level, (h) restored image for 80% noise level. All experiments are computed by $\alpha = 1.1$.

4 Continuous Max-Flow Approach to Image Segmentation

In theory of discrete optimization, performing a minimum cut, a.k.a. min-cut, to the connected weighted graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$, together with two special nodes: the source s and the sink t and associate edges, is a fundamental problem, which is of utmost interest in the applications of computer vision and image processing, especially image segmentation. Taken such graph as a single-source, single-sink flow network with flow capacities along edges, the min-cut problem tries to find the minimum capacity passing through the flow network. It is well-known that such min-cut problem is equal to the max-flow problem which is to find the maximum feasible flow through such single-source, single-sink flow network by the *max-flow and min-cut theorem* [64]. In addition, most fast min-cut solvers are designed in the manner of recovering max-flow from the source, e.g. Edmonds-Karp algorithm [57] in $O(VE^2)$, push-relabel algorithm [69] in $O(V^2E)$.

In analogy with the graph based min-cut approach to image segmentation (see Fig. 4.1) where max-flow presents most fast solvers, we study the min-cut problem over the spatially continuous image domain, namely the *continuous min-cut model*

$$\min_{u(x)\in[0,1]} \int_\Omega (1-u(x))C_s(x)\,dx + \int_\Omega u(x)C_t(x)\,dx + \alpha \int_\Omega |\nabla u(x)|\,dx\,. \tag{4.1}$$

and introduce the new max-flow formulation, so-called the *continuous max-flow model*. Like the classical *max-flow and min-cut theorem*, we prove the proposed continuous max-flow model is dual and equivalent to the continuous min-cut problem (4.1). Meanwhile, we show the nonconvex continuous min-cut version (1.12) can be solved exactly and globally by exploring the new continuous max-flow model. We also revisit and explain the fundamental terminologies of max-flow and min-cut over graphs, e.g. 'saturated flows'/'unsaturated flows' and 'cuts',and present their close connections under the new variational perspective. For the interative image segmentation which proposes the min-cut problem with supervision constraints, we show the proposed continuous max-flow approach can easily adapt such constraints without introducing any additional computational and memory load. In addition, the new fast continuous max-flow based algorithm can be derived by standard convex optimization theories.

4.1 Max-Flow and Min-Cut: a Classical Viewpoint

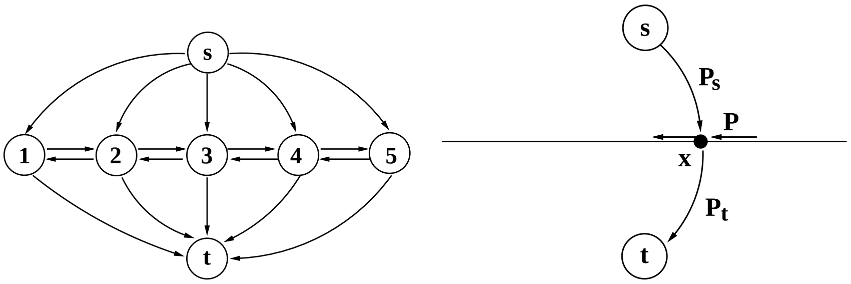


Figure 4.1: **Settings of Max-Flow and Min-Cut**, Discrete (left) vs. Continuous (right)

Now we revisit basic conceptions of the classical max-flow and min-cut. Let the graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$ consist of the node set $\mathcal{V}$ and the edge set $\mathcal{E}$. The node set of graphs used in image processing and computer vision usually includes all vertices of the 2-D or 3-D nested image grid, together with two terminal nodes: the source s and the sink t, e.g. see the left graph of Fig. 4.1. The edge set is comprised of two types of edges:

- Spatial edges $(i, j) \in \mathcal{E}$ stick to the given grid and link two neighbour stencils $i, j \in \mathcal{V} \backslash \{s, t\}$, except s and t. For the left graph of fig. 4.1, the spatial edges are the edges between nodes $\{1, 2, 3, 4, 5\}$, e.g. $(1, 2)\,(2, 3)\,(3, 4)\,(4, 5)$ (two arrows at each spatial edge represent the two directions of the flow).
- Terminal edges, e.g. (s, i) or (i, t) where $i \in \mathcal{V} \backslash \{s, t\}$, which link the specified terminal s or t to the grid node i. For the left graph of fig. 4.1, the terminal edges include the source edges which link the source s and nodes $\{1, 2, 3, 4, 5\}$, e.g. $(s, 1) \ldots (s, 5)$, and the sink edges which link nodes $\{1, 2, 3, 4, 5\}$ and the sink t, e.g. $(1, t) \ldots (5, t)$ (the arrows of the terminal edges indicate flows along these edges are directed which will be discussed in more details at the following parts).

We assign the cost $C(e)$ to each edge e, which is assumed to be nonnegative i.e. $C(e) \geq 0$.

4.1.1 Min-Cut Formulation

Upon the above configurations, a s-t cut assigns two disjoint partitions to the node set $\mathcal{V}$, i.e.

$$\mathcal{V} = \mathcal{V}_s \bigcup \mathcal{V}_t \,, \quad \mathcal{V}_s \cap \mathcal{V}_t = \emptyset \,,$$

where $\mathcal{V}_s$ includes the source s and $\mathcal{V}_t$ includes the sink t. Obviously, it segments the spatial grid vertices $\mathcal{V}\backslash\{s,t\}$ into two disjoint groups: one relates to the source s and the other one to the sink t (see the right graph of Fig. 1.4).

To each cut, its energy is the total cost of all edges $e \in \mathcal{E}_{st} \subset \mathcal{E}$, whose end-points belong to two different partitions $\mathcal{V}_s$ and $\mathcal{V}_t$ respectively. Hence the problem of min-cut is to find the s-t cut whose cut-energy is minimal. It can be mathematically formulated as

$$\min_{\mathcal{E}_{st} \subset \mathcal{E}} \sum_{e \in \mathcal{E}_{st}} C(e) \,. \tag{4.2}$$

4.1.2 Max-Flow Formulation

On the other hand, each edge $e \in \mathcal{E}$ can be viewed as a pipe and its edge cost $C(e)$ can be taken as the capacity of this pipe, for which the maximal flow is allowed. For such a 'pipe network', the following constraints on flows are applied:

- Capacity of Spatial Flows p: for each spatial edge $e_n = (i,j) \in \mathcal{E}$, $i,j \in \mathcal{V}\backslash\{s,t\}$, the spatial flow $p(e_n)$ along e_n is constrained by:

  $$|p(e_n)| \leq C(e_n) \,. \tag{4.3}$$

  Note that we consider the simplified case (4.3) to ease exposition, for which flow capacities of both directions are the same [1]. For the 2-D image grid with 4 connected neighbours, (4.3) corresponds to the well-known anisotropic total-variation regularizar when $C(e_n)$ is constant (as discussed below).

- Capacity of Source Flows p_s: for each edge $(s,i) \in \mathcal{E}$, $i \in \mathcal{V}\backslash\{s,t\}$, the source flow $p_s(i)$ is directed from s to i. Its capacity $C_s(i)$ indicates

  $$p_s(i) \leq C_s(i) \,. \tag{4.4}$$

- Capacity of Sink Flows p_t: for each edge $(i,t) \in \mathcal{E}$, $i \in \mathcal{V}\backslash\{s,t\}$, the sink flow $p_t(v)$ is directed from i to t. Its capacity $C_t(i)$ indicates

  $$p_t(i) \leq C_t(i) \,. \tag{4.5}$$

[1] More general case may also be considered, where the flow capacities at the two directions are different.

- Flow Conservation: at each node $i \in V \backslash \{s,t\}$, incoming flows should be balanced by outgoing flows, i.e. all the flows passing i including the spatial flows $p(e_n)$ where $e_n := (\mathcal{N}(i), i)$, the source flow $p_s(i)$ and the sink flow $p_t(i)$, should be constrained by the following linear equality:

$$\big(\sum_{e_n = (\mathcal{N}(i), i)} p(e_n) \big) - p_s(i) + p_t(i) \, = \, 0 \, , \tag{4.6}$$

  where $\mathcal{N}(i)$ denotes the neighbour system of the node $i \in V \backslash \{s,t\}$.

The max-flow problem over the above specified 'pipe network' tries to push flow from the source s as much as possible, which can be written as the following form:

$$\max_{p_s, p_t, p} \sum_{i \in \mathcal{V} \backslash \{s,t\}} p_s(i) \tag{4.7}$$

subject to the above conditions (4.3), (4.4), (4.5) and (4.6).

4.1.3 Duality btw. Max-Flow and Min-Cut

It is well-known that the max-flow problem (4.7) is equivalent to the min-cut problem (4.2) by the max-flow and min-cut theorem [64]. The proof can be easily found in the classical textbooks of combinatorial optimization [129, 105] or linear programming [162].

By the graph-cut terminologies, when a flow $p(e)$ on the edge $e \in E$ reaches its corresponding capacity $C(e)$, given in (4.3), (4.4) or (4.5), we call it 'saturated'; otherwise, 'unsaturated'. Reaching the status with the maximum flow, flows are saturated uniformly on the cut edges $e \in \mathcal{E}_{st}$ whose two end-points locate in different partitions, i.e. the total flow is bottlenecked by the 'saturated pipes'. We will revisit these conceptions under a variational perspective in the following sections.

4.2 Min-Cut in Spatially Continuous Setting

Under the spatially continuous context, the min-cut problem amounts to finding the optimal sub-domain $\Omega_s \subset \Omega$ such that

$$\min_{\Omega_s \subset \Omega} \int_{\Omega_s} C_t(x) \, dx \, + \int_{\Omega \backslash \Omega_s} C_s(x) \, dx \, + \, \alpha \, |\partial \Omega_s| \tag{4.8}$$

where $|\partial\Omega_s|$ denotes the perimeter of Ω_s.

Given the indicator function $u(x) \in \{0,1\}$ of Ω_s, the continuous min-cut formulation (4.8) can be rewritten as

$$\min_{u(x)\in\{0,1\}} \int_\Omega u(x)C_t(x)\,dx \,+\, \int_\Omega (1-u(x))C_s(x)\,dx \,+\, \alpha\int_\Omega |\nabla u(x)|\,dx\,, \qquad (4.9)$$

where the total-variation term of the indicator function $u(x)$ measures the perimeter of Ω_s, i.e.

$$|\partial\Omega_s| \,=\, \int_\Omega |\nabla u(x)|\,dx\,.$$

Chan et al [42, 124] proposed the convex relaxation formulation of (4.9), i.e. (4.1), which relaxes the nonconvex binary constraint $u(x) \in \{0,1\}$ to the convex one $u(x) \in [0,1]$, and proved that (4.1) solves the nonconvex optimization problem (4.8) or (4.9) exactly and globally. It results in a global optimization framework for the well-known active contour/snake model [91, 36] with region priors, e.g. active contour without edges [40]. Experiments in [124, 31] showed the proposed convex relaxation scheme properly avoided the trap of local optima and was reliable with respect to the given data and initial condition.

In this thesis, we also call (4.8) or (4.9) the *nonconvex continuous min-cut model* and (4.1) the *convex continuous min-cut model* if necessary. Otherwise, all of them are called the *continuous min-cut model* to ease reading, if not confused.

For the numerical solver of the continuous min-cut model (4.1), Chan et al [42, 124] applied a PDE-descent scheme in numerics, together with an exact penalty term to enforce the pointwise $[0,1]$ constraints at each iteration, which is sensitive to the chosen step-size and converges slowly to the optimum.

Bresson et al [31] extended Chan et al's work [42, 124] by applying the generalized weighted total-variation term. They also proposed a fast algorithm for (4.1) based on its approximation:

$$\min_{\lambda,\mu} \Big\{ \alpha\int_\Omega |\nabla\lambda(x)|\,dx \,+\, \frac{1}{2\theta}\|\lambda-u\|^2 \,+\, \int_\Omega u(x)\big(C_t(x)-C_s(x)\big)\,dx \,+\, \beta P(u)\Big\} \qquad (4.10)$$

where

$$P(u) \,:=\, \int_\Omega \max\{0, 2\,|u-0.5|-1\}\,dx$$

is an exact penalty function which forces $u(x)$ to the interval $[0,1]$ pointwise. Clearly, when $\theta > 0$ is chosen small enough, it is expected that $\lambda \simeq u$ and (4.10) solves (4.1) given $u(x) \in [0,1]$. Therefore, the constrained convex optimization problem (4.1) is approximated by a relatively simple unconstrained optimization formulation (4.10). Experiments show Bresson et al's algorithm is significantly superior than Chan et al's in terms of accuracy and efficiency [31].

In view of (4.10), the authors introduced a fast alternation-descent scheme which includes two inner steps concerning the two variables λ and u within each outer iteration.

For the k-th iteration,

- fix u^k and solve

$$\lambda^{k+1} := \arg\min_{\lambda} \Big\{ \alpha \int_{\Omega} |\nabla \lambda(x)| \, dx \, + \, \frac{1}{2\theta} \|\lambda(x) - u^k(x)\|^2 \Big\}$$

  which can be computed by the standard Chambolle's projection algorithm [39];

- fix λ^{k+1} and solve

$$u^{k+1} := \arg\min_{u} \Big\{ \frac{1}{2\theta} \|u(x) - \lambda^{k+1}\|^2 \, + \, \int_{\Omega} u(x) \big(C_t(x) - C_s(x) \big) \, dx \, + \, \beta P(u) \Big\}$$

  which can be simply solved in closed form by shrinkage (see Prop. 4 of [31]).

4.3 Continuous Max-Flow Model

Now we introduce the new *continuous max-flow model* and build up the duality between the two models: *continuous max-flow* and *continuous min-cut*.

We take the following spatially continuous settings: let Ω be the closed 2-D image domain and s, t be the source and sink terminals (see the right graph of Fig. 4.1); at each position $x \in \Omega$, we denote the connected source edge and sink edge by (s, x) and (x, t) respectively; the source flow $p_s(x)$ is directed from the source s to x along the source edge (s, x) and the sink flow is directed from x to the sink t along the sink edge (x, t); under this spatially continuous perspective, the spatial neighbours of x densely distribute around x and the spatial flows is given by the local flow field $p(x) := (p_1(x), p_2(x))$ around x; the flow excess at x is evaluated by the divergence $\operatorname{div} p(x)$, which actually corresponds to its dicrete version at some node $i \in \mathcal{V} \backslash \{s, t\}$:

$$\sum_{e_n = (\mathcal{N}(i), i)} p(e_n) \,, \quad \forall i \in \mathcal{V} \backslash \{s, t\} \,.$$

More technically, the divergence computes the volume density of the outward flux of a vector field from an infinitesimal volume around the given point. It can be properly approximated by the sum of flows over edges around the given node when the grid turns to be infinitely dense.

4.3.1 Primal Model: Continuous Max-Flow

In view of the flow constraints (4.3), (4.4), (4.5) and (4.6) over the graph setting, we consider similar constraints on the flows $p(x)$, $p_s(x)$ and $p_t(x)$ at each $x \in \Omega$:

$$\begin{aligned} |p(x)| \,=\, \sqrt{p_1(x)^2 + p_2(x)^2} \,&\leq\, C(x)\,; && (4.11)\\ p_s(x) \,&\leq\, C_s(x)\,; && (4.12)\\ p_t(x) \,&\leq\, C_t(x)\,; && (4.13)\\ \operatorname{div} p(x) - p_s(x) + p_t(x) \,&=\, 0\,; && (4.14) \end{aligned}$$

where $C(x)$, $C_s(x)$ and $C_t(x)$ give the concerning flow capacity functions and the flow conservation condition at each $x \in \Omega$ is written in the form of (4.14).

Like the discrete max-flow model (4.7), we try to push flows from the source s as much as possible which gives rise to the *continuous max-flow* by

$$\max_{p_s, p_t, p} \, P(p_s, p_t, p) \,=\, \int_\Omega p_s(x)\, dx \tag{4.15}$$

subject to the flow constraints (4.11), (4.12), (4.13) and (4.14).

In this chapter, we also call (4.15) the *primal model* and all flow variables p_s, p_t and p *primal variables*.

4.3.2 Primal-Dual Model

Introducing the multiplier $u(x)$, which is also called the *dual variable*, to the linear equality associated to the flow conservation condition (4.14), the continuous max-flow model (4.15) can then be reformulated as its equivalent *primal-dual model* :

$$\begin{aligned} \max_{p_s, p_t, p} \min_u \, E(p_s, p_t, p; u) \,&=\, \int_\Omega p_s(x)\, dx + \int_\Omega u(x)\big(\operatorname{div} p(x) - p_s(x) + p_t(x)\big)\, dx && (4.16)\\ \text{s.t.} \quad p_s(x) \leq C_s(x)\,, &\quad p_t(x) \leq C_t(x)\,, \quad |p(x)| \,\leq\, C(x)\,. \end{aligned}$$

We rearrange the primal-dual formulation (4.16) and write it by its another identical form:

$$\begin{aligned} \max_{p_s, p_t, p} \min_u \, E(p_s, p_t, p; u) \,&=\, \int_\Omega \big\{\big(1 - u(x)\big)p_s(x) + u(x)p_t(x) + u(x) \operatorname{div} p(x)\big\}\, dx && (4.17)\\ \text{s.t.} \quad p_s(x) \,\leq\, C_s(x)\,, &\quad p_t(x) \leq C_t(x)\,, \quad |p(x)| \,\leq\, C(x)\,. \end{aligned}$$

For the primal-dual model (4.17), the conditions of the minimax theorem (see e.g., [58] Chapter 6, Proposition 2.4) are all satisfied. That is, the constraints of flows are convex, and the energy function is linear to both the dual variable u and the primal functions p_s, p_t and p, hence convex l.s.c. for fixed u and concave u.s.c. for fixed p_s, p_t and p. This confirms the existence of at least one saddle point, see [58]. It also follows that the min and max operators in the above primal-dual model (4.17) can be interchanged, i.e.

$$\max_{p_s,p_t,p} \min_u E(p_s,p_t,p;u) \,=\, \min_u \max_{p_s,p_t,p} E(p_s,p_t,p;u)\,. \tag{4.18}$$

Clearly, the minimization of the primal-dual problem over the dual variable $u(x)$ gives back to the continuous max-flow model (4.15), i.e.

$$P(p_s,p_t,p) \,=\, \min_u E(p_s,p_t,p;u)\,.$$

4.3.3 Dual Model: Continuous Min-Cut

Now we consider performing the optimization of the minimax formulation (4.18) in another order, i.e. first maximize (4.17) over the flow functions p_s, p_t and p, in order to derive its equivalent *dual model*:

$$\min_{u(x)\in[0,1]} D(u) \,=\, \int_\Omega \big\{\big(1-u(x)\big)C_s(x) + u(x)C_t(x)\,dx + C(x)\,|\nabla u(x)|\,\big\}\,dx\,. \tag{4.19}$$

Obviously, when $C(x)=\alpha$ is constant, (4.19) just leads to the continuous min-cut model (4.1) proposed by Chan te al [42, 124], i.e.

$$\min_{u(x)\in[0,1]} D(u) \,=\, \int_\Omega \big\{\big(1-u(x)\big)C_s(x) + u(x)C_t(x) + \alpha\,|\nabla u(x)|\,\big\}\,dx \tag{4.20}$$

Therefore, we can prove

Proposition 17. *The continuous max-flow model* (4.15), *the primal-dual models:* (4.16) *and* (4.17), *and the continuous min-cut model* (4.19) *are equivalent to each other.*

It is clear that the continuous max-flow model equals to the primal-dual models (4.16) and (4.17). The proof of the equivalence between the primal-dual model (4.17) and the continuous min-cut model (4.19) directly comes from the following observations:

Optimization over Flow Functions p_s and p_t

In order to maximize the primal-dual model (4.17) over the flow functions p_s and p_t, let us first consider the following maximization problem

$$f(q) \,=\, \max_{p \leq C} \, p \cdot q \, . \tag{4.21}$$

Observe that if $q < 0$, p can be a negative infinity value in order to maximize the value $p \cdot q$, hence $f(q) = +\infty$; it follows that $q \geq 0$ so as to make the function $f(q)$ meaningful:

$$\begin{cases} \text{if } q = 0 \, , \quad \text{then } p < C \text{ and } f(q) \text{ reaches maximum } 0 \\ \text{if } q > 0 \, , \quad \text{then } p = C \text{ and } f(q) \text{ reaches maximum } q \cdot C \end{cases} . \tag{4.22}$$

Therefore, we can equally express $f(q)$ as

$$f(q) \,=\, q \cdot C \, , \quad q \geq 0 \, . \tag{4.23}$$

Obviously, the maximization of $q \cdot p$ over $p \leq C$ in (4.21) provides a prototype to maximize the *primal-dual model* (4.17) over the source flow function $p_s(x)$ and the sink flow function $p_t(x)$. In view of (4.23), at each position $x \in \Omega$ we have

$$\begin{aligned} f_s(x) &= \max_{p_s(x) \leq C_s(x)} \, \big(1 - u(x)\big) \cdot p_s(x) \, , \\ \Longrightarrow f_s(x) &= \big(1 - u(x)\big) \cdot C_s(x) \; , \; 1 - u(x) \, \geq \, 0 \, . \end{aligned} \tag{4.24}$$

and

$$\begin{aligned} f_t(x) &= \max_{p_t(x) \leq C_t(x)} \, u(x) \cdot p_t(x) \\ \Longrightarrow f_t(x) &= \, u(x) \cdot C_t(x) \, , \quad u(x) \, \geq \, 0 \, . \end{aligned} \tag{4.25}$$

Optimization over Flow Function p

For the maximization of (4.17) over the spatial flow $p(x)$, it is well-known that [68]

$$\max_{|p(x)| \leq C(x)} \int_\Omega u \, \operatorname{div} p \, dx \,=\, \int_\Omega C \, |\nabla u| \, dx \tag{4.26}$$

where the boundary condition $p_n|_{\partial\Omega} = 0$ is assumed in this work. By (4.24), (4.25) and (4.26),

the maximization of the primal-dual model (4.17) over flow functions $p_s(x)$, $p_t(x)$ and $p(x)$ leads to its equivalent dual model (4.19).

4.3.4 Global and Exact Optimums of Min-Cut

Chan and Esedoglu [42] proved that the convex continuous min-cut model (4.1) solves the nonconvex binary partition problem (4.8), i.e. (4.9)

$$\min_{u(x)\in\{0,1\}} \int_\Omega u(x)C_t(x)\,dx + \int_\Omega (1-u(x))C_s(x)\,dx + \alpha\int_\Omega |\nabla u(x)|\,dx \tag{4.27}$$

where $u(x)$ gives the indicator function of some partition $\Omega_s \subset \Omega$. Chan and Esedoglu [42] showed that threshhold the optimum $u^*(x)$ of the continuous min-cut model (4.1) by any value $\gamma \in (0,1]$ gives a global optimum $u^*(x) \in \{0,1\}$ of (4.27), which solves (4.8) exactly at the same time.

When $C(x)$ is a generalized function, e.g. the edge detector $C(x) = 1/(1+|\nabla f|^t)$ $t \geq 1$, the same result holds such that threshholding the optimum of (4.19) by $\gamma \in (0,1]$ results in a global optimum of the concerning binary partition problem (see Theorem 2, Bresson et al [31]).

Now we focus on the case when $C(x) = \alpha$ is constant and show that the same result as above can also be proved by means of the proposed continuous max-flow model (4.15). In contrast to [42], our proof is simpler and more elegant. In addition, we show that every continuous minimum cut of (4.8) contains the same energy as the proposed continuous max-flow model (4.15). The results can be easily extended to a more general version of non-constant $C(x)$.

Proposition 18. *Let p_s^*, p_t^*, p^* and $u^*(x)$ be the global optimum of the primal-dual model* (4.16) *when $C(x) = \alpha$. Then each $\gamma-$upper level set $S^\gamma := \{x \,|\, u^*(x) \geq \gamma\,,\ \gamma \in (0,1]\,\}$, $\gamma \in (0,1]$, of $u^*(x)$ and its indicator function u^γ*

$$u^\gamma(x) := \begin{cases} 1\,, & u^*(x) \geq \gamma \\ 0\,, & u^*(x) < \gamma \end{cases}\,,$$

solves the nonconvex min-cut problem (4.8) *globally and exactly.*

Moreover, each cut energy given by S^γ has the same energy as the optimal max-flow energy, i.e.

$$P(p_s^*, p_t^*, p^*) = \int_\Omega p_s^*(x)\,dx\,.$$

Proof. Let p_s^*, p_t^*, p^* and $u^*(x)$ be the optimal primal-dual pair of (4.16), then p_s^*, p_t^*, p^* optimize the max-flow problem (4.15) and $u^*(x)$ optimizes its dual problem (4.1). Clearly, the max-flow energy is

$$P(p_s^*, p_t^*, p^*) = \int_\Omega p_s^*(x)\,dx \tag{4.28}$$

and satisfies

$$P(p_s^*, p_t^*, p^*) \;=\; E(p_s^*, p_t^*, p^*; u^*) \;=\; D(u^*)\,.$$

For the max-flow problem (4.15), the optimal flow variables p_s^*, p_t^*, p^* must suffice the flow conservation condition (4.14), i.e.

$$\operatorname{div} p^*(x) - p_s^*(x) + p_t^*(x) \;=\; 0\,. \tag{4.29}$$

Given $\gamma \in (0, 1]$, let S^γ be the $\gamma-$upper level set of $u^*(x)$ and $u^\gamma(x)$ be its indicator function, i.e. $u^\gamma(x) \in \{0, 1\}$.

In view of (4.22), for any point $x \in \Omega \backslash S^\gamma$, i.e. $u^*(x) < \gamma \leq 1$, it is easy to see that

$$p_s^*(x) \;=\; C_s(x)\,. \tag{4.30}$$

Likewise, for any point $x \in S^\gamma$, i.e. $u^*(x) \geq \gamma > 0$, we have

$$p_t^*(x) \;=\; C_t(x)\,.$$

Then by (4.29), it follows that

$$p_s^*(x) \;=\; C_t(x) + \operatorname{div} p^*(x)\,, \quad \forall\, x \in S^\gamma\,. \tag{4.31}$$

Therefore, by (4.30) and (4.31), the total energy defined in (4.28), for each level set S^γ, is

$$\begin{aligned} P(p_s^*, p_t^*, p^*) \;&= \int_{\Omega \backslash S^\gamma} C_s(x)\, dx + \int_{S^\gamma} \big(C_t(x) + \operatorname{div} p^*(x)\big)\, dx \\ &= \int_{\Omega \backslash S^\gamma} C_s(x)\, dx + \int_{S^\gamma} C_t(x)\, dx + \int_{S^\gamma} \operatorname{div} p^*(x)\, dx \\ &= \int_{\Omega \backslash S^\gamma} C_s(x)\, dx + \int_{S^\gamma} C_t(x)\, dx + L_{S^\gamma}\,. \end{aligned}$$

Observe the facts that the energy of the nonconvex min-cut problem (4.8) over the partition S^γ, indicated by the given binary function $u^\gamma(x)$, is the same as the energy $P(p_s^*, p_t^*, p^*)$ of the convex max-flow model (4.15) which is definitely global. It follows that S^γ solves the nonconvex min-cut problem (4.8) globally. □

In comparison to the proof of Chan and Esedoglu [42], our proof is simpler. It can also be easily extended to the case when $C(x)$ is not constant. Moreover, we further indicate each cut given by S^γ shares the same energy of the corresponding max-flow model (4.15).

Observe Prop. 18, the continuous max-flow problem (4.15) equally solves the segmentation of Ω together with a minimal length, i.e. the minimum cut, where the optimal multiplier u^* works as the multiplier function to the flow conservation condition at the same time. This also gives a clue to design the max-flow based algorithm (see Alg. 3) to compute the continuous min-cut problem (4.19), which is in contrast to previous works, e.g. [124, 31, 72] etc.

4.3.5 Variational Interpretations of Flows and Cuts

In fact, analyzes of the function given in (4.21) provides a variational interpretation of the close relationships between flows and cuts, which also recovers conceptions and terminologies of graph cuts.

By means of variations, for any point $p < C$ strictly, its variation directly leads to $q = 0$ as its variation δp can be both negative and positive. On the other hand, for $p = C$, its variation under the constraint gives $\delta p < 0$ and $q > 0$. In other words, if $p < C$, i.e. does not reach its maximum which means 'unsaturated' or can be increased, then $q = 0$ which leads to the so-called 'cut' in the sense of graph-cut.

In the same manner, through (4.12), it is easy to see that when the source flow $p_s(x) < C_s(x)$ at $x \in \Omega$, i.e. 'unsaturated', we must have $1 - u(x) = 0$, i.e. $f_s(x) = (1 - u(x))p_s(x) = 0$, which means that at the position x, the source flow $p_s(x)$ has no contribution to the energy function and the flow $p_s(x)$, from the source s to x, can be 'cut' off from the energy function of (4.17). The same holds for the sink flows p_t, i.e. 'unsaturated' flow $p_t(x) < C_t(x)$ can be cut off and means $u(x) = 0$.

On the other hand, only 'saturated' source and sink flows have contributions to the total energy. For the spatial flows $p(x)$, let $C^\alpha_{\mathrm{TV}} := \{p \mid \|p\|_\infty \leq \alpha\,,\, p_n|_{\partial\Omega} = 0\,\}$. Obviously,

$$\max_{p \in C^\alpha_{\mathrm{TV}}} \langle \operatorname{div} p, u \rangle \;=\; \max_{p \in C^\alpha_{\mathrm{TV}}} \langle p, \nabla u \rangle \;. \tag{4.32}$$

The extremum of the inner product $\langle p, \nabla u \rangle$ in (5.46) just indicates the normal cone-based condition [79] of ∇u such that

$$\nabla u \;\in\; N_{C^\alpha_{\mathrm{TV}}}(p)\,. \tag{4.33}$$

Then we simply have:

$$\nabla u(x) \neq 0\,, \qquad \text{if} \quad |p^*(x)| = \alpha\;, \tag{4.34a}$$

$$\nabla u(x) = 0\,, \qquad \text{if} \quad |p^*(x)| < \alpha \tag{4.34b}$$

where p^* is the optimal value maximizing (5.46).

In other words, for some local area around $x \in \Omega$ where the flow $p^*(x)$ is 'saturated', i.e. $|p^*(x)| = \alpha$, we must have $\nabla u(x) \neq 0$, i.e. there exists jumps of $u(x)$ locally, i.e. a 'cut' locally. For some local area $x \in \Omega$ where the flow variable $p^*(x)$ is not 'saturated', i.e. $|p^*(x)| < \alpha$, we must have $\nabla u(x) = 0$, i.e. $u(x)$ is locally constant.

4.4 Supervised Continuous Max-Flow and Min-Cut

In this part, we study continuous max-flow and min-cut models with priori given supervision constraints. In contrast to the continuous max-flow and min-cut introduced previously, the supervised max-flow and min-cut computes the optimal partition subject to the user-input constraints, e.g. some image pixels are labeled in advance as foreground or background. Such supervised image partitioning problem can be modeled as the following *supervised continuous min-cut problem*

$$\begin{aligned} &\min_S \; \int_{S \backslash \Omega_f} C_s(x)\, dx + \int_{(\Omega \backslash \Omega_b) \backslash S} C_t(x)\, dx + \alpha\, |\partial S| \\ &\text{s.t. } \; \Omega_f \subset S \subset \Omega \backslash \Omega_b \end{aligned} \tag{4.35}$$

where $\Omega_f, \Omega_b \subset \Omega$ are the two disjoint regions marked a priori by the user: Ω_f belongs to the foreground or objects and Ω_b belongs to the background.

The supervised continuous min-cut formulation can be equivalently written in terms of the binary indicator function $u(x) \in \{0, 1\}$:

$$\min_{u(x) \in \{0,1\}} \int_\Omega (1 - u(x)) C_s(x)\, dx \, + \, \int_\Omega u(x) C_t(x)\, dx \, + \, \alpha \int_\Omega |\nabla u(x)|\, dx \tag{4.36}$$

subject to the labeling constraints

$$u(\Omega_f) \, = \, 1\,, \quad u(\Omega_b) \, = \, 0\,. \tag{4.37}$$

Consider the above discussions in Sec. 4.3, we may simply adopt the constraints (4.37) into the max-flow framework (4.15) by setting flow capacities as

$$C_s(\Omega_f) \, = \, +\infty\,, \quad C_t(\Omega_b) \, = \, +\infty\,. \tag{4.38}$$

This says that the source flow $p_s(x)$ is not constrained at $x \in \Omega_f$ and the sink flow $p_t(x)$ is not constrained at $x \in \Omega_b$. In view of discussions of Sec. 4.3, the labeling constraints (4.37)

directly follows. As in [28], this provides an easy way to couple the max-flow approach to the min-cut problem with the supervised constraints (4.37).

In this section, we propose new supervised max-flow and min-cut models without such artificial flow configurations (4.38), which implicitly encode the supervised information (4.37) and share the same computational complexities and load as the unsupervised ones: (4.15) and (4.19). Actually, it gives a new way to incorporate the overlap prior of regions in (4.35). It is also flexible in case the supervised information is not given in a determinant way as (4.37): for example the marked areas Ω_f and Ω_b may be provided in a 'probability' manner:

$$u(\Omega_f) \,=\, t_f \,\in\, (0,1)\,, \quad u(\Omega_b) \,=\, t_b \,\in\, (0,1) \tag{4.39}$$

where t_f and t_b are positive constants but less than 1. It is easy to see that manually modifying flow capacities as (4.38) does not work in this case.

To motivate the following approach, we first define two indicator functions concerning the label constraints (4.37):

$$u_f(x) = \begin{cases} 1, & x \in \Omega_f \\ 0, & x \notin \Omega_f \end{cases}, \quad u_b(x) = \begin{cases} 1, & x \in \Omega \backslash \Omega_b \\ 0, & x \notin \Omega \backslash \Omega_b \end{cases}. \tag{4.40}$$

Observe that Ω_f and Ω_b are disjoint, it follows that

$$u_f(\Omega_b) \,=\, 0\,, \quad u_b(\Omega_f) \,=\, 1\,. \tag{4.41}$$

For the 'soft' version of the constraints (4.39), we define

$$u_f(x) = \begin{cases} t_f, & x \in \Omega_f \\ 0, & x \notin \Omega_f \end{cases}, \quad u_b(x) = \begin{cases} 1, & x \in \Omega \backslash \Omega_b \\ 1 - t_b, & x \notin \Omega \backslash \Omega_b \end{cases}. \tag{4.42}$$

It is easy to see that the functions $u_f(x)$ and $u_b(x)$ describe the lower and upper bounds of the probability of labeling the image pixel $x \in \Omega$ as foreground objective. This is further shown in Sec. 4.4.3.

In the following discussions, we still focus on the case when (4.37) to ease the derivions. The results can be simply extended to the case of (4.39).

4.4.1 Supervised Max-Flow Model

We propose the new supervised max-flow model as follows: Consider the source flow $p_s(x)$, which flows from the source s to each pixel $x \in \Omega$; when $x \in \Omega_b$, the flow should have no contribution to the energy as it passes through the known background pixel; otherwise, it is valued as the full flow $p_s(x)$. Therefore, in view of (4.40) which implies $u_b(\Omega_b) = 0$ and $u_b(\Omega \backslash \Omega_b) = 1$, the total source flow p_s in Ω is given by $\int_\Omega u_b(x) p_s(x)\, dx$. Concerning the sink flow $p_t(x)$: it flows from each spatial pixel x to the sink t; likewise, in view of (4.40) where $u_f(\Omega_f) = 1$ and $u_f(\Omega \backslash \Omega_f) = 0$, we evaluate the total out-flow of p_t in Ω as $- \int_\Omega u_f(x) p_t(x)\, dx$.

In contrast to the *continuous max-flow* problem (4.15), we formulate the related *supervised max-flow model* as the gap of flow-in and flow-out by

$$\sup_{p_s, p_t, p} \; P_S(p_s, p_t, p) \; = \; \int_\Omega u_b(x) p_s(x)\, dx - \int_\Omega u_f(x) p_t(x)\, dx \tag{4.43}$$

subject to the same flow constraints (4.11), (4.12), (4.13) and (4.14) on p_s, p_t and p. Likewise, (4.43) is also called the primal model of the supervised max-flow / min-cut problem.

As the special case when no priori information about foreground and background is given, then we have the two characteristic functions $u_f(x) \equiv 0$ and $u_b(x) \equiv 1$ for $\forall x \in \Omega$. It is easy to check that the supervised max-flow problem (4.43) coincides with the max-flow formulation (4.15) without supervised constraints.

4.4.2 Supervised Primal-Dual Model

In analogue with (4.16), we can construct the equivalent primal-dual formulation of (4.43) by introducing the multiplier function u

$$\begin{aligned} \sup_{p_s, p_t, p} \min_u \; E_S(p_s, p_t, p; u) \; = \; & \int_\Omega u_b(x) p_s(x)\, dx - \int_\Omega u_f(x) p_t(x)\, dx + \\ & \int_\Omega u(x) \big(\operatorname{div} p(x) - p_s(x) + p_t(x) \big)\, dx \\ \text{s.t.} \quad & p_s(x) \le C_s(x)\, , \; p_t(x) \le C_t(x)\, , \; |p(x)| \; \le \; C(x)\, , \end{aligned} \tag{4.44}$$

which can be equivalently be formulated by

$$\sup_{p_s, p_t, p} \min_u \; E_S(p_s, p_t, p; u) \; = \; \int_\Omega (u_b - u) p_s\, dx + \int_\Omega (u - u_f) p_t\, dx \; + \; \int_\Omega u(x) \operatorname{div} p(x)\, dx \tag{4.45}$$

$$\text{s.t.} \quad p_s(x) \le C_s(x)\, , \; p_t(x) \le C_t(x)\, , \; |p(x)| \; \le \; C(x)\, .$$

As discussed in section 4.3.2, we have the same minimax relationship as (4.18), i.e.

$$\sup_{p_s, p_t, p} \min_u \; E_S(p_s, p_t, p; u) \; = \; \min_u \; \sup_{p_s, p_t, p} \; E_S(p_s, p_t, p; u)\, ,$$

and at least one optimal primal-dual saddle point exist.

4.4.3 Dual Model: Supervised Min-Cut Model

Maximizing all the flow functions p_s, p_t and p in $E_S(p_s, p_t, p; u)$ of (4.45), in the same manner as (4.24), (4.25) and (4.26), leads to the equivalent dual model to (4.43), also called the *supervised min-cut model*:

$$\min_u D_S(u) \, = \int_\Omega \big(u_b - u\big) C_s \, dx + \int_\Omega \big(u - u_f\big) C_t \, dx + \int_\Omega C(x) \left|\nabla u(x)\right| \, dx \qquad (4.46)$$
$$\text{s.t.} \;\; u_f(x) \, \leq \, u(x) \, \leq \, u_b(x) \, .$$

In this paper, we focus on the case that $C(x) = \alpha$, $\forall x \in \Omega$, then (4.46) can be equally written as

$$\min_u D_S(u) \, = \int_\Omega \big(u_b - u\big) C_s \, dx + \int_\Omega \big(u - u_f\big) C_t \, dx + \alpha \int_\Omega \left|\nabla u(x)\right| \, dx \qquad (4.47)$$
$$\text{s.t.} \;\; u_f(x) \, \leq \, u(x) \, \leq \, u_b(x) \, ;$$

or, observe u_b and u_f are given in advance, be shortened by

$$\min_u D_S(u) \, = \int_\Omega u\big(C_t - C_s\big) \, dx + \alpha \int_\Omega \left|\nabla u(x)\right| \, dx \qquad (4.48)$$
$$\text{s.t.} \;\; u_f(x) \, \leq \, u(x) \, \leq \, u_b(x) \, .$$

We see that (4.48) is just the convex relaxed model of the nonconvex supervised min-cut problem (4.36), where the subset ordering

$$\Omega_f \subset S \subset \Omega \backslash \Omega_b$$

in (4.35) is expressed by the inequality ordering

$$u_f(x) \leq u(x) \leq \, u_b(x) \, , \quad x \in \Omega \, .$$

Moreover, the applied inequality constraint of u in (4.48), in view of (4.40) and (4.41), just gives

$$u(\Omega_f) \, = \, 1 \, , \quad u(\Omega_b) \, = \, 0 \, . \qquad (4.49)$$

This coincides with the priori information that Ω_f is already labeled as foreground objects and Ω_b is labeled as the background. It follows that the inequality constraint of $u(x)$ implicitly encodes the priori supervision information.

In the special case when no priori information about foreground and background is given, i.e. $u_f(x) \equiv 0$ and $u_b(x) \equiv 1$, $\forall x \in \Omega$, it is easy to see that the supervised min-cut problem (4.47) is equivalent to the continuous min-cut problem (4.1) such that $u(x) \in [0, 1]$.

4.4.4 Global Binary Supervised Min-Cuts

Now we prove that global optimums of the nonconvex supervised min-cut model (4.35) can also be obtained by threshholding the global optimum u^* of its convex relaxed version (4.47) or (4.48), in a similar manner as Prop. 18.

Proposition 19. *Let p_s^*, p_t^*, p^* and $u^*(x)$ be a global optimum of the primal-dual problem* (4.44) *with $C(x) = \alpha$. Then each $\ell-$upper level set $S^\ell := \{x \,|\, u(x) \,\geq\, \ell\}$ of $u^*(x)$ where $\ell \in (0,1]$, and its indicator function u^ℓ:*

$$u^\ell(x) \,:=\, \begin{cases} 1\,, & u^*(x) \geq \ell \\ 0\,, & u^*(x) < \ell \end{cases} \,,$$

is a global solution of the nonconvex supervised min-cut problem (4.35).

Moreover, each supervised cut given by S^ℓ has the same energy as the supervised max-flow energy, i.e.

$$P_S(p_s^*, p_t^*, p^*) \,=\, \int_\Omega u_b(x) p_s^*(x)\, dx \,-\, \int_\Omega u_f(x) p_t^*(x)\, dx\,.$$

Proof. Let p_s^*, p_t^*, p^* and $u^*(x)$ be a global optimum of (4.44). Then p_s^*, p_t^*, p^* optimize the primal problem (4.43) and $u^*(x)$ optimizes (4.47) or (4.48). Meanwhile, the two energies are equal, i.e.

$$P_S(p_s^*, p_t^*, p^*) \,=\, E_S(p_s^*, p_t^*, p^*, u^*) \,=\, D_S(u^*)\,.$$

By the definition of u_b and u_f in (4.40), the energy of (4.43) is

$$\begin{aligned} P_S(p_s^*, p_t^*, p^*) \,&=\, \int_\Omega u_b(x) p_s^*(x)\, dx \,-\, \int_\Omega u_f(x) p_t^*(x)\, dx \\ &=\, \int_{\Omega\backslash\Omega_b} p_s^*(x)\, dx \,-\, \int_{\Omega_f} p_t^*(x)\, .dx \end{aligned} \qquad (4.50)$$

Concerning the supervised min-cut problem, the constraints of (4.49) simply indicates that

$$u^*(\Omega_f) \,=\, 1\,, \quad u^*(\Omega_b) \,=\, 0\,. \qquad (4.51)$$

Then each upper-level set S^ℓ, $\ell \in (0,1]$, of u^*:

$$S^\ell \,:=\, \{x \,|\, u^*(x) \,\geq\, \ell\,\}\,,$$

contains Ω_f and excludes Ω_b, i.e. we have

$$\Omega_f \,\subset\, S^\ell \,\subset\, \Omega\backslash\Omega_b\,. \qquad (4.52)$$

As $u^*(x)$ is the optimal multiplier, we must have the flow conservation condition (4.14), i.e.

$$\operatorname{div} p^*(x) - p_s^*(x) + p_t^*(x) \;=\; 0, \quad \text{a.e. } x \in \Omega. \tag{4.53}$$

For any point $x \in S^\ell \backslash \Omega_f$, where $u^*(x) \geq \ell$, it follows from (4.40) that $u^*(x) > u_f(x)$, therefore

$$p_t^*(x) \;=\; C_t(x)\,.$$

Then by (4.53), we have

$$p_s^*(x) \;=\; C_t(x) + \operatorname{div} p^*(x)\,, \quad \text{a.e. } x \in S^\ell \backslash \Omega_f\,. \tag{4.54}$$

And for any point $x \in (\Omega \backslash \Omega_b) \backslash S^\ell$, i.e. $u^*(x) < \ell$, hence $u^*(x) < u_b(x)$ and it is easy to see that

$$p_s^*(x) \;=\; C_s(x)\,. \tag{4.55}$$

Therefore, in view of (4.55) and (4.54), the total energy of (4.50) is

$$\begin{aligned} P_S(p_s^*, p_t^*, p^*) \;&=\int_{(\Omega \backslash \Omega_b) \backslash S^\ell} C_s(x)\,dx + \int_{S^\ell} \big(C_t(x) + \operatorname{div} p^*(x)\big)\,dx - \int_{\Omega_f} p^*(x)dx \\ &= \int_{(\Omega \backslash \Omega_b) \backslash S^\ell} C_s(x)\,dx + \int_{S^\ell \backslash \Omega_f} C_t(x)\,dx + \int_{S^\ell} \operatorname{div} p^*(x)\,dx \\ &= \int_{(\Omega \backslash \Omega_b) \backslash S^\ell} C_s(x)\,dx + \int_{S^\ell \backslash \Omega_f} C_t(x)\,dx + \alpha \left|\partial S^\ell\right|\,, \end{aligned}$$

which obviously gives a solution u^ℓ of the nonconvex supervised min-cut problem (4.35).

The above binary solution u^ℓ is contained in the relaxed convex set $u(x) \in [0,1]$ and reaches the globally optimal energy E^*. It follows that such binary solver is globally optimal. □

4.5 Continuous Max-Flow Algorithm and Experiments

In this section, we propose the new algorithms for the continuous min-cuts (4.1) and (4.48) based their respective max-flow formulations (4.15) and (4.43).

4.5.1 Continuous Max-Flow Based Algorithm

Now we consider the algorithm for the constrained optimization problem (4.15). The energy formulation of (4.16) is just the lagrangian function of (4.15). To this end, we define its respective augmented lagrangian function as

$$L_c(p_s, p_t, p, u) := \int_\Omega p_s \, dx + \int_\Omega u\big(\operatorname{div} p - p_s + p_t\big) \, dx - \frac{c}{2} \left\| \operatorname{div} p - p_s + p_t \right\|^2 , \tag{4.56}$$

where $c > 0$.

Therefore, we derive the algorithm, see Alg. 3, for the continuous maximal flow problem (4.15) based on the augmented lagrangian method, see [21, 20, 138] for references. The labeling function u is updated as the multiplier at each iteration.

The step (4.58) to update $p(x)$ gives a projection problem, which can be easily implemented by Chambolle's algorithm [39]. In this work, we solve (4.58) by just one step of its projected-gradient approximation:

$$p_{i,j}^{k+1} = \operatorname{Proj}_{|p(x)| \le \alpha} (p^k - \beta \big(\nabla (\operatorname{div} p^k - F^k) \big)) , \tag{4.57}$$

where $\beta \in (0, 1/4]$. Detailed numerical implementation of (4.57) can be found in the Appendix 7.2. This leads to a very fast implementation in numerics, which significantly outperforms the state of art algorithms, e.g. Bresson et al's [31], for the continuous min-cut problem.

4.5.2 Supervised Continuous Max-Flow Based Algorithm

Now we propose the algorithm for the supervision-constrained min-cut problem (4.48) based upon its equivalent continuous max-flow formulation (4.43). Likewise, its equivalent primal-dual formulation of (4.44) is obviously the lagrangian function of (4.43). We define its respective augmented lagrangian function as

$$\begin{aligned} L_c(p_s, p_t, p, u) = & \int_\Omega u_b p_s \, dx - \int_\Omega u_f p_t \, dx + \int_\Omega u\big(\operatorname{div} p - p_s + p_t\big) \, dx \\ & - \frac{c}{2} \left\| \operatorname{div} p - p_s + p_t \right\|^2 . \end{aligned}$$

where $c > 0$.

The supervised continuous max-flow based algorithm is stated in Alg. 4. The step (4.59) to update $p(x)$ gives a projection problem computed by Chambolle's algorithm [39]. In this work, we apply a single projection-descent step as (4.57) to implement Alg. 4 in a fast way (see the Appendix 7.2).

Algorithm 3 Multiplier-Based Maximal-Flow Algorithm

Set the starting values p_s^1, p_t^1, p^1 and u^1, let $k = 1$ and start $k-$th iteration, which includes the following steps, till convergence:

- Optimizing p by fixing other variables

$$\begin{aligned} p^{k+1} &:= \arg \max_{\|p\|_\infty \le \alpha} \; L_c(p_s^k, p_t^k, p, u^k) \\ &= \arg \max_{\|p\|_\infty \le \alpha} \; -\frac{c}{2} \left\| \operatorname{div} p(x) - F^k \right\|^2 \end{aligned} \tag{4.58}$$

  where F^k is fixed.

- Optimizing p_s by fixing other variables

$$\begin{aligned} p_s^{k+1} &:= \arg \max_{p_s(x) \le C_s(x)} \; L_c(p_s, p_t^k, p^{k+1}, u^k) \\ &:= \arg \max_{p_s(x) \le C_s(x)} \; \int_\Omega p_s \, dx - \frac{c}{2} \left\| p_s - G^k \right\|^2 \end{aligned}$$

  where G^k is a fixed variable and optimizing p_s can be easily computed in a closed form such that

$$p_s(x) \; = \; \min(G^k(x) + 1/c, C_s(x)) \, ;$$

- Optimizing p_t by fixing other variables

$$\begin{aligned} p_t^{k+1} &:= \arg \max_{p_t(x) \le C_t(x)} \; L_c(p_s^{k+1}, p_t, p^{k+1}, u^k) \\ &:= \arg \max_{p_t(x) \in C_t(x)} \; -\frac{c}{2} \left\| p_t - H^k \right\|^2 \, , \end{aligned}$$

  where H^k is a fixed variable and optimizing p_t can be simply solved by

$$p_t(x) \; = \; \min(H^k(x), C_t(x)) \, ;$$

- Update u by

$$u^{k+1} \; = \; u^k - c \, (\operatorname{div} p^{k+1} - p_s^{k+1} + p_t^{k+1}) \, ;$$

- Let $k = k + 1$ return to the $k + 1$ iteration till converge.

4.6 Experiments

We show two types of experiments for the proposed continuous max-flow / min-cut models: unsupervised image segmentation and supervised image segementation. all the experiments are

Algorithm 4 Multiplier-Based Supervised Max-Flow

Set the starting values p_s^1, p_t^1, p^1 and u^1, let $k = 1$ and start $k-$th iteration, which includes the following steps, till convergence:

- Optimizing p by fixing other variables

$$\begin{aligned} p^{k+1} &:= \arg \max_{\|p\|_\infty \leq \alpha} \; L_c(p_s^k, p_t^k, p, u^k) \\ &:= \arg \max_{\|p\|_\infty \leq \alpha} \; -\frac{c}{2} \left\| \operatorname{div} p - F^k \right\|^2 \; ; \end{aligned} \tag{4.59}$$

  where F^k is some fixed variable;

- Optimizing p_s by fixing other variables

$$\begin{aligned} p_s^{k+1} &:= \arg \max_{p_s(x) \leq C_s(x)} \; L_c(p_s, p_t^k, p^{k+1}, u^k) \\ &:= \arg \max_{p_s(x) \leq C_s(x)} \; \int_\Omega u_b p_s \, dx - \frac{c}{2} \left\| p_s - G^k \right\|^2 \, , \end{aligned}$$

  where G^k is a fixed variable and optimizing p_s can be easily computed in a closed form such that

$$p_s(x) \; = \; \min(G^k(x) + u_b/c, C_s(x)) \, ;$$

- Optimizing p_t by fixing other variables

$$\begin{aligned} p_t^{k+1} &:= \arg \max_{p_t(x) \leq C_t(x)} \; L_c(p_s^{k+1}, p_t, p^{k+1}, u^k) \\ &:= \arg \max_{p_t(x) \in C_t(x)} \; - \int_\Omega u_f p_t \, dx - \frac{c}{2} \left\| p_t - H^k \right\|^2 \, , \end{aligned}$$

  where H^k is a fixed variable and optimizing p_t can be also simply solved by

$$p_t(x) \; = \; \min(G^k(x) - u_f/c, C_t(x)) \, ;$$

- Update u by

$$u^{k+1} \; = \; u^k - c \, (\operatorname{div} p^{k+1} - p_s^{k+1} + p_t^{k+1}) \, ;$$

- Let $k = k + 1$ return to the $k + 1$ iteration till converge.

made on a Linux desktop with AMD Athlon 64x2 5200+ and 3 Gb memory.

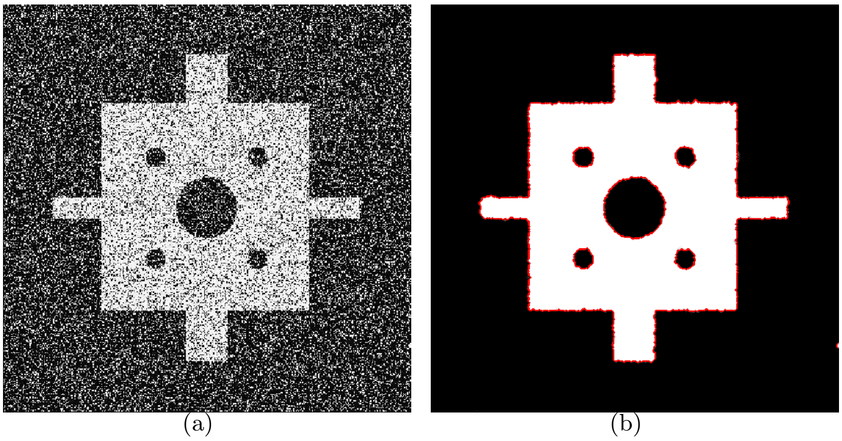


Figure 4.2: (a) Input Image, (b) the labeling function $u^*(x)$ computed by Alg. 3 and its threshold by $u^*(x) \geq 0.5$ (red contour).

In view of Alg. 3 and Alg. 4, the update step of the labeling function $u(x)$ at each iteration is given by the function $c\,(\operatorname{div} p^{k+1} - p_s^{k+1} + p_t^{k+1})$. Therefore, in all the experiments, we evaluate the avarage difference of the labeling functions at two sequential iterations as the convergence criterion, i.e.

$$\text{err}^k \;=\; c\,\frac{\int_\Omega \left|\operatorname{div} p^{k+1} - p_s^{k+1} + p_t^{k+1}\right| \, dx}{|\Omega|}\,. \tag{4.60}$$

4.6.1 Validation of Algorithm Parameters

All the experiments in this section are performed on the same 400×400 image $f(x)$ shown in Fig. 5.13, where the data terms of (4.1) are given as follows

$$C_s(x) \;=\; |f(x) - C_1|^p \;, \quad C_t(x) \;=\; |f(x) - C_2|^p \;, \quad p \;=\; 1 \text{ or } 2$$

together with the same parameter setting: $C_1 = 0.9$, $C_2 = 0.1$, $p = 1$ and the penalty parameter $\alpha = 0.75$.

To validate the parameter settings of Alg. 3, we first try different choices of $\beta = \{0.10,\ 0.14,\ 0.16,\ 0.18,\ 0.19,\ 0.2,\ 0.21,\ 0.22,\ 0.24\}$, by the fixed augmented parameter $c = 0.25$ and error bound $\epsilon < 5 \times 10^{-4}$ evaluated by (4.60). Experiment results show that around $\beta \in [0.18, 0.21]$ gives the best convergence result (with 8 iterations and 0.32 sec.), see Tab. 4.1. The first graph of Fig. 4.3 shows the respective convergences, where the bold black line for $\beta = 0.10$, the bold blue line for $\beta = 0.19$ and the bold red line for $\beta = 0.24$.

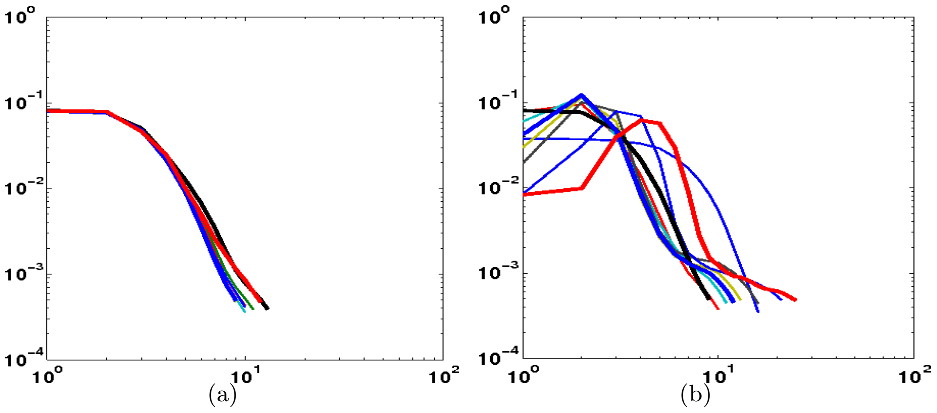


Figure 4.3: (a) Convergence results for $\beta = \{0.10, 0.14, 0.16, 0.18{,}0.19, 0.2, 0.21, 0.22, 0.24\}$ by fixed $c = 0.25$, where **bold black:** $\beta = 0.10$, **bold blue:** $\beta = 0.19$ and **bold red** $\beta = 0.24$. (b) Convergence results for $c = \{0.1, 0.25, 0.35, 0.5, 0.65, 0.8, 1, 2, 3\}$ by fixed $\beta = 0.19$, where **bold black:** $c = 0.1$, **bold blue:** $c = 0.25$ and **bold red** $c = 3$.

Tab. 4.2 and the right graph of Fig. 4.3 show the experiment results for para. settings of $c = \{0.1, 0.25, 0.35, 0.5, 0.65, 0.8, 1, 2, 3\}$ by the fixed $\beta = 0.19$. It shows that aound $c = 0.25$ gives the fastest convergence, e.g. 8 iterations and 0.31 sec. converges below the error bound 5×10^{-4}. The right graph of Fig. 4.3 presents respective convergence results, where the bold black line for $c = 0.1$, the bold blue line for $c = 0.25$ and the bold red line for $c = 3$.

Table 4.1: Validation for the step-size β when $c = 0.25$

β	0.10	0.14	0.16	0.18	0.19	0.20	0.21	0.22	0.24
Iter.	12	10	9	9	8	8	8	9	11
Time (sec.)	0.69	0.40	0.37	0.37	0.32	0.33	0.33	0.37	0.44

4.6.2 Unsupervised Image Segmentation

For image segmentation without user inputs, we adopt piecewise constant functions as the image model: i.e. two grayvalues C_1 and C_2 are chosen priori for clues to build data terms:

$$C_s(x) = |f(x) - C_1|^p \,, \quad C_t(x) = |f(x) - C_2|^p \,, \quad p = 1 \text{ or } 2$$

Table 4.2: Validation for the augmented parameter c when $\beta = 0.19$

c	0.10	0.25	0.35	0.50	0.65	0.8	1	2	3
Iter.	15	8	9	10	11	12	15	20	24
Time (sec.)	0.64	0.31	0.38	0.41	0.43	0.48	0.60	0.76	0.92

together with the same parameter setting: $C_1 = 0.15$, $C_2 = 0.6$, $p = 1$ and the penalty parameter $\alpha = 0.75$.

Fig. 4.4 and Fig. 4.5 show two experiments. Each is computed by the proposed continuous max-flow based method Alg. 3 and Bresson et al [31] for comparisons. For the experiment shown in Fig. 4.4, we chose $\alpha = 0.4$ and threshhold value $\ell = 0.5$. Our method converges to a result (see graphs at the second row of Fig. 4.4), which takes the value 0 or 1 nearly everywhere. This is in contrast to the result by Bresson's method (see graphs at the first row of Fig. 4.4). For the experiment shown in Fig. 4.5, we chose $\alpha = 0.4$ and threshhold value $\ell = 0.02$. Both results look quite the same, but our method converges significantly faster than Bresson's algorithm [31].

In contrast to Bresson et al [31], the proposed Alg. 3 converges within 13 iterations and about 0.19 sec (the accuracy below 2×10^{-4}). It greatly outperforms Bresson's in terms of convergence rate, see Fig. 4.6: Bresson's (blue line) and ours (red line). In addition, our algorithm is also reliable for a wide range of c.

4.6.3 Supervised Image Segmentation

For supervised image segmentation, we use the Middlebury data set [153] for experiments, see images in Fig. 4.7. The corresponding data term, i.e. $C_s(x)$ and $C_t(x)$, is based on Gaussian mixture color models of foreground and background and provided in advance. It is not required for us to put very large flow capacities artificially at the marked areas Ω_f and Ω_b as proposed in the supervised continuous max-flow method (4.43). This in contrast to graph-based supervised image segmentation [167, 97, 30].

Here the tree-reweighted message passing method [167, 97] and α expansion method [30, 28] are applied for comparisons. As we see, there are no visual artifact, like metrication errors, in our results (see details of the results, e.g. the left-bottom pedal of the flower (middle column)).

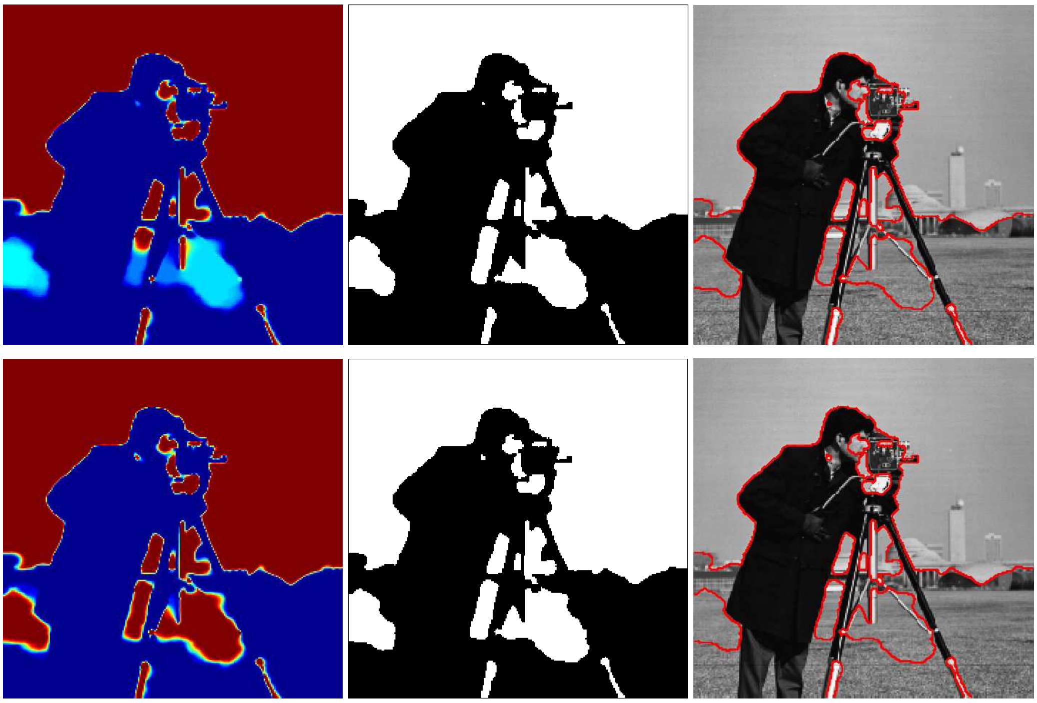


Figure 4.4: At this experiment, we chose $\alpha = 0.4$ and $\ell = 0.5$. Graphs of the first row show the results by Bresson's method: (left) computed $\lambda^*(x)$, (middle) threshholded $u^\ell(x)$, (right) segmented image. Graphs of the second row show the results by our method: (left) computed $\lambda^*(x)$, (middle) threshholded $u^\ell(x)$, (right) segmented image.

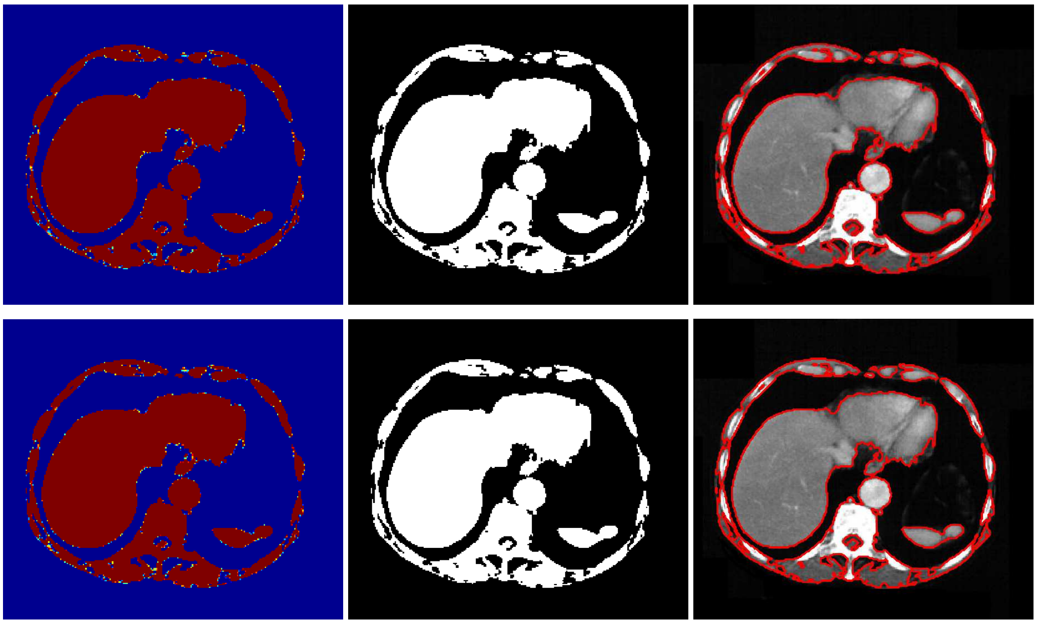


Figure 4.5: At this experiment, we chose $\alpha = 0.02$ and $\ell = 0.5$. Graphs of the first row show the results by Bresson's method: (left) computed $\lambda^*(x)$, (middle) threshholded $u^\ell(x)$, (right) segmented image. Graphs of the second row show the results by our method: (left) computed $\lambda^*(x)$, (middle) threshholded $u^\ell(x)$, (right) segmented image.

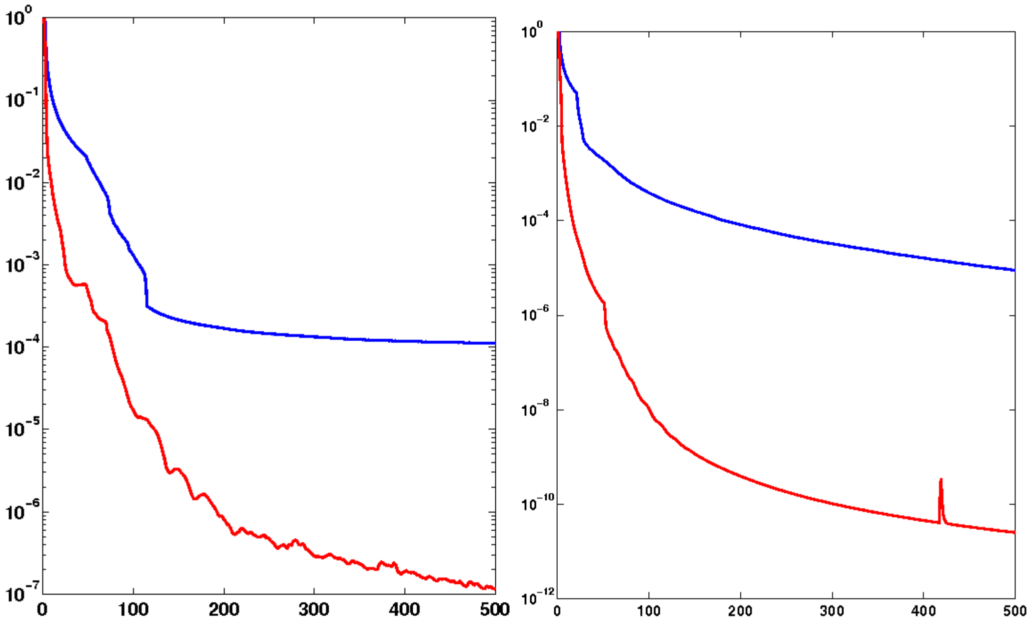


Figure 4.6: Comparisons of convergence: (**left**) for the experiment shown in Fig. 4.4, Bresson's method (blue line) converges much slower than the proposed continuous max-flow method (4.15)(red line); (**right**) for the experiment shown in Fig. 4.5, Bresson's method (blue line) also converges much slower than the proposed continuous max-flow method (4.15)(red line).

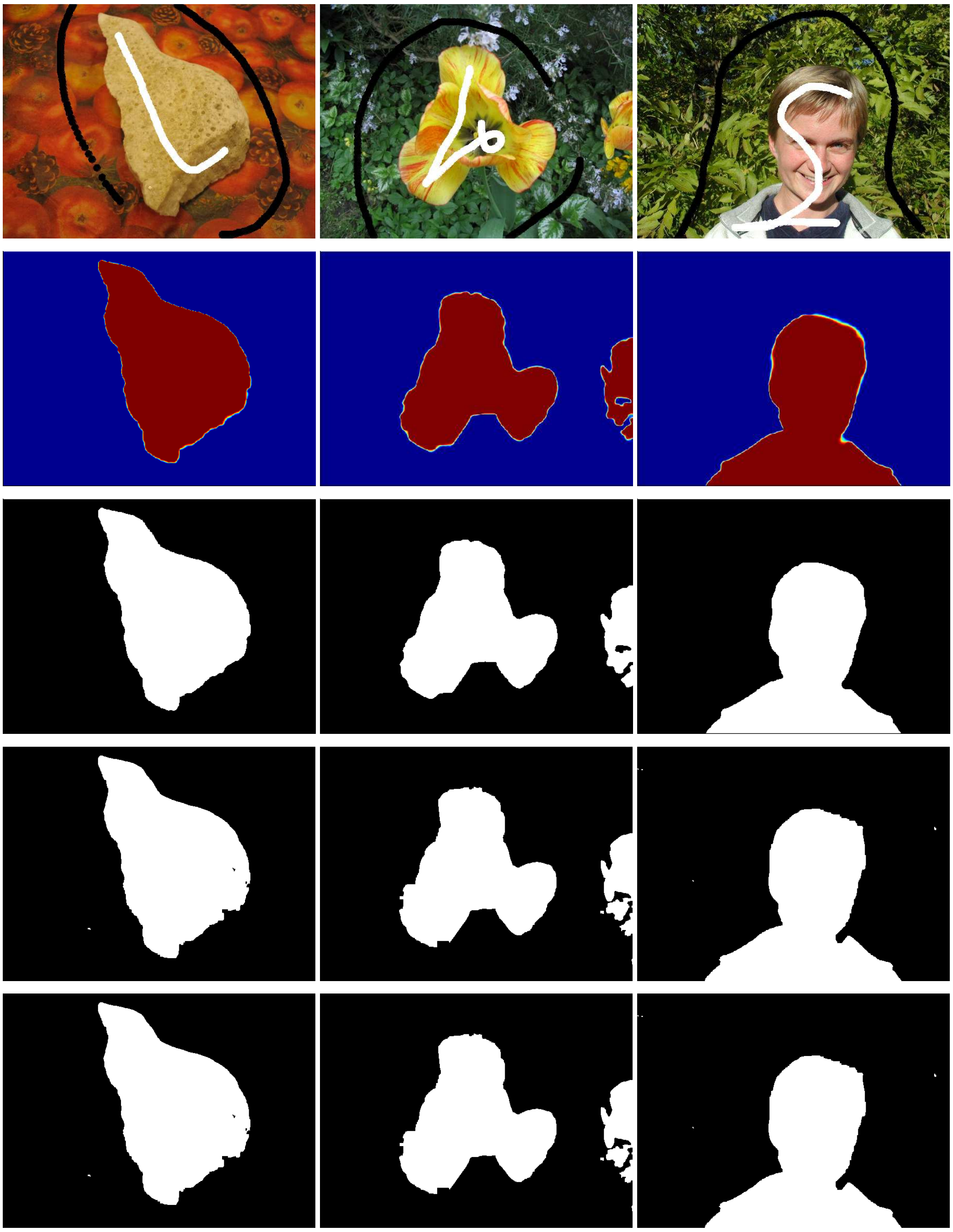


Figure 4.7: **1st. row:** The three given images, from the Middlebury data set, with pixels marked as foreground (white) and background (black). **2nd row:** computation result of λ^* to each image shown by color images, 0: blue and 1: red. **3rd row:** the black-white segmentation result by a threshhold of λ^*. **4th and 5th rows:** respective results computed from tree-reweighted message passing method [167, 97] and α expansion algorithm [30, 28].

5 Convex Optimization Approach to Potts Model

In this chapter, we discuss *Potts model* (1.14) to multi-class image partition especially in the spatially continuous setting, which can be mathematically formulated by

$$\min_{\{\Omega_i\}_{i=1}^n} \sum_{i=1}^n \int_{\Omega_i} \rho(l_i, x)\, dx + \alpha \sum_{i=1}^n |\partial\Omega_i| \ , \tag{5.1}$$

$$\text{s.t.} \quad \cup_{i=1}^n \Omega_i \, = \, \Omega \, , \quad \Omega_k \cap \Omega_l \, = \, \emptyset \, , \forall k \neq l \tag{5.2}$$

where $\rho(l_i, x), i = 1, \ldots, n,$ evaluates the cost of each label assignment l_i to the specified position $x \in \Omega$ and $|\partial\Omega_i|$ measures the perimeter of each disjoint subregion $\Omega_i \subset \Omega$, $i = 1, \ldots, n.$ Potts model (5.1) favors partitions of the given image domain Ω with the minimum total data costs and total perimeter.

In this chapter, we study Potts model (5.1) by means of convex relaxation, namely the *convex relaxed Potts model* which gives a nonsmooth convex optimization problem with complex constraints. Previous approaches focus on computing the labeling functions or partitions by tackling the resulted pixelwise simplex constraints directly, which leads to additional algorithmic steps with the extra computational load, e.g. projection to the pixelwise simplex. In contrast to these methods, we introduce two dual models to the concerned convex relaxed Potts model, which result in two different fast convex optimization approaches: the *smoothed entropy-maximization method* and the *continuous max-flow method*. Both approaches successfully avoid directly tackling the nonsmoothness and hard constraints of the given *convex relaxed Potts model*, and construct their respective fast solver in numerics.

5.1 Convex Relaxation Approaches to Potts Model

Let $u_i(x)$, $i = 1, \dots, n$, denote the indicator function of the disjoint subdomain Ω_i, i.e.

$$u_i(x) := \begin{cases} 1\,, & x \in \Omega_i \\ 0\,, & x \notin \Omega_i \end{cases} \,, \quad i = 1, \dots, n$$

The perimeter of each disjoint subdomain can be evaluated by

$$|\partial \Omega_i| = \int_\Omega |\nabla u_i| \, dx\,, \quad i = 1 \dots n\,. \tag{5.3}$$

The Potts model (1.14) can then be rewritten as

$$\min_{u_i(x) \in \{0,1\}} \sum_{i=1}^n \int_\Omega \{ u_i(x) \rho(l_i, x) \, + \, \alpha \, |\nabla u_i| \, \} \, dx\,, \text{ s.t. } \sum_{i=1}^n u_i(x) \, = \, 1\,, \, \forall x \in \Omega \tag{5.4}$$

where the constraints to $u_i(x)$, $i = 1 \dots n$, just emphasizes the segmentation principle (5.2) of subdomains Ω_i, $i = 1 \dots n$, i.e. each image pixel can be assigned by only one label from the label set $\{l_1, \dots, l_n\}$.

Clearly, the Potts model (5.4) proposes a non-convex optimization problem due to the binary configuration of each labeling function $u_i(x) \in \{0, 1\}$, $i = 1, \dots, n$. The *convex relaxed Potts model* [37, 108] proposes to relax such binary constraints to the convex interval $u_i(x) \in [0, 1]$, $i = 1, \dots, n$, and approximates (5.4) by the reduced convex optimization problem:

$$\min_{u \in S} \sum_{i=1}^n \int_\Omega u_i(x) \, \rho(l_i, x) \, dx \, + \, \alpha \sum_{i=1}^n \int_\Omega |\nabla u_i| \, dx \tag{5.5}$$

where S is the convex constrained set of $u(x) := (u_1(x), \dots, u_n(x))$:

$$S \, = \, \{ u(x) \, | \, (u_1(x), \dots, u_n(x)) \, \in \, \triangle_+\,, \, \forall x \in \Omega \, \}\,, \tag{5.6}$$

$\triangle_+$ is the simplex set, i.e.

$$\text{for } \forall x \in \Omega\,, \quad \sum_{i=1}^n u_i(x) \, = \, 1\,; \quad u_i(x) \in [0, 1]\,, \quad i = 1 \dots n\,.$$

The computation result of the convex relaxed Potts model (5.5) gives an approximation to the original Potts model (5.4) which proposes a multi-terminal 'cut' to the given image domain Ω. If the minimizer of (5.5) happens to be binary everywhere, it is definitely a global minimizer of the original problem (5.4). However, unlike the two label problem (4.1), if the computed minimizer of (5.5) is not binary, there is no thresholding scheme which can be used to generate the binary global minimizer of (5.4). Even if such a binary minimizer exists, (5.5) may result in the non-binary solution. Zach et al [184] and Lellmann et al [108] proposed to use the indicator function of the largest component $u_i^*(x)$ at each pixel $x \in \Omega$ as the approximate binary solution, i.e. the thresholded solution was selected as

$$\tilde{u}_k(x) = \begin{cases} 1 & \text{if } k = \arg\max_{i=1\ldots n}\, u_i^*(x) \\ 0 & \text{otherwise.} \end{cases} .$$

If the above maximizer is not unique, the convention of using the maximizer with smallest index was suggested.

5.1.1 Previous Approaches

In [184], Zach et al proposed an alternating optimization way to solve (5.5) approximately through

$$\min_{u,v} \sum_{i=1}^n \int_\Omega v_i\, \rho(l_i, x)\, dx + \frac{1}{2\theta} \left\| u - v \right\|^2 + \alpha \sum_{i=1}^n \int_\Omega \left| \nabla u_i \right|\, dx$$

subject to $v(x) \in \triangle_+$ for $\forall x \in \Omega$. Obviously, when θ takes a value small enough, the above convex optimization formulation approximates the convex relaxed Potts model (5.5) within certain accuracy. Within each iteration, two substeps are taken to tackle the total-variation term and explore the pointwise simplex constrained set $\triangle_+$ respectively, where one solve for $k = 1, ...$

$$u^{k+1} = \arg\min_u \; E_1(u) = \sum_{i=1}^n \int_\Omega \frac{1}{2\theta}(u_i - v_i^k)^2 + |\nabla u_i|\, dx \tag{5.7}$$

$$v^{k+1} = \arg\min_{v \in S} \; E_2(v) = \sum_{i=1}^n \int_\Omega \frac{1}{2\theta}(u_i^{k+1} - v_i)^2 + \alpha f_i v_i\, dx, \tag{5.8}$$

Here θ is some small parameter. The second problems can be optimized pointwise and has closed form solutions. However, the first subproblems are TV optimization problems, and must be solved by some iterative technique such as Chambolle's algorithm [39].

Lellmann et al [108, 106] applied a Douglas-Rachford like splitting approach [118] to (5.5) along with a variant of the total-variation term:

$$\int_\Omega \sqrt{|\nabla u_1(x)|^2 + \ldots + |\nabla u_n(x)|^2}\, dx \, .$$

After some derivations, such a scheme can be written

$$u^k = \arg\min_u \; \frac{1}{2} \sum_{i=1}^n \int_\Omega (u_i - (z_i^k - \tau f_i))^2\, dx + (\tau\alpha) \int_\Omega |\nabla u_i|\, dx \tag{5.9}$$

$$w^k = \arg\min_{w\in S} \frac{1}{2\tau}\int_\Omega (w-(2u^k - z^k))^2\, dx,\ \ z^{k+1} = z^k + w^k - u^k \tag{5.10}$$

As we see, this scheme also involves a substep (5.9) where n TV minimization problems need to be solved iteratively. The second subproblems (5.10) have closed form solutions and can be solved efficiently. In contrast to (5.7), (5.8), such a scheme can be proved to convergence provided the subproblems are solved exactly. As we see, both these algorithms require one outer loop and one inner loop. Also in [106], the similar algorithm was applied to solve the given convex relaxation problem within a suboptimality bound.

In [37, 131], Pock et al. introduced a variant implementation of the convex constraint $u(x) \in \triangle_+$, i.e. a tighter relaxation based on the multi-layered configuration, and this gave a more complex constraint on the concerning dual variable p to avoid multiple counting. In addition, a primal-dual projection-descent scheme is applied to approximate the minimum.

In this part, we call the continuous optimization problem (5.5) the *primal formulation* or *primal model* and u_i, $i = 1, \ldots, n$, primal variables, in comparison to its equivalent convex models discussed in the following sections.

5.2 Duality-Based Entropy Maximization Approach

In this section, we introduce an equivalent dual model to the studied convex relaxed Potts model (5.5) and then build up the corresponding duality based approach. By analyzing the dual formulation, sufficient conditions can be derived. They show that the relaxation is often exact, i.e. there exists optimal solutions that are also globally optimal to the original nonconvex Potts model (5.4).

In order to deal with the highly nonsmooth dual problem, we develop a smoothing method based on the entropy-maximization regularizar. It leads to a novel smoothed primal-dual model and suggests labelings with the maximum entropy. Analysis gives the approximation bound of the proposed smoothing scheme. Such a smoothing method for the dual model also yields a new thresholding scheme to obtain approximate solutions. An expectation maximization like algorithm is proposed based on the smoothed version of the dual formulation, which is shown to be superior in numerical efficiency compared to earlier convex relaxation approaches and outperforms in various aspects, such as achieving lower energies and better visual quality.

5.2.1 Equivalent Convex Models

Primal-Dual Model

By using integration by parts, it is well known that the total variation term in (5.5) can equivalently be formulated as a maximization problem

$$\alpha \int_\Omega |\nabla u| \, dx \, = \, \max_{p \in C_\alpha} \, - \int_\Omega \nabla u \cdot p \, dx \, = \, \max_{p \in C_\alpha} \int_\Omega u \operatorname{div} p \, dx \tag{5.11}$$

in terms of the dual variable p over the convex set C_α given by

$$C_\alpha \, := \, \{ p \, : \, \Omega \mapsto \mathbb{R}^2 \, | \, |p(x)| \leq \alpha \, , \, \forall x \in \Omega \, ; \, p_n|_{\partial\Omega} = 0 \, \} \, , \tag{5.12}$$

see e.g. [115].

By inserting this expression, the primal problem (5.5) can be identically formulated as

$$\min_{(u_1(x),\ldots,u_n(x)) \in \triangle_+} \, \max_{p_i \in C_\alpha} \, E(u,p) = \sum_{i=1}^n \int_\Omega u_i(x) \big(\rho(l_i, x) \, + \, \operatorname{div} p_i(x) \big) \, dx \, . \tag{5.13}$$

The variables p_i $i = 1, \ldots, n$ are named as *dual variables* in this section. The min-max problem (5.13) gives an equivalent *primal-dual formulation* or *primal-dual model* of (5.5), which can be optimized over both the primal variables u_i and the dual variables p_i.

Note that the min and max operators in the above primal-dual model (5.13) can be interchanged

$$\min_{(u_1(x),\ldots,u_n(x)) \in \triangle_+} \, \max_{p_i \in C_\alpha} \, E(u,p) \, = \, \max_{p_i \in C_\alpha} \, \min_{(u_1(x),\ldots,u_n(x)) \in \triangle_+} \, E(u,p) \tag{5.14}$$

because the conditions of the minimax theorem (see e.g., [58] Chapter 6, Proposition 2.4), also [62]) are all satisfied. That is, C_α and Δ_+ are convex, and the energy function $E(u,p)$ is linear in both variables u and p, hence convex l.s.c. for fixed p and concave u.s.c. for fixed u. This also implies the existence of at least one saddle point, see [58].

Dual Model

We will now derive another equivalent formulation of (5.5) by optimizing the primal-dual model (5.13) via the primal variable $(u_1(x), \ldots, u_n(x)) \in \triangle_+$ at each position $x \in \Omega$.

Observe that for any vector $q = (q_1, \ldots, q_n) \in \mathbb{R}^n$ and $v = (v_1, \ldots, v_n) \in \triangle_+$, we have

$$\min_{(v_1,\ldots,v_n) \in \triangle_+} \, \sum_{i=1}^n v_i q_i \, = \, \min(q_1, \ldots, q_n) \, , \tag{5.15}$$

Therefore, minimizing (5.13) over the primal variables $u_i(x)$ $i = 1, \ldots, n$, at each position $x \in \Omega$, leads to

$$\max_{p_i \in C_\lambda} \; E^D(p) \; := \; \int_\Omega \big\{ \min(\rho(l_1, x) + \operatorname{div} p_1, \ldots, \rho(l_n, x) + \operatorname{div} p_n) \big\} \, dx \, . \tag{5.16}$$

We call (5.16) the *dual model* of the convex relaxed Potts model (5.5). $E^D(p)$ is called the dual energy functional.

By regarding $d(l_i, x) \; = \; \rho(l_i, x) + \operatorname{div} p_i(x)$, $x \in \Omega$, as the proximity measure of labeling x as l_i, $i = 1, \ldots, n$, the minimal distance indicates which label should be assigned at x by the dual model (5.16). In this sense, the dual formulation (5.16) can be viewed as a generalized center-based clustering formulation [159, 15], where $f_i(x)$ are the data and $\operatorname{div} p_i(x)$ are the centroids. In contrast to the classical clustering problem, the spatial centroids are formally constrained by a convex set. Moreover, updating $p_i(x)$ over (5.11) minimizes the perimeter of the spatial partitions Ω_i, $i = 1, \ldots, n$, implicitly! This gives a geometrical explanation of the dual model (5.16) in the sense of minimal length clusterings.

Discussions and Comments

Now we have two equivalent optimization models, the primal-dual model (5.13) and the dual model (5.16), to the primal optimization problem (5.5). Clearly, the energy functional $E^P(u)$ of the primal model (5.5) is given by maximizing $E(u, p)$ of the primal-dual problem (5.13) over the dual variable p first, i.e.

$$E^P(u) \; := \; \max_{p_i \in C_\lambda} \; E(u, p)$$

Likewise, the energy functional $E^D(p)$ of the dual model (5.16) is resulted by minimizing $E(u, p)$ first over the primal variable u, i.e.

$$E^D(p) \; := \; \min_{(u_1(x), \ldots, u_n(x)) \in \triangle_+} \; E(u, p)$$

As a consequence, we always have

$$E^P(u) \; \geq \; E(u, p) \; \geq \; E^D(p) \, . \tag{5.17}$$

Let (u^*, p^*) be the optimal saddle point to the primal-dual model (5.13)), then by (5.14) we have

$$E^P(u^*) \; = \; E(u^*, p^*) \; = \; E^D(p^*) \, . \tag{5.18}$$

(u^*, p^*) is also called the optimal primal-dual pair.

The equivalences between these models implies that we can also solve the convex relaxed Potts problem (5.5) by optimizing its dual model (5.16). In fact, when the n values $(\rho(l_1, x) + \operatorname{div} p_1^*(x), ..., \rho(l_n, x) + \operatorname{div} p_n^*(x))$ at $x \in \Omega$ have a unique minimum, an optimal primal variable $u^*(x)$ of $\min_{u(x)\in\Delta_+} E(u, p^*)$ at x can be uniquely recovered, in view of (5.15), by

$$u_k^*(x) \;=\; \begin{cases} 1 & \text{if } k = \arg\min_{i=1,\dots,n} \left(f_i(x) + \operatorname{div} p_i^*(x)\right) \\ 0 & \text{otherwise} \end{cases} \,, \tag{5.19}$$

which is a binary indicator vector and exactly indicates the optimal labeling value of u^* at position x. Such a binary $u^*(x)$ is globally optimal both to the convex relaxed Potts model (5.5) and the nonconvex Potts model (5.4).

Based on the above consideration, we propose such a duality-based approach by maximizing the dual functional $E^D(p)$, in contrast to previous works which tackle the primal unknowns $u_i(x)$, $i = 1, \dots, n$, directly. Moreover, the dual model (5.16) also provides a powerful tool to analyze the connections between the global optimums of the non-convex Potts model (5.4) and its relaxed version (5.5).

5.2.2 Global Optimums of Convex Relaxed Potts Model

The existence of a global binary optimum of the convex relaxed formulation (5.4), i.e. the exactness of (5.4), is still open. However, we can show (5.4) is exact under specified conditions. To do this, we first state the relationship between a maximum p^* of the dual model (5.16) and a minimum u^* of the primal model (5.4) as follows

Theorem 20. *Given any maximum p^* of the dual problem (5.16), the primal variables u^* are defined such that (u^*, p^*) is an optimal primal-dual pair of (5.13). If the n values $(\rho(l_1, x) + \operatorname{div} p_1^*(x), \dots, \rho(l_n, x) + \operatorname{div} p_n^*(x))$, at some $x \in \Omega$, have a unique minimum, e.g. $\rho(l_k, x) + \operatorname{div} p_k^*(x)$, then $u^*(x)$ at x must be valued*

$$u_k^*(x) \;=\; 1 \;\; \textit{and} \quad u_i^*(x) \;=\; 0 \,,\, i \neq k \,. \tag{5.20}$$

If the n values $(\rho(l_1, x) + \operatorname{div} p_1^*(x), \dots, \rho(l_n, x) + \operatorname{div} p_n^*(x))$ at some $x \in \Omega$ have $k > 1$ minimums, e.g. $\rho(l_j, x) + \operatorname{div} p_j^*(x)$ $j \in \mathcal{T} = \{t_1, \dots, t_k\}$, then $u^*(x)$ at x must satisfy

$$\sum_{i=1}^{k} u_{t_i}^*(x) \;=\; 1 \;\; \textit{and} \quad u_j^*(x) \;=\; 0 \,,\, j \notin \{t_1, \dots, t_k\} \,. \tag{5.21}$$

Proof. Assume there exists a primal variable u^* such that (u^*, p^*) is an optimal primal-dual pair and u^* does not satisfy (5.20) and (5.21) for all $x \in \Omega$. Let $x \in \Omega$ be a point where (5.20) or (5.21) are violated, then for some $\varepsilon \in (0, 1]$, $u^*(x)$ satisfies

$$\sum_{i \in \mathcal{T}} u_i^*(x) = 1 - \varepsilon, \qquad \sum_{i \in \{1,...,n\} \backslash \mathcal{T}} u_i^*(x) = \varepsilon.$$

Let $\min^{\text{2nd}}_{i \in \{1,...,n\}}(a_i)$ denote the second smallest component of $(a_1, ..., a_n)$, then

$$\sum_{j \in \mathcal{T}} u_j^*(x)(\rho(l_j, x) + \operatorname{div} p_j^*(x)) + \sum_{i \in \{1,...,n\} \backslash \mathcal{T}} u_i^*(\rho(l_i, x) + \operatorname{div} p_i^*(x))$$

$$= \sum_{j \in \mathcal{T}} u_j^*(x) \min_{i \in \{1,...,n\}} (\rho(l_i, x) + \operatorname{div} p_i^*(x)) + \sum_{i \in \{1,...,n\} \backslash \mathcal{T}} u_i^*(x)(\rho(l_i, x) + \operatorname{div} p_i^*(x))$$

$$\geq \sum_{j \in \mathcal{T}} u_j^*(x) \min_{i \in \{1,...,n\}} (\rho(l_i, x) + \operatorname{div} p_i^*(x)) + \sum_{j \in \{1,...,n\} \backslash \mathcal{T}} u_j^*(x) \overset{\text{2nd}}{\min_{i \in \{1,...,n\}}} (\rho(l_i, x) + \operatorname{div} p_i^*(x)),$$

$$(1 - \varepsilon) \min_{i \in \{1,...,n\}} (\rho(l_i, x) + \operatorname{div} p_i^*(x)) + \epsilon \overset{\text{2nd}}{\min_{i \in \{1,...,n\}}} (\rho(l_i, x) + \operatorname{div} p_i^*(x)),$$

$$= \min_{i \in \{1,...,n\}} (\rho(l_i, x) + \operatorname{div} p_i^*(x)) + \varepsilon \big(\overset{\text{2nd}}{\min_{i \in \{1,...,n\}}} (\rho(l_i, x) + \operatorname{div} p_i^*(x)) - \min_{i \in \{1,...,n\}} (\rho(l_i, x) + \operatorname{div} p_i^*(x)) \big)$$

$$> \min_{i \in \{1,...,n\}} (\rho(l_i, x) + \operatorname{div} p_i^*(x)).$$

Therefore, integrating over all $x \in \Omega$

$$E(u^*, p^*) = \sum_{i=1}^{n} \int_\Omega u_i^*(x)(\rho(l_i, x) + \operatorname{div} p_i^*(x))\, dx > \int_\Omega \min_{i \in \{1,...,n\}} (f_i(x) + \operatorname{div} p_i^*(x))\, dx = E^D(p^*),$$

a contradiction to the fact that (u^*, p^*) is an optimal primal-dual pair of (5.13). □

Then it follows directly from Theorem 20 such that

Proposition 21. *Let* p^* *be one optimum of the dual problem* (5.16)*. If the values* $(\rho(l_1, x) + \operatorname{div} p^*(x), ..., \rho(l_n, x) + \operatorname{div} p^*(x))$ *have a unique minimum at all* x *in* Ω*, then the primal variable* u^* *given by* (5.19) *is a binary global optimum to the convex relaxed Potts problem* (5.5) *and to the original non-convex Potts model* (5.4)*.*

Proof. By Prop (20) any primal-dual pair must satisfy (5.20) and (5.21). Since the primal variable u^* given by (5.19) is the only variable that satisfies these constraints for the given p^*, it follows that (u^*, p^*) is an optimal primal-dual pair by the existence of such a pair. Hence u^* is an optimum of the primal problem (5.5).

That u^* is also a global optimum of (5.4) follows from the fact that the feasible set of the convex relaxed Potts model (5.5) contains the feasible set of the nonconvex Potts model (5.4). Therefore, any global binary optimum of the convex relaxed Potts model, which is feasible in the nonconvex Potts model (5.5), is also globally optimal to the nonconvex Potts model. □

5.2.3 Smoothed Models

In order to solve the optimal labeling problem (5.5) through its nonsmooth dual model (5.16), we propose a smoothing method in this section. It leads to the *smoothed primal-dual model* and *smoothed dual model*, associated with (5.13) and (5.16). We will show the smoothed dual model also gives rise to a simple and efficient numerical implementation to solve the studied continuous Potts model.

Asymptotic Function and Smoothed Dual Model

We first introduce the asympototic function in order to derive the smoothing method. The asymptotic function g_∞ of a proper convex function $g(u)$ is also a proper convex function, positively homogeneous and defined in an approximation way [137, 159] as

$$g_\infty(z) \;=\; \lim_{s \to 0^+} \left\{ g_s(z) := s g(s^{-1} z) \right\}.$$

For example,

$$g(u) \;=\; \sqrt{1 + \|u\|^2}\,, \quad g_\infty(z) \;=\; \|z\| \;;$$

and

$$g(u) \;=\; \log \sum_{j=1}^{k} e^{u_j}\,, \quad g_\infty(z) \;=\; \max_{1 \leq j \leq k} z_j\,. \tag{5.22}$$

We use an example to show the smoothing effects of the Log-Sum exponential function (5.22) for the highly nonsmooth function $\max_{1 \leq j \leq k} z_j$. In Fig. 5.1, the nonsmooth function $f(x) = \max(1 - x, x)$ is given in the first graph. We use the Log-Sum exponential function (5.22) to approximate it by

$$f_s(x) \;=\; s \log(\exp((1 - x)/s) + \exp(x/s))\,,$$

where $s > 0$. We see, by the two blue lines on the right graph, that the approximation becomes better when s is chosen smaller.

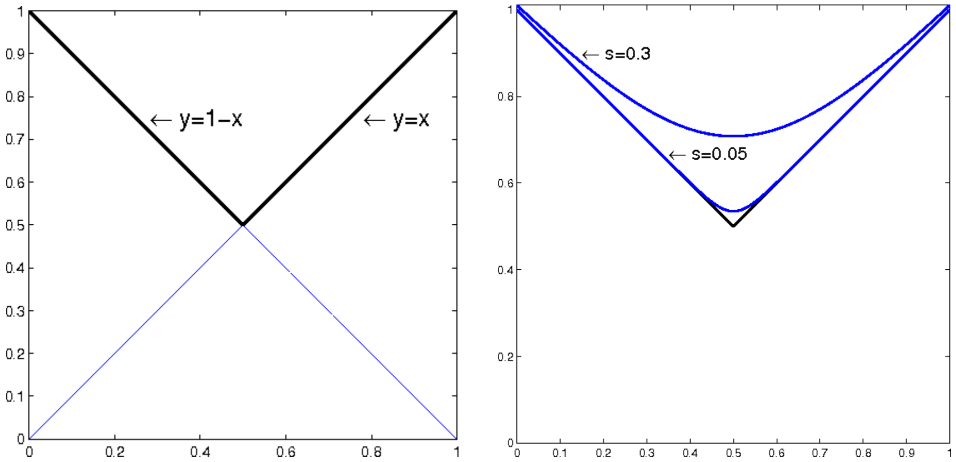


Figure 5.1: The left graph, the black and bold line, gives the function $f(x) = \max(1-x, x)$, in the interval $x \in [0,1]$. The right graph shows the the approximation of $f(x)$ by the Log-Sum exponential function $f_s(x) = s\log(\exp((1-x)/s) + \exp(x/s))$ where $s = 0.3$: the upper blue line, $s = 0.05$: the lower blue line.

Likewise, we apply (5.22) to approximate the min function in (5.16) by chosing a small parameter $s > 0$. In this way, the nonsmooth optimization problem (5.16) can be approximated by

$$\max_{p_i \in C_\lambda} \; E^D_{s>0}(p) \; := \; -s \int_\Omega \Big\{ \log \sum_{i=1}^n \exp(\frac{-\rho(l_i,x) - \operatorname{div} p_i}{s}) \Big\} \, dx \, . \qquad (5.23)$$

We call the new optimization problem (5.23) the *smoothed dual model* in comparison to the original dual one (5.16).

Equivalent Smoothed Models and Maximum Entropy Labelings

Actually, it is well known that the smooth log-sum function has an identical expression [137]:

Lemma 22. *For any given $\mu \in \triangle_+$ and $h \in \mathbb{R}^n$,*

$$\log \sum_{i=1}^n \mu_i e^{h_i} \; = \; \max_{u \in \triangle_+} \; \{ \langle u, h \rangle \; - \; \sum_{i=1}^n u_i \log \frac{u_i}{\mu_i} \} \, .$$

Let $\mu_i = 1/n$, $i = 1 \ldots n$. By the results of lemma 22, we see that the *smoothed dual model* (5.23) is just equivalent to

$$\max_{p_i \in C_\lambda} \min_{u(x) \in \triangle_+} E_s(u,p) = \int_\Omega \Big\{ \sum_{i=1}^n u_i(\rho(l_i,x) + \operatorname{div} p_i) \; + \; s \sum_{i=1}^n u_i \log u_i \Big\} \, dx \, . \qquad (5.24)$$

In view of the primal-dual model (5.13), the energy functional in the optimization problem (5.24) is just the energy functional of (5.13) plus an entropy-penalizing term. Such entropy penalization provides a proper regularization or smoothing of the original function. We, likewise, call the optimization problem (5.24) the *smoothed primal-dual model*.

Correspondingly, optimizing the dual variables p in (5.24) leads to the equivalent *smoothed primal model*:

$$\min_{u(x)\in\triangle_+} E_s^P(u) = \int_\Omega \big\{ \sum_{i=1}^n u_i\rho(l_i,x) + \lambda\sum_{i=1}^n |\nabla u_i| + s\sum_{i=1}^n u_i \log u_i \big\}\, dx\,. \tag{5.25}$$

Clearly, the positive value s here works as a penalization parameter. When s approaches 0, the optimization problem (5.24) approaches the original primal-dual problem (5.13) and the smoothed primal model (5.25) approaches its nonsmooth version (5.5). At this, the smoothed primal-dual model (5.24) shares the same formulation of the *maximum entropy clustering algorithms* [139]. To this end, we also call our smoothing approach given by (5.23) or (5.24) the method of *maximum entropy labelings*.

Approximation Bounds of Smoothed Models

In fact, the Log-Sum exponential function gives the following approximation bound of the maximum function $\max_{1\le i\le k} z_i$ [159]

Lemma 23. *For each $\mu \in \triangle_+$, the following inequalities hold,*

$$\sum_{i=1}^k \mu_i z_i \le \log\sum_{i=1}^k \mu_i e^{z_i} \le \max_{1\le i\le k} z_i\,.$$

Moreover, for $s > 0$

$$\sum_{i=1}^k \mu_i z_i \le \lim_{s\to 0^+} \{s\log\sum_{i=1}^k \mu_i e^{z_i/s}\} \le \max_{1\le i\le k} z_i\,.$$

The proof is referred to [159].

Then in view of Lemma 23, we have the approximation bound of the smoothed dual model:

Proposition 24. *For any $s > 0$, the smoothed dual model* (5.23) *gives an approximation of* (5.16), *which has the bound:*

$$0 \le E^D(p) - E^D_{s>0}(p) \le s\log n\,|\Omega|$$

where the functions $E^D(p)$ and $E^D_{s>0}(p)$ are the energy functional of (5.16) *and* (5.23) *respectively, $|\Omega|$ is the area of the domain Ω.*

Proof. Define the function $G_s(x)$ as

$$G_s(x) \;:=\; -s \log \big\{ \sum_{i=1}^{n} \exp(\frac{-\rho(l_i, x) - \operatorname{div} p_i(x)}{s}) \big\} \,,$$

i.e. the component function of (5.23) to be integrated.

Let $\mu_i = 1/n$ and $z_i = -(\rho(l_i, x) + \operatorname{div} p_i(x))$ for each $x \in \Omega$. By Lemma 23, we have

$$\sum_{i=1}^{n} \frac{\rho(l_i, x) + \operatorname{div} p_i(x)}{n} \;\geq\; s \log n + G_s(x) \;\geq\; \min_{1 \leq i \leq k} \left(\rho(l_i, x) + \operatorname{div} p_i(x)\right).$$

Therefore,

$$\min_{1 \leq i \leq k} \left(\rho(l_i, x) + \operatorname{div} p_i(x)\right) - G_s(x) \;\leq\; s \log n \,,$$

and

$$E^D(p) - E^D_{s>0}(p) \;\leq\; s \log n \, |\Omega| \;.$$

On the other hand, through lemma 22 and (5.24), we have

$$G_s(x) \;=\; \min_{u \in \triangle_+} \sum_{i=1}^{n} u_i (\rho(l_i, x) + \operatorname{div} p_i) + s \sum_{i=1}^{n} u_i \log u_i \,.$$

Hence

$$0 \;\leq\; -s \sum_{i=1}^{n} u_i \log u_i \;\leq\; \min_{1 \leq i \leq k} \left(\rho(l_i, x) + \operatorname{div} p_i(x)\right) - G_s(x) \,;$$

then

$$E^D(p) - E^D_{s>0}(p) \;\geq\; 0 \,.$$

□

By Prop. 24, the approximation bound of the smoothed model (5.23) depends on the smoothing parameter s. Hence by choosing s small enough, the smoothed dual model (5.23) solves the original nonsmooth dual model (5.16) within an expected error bound.

5.2.4 Entropy-Maximization Based Algorithm

The smooth energy function considered in the smoothed dual model (5.23) provides the feasibilty to build up an efficient and simple numerical scheme over dual variables $p_i(x)$, $i = 1 \dots n$. In order to maximize the energy functional of (5.23), we propose a projected gradient algorithm, see Alg. 5, which contains the same steps as the algorithms suggested in [39, 92].

The two main steps at each iteration can also be explained as the Expectation Maximization (EM) steps:

Algorithm 5 Projection-based Smoothing Algorithm

- Let $\delta > 0$ be chosen as some suitable step-size and let p_i^0, $i = 1, \ldots, n$ be chosen as the starting values, set $k = 0$ then start;
- Compute

$$u_i^k(x) \;=\; \frac{e^{\frac{-\rho(l_i,x) - \operatorname{div} p_i^k(x)}{s}}}{\sum_{i=1}^n e^{\frac{-\rho(l_i,x) - \operatorname{div} p_i^k(x)}{s}}}\,, \quad i = 1, \ldots, n\,; \tag{5.26}$$

- Update p_i^{k+1}, $i = 1, \ldots, n$ by

$$p_i^{k+1} \;=\; \mathrm{Proj}_{C_\alpha}(p_i^k + \delta \nabla u_i^k)\,, \quad i = 1, \ldots, n\,,$$

  where Proj_{C_α} is the projection operator to the convex set C_α;

- Let $k = k + 1$ and restart $k + 1$ iteration until convergence. When convergence is achieved, the primal variable u is recovered by

$$u_l \;=\; \begin{cases} 1 & \text{if } l = \arg\min_{i=1,\ldots,n} (f_i + \operatorname{div} p_i) \\ 0 & \text{otherwise.} \end{cases} \qquad .$$

- **Expectation Step**, compute the conditional probabilities by fixing the dual variables p_i^k, $i = 1 \ldots n$:

$$u_i^k(x) \;=\; \frac{e^{\frac{-\rho(l_i,x) - \operatorname{div} p_i^k(x)}{s}}}{\sum_{i=1}^n e^{\frac{-\rho(l_i,x) - \operatorname{div} p_i^k(x)}{s}}}\,, \quad i = 1, \ldots, n\,;$$

- **Maximization Step**, maximize the energy functional by fixing u_i^k, $i = 1 \ldots n$:

$$p_i^{k+1} \;=\; \mathrm{Proj}_{C_\alpha}(p_i^k + \delta \nabla u_i^k)\,, \quad i = 1, \ldots, n\,.$$

The above *maximization step* is implemented by the following projected descent steps:

- **Gradient-Descent Step**, compute

$$\tilde{p}_i^{k+1} \;=\; p_i^k + \delta \nabla u_i^k\,, \quad i = 1, \ldots, n$$

  where ∇u_i^k is the gradient of the energy functional of (5.23)

- **Projection Step**, compute the projection to the convex set C_α:

$$p_i^{k+1} \;=\; \mathrm{Proj}_{C_\alpha}(\tilde{p}_i^{k+1})\,, \quad i = 1, \ldots, n\,.$$

This algorithm can also be seen as a forward-backward splitting algorithm. Convergence proofs for such algorithms have been established in [45].

Implementation

In this work, we apply the *mimetic finite-difference method* to implement the two steps for the proposed Alg. 5, i.e. gradient step and projection step:

- **Gradient Step:** At each iteration of the algorithm, given $p_i \in H_S$, $f_i \in H_V$, $i = 1, \ldots, n$, we have

$$\omega_i(\alpha,\beta) \;=\; \exp \frac{-f_i(\alpha,\beta) - (\overline{\mathrm{div}^h} p_i)_{(\alpha,\beta)}}{s}\,, \quad i = 1, \ldots, n\,,$$

  then

$$\rho_i(\alpha,\beta) \;=\; \frac{\omega_i(\alpha,\beta)}{\sum_{i=1}^n \omega_i(\alpha,\beta)}\,, \quad i = 1, \ldots, n\,,$$

  for each cell (α,β).

  Therefore, the gradient of ρ_i $i = 1, \ldots, n$ is given by

$$d_i \;=\; \overline{\nabla^h} \rho_i\,, \quad i = 1, \ldots, n\,, \tag{5.27}$$

  and

$$\tilde{p}_i \;=\; p_i + \delta\, d_i\,, \quad i = 1, \ldots, n\,.$$

- **Projection Step:** Recall the convex constraint set of the dual variables (5.12). In the discrete setting, by the *mimetic finite-difference method*, any vector field $p \in C_\lambda$, at each cell (α,β) should satisfy

$$\ell_{\alpha,\beta}(p) \;:=\; \sqrt{\frac{1}{2}\big((p^b)^2 + (p^r)^2 + (p^t)^2 + (p^l)^2\big)_{(\alpha,\beta)}} \;\leq\; \lambda\,.$$

  The projection of any vector field p to the convex set C_λ can be approximated by the following two steps:

– Define $\tilde{\ell}(p) \in H_V$:

$$\tilde{\ell}_{\alpha,\beta}(p) := \begin{cases} \lambda/\ell_{\alpha,\beta}(p) & \text{when } \ell_{\alpha,\beta}(p) \geq \lambda \\ 1 & \text{when } \ell_{\alpha,\beta}(p) < \lambda \end{cases} ,$$

and define the discrete vector field $q \in H_S$:

$$q^t_{(\alpha,\beta)} := (\tilde{\ell}_{(\alpha,\beta)} + \tilde{\ell}_{(\alpha,\beta-1)})/2, \; q^b_{(\alpha,\beta)} := (\tilde{\ell}_{(\alpha,\beta+1)} + \tilde{\ell}_{(\alpha,\beta)})/2$$

$$q^l_{(\alpha,\beta)} := (\tilde{\ell}_{(\alpha,\beta)} + \tilde{\ell}_{(\alpha-1,\beta)})/2 \;, q^r_{(\alpha,\beta)} := (\tilde{\ell}_{(\alpha+1,\beta)} + \tilde{\ell}_{(\alpha,\beta)})/2 \,;$$

– $\tilde{p} := \mathrm{Proj}_{C_\lambda}(p) \in H_S$ is computed by:

$$\tilde{p}^t_{(\alpha,\beta)} := p^t_{(\alpha,\beta)}\, q^t_{(\alpha,\beta)} \,, \quad \tilde{p}^b_{(\alpha,\beta)} := p^b_{(\alpha,\beta)}\, q^b_{(\alpha,\beta)}$$

$$\tilde{p}^l_{(\alpha,\beta)} := p^l_{(\alpha,\beta)}\, q^l_{(\alpha,\beta)} \,, \quad \tilde{p}^r_{(\alpha,\beta)} := p^r_{(\alpha,\beta)}\, q^r_{(\alpha,\beta)} \,.$$

When the algorithm converges to some optimal p^*_i $i = 1, \ldots, n$, evaluate u_l by

$$u_l(\alpha, \beta) = \begin{cases} 1 & \text{if } l = \arg\min_{i=1,\ldots,n} (f_i + \overline{\mathrm{div}^h} p_i)_{(\alpha,\beta)} \\ 0 & \text{otherwise} \end{cases} .$$

5.2.5 Experiments

We demonstrate the performance of the smoothed dual model by several experiments and compare with established methods. Alpha expansion and alpha-beta swap [30] are widely considered as state of the art for approximately minimizing the discrete version of (5.1) with anisotropic total variation (TV) term. The method proposed in this paper instead minimizes the more ideal energy functional with isotropic TV term, i.e. the euclidian length of the boundaries. Because of this difference, energy comparison is not straight forward. However, there exists a result which allows to approximate the euclidian curve length on a discrete grid. This result is called the Cauchy-Crofton formula and was specialized for computer vision problems in [29]. In short, it gives a formula for edge weights between neighboring grid points such that the discrete boundary length converges to the euclidean boundary length as the mesh size goes to zero and the number of neighbors goes to infinity. This result can therefore be used to determine weights on regularization edges in the discrete model, such that it correctly corresponds to the continuous model. It is also used to compute the final energy of the outputs produced by the different methods, i.e. it can be used to compare energy of the thresholded solutions. Secondly, we evaluate quality and efficiency with the approaches of [184, 108].

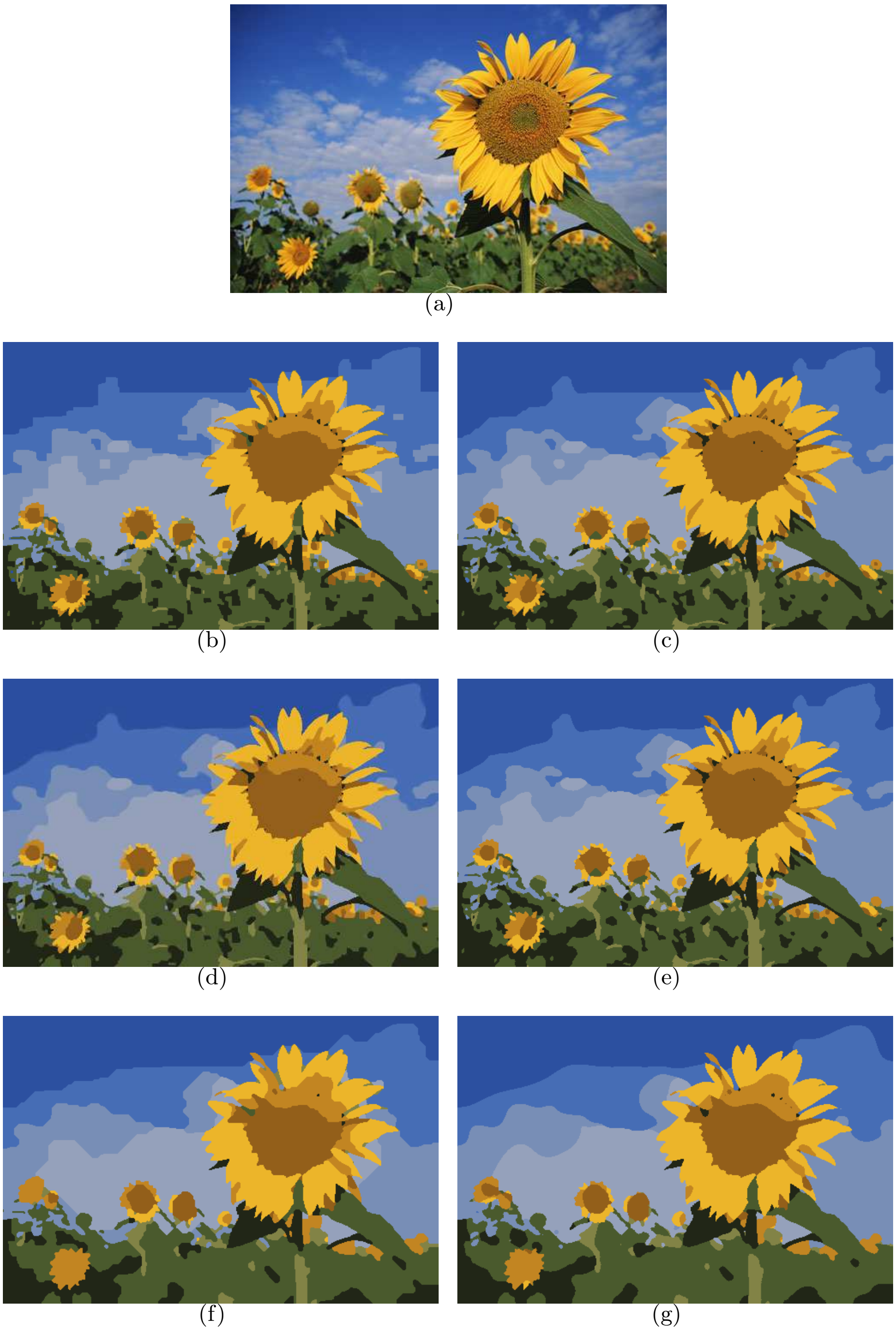


Figure 5.2: (a) Input, (b) alpha expansion 4 neighbors, (c) alpha expansion 8 neighbors, (d) Pock et. al. (e) dual model. (f)-(g): $\alpha = 40$, (f) alpha expansion 8 neighbors, (g) dual model.

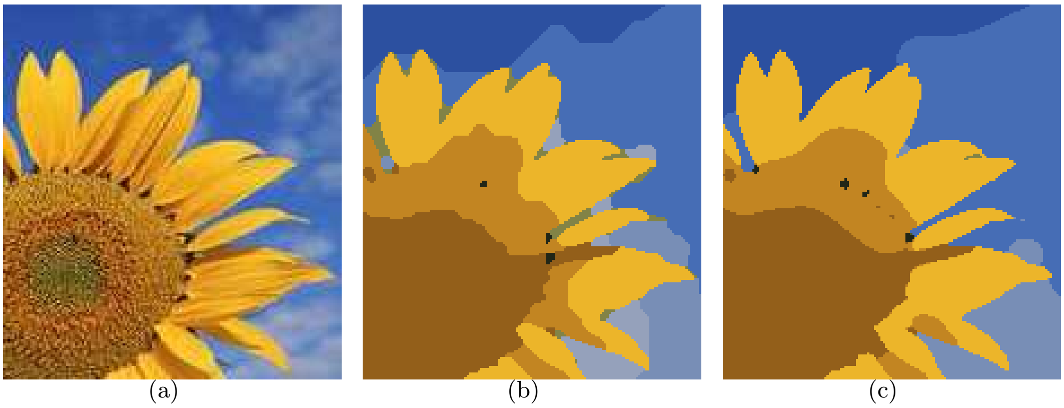


Figure 5.3: Segmentation into 10 regions by using Potts model. (a) Input, (b) result graph cut-based alpha-expansion [30] with "8-neighborhood system", (c) result our approach. The full images and experiment are presented in Fig. 5.2.

Energy plots for all experiments can be found in Figure 5.10. The final energies of the different methods are plotted as a function of the regularization parameter α. Some comparisons are also made to the very recent convex relaxation approach [133] for minimizing the isotropic variant of the energy functional, however an extensive experimental comparison with this approach is out of the scope of this work. The relaxation [133] can be shown to be tighter, but is more computationally complex, especially when the number of labels is large.

In experiments where the correct solution is known, we have also compared the percentage of misclassified pixels, Table 5.1. The regularization parameter α has here been manually selected for each method to minimize the percentage of misclassified pixels. The implementation of the proposed method is made in matlab and the implementations of alpha expansion and alpha-beta swap are made in C++ [30]. The input images in Figure 5.2 and 5.8 was first used by Pock et. al. [132], and the input images in Figure 5.6 and 5.7 was first used by Lellmann et. al. [108].

Qualitative evaluation

In Figure 5.2, it presents the full experiments on the flower image. 10 labels/phases have been used, with color data fidelity

$$\rho(l_i, x) = \sum_{j=1}^{3} |I^j(x) - c_i^j|, \quad i = 1, ..., 10, \tag{5.28}$$

where $\{\mathbf{c}_i\}_{i=1}^{10}$ labeled by l_i, $i = 1 \dots 10$, are predefined color vectors.

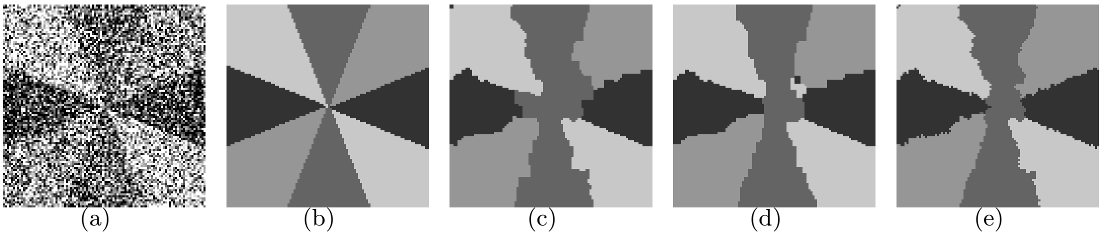


Figure 5.4: Experiment 1: (a) Input, (b) ground truth, (c) alpha expansion, (d) alpha-beta swap, (e) dual model. Size 100×100.

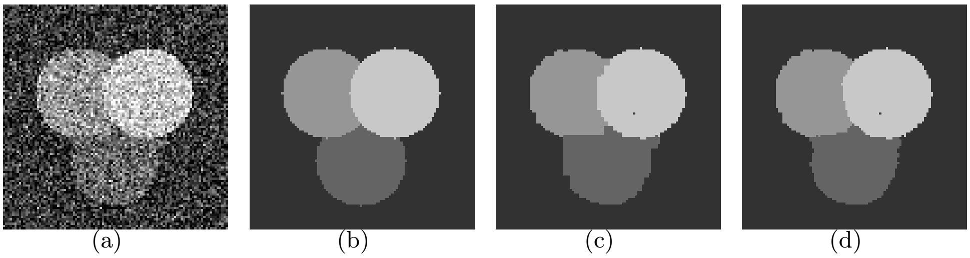


Figure 5.5: Experiment 2: (a) Input, (b) ground truth, (c) alpha expansion, (d) dual model. Size: 100×100.

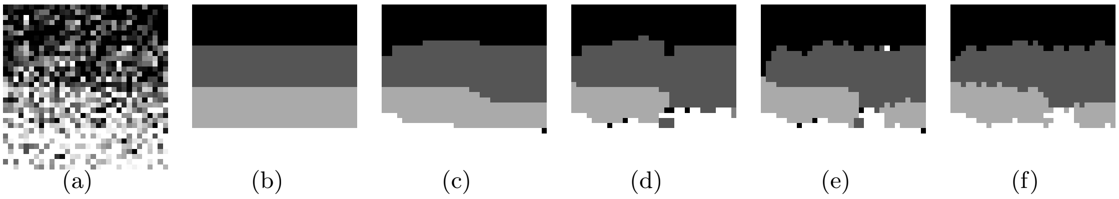


Figure 5.6: Experiment 3: (a) Input, (b) ground truth, (c) alpha expansion, (d) alpha-beta swap, (e) Lellmann et. al., (f) dual model. Size: 32×32.

In Figure 5.2 (b)-(g) a low regularization ($\alpha = 10$) has been chosen. In Figure 5.2 (f)-(e), a higher regularization ($\alpha = 40$) is used. Alpha expansion and alpha beta swap leads to metrication errors, which is particularly visible with 4 neighbors and low regularization (b) and 8 neighbors and high regularization (f). In addition artifacts are introduced as the energy is not minimized exactly, see e.g. the transition between flower and sky. In terms of energy, the smoothed dual model outperforms the graph cut based approaches, see Figure 5.10 (a), especially when α is large. The results in the introduction, Figure 5.3, were generated with the largest α in the energy plot. In Figure 5.2 (c) a comparison with the recent method of Pock et. al. is made. Their method seems to recover almost integer valued solutions up to some blurring of the boundaries.

Some artificial examples are presented next in experiment 1-4, Fig. 5.4 - 5.7. The leftmost gray scale image I is to be classified into 4 classes by using the L^1 norm in the data fidelity term

$$\rho(l_i, x) = |I(x) - c_i|, \quad i = 1, ..., 4, \tag{5.29}$$

where $\{c_i\}_{i=1}^4$ labeled by l_i, $i = 1 \dots 4$, are predefined real values. We observe that in experiment 1,2 and 4 the new method with $s = 0.01$ outperforms alpha expansion and alpha-beta swap implemented with 4 neighbors, both in terms of visual quality and number of misclassified pixels (Table 5.1). In experiment 3, alpha expansion performs best. This is due to the fact that the correct solution only consists of horizontal boundaries, which are favored by the anisotropic 4-neighborhood model. However, the proposed method outperforms alpha-beta swap and the primal model for this example. In experiment 4, where the boundaries are diagonal, the dual model performs best. For energy plots, see Figure 5.10, where we also have used 8 neighborhoods in the discrete models. In terms of energy, our approach performs about equally well as alpha expansion for these two examples. Observe also that our approach can obtain solutions of lower energy than the approaches [184, 108]. This is particularly visible in Figure 5.10(c).

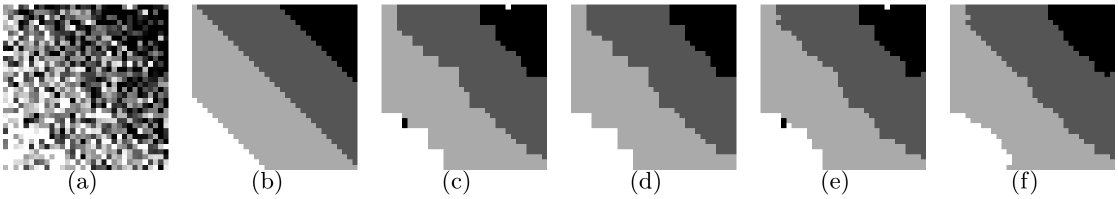


Figure 5.7: Experiment 4: (a) Input, (b) ground truth, (c) alpha expansion, (d) alpha-beta swap, (e) Lellmann et. al., (f) dual model. Size: 32×32.

The advantage of the smoothing is illustrated in the next example, Figure 5.8, where we want to recover a triple junction by filling in the gray area. The data term is given by $\rho(l_i, x) = 0$ for $i = 1, 2, 3$ inside the gray disk, and by the color distance (5.28) outside the gray disk. This is a typically difficult example as the data term is equal for all labels. The global minimum of Potts model will fill in the gray area such that the total length of the boundaries between the labels are minimized, i.e. the boundaries meet with 120 degree angles in the center. In this example we expect that for the non-smooth model $(\rho(l_1, x) + \operatorname{div} p_1^*(x), ..., \rho(l_3, x) + \operatorname{div} p_3^*))$ does not have a unique minimum for some points inside the gray area, which makes it difficult to determine the label at such points. However, for the smooth model a unique minimum can be obtained at each point. The final result is shown in Figure (5.8), where we also compare with other methods. The difference between the components of $(\rho(l_1, x) + \operatorname{div} p_1^s(x), ..., \rho(l_3, x) + \operatorname{div} p_3^s))$ is small near the center of the image, hence it is difficult to verify whether the reconstructed solution is also globally optimal to the original problem, although it can be verified visually, since the global solution is known. As seen in the following subfigures, the approach of Lellmann et. al. does not recover a binary solution. Alpha expansion yields a binary, but incorrect result, Figure (5.8) (d). It can easily be seen geometrically that this is a local optimum, i.e. no alpha expansion move can yield a result of lower energy. We also compare with the convex relaxation of Pock et. al. [132], who first tested their method on this image. As can be seen, they are not able to recover the integer valued global minimum, although they are close for this particular example. After thresholding, they are also able reconstruct the triple junction. Numerical calculations for triple junctions have also been tested in [109] showing that the piecewise constant level set method is able to produce 120 degrees for the junctions.

Figure 5.11 (b) shows the result of 4 class segmentation of a brain MRI image. One would like to classify the input image in Figure 5.11 (a) into the classes: background, cerebrospinal fluid, gray matter and white matter. For this example we have used the Mumford-Shah model with

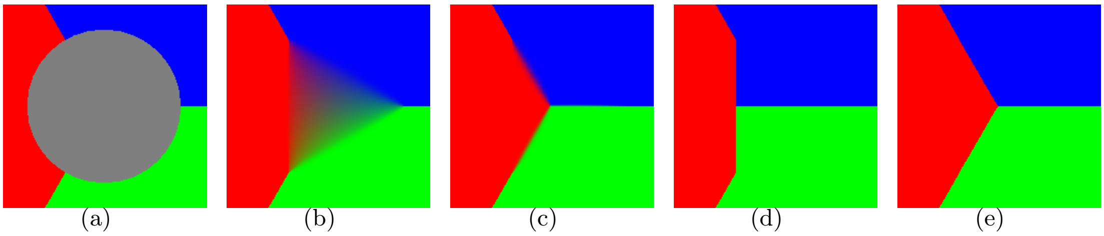


Figure 5.8: (a) Input, (b) Lellmann et. al., (c) Pock et. al., (d) Alpha expansion (e) dual model

L^2 data term

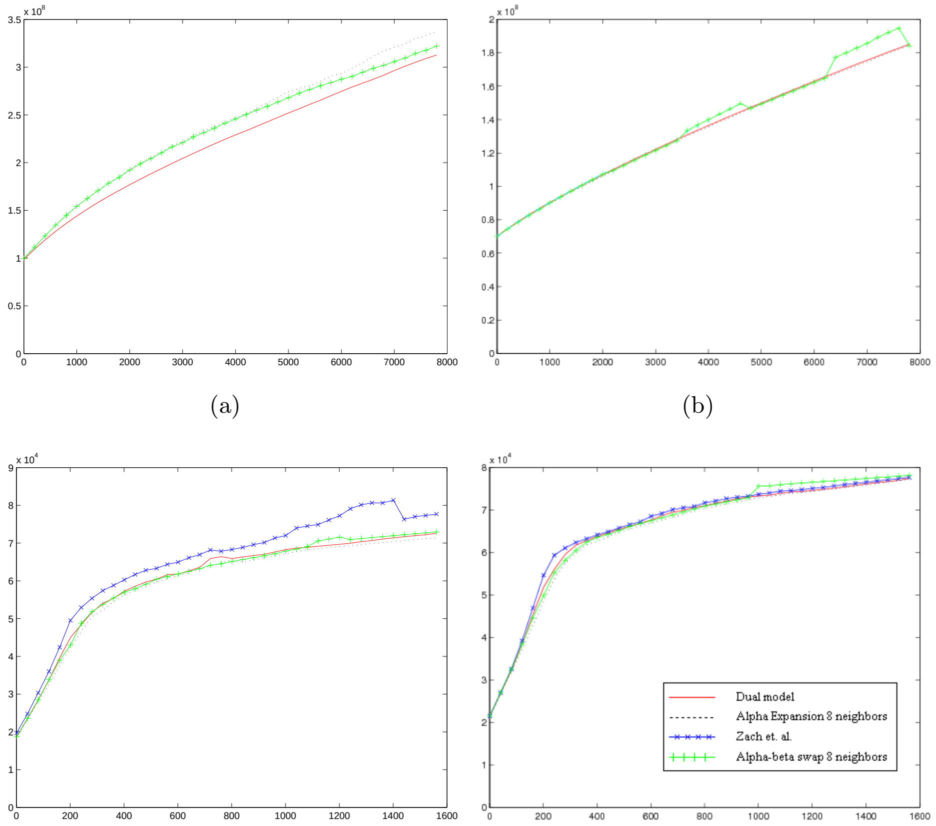
$$\rho(l_i, x) = |I(x) - c_i|^2, \quad i = 1, ..., 4.$$

In order to estimate the optimal constant values $\{c_i\}_{i=1}^n$, we alternate optimization with respect to $\{c_i\}_{i=1}^n$ and the labeling function as described in more details in [11]. This algorithm finds a local minimum with respect the constant values. For energy plots, see Fig. 5.10.

The positive parameter s controls how well the dual model is approximated. The lower s is the better the dual model is approximated. We found that setting $s = 0.01$ or $s = 0.005$ is sufficient and often optimal: setting s lower does not seem to lower the energy of the binary result. This indicates there is a certain benefit of the smoothing in connection with the thresholding scheme. This benefit can also be observed in the energy plots of Figure 5.10: we can obtain binary solutions of lower energy than the approaches of [184, 108].

Table 5.1: Percentage of misclassified pixels for experiment 1-4 (α-expansion and $\alpha - \beta$-swap implemented with 4 neighborhood system)

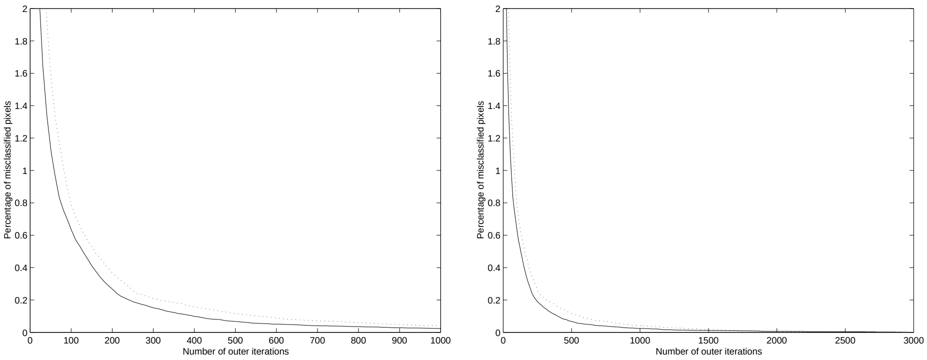
	α-exp	$\alpha - \beta$-swap	Lellmann et. al.	dual
Experiment1	8.89	6.12	-	5.51
Experiment2	1.17	1.17	-	1.06
Experiment3	7.42	15.72	12.30	11.72
Experiment4	6.64	7.23	6.25	5.86



(a) (b)

(c) (d)

Figure 5.9: Energy plot as a function of regularization parameter α of binary solutions obtained by each method. Red: smoothed dual model, black dotted: alpha expansion 8 neighbors, light blue: alpha expansion 4 neighbors, green $+$: alpha-beta swap 8 neighbors, blue x: Zach et. al. [184]. (a) Flower, (b) brain (c)-(d) Experiment 3 and 4. In all experiments the smoothed dual model (red) performs better than or as good as competitive approaches. Fig (a) is a typically difficult example with a large number of labels, where the smoothed dual model clearly performs best.



(a)

(b)

Figure 5.10: Convergence rate for flower image. Solid: percentage of misclassified pixels as a function of the iteration count for the smoothed dual model. Dotted: percentage of misclassified pixels as a function of the *outer* iteration count for the Douglas-Rachford splitting approach.

Evaluation of efficiency and convergence

We will now compare the cpu time and convergence with the approaches [184, 108]. In order to deal with the simplex constraint in the primal optimization problem (5.5), an alternating optimization approach was used in [184] where each iteration includes two substeps (5.7) and (5.8). In [108] a Douglas-Rachford splitting scheme was applied to deal with the simplex constraint, see (5.9) and (5.10). As we see, both these schemes require one outer loop and one inner loop. In contrast, the simplex constraint is inherent in our dual formulation, therefore only one loop is enough. Furthermore, each iteration of this loop has a computational cost approximately equal to one inner loop iteration of [184, 108]. When θ is low, the problem is solved with high accuracy, but more iterations are required. Therefore, one could say θ plays the same role as the smoothing parameter s in our approach. When s is low, the relaxed problem is solved with higher accuracy, but more iterations are required as the time step size δ depends

Table 5.2: Number of iterations to reach exact solution

	Zach et. al.	Douglas Rachford	dual
Experiment 3	1178 ($\times$ 30)	610 ($\times$ 30)	481
Experiment 4	1504 ($\times$ 30)	540 ($\times$ 30)	425

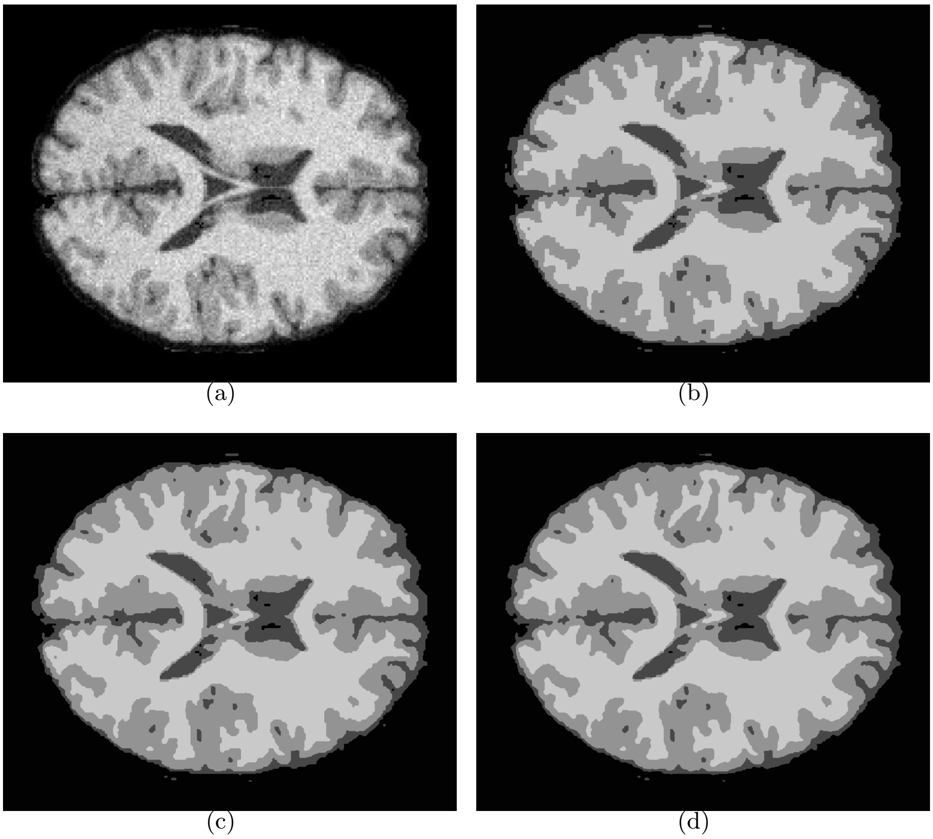


Figure 5.11: (a) Input, (b) alpha expansion 4 neighbors, (c) alpha expansion 8 neighbors, (d) dual model. Size: 709×591.

on s to have stability. Trial and error indicate that this dependency is given by $\delta \leq \frac{1}{2}s$ when images are scaled between 0 and 1. The approaches [108, 184] require more parameters, like the outer time step θ, inner time step, accuracy of solving inner problem etc. We have done our best to optimize these parameters such that the algorithms converge as fast as possible.

Convergence is measured as the number of iterations required to reach the exact thresholded solutions. Since there are finitely many possibilities for such thresholded solutions, converge will occur in a finite number of steps. The exact solutions are determined by running each method for 5000 iterations. The iteration counts for the experiments in Figure 5.6 and 5.7 are shown in Table 5.2. Around 400 iterations are required for the smoothed dual model. The Douglas-Rachford method requires slightly more outer iterations, but also contains an inner loop for each such outer iteration. As indicated in the parenthesis, the inner problems are solved approximately by 30 iterations of Chambolle's algorithm. Overall our approach is therefore significantly faster. The splitting approach (5.7) (5.8) falls even further behind. The flower image in figure 5.10 is larger (508×336), hence such a pixel-wise termination criterion is more strict. Figure 5.10 shows the percentage of incorrect pixels compared to the exact solution, as a function of the iteration count for the dual model (solid) and the outer iteration count of the Douglas-Rachford splitting approach (dotted). In this example the dual model also outperforms the Douglas Rachford scheme in terms of outer iterations. After 70 iterations, more than 99% of the pixels have reached steady state. After 300 iterations more than 99.8% have reached steady state.

Comparison with graph cut based alpha expansion and alpha beta swap is more difficult. We have used the highly optimized c++ implementation [30, 28] of these methods, while the algorithm for the dual model is implemented in a simple matlab program. There has recently been much effort on comparing continuous and discrete (graph cut) techniques in computer vision, see e.g. [93] for an extensive discussion about two phase partitioning problems. Continuous convex optimization techniques consist mainly of floating point matrix/vector arithmetic, which is highly suited for massive parallel implementation on GPU. In contrast, combinatorial max-flow algorithms have a much more serial nature. In this regard, continuous convex optimization has been shown to be faster [93]. This is especially evident in 3D. The development of processor technology is expected to be largely of the parallel aspect in the future. Hence we see our work as more suitable for current and future generations of hardware. Unfortunately, we don't have the resources to implement our method in such a parallel manner on GPU. We can mention that Zach et. al. implemented their algorithm on GPU and thereby claimed to beat the graph cut based approaches in terms of efficiency by a factor of 30. The cpu times for our simple matlab implementation are as follows: For the 709×591 brain image convergence was reached in 1 minutes and 32 seconds for our implementation. For the 32×32 images in Figure 5.6 and 5.7 convergence averaged around 2.5 seconds. For the 100×100 images in Figure 5.4 and 5.5 convergence took 10.21 and 4.68 seconds respectively. Due to the extreme amount of noise on these small images the regularization parameter must be set very high, which increases cpu time compared to images of the same size with lower noise level.

5.3 Continuous Max-Flow Approach

In this section, we propose the new continuous max-flow formulation and prove it is equivalent to the convex relaxed Potts model (5.5) in terms of primal and dual. Similar as introduced

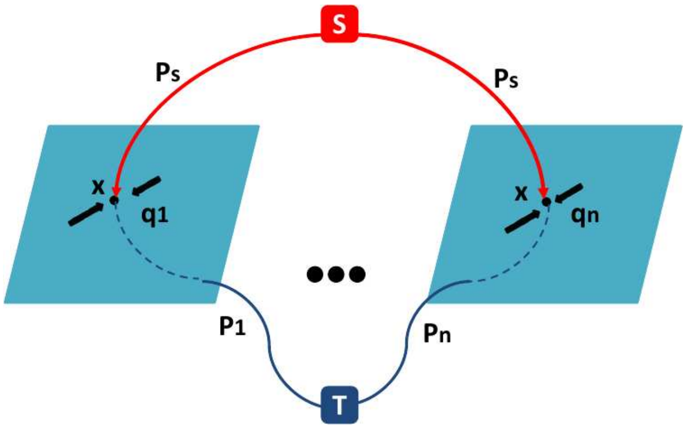


Figure 5.12: Continuous configuration of max-flow with n labels.

in the continuous max-flow and min-cut models (see Chapter. 4), we can also provide a new variational perspective of 'flows' and 'cuts'.

Comparing to the entropy-maximization method introduced in the previous section, the new continuous max-flow approach solves the convex relaxed Potts model (5.5) exactly in stead of its smoothed version.

5.3.1 Continuous Max-Flow Model

Spatially Continuous Configurations

We propose the continuous configuration of the new max-flow model with n labels, see Fig. ??:

1. n copies Ω_i, $i = 1, \ldots, n$, of the image domain Ω are given in parallel;
2. For each position $x \in \Omega$, the same source flow $p_s(x)$ tries to stream from the source s to x at each copy Ω_i, $i = 1, \ldots, n$, simultaneously, i.e. the source flow field $p_s(x)$ is the same for each Ω_i, $i = 1, \ldots, n$;
3. For each position $x \in \Omega$, the sink flow $p_i(x)$ is directed from x at each copy Ω_i, $i = 1, \ldots, n$, of Ω to the sink t. The sink flow fields $p_i(x)$, $i = 1, \ldots, n$, may be different to each other;
4. The spatial flow field $q_i(x)$, $i = 1, \ldots, n$, is defined within each Ω_i, $i = 1, \ldots, n$.

Continuous Max-Flow Model

Under such a contiuous setting, we define the constraints of flows $p_i(x)$ and $q_i(x)$, at $x \in \Omega$, as follows:

$$|q_i(x)| \,\leq\, C_i(x)\,, \;\; p_i(x) \,\leq\, \rho(l_i,x)\,, \quad i = 1 \ldots n\,; \tag{5.30}$$

$$\big(\operatorname{div} q_i - p_s + p_i\big)(x) \,=\, 0\,, \quad i = 1 \ldots n\,. \tag{5.31}$$

Note: there is no constraint for the source flow $p_s(x)$. (5.30) gives the flow capacity constraints. The equality constraint (5.31) is anologous to the flow conservation condition of graph cuts such that $\operatorname{div} q_i(x)$ corresponds to the excess of the spatial flow at each image pixel x of Ω_i, $i = 1 \ldots n$.

We formulate the continuous max-flow model, over all flows $p_s(x)$, $p(x) := (p_1(x), \ldots, p_n(x))$ and $q(x) := (q_1(x), \ldots, q_n(x))$, as follows:

$$\max_{p_s,p,q} \; \big\{ P_{\mathrm{ML}}(p_s,p,q) \,:=\, \int_\Omega p_s(x)\,dx \big\} \tag{5.32}$$

subject to (5.30) and (5.31).

Preliminary Remarks and Connections

The continuous max-flow model (5.32) tries to perform the maximization of the total source flow $p_s(x)$ over the whole image domain Ω subject to certain flow capacity and conservation conditions. It is easy to notice that, in view of the flow conservation condition (5.31), at the same place x of each Ω_i, $i = 1 \ldots n$, we have

$$p_s(x) \,=\, \operatorname{div} q_i(x) + p_i(x)\,, \quad i = 1 \ldots n\,. \tag{5.33}$$

Observe the righthand of (5.33) and the flow capacity constraints given in (5.30), $p_s(x)$ is thus constrained and should be consistent to all the n flow configurations of $\operatorname{div} q_i(x) + p_i(x)$, $i = 1 \ldots n$, at x. It naturally boils down to

$$p_s(x) \,=\, \min\big(\operatorname{div} q_1(x) + p_1(x), \ldots, \operatorname{div} q_n(x) + p_n(x)\big)\,, \quad \forall x \in \Omega\,. \tag{5.34}$$

In this regard, we can prove

Proposition 25. *The proposed continuous max-flow model* (5.32) *is equivalent to*

$$\max_{|q_i(x)| \leq C_i(x)} \; \int_\Omega \big\{ \min\big(\rho(l_1,x) + \operatorname{div} q_1(x), \ldots, \rho(l_n,x) + \operatorname{div} q_n(x)\big) \big\}\,dx\,. \tag{5.35}$$

Proof. Following the result (5.34), the continuous max-flow model (5.32) can be equally reformulated by

$$\max_{p(x),q(x)} \int_\Omega \big\{ \min\big(p_1(x) + \operatorname{div} q_1(x), \dots, p_n(x) + \operatorname{div} q_n(x)\big) \big\}\, dx$$

subject to the flow capacity constraints (5.30).

Given $u(x) := (u_1(x), \dots, u_n(x)) \in S$ where S denotes the piecewise simplex constraint (5.6), the above formulation can then be rewritten as

$$\max_{p(x),q(x)} \min_{u(x)} \int_\Omega \Big\{ \sum_{i=1}^n u_i(x)\, \big(p_i(x) + \operatorname{div} q_i(x)\big) \Big\}\, dx\,. \tag{5.36}$$

Then it is easy to see that the maximization over $p_i(x) \leq \rho(l_i, x)$, $i = 1 \dots n$, is consistent to the constraint $u(x) \in S$. By simple variation computations over $p(x)$ and $u(x)$, (5.36) just amounts to (5.35). □

The result (5.35) of Prop. (25) simply discovers the nonsmooth dual model (5.16) introduced in Sec. 5.2, when $C_i(x) = \alpha$ are constant.

In addition, observing the conclusion (5.34), we can regard each image copy Ω_i, $i = 1 \dots n$, together with the constrained sink flow field $p_i(x)$ and the spatial flow field $q_i(x)$ given in (5.30), as a 'filter' F_i whose filtering capacity at $x \in \Omega$ is constrained by $\operatorname{div} q_i(x) + p_i(x)$ along with (5.30), i.e. the passing source flow $p_s(x)$ at each x does not overflow the minimal flow allowed by the n 'filter' configurations. Then one can explain the continuous max-flow model (5.32) such that all the filters F_i, $i = 1, \dots, n$, are layered one by one and the source flow $p_s(x)$ tries to pass such a stack of such 'filters' in one time. It is obvious that $p_s(x)$ is bottlenecked by the minimum capacity of $\operatorname{div} q_i(x) + p_i(x)$, $i = 1 \dots n$. In such a 'filter' configuration, (5.32) aims to maximize the total flow passing through this 'filter' set.

In the following section, we introduce the equivalent representations of the continuous max-flow formulation (5.32) and show its equivalent dual formulation just amounts to the convex relaxed Potts model (5.5) when $C_i(x) = \alpha$ are constant.

5.3.2 Equivalent Primal-Dual Formulation

By the introdution of the multiplier functions $u_i(x)$, $i = 1 \dots n$, to the n flow conservation equalities (5.31), then we have the equivalent primal-dual model of (5.32) as follows:

$$\max_{p_s,p,q} \min_u \Big\{ E(p_s,p,q;u) \, := \, \int_\Omega p_s \, dx \, + \, \sum_{i=1}^n \int_\Omega u_i(\operatorname{div} q_i - p_s + p_i) \, dx \Big\} \tag{5.37}$$
$$\text{s.t.} \quad p_i(x) \, \leq \, \rho(\ell_i, x) \, , \quad |q_i(x)| \, \leq \, C_i(x) \, ; \quad i = 1 \dots n$$

where $u(x) := (u_1(x), \dots, u_n(x)) \in \mathbb{R}^n$ for $\forall x \in \Omega$.

Rearranging the energy function $E(p_s,p,q;u)$ of (5.37), we have

$$E(p_s,p,q;u) \, = \int_\Omega \big\{ (1 - \sum_{i=1}^n u_i) \, p_s \, + \, \sum_{i=1}^n u_i \, p_i \, + \, \sum_{i=1}^n u_i \, \operatorname{div} q_i \big\} \, dx \tag{5.38}$$

For the primal-dual model (5.37), the conditions of the minimax theorem (see e.g., [58] Chapter 6, Proposition 2.4) are all satisfied. That is, the constraints of flows are convex, and the energy function is linear in both the multiplier u and the flow functions p_s, p and q, hence convex l.s.c. for fixed u and concave u.s.c. for fixed p_s, p and q. This confirms the existence of at least one saddle point, see [58, 62]. It also follows that the min and max operators of the primal-dual model (5.37) can be interchanged, i.e.

$$\max_{p_s,p,q} \big\{ \min_u \, E(p_s,p,q;u) \big\} \, = \, \min_u \big\{ \max_{p_s,p,q} \, E(p_s,p,q;u) \big\} \, . \tag{5.39}$$

5.3.3 Equivalent Dual Formulation

Now we investigate the optimization of (5.37) by the min-max order as the righthand side of (5.39), i.e. we first maximize $E(p_s,p,q;u)$ over the flow functions p_s, p and q then minimize over the multiplier function u. We show that this leads to the equivalent dual model of the continuous max-flow formulation (5.32), i.e.

$$\min_u \, \big\{ D(u) \, := \sum_{i=1}^n \big(\int_\Omega u_i(x) \, \rho(\ell_i, x) \, dx \, + \, \int_\Omega C_i(x) \, |\nabla u_i| \, dx \big) \big\} \tag{5.40}$$
$$\text{s.t.} \quad \sum_{i=1}^n u_i(x) \, = \, 1 \, , \quad u_i(x) \, \geq \, 0 \, .$$

Optimization of Flow Functions p, q and p_s

In order to optimize the flow function $p(x)$ in (5.38), let us consider the maximization problem (4.21), i.e.

$$f(q) \,=\, \max_{p \le C} \, p \cdot q \,. \tag{5.41}$$

where p, q and C are scalars.

By virtue of (4.22), we can equally express $f(q)$ by

$$f(q) \,=\, q \cdot C \,, \quad q \geq 0 \,. \tag{5.42}$$

Observe (4.21) and (4.23), we can maximize $E(p_s, p, q; u)$ of (5.38) over the sink flows $p_i(x)$, $i = 1 \dots n$, then we have

$$\max_{p_i(x) \le \rho(l_i, x)} \int_\Omega u_i p_i \, dx \,=\, \int_\Omega u_i(x) \rho(l_i, x) \, dx \,, \quad u_i(x) \,\geq\, 0 \,,\; i = 1, \dots, n \,. \tag{5.43}$$

For the maximization over the spatial flow functions $q_i(x)$, $i = 1, \dots, n$, it is well-known [68] that

$$\max_{|q_i(x)| \le C_i(x)} \int_\Omega u_i \operatorname{div} q_i \, dx \,=\, \int_\Omega C_i(x) \, |\nabla u_i| \; dx \,. \tag{5.44}$$

Furthermore, observe the source flow function $p_s(x)$ is unconstrained, the maximization of (5.38) over p_s simply leads to

$$1 - \sum_{i=1}^n u_i(x) \,=\, 0 \,, \quad \forall x \in \Omega \,. \tag{5.45}$$

By the results of (5.45), (5.43) and (5.44), it is easy to conclude that the maximization of the primal-dual model (5.38) over flow functions p_s, p and q boils down to its equivalent dual model (5.40). Therefore, we have

Proposition 26. *The continuous max-flow model* (5.32)*, the primal-dual model* (5.37) *and the dual model* (5.40) *are equivalent to each other, i.e.*

$$(5.32) \iff (5.37) \iff (5.40) \,.$$

The proof of Prop. 26 follows the above statements.

In this work, we focus on the case when $C_i(x) = \alpha$, $\forall x \in \Omega$ and $i = 1, \dots, n$. Obviously,

Proposition 27. *When $C_i(x) = \alpha$, $\forall x \in \Omega$ and $i = 1 \dots n$, the dual model* (5.40) *amounts to the* convex relaxed Potts model (5.5). *Hence, in this case,*

$$\textit{continuous max-flow model } (5.32) \iff \textit{convex relaxed Potts model } (5.5)\,.$$

Its proof simply follows by Prop. 26, which is omitted here.

5.3.4 Variational Perspective of Flows and Cuts

Through the above analytical procedures, we can build up a variational perspective of flows and cuts, which recovers conceptions and terminologies used in graph-cuts.

Consider the maximization problem (4.21), for any fixed q, let some optimal p^* maximize $q \cdot p$ over $p \leq C$. By means of variations, if such $p^* < C$ strictly, its variation directly leads to $q = 0$ since the variation δp can be both negative and positive. On the other hand, for $p^* = C$, its variation under the constraint $p \leq C$ gives $\delta p < 0$, then we must have $q > 0$.

In terms of graph-cuts, some maximum flow $p^*(e) < C(e)$, over the edge $e \in \mathcal{E}$, just means the considered flow $p(e)$ does not reach its maximum or capacity $C(e)$ along the edge e, i.e. 'unsaturated'; which results the so-called 'cut' over the edge e.

In the same manner, for the maximum sink flow function $p_i^*(x)$, $i = 1 \dots n$, it is easy to see that when the flow $p_i^*(x) < \rho(l_i, x)$ at some $x \in \Omega$, i.e. 'unsaturated', we must have $u_i(x) = 0$, i.e. $u_i(x) p_i^*(x) = 0$. This means that at the position x, the flow $p_i(x)$ has no contribution to the energy function and the flow $p_i(x)$, from $x \in \Omega_i$ to the sink t, can be 'cut' off from the energy function of (5.37). On the other hand, the indicator function $u_i(x) = 0$ definitely means the position x is not labeled as l_i.

For the spatial flows $q_i^*(x)$, $i = 1 \dots n$, let

$$C_{\mathrm{TV}}^i := \{ q_i(x) \mid |q_i(x)| \leq C_i(x)\,,\, n \cdot q_i|_{\partial\Omega} = 0 \}\,.$$

Observe that

$$\sup_{q_i \in C_{\mathrm{TV}}^i} \int_\Omega u_i(x)\, \mathrm{div}\, q_i(x)\, dx \;=\; \sup_{p \in C_{\mathrm{TV}}^\alpha} \int_\Omega q_i(x)\, \nabla u_i(x)\, dx\,, \tag{5.46}$$

the extremum of the righthand in (5.46) just indicates the normal cone-based condition [79] of $\nabla u_i^*(x)$, i.e.

$$\nabla u_i^* \;\in\; N_{C_{\mathrm{TV}}^i}(q_i^*)\,, \tag{5.47}$$

for $i = 1 \dots n$.

Then we simply have:

$$\text{if} \quad \nabla u_i^*(x) \quad \neq 0\,, \qquad \text{then} \quad |q_i^*(x)| = C_i(x)\;, \tag{5.48a}$$

$$\text{if} \quad |q_i^*(x)| < C_i(x)\,, \qquad \text{then} \quad \nabla u_i^*(x) \quad = 0\,. \tag{5.48b}$$

In other words, at some locations $x \in \Omega$ where $\nabla u_i^*(x) \neq 0$, the spatial flow $q_i^*(x)$ is 'saturated' (5.48a), i.e. $|q_i^*(x)| = C_i(x)$; at some locations $x \in \Omega$ where $|q_i^*(x)| < \alpha$ is not saturated, we must have $\nabla u_i^*(x) = 0$, i.e. no variances of $u_i^*(x)$ around x, and therefore the 'cut' does not appear around the spatial domain at x.

5.3.5 Multiplier-Based Max-Flow Algorithm

Observe that the energy function of the primal-dual model (5.37) just gives the Lagrangian function of the continuous max-flow model (5.32) where $u_i(x)$, $i = 1 \dots n$, are the corresponding multiplier functions to the flow conservation equalities (5.31). Now we introduce our multiplier-based max-flow algorithm, which is based on the augmented lagrangian method [21, 20, 138]. In this respect, we define the augmented Lagrangian function

$$L_c(p_s, p, q, u) \;=\; \int_\Omega p_s\,dx \;+\; \sum_{i=1}^n \langle u_i, \operatorname{div} q_i - p_s + p_i \rangle \;-\; \frac{c}{2} \sum_{i=1}^n \|\operatorname{div} q_i - p_s + p_i\|^2$$

where $c > 0$.

By the standard augmented Lagrangian method, each iteration of the algorithm can then be generalized as follows:

- Optimize spatial flows q_i, $i = 1 \dots n$, by fixing other variables:

$$q_i^{k+1} := \arg \max_{\|q_i\|_\infty \leq \alpha} \; -\frac{c}{2} \left\| \operatorname{div} q_i + p_i^k - p_s^k - u_i^k/c \right\|^2 \;, \tag{5.49}$$

  which can be solved by Chambolle's projection algorithm [39].

- Optimize sink flows p_i, $i = 1...n$, by fixing other variables

$$p_i^{k+1} := \arg \max_{p_i(x) \leq \rho(l_i, x)} \; -\frac{c}{2} \left\| p_i + \operatorname{div} q_i^{k+1} - p_s^k - u_i^k/c \right\|^2 \;, \tag{5.50}$$

  which can be computed at each $x \in \Omega$ in a closed form.

- Optimize the source flow p_s and update multipliers u_i, $i = 1 \dots n$

$$p_s^{k+1} := \arg\max_{p_s} \int_\Omega p_s\,dx - \frac{c}{2} \sum_{i=1}^n \left\| p_s - (p_i^{k+1} + \operatorname{div} q_i^{k+1}) + u_i^k/c \right\|^2 \;, \tag{5.51}$$

$$u_i^{k+1} = u_i^k - c\,(\operatorname{div} q_i^{k+1} - p_s^{k+1} + p_i^{k+1})\,. \tag{5.52}$$

Both (5.51) and (5.52) can be obtained in the closed form.

Consider the above numerical steps, it is easy to see that the two flows q_i and p_i, $i = 1 \ldots n$, computed by (5.49) and (5.50) can be handled independently for each label i. Hence, (5.49) and (5.50) can be implemented in a parallel way. Once such two steps are finished, the source flow $p_s(x)$ and the labeling functions $u_i(x)$, $i = 1 \ldots n$, are updated. Obviously, such parallelism naturally originates the configuration shown in Fig. 5.12.

Fast Linearized Max-Flow Based Algorithm

Actually, the sub-step (5.49) at each iteration can be solved in an inexact manner, i.e. without solving the Chambolle-projection exactly which is time-consuming. Now, we consider the minimizaion problem

$$q_i^{k+1}(x) := \arg \min_{|q_i(x)| \leq \alpha} \left\| \operatorname{div} q_i - D_i^k \right\|^2 \tag{5.53}$$

where $D_i^k(x) = (p_s^k + u_i^k/c - p_i^k)(x)$ for $i = 1 \ldots n$. We apply a linearized solver which just performs a simple projection-gradient step to the proposed problem (5.53) such that

$$q_i^{k+1} := \operatorname{Proj}_{|q_i(x)| \leq \alpha} \Big(q_i^k - \gamma \nabla (\operatorname{div} q_i^k - D_i^k) \Big) \tag{5.54}$$

where γ is the step-size and its maximum value depends on the largest eigen-value of the discretized $\nabla \operatorname{div}$, i.e. the matrix $\overline{\nabla^h} \operatorname{div}^h$ after discretization. In this work, we apply the mimetic finite-difference method [83, 82] over the regular image grid and its detailed implementation of (5.54) can be found in the Appendix 7.2. The largest eigen-value of the resulted matrix $\overline{\nabla^h} \operatorname{div}^h$ is just $1/8$. Hence we apply $\gamma \leq 1/4$ in the following experiments, i.e. two times of the largest eigen-value, in order to construct a nonexpansive operator for the proposed iterative updating step (5.55) of $q_i(x)$, $i = 1 \ldots n$. Similar linearized solver appeared in the recent study of the Bregman-Splitting algorithm [72], which results in a fast solver to the continuous min-cut problem.

In this regard, we propose the fast linearized max-flow based algorithm as Alg. 6, where every substep at each iteration only performs one simple computation.

5.3.6 Experiments

In this section, we first make experiments to validate the proposed continuous max-flow algorithm, i.e. Alg. 6, for its associated parameter settings and convergence. We then show its

Algorithm 6 Multiplier-Based Maximal Potts Flow Algorithm

Set the starting values $p_s^1(x)$, $p^1(x)$, $q^1(x)$ and $u^1(x)$, let $k = 1$ and start k-th iteration, which includes the following steps, till convergence:

- Update q_i, $i = 1 \ldots n$, by fixing other variables

$$q_i^{k+1} = \mathrm{Proj}_{|q_i(x)| \leq \alpha}\Big(q_i^k - \gamma \nabla (\operatorname{div} q_i^k - D_i^k)\Big)\,, \tag{5.55}$$

  where $D_i^k(x) = (p_s^k + u_i^k/c - p_i^k)(x)$ for $i = 1 \ldots n$.

- Update p_i, $i = 1 \ldots n$, by solving the substep (5.50) which results in

$$p_i^{k+1}(x) \; = \; \min\big(\rho(l_i, x), F_i^k(x)\big)$$

  where $F_i^k(x) = (p_s^k + u_i^k/c - \operatorname{div} q_i^{k+1})(x)$ for for $i = 1 \ldots n$;

- Update p_s by solving the substep (5.51)

$$p_s^{k+1}(x) \; = \; (1 + c\sum_{i=1}^n G_i^k(x))/n\,c\,,$$

  where $G_i^k(x) = (p_i^{k+1} + \operatorname{div} q_i^{k+1} - u_i^k)(x)/c$ for $i = 1 \ldots n$.

- Update multipliers u_i, $i = 1, \ldots, n$, by

$$u_i^{k+1} \; = \; u_i^k - c\,(\operatorname{div} q_i^{k+1} - p_s^{k+1} + p_i^{k+1})\,;$$

- Let $k = k + 1$ return to the $k + 1$ iteration till converge.

significant outperformance over other state of art convex optimization approaches. In comparison to graph-cuts, e.g. alpha-expansion, the studied convex relaxation model comes with the important advantage of rotational invariance, which means that metrication errors are properly avoided. The quality of the relaxation approach (5.5) has been evaluated extensively in [184, 108, 13] where its outperformance over the state of art methods from discrete optimization, e.g. alpha expansion and alpha-beta swap [30] has been shown for effectively minimizing the Pott's energy. All the experiments are made on a Linux desktop with AMD Athlon 64x2 5200+ and 3 Gb memory.

Algorithm Validations

For the proposed Alg. 6, there are two parameters: the step-size γ and the augmented parameter c. In view of (5.52), the update of the labeling function $u_i(x)$, $i = 1 \ldots n$, at each iteration gives us an appropriate criterion of convergence

$$\epsilon_a = \frac{\sum_{i=1}^n \left| c\,(\operatorname{div} q_i^{k+1} - p_s^{k+1} + p_i^{k+1}) \right|}{n\ |\Omega|}$$

which evaluates the avarage change of the labeling function for each pixel x and each label. In the following experiments, we apply ϵ_a to be less than some small positive value as the convergence criterion.

We make labeling experiments in this part with the same input image (see Fig. 5.13(a) and Fig. 5.13(b) for the input and ground-truth images). Four data terms $\rho(l_i, x) = |I(x) - l_i|^p$, $i = 1 \ldots 4$, are used, where $I(x)$ and l_i, $i = 1 \ldots 4$, take the triple RGB values and $p = 1$.

In this respect, we make labeling experiments of the input image (see Fig. 5.13(a) and Fig. 5.13(b) for the input and ground-truth images) together with various step-size $\gamma = 0.1, 0.11, \ldots, 0.18$ and the fixed augmented parameter $c = 0.25$. We set $\epsilon_a < 5 \times 10^{-4}$ as the stopping criterion. When $\gamma > 0.18$, the proposed algorithm fails to achieve convergence within 300 iterations and more than one updating of q_i, $i = 1 \ldots n$, for each iteration are required to obtain a faster convergence. Tab. 5.3 list detailed results in terms of the total number of iterations and computation time. Fig. 5.13(d) shows their respective convergence graph with log-log illustration. Clearly, when the step-size γ takes values between 0.1 and 0.17, Alg. 6 performs very similarly and obtains convergence within about 35 iterations.

To evaluate Alg. 6 with various settings of the augmented parameter c, we make experiments for the same input along with $c = 0.1$, 0.25, 0.4, 0.55, 0.7, 0.85, 1, 2, 3 and the fixed step-size $\gamma = 0.17$. We set $\epsilon_a < 5 \times 10^{-4}$ as the stopping criterion. Tab. 5.4 list detailed results in terms of the total number of iterations and computation time. Fig. 5.13(e) shows their respective convergence graph with log-log illustration. Clearly, when the augmented parameter c takes values between 0.1 and 0.85, Alg. 6 converges relatively faster and obtains convergence within 40 iterations. In Fig. 5.13(e), the bolded black line shows the convergence result (fastest) of $c = 0.25$ and the bolded black line shows the convergence result (slowest) of $c = 3$.

Table 5.3: Validation for the step-size γ when $c = 2.5$

γ	0.10	0.11	0.12	0.13	0.14	0.15	0.16	0.17	0.18
Iter.	36	35	35	34	34	34	34	34	195
Time (sec.)	5.56	5.03	4.72	4.30	4.20	4.56	4.10	4.33	26.28

Table 5.4: Validation for the augmented parameter c when $\beta = 0.18$

c	0.1	0.25	0.4	0.55	0.7	0.85	1	2	3
Iter.	37	34	34	38	39	40	42	63	83
Time (sec.)	5.97	5.50	5.88	6.11	6.57	6.74	6.87	9.64	13.39

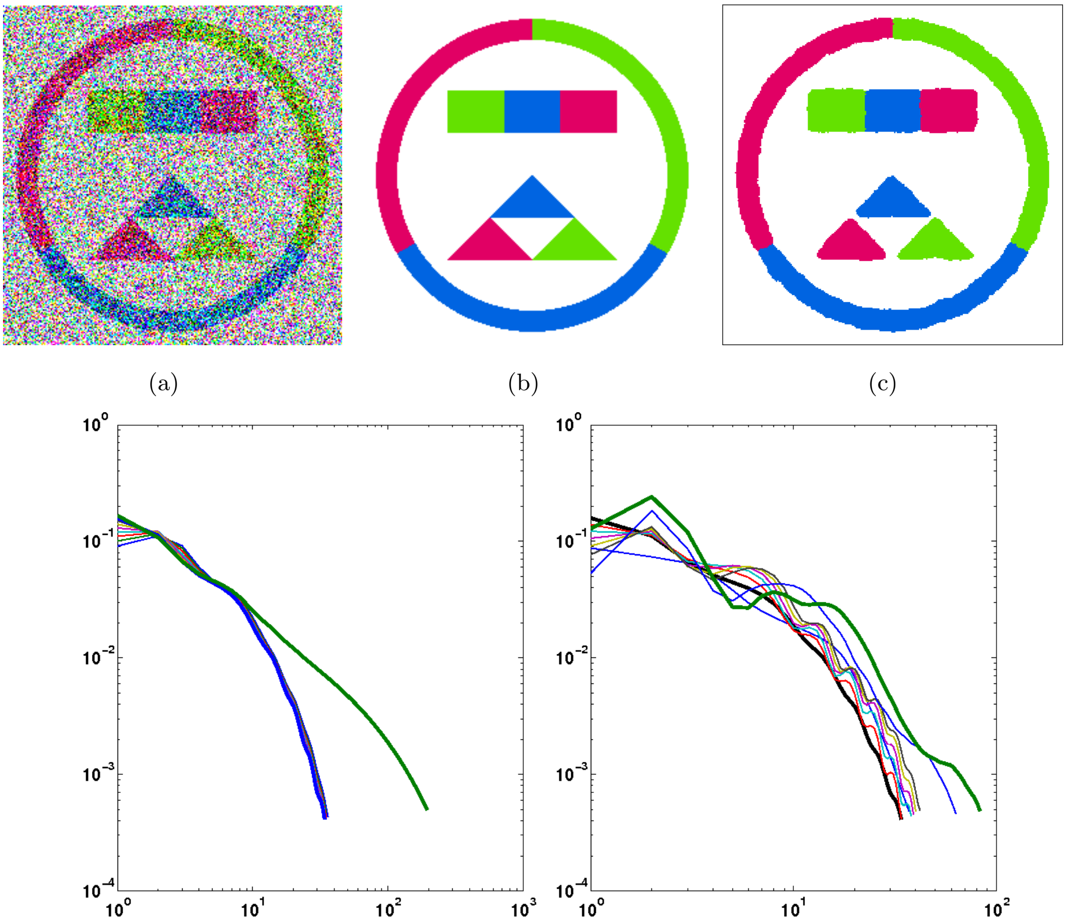
5.3.7 Comparisons to Other Approaches

Examples are given in Figure (5.14), where we have used the Mumford-Shah data term $\rho(\ell_i, x) = |I(x) - \ell_i|^2$, $i = 1, ..., n$. As we see, equally good solutions as alpha expansion are produced, but without the metrication artifacts.

In contrast to the minimization approach of Zach et. al. [184], the proposed algorithm can be proved to converge by classical optimization theories. The Douglas-Rachford splitting approach given in [108] can also be proved to converge (in the discrete setting), but we experienced that our approach was more efficient than both these approaches. The inner problem has the same complexity for all approaches, since it is dominated by the process of iteratively solve a tv minimization problem. However, in contrast to [184, 108] our approach avoids iterative projections to the convex set S and consequently require much less outer iterations. Convergence is reached for a wide range of the outer 'step-size' c. To measure converge, we find a good estimate of the final energy E^* by solving the problem with 10000 outer iterations. The energy precision at iteration k is then measured by $\epsilon = \frac{E^k - E^*}{E^*}$. For the three images (see Fig. 2), different precision ϵ are taken and the total number of iterations to reach convergence is evaluated, see Tab 5.5: clearly, our method is about 4 times faster than the Douglas-Rachford-splitting [108], the approach in [184] is even slower and failed to reach such a low precision.

	Brain $\epsilon \leq 10^{-5}$	Flower $\epsilon \leq 10^{-4}$	Bear $\epsilon \leq 10^{-4}$
Zach et al [184]	fail to reach such a precision		
Lellmann et al [108]	421 iter.	580 iter.	535 iter.
Proposed algorithm	88 **iter.**	147 **iter.**	133 **iter.**

Table 5.5: Comparisons between algorithms: Zach et al [184], Lellmann [108] and the proposed max-flow algorithm: for the three images (see Fig. 2), different precision ϵ are taken and the total number of iterations to reach convergence is evaluated.



(a) (b) (c)

(d)

Figure 5.13: (a) Input Image, (b) Ground-truth Image (c) Computation Result $u(x)$ with $\gamma = 0.17$ and $c = 0.55$, (d) Convergence results associated to different step-size settings of γ, (e) Convergence results associated to different settings of the augmented parameter c.

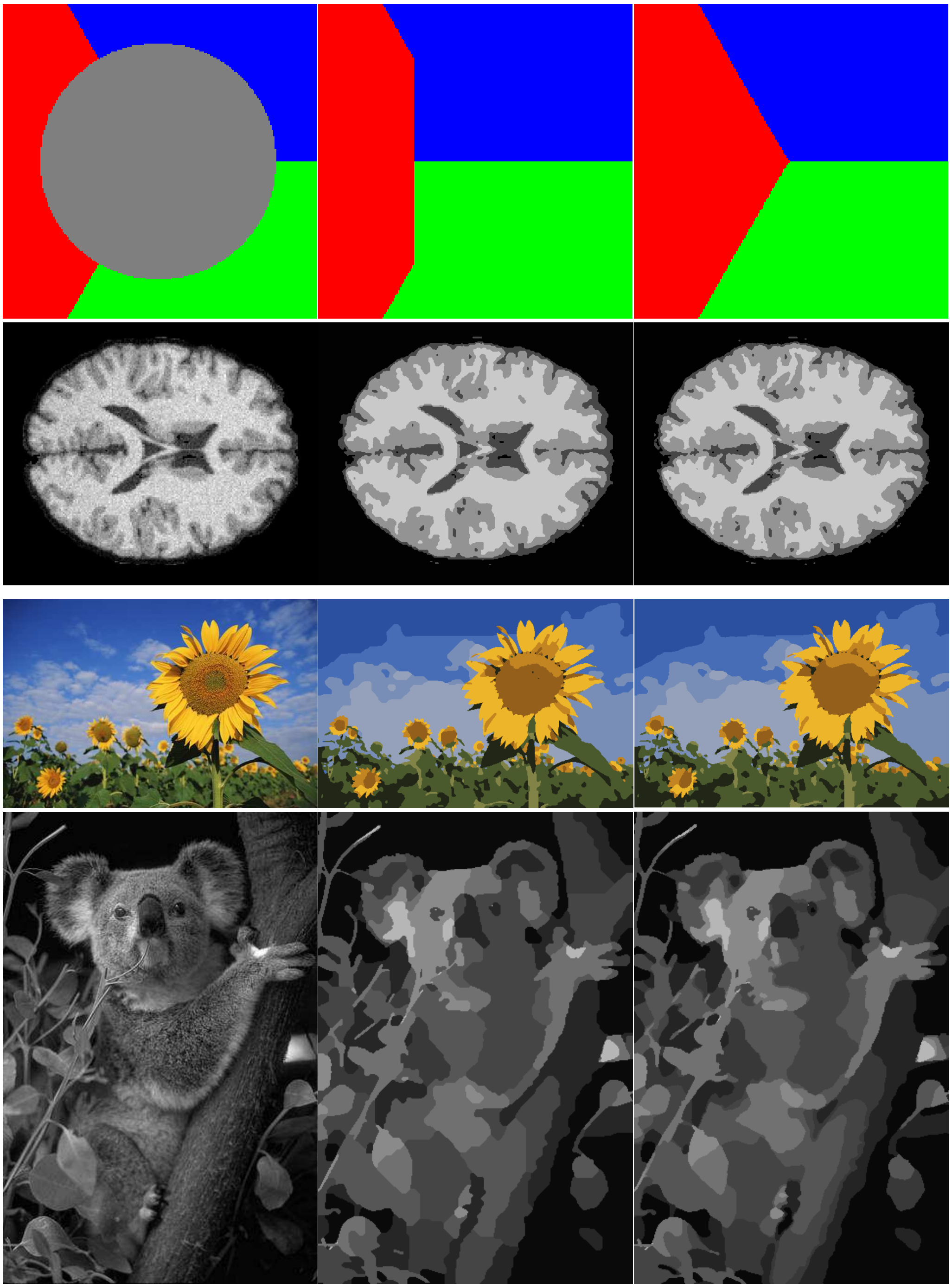


Figure 5.14: **Each row (from left to right):** the input image, result by Alpha expansion with 8 neighbors, result by the proposed max-flow approach. For the experiment in 1st row (inpainting in gray area), $\alpha = 0.03$ and $n = 3$; 2nd row, $\alpha = 0.04$ and $n = 4$, 3rd row, $\alpha = 0.047$ and $n = 10$; 4th row, $\alpha = 0.02$ and $n = 8$.

6 Conclusions and Future Topics

In this thesis, we explored four challenging topics of computer vision and image processing: *non-rigid flow estimation, TV-L1 image approximation, image segmentation and multi-class image partition*, by means of modern convex optimization techniques. Due to their diverse backgrounds and mathematical formulations, we study and solve these problems through different ways.

Non-rigid Flow Estimation

For non-rigid flow decomposition and estimation, we propose a flow decomposition based approach to compute the high-order div-curl regularized non-rigid flow estimation, where the variational flow estimation problem is reformulated by the optimization over two potential fields associated to divergence and curl respectively. A space-decomposition algorithm is proposed to explore both two subproblems, each of which has much less unknowns, at each iteration.

In order to achieve numerical compatibilty and stability, a mimetic finite difference method is introduced to build up accurate and reliable discretization schemes of flow decomposition and estimation. The nonsmooth functions, e.g. total-variation function, are defined as well over the proposed discretization method. Experiments show its outstanding performance in implementing fast and stable algorithms in numerics.

Moreover, we also introduce and study the new nonsmooth div-curl regularizar and show its resulting structure-texture decomposition of non-rigid flows. In addition, the novel flow estimation joint with the proposed nonsmooth flow decomposition is also studied. We use the second-order cone programming to compute the reduced highly nonsmooth convex minimization problems. Experiments show the proposed nonsmooth flow decomposition is superior in achieving high accuracy of flow estimation and keeping small-scale flow patterns.

We would mention the primal-dual optimization approach also works for the flow estimation problem. In the following, we draft the main ideas.

Now we consider the flow estimation problem as follows:

$$\min_{\mathbf{u}} D(\partial_t g - \nabla g \cdot \mathbf{u}) + R_d(\operatorname{div} \mathbf{u}) + R_c(\operatorname{curl} \mathbf{u}) , \tag{6.1}$$

where $D(\cdot)$ is the convex penalty function to force fitting the brightness-constant condition and the regularization functions $R_d(\cdot)$ and $R_c(\cdot)$ are also convex. Let D^*, R_d^* and R_c^* be their respective conjugate function.

Then after simple computation, we have the equivalent primal-dual model of (6.1):

$$\min_{\mathbf{u}} \max_{q,\psi,\phi} \langle q, \partial_t g \rangle - D^*(q) - R_d^*(\psi) - R_c^*(\phi) + \left\langle \nabla\psi + \nabla^{\perp}\phi - q\nabla g, \mathbf{u} \right\rangle . \tag{6.2}$$

Optimize the minimax problem (6.2) over $\mathbf{u}$ directly gives its equivalent dual model such that (6.1):

$$\max_{q,\psi,\phi} \quad \langle q, \partial_t g \rangle - D^*(q) - R_d^*(\psi) - R_c^*(\phi) \tag{6.3}$$

$$\text{s.t.} \quad \nabla\psi + \nabla^{\perp}\phi \; = \; q\nabla g \, . \tag{6.4}$$

Clearly, the constraint (6.4) is just the orthogonal decomposition of the vector field $q\nabla g$. In this sense, the dual model (6.3) just corresponds to the optimal control formulation of such orthogonal decomposition (6.4).

Similarly, the multiplier-based algorithmic scheme works for the linear equality constrained optimization problem (6.3).

Let

$$L(\mathbf{u}, q, \psi, \phi) \; = \; \langle q, \partial_t g \rangle - D^*(q) - R_d^*(\psi) - R_c^*(\phi) + \left\langle \nabla\psi + \nabla^{\perp}\phi - q\nabla g, \mathbf{u} \right\rangle ,$$

and its augmented Lagrangian function be given by

$$L_c(\mathbf{u}, q, \psi, \phi) \; = \langle q, \partial_t g \rangle - D^*(q) - R_d^*(\psi) - R_c^*(\phi) + \left\langle \nabla\psi + \nabla^{\perp}\phi - q\nabla g, \mathbf{u} \right\rangle \tag{6.5}$$

$$- \frac{c}{2} \left\| \nabla\psi + \nabla^{\perp}\phi - q\nabla g \right\|^2 \tag{6.6}$$

where $c > 0$. Therefore, we have the duality based flow estimation algorithm, see Alg 7.

The great advantage of performing (6.3) instead of the primal problem (6.1) is that it avoids the possible nonsmoothness of the energy function. For example, for the L_1 data fidelity term

$$D(\partial_t g - \nabla g \cdot \mathbf{u}) \; = \; \int_{\Omega} \left| \partial_t g - \nabla g \cdot \mathbf{u} \right| \, dx$$

Algorithm 7 Duality-Based Flow Estimation Algorithm for

Set the starting values of q^1, ψ^1, ϕ^1 and $\mathbf{u}^1$, and let $k = 1$ then start;

- Start k-th iteration, optimize q by fixing other variables:

$$\begin{aligned} q^{k+1} &:= \arg\max_{q} L_c(\mathbf{u}^k, q, \psi^k, \phi^k) \\ &= \arg\max_{q} \langle q, \partial_t g \rangle - D^*(q) - \frac{c}{2} \left\| q\nabla g - (\nabla\psi^k + \nabla^\perp\phi^k - \mathbf{u}^k/c) \right\|^2 , \end{aligned}$$

which can be computed in the following closed form

$$q^{k+1} = \frac{g_t + c\nabla g \cdot (\nabla\psi^k + \nabla^\perp\phi^k - \mathbf{u}^k/c)}{1 + c\left|\nabla g\right|^2} ; \tag{6.7}$$

optimize (ψ, ϕ) by fixing other variables:

$$\begin{aligned} (\psi, \phi)^{k+1} &:= \arg\max_{\psi,\phi} L_c(\mathbf{u}^k, q^{k+1}, \psi, \phi) \\ &= \arg\max_{\psi,\phi} -R_d^*(\psi) - R_c^*(\phi) - \frac{c}{2} \left\| \nabla\psi + \nabla^\perp\phi - (q^{k+1}\nabla g + \mathbf{u}^k/c) \right\|^2 ; \end{aligned} \tag{6.8}$$

- Update $\mathbf{u}$ by

$$\mathbf{u}^{k+1} := \mathbf{u}^k + c(q^{k+1}\nabla g - \nabla\psi^{k+1} + \nabla^\perp\phi^{k+1}) ;$$

- Let $k = k + 1$ and repeat the above steps till convergence is achieved.

we have

$$D^*(q) := I_S(q)$$

where I_S is the characteristic function of the convex set S and S is

$$S := \{ q \,|\, q(x) \in [0, 1] \} .$$

Obviously, to explore such nonsmooth data fitting term one only need to perform the simple projection operation to the convex set S at each iteration. The same for the nonsmooth div-curl regularizar. This algorithm appears when I changed my research topics to primal-dual convex optimization approaches. However, I have no time to implement it, especially for the nonsmooth high-order div-curl regularizar (2.1b).

TV-L1 Image Approximation

This topics studies both the convex TV-L1 image approximation model and the discrete constrained TV-L1 image approximation, with applications to image denoising. We prove that the convex TV-L1 approximation model (3.1) can be applied to solve such nonconvex optimization problem (3.2) exactly and globally, in the spatially continuous context. This greatly extends recent studies of Chan et al. [41, 42], from the simplest binary case to the general gray-scale case. In numerics, the proposed fast multiplier-based algorithm upon the constructed equivalent convex formulations, which properly avoids the nonsmoothness of the studied TV-L1 energy function. Its numerical reliability and efficiency have been verified by experiments and comparisons to the state of art method, e.g. [169]. In contrast to the graph-cut based approach [51], the proposed approach also avoids heavy memory and computation load especially when the total number of discrete values is large.

The TV-L1 scheme has a close connection to the shape or surface regularization. It application to the shape regularization, especially in 3-D, could be an interesting topics in the future. In addition, the duality based algorithm to TV-L1 color image approximation also seems open.

Image Segmentation: Continuous Min-Cut Model

For this topics, we study continuous max-flow and min-cut models, with or without supervised constraints. Dualities between max-flow and min-cut in the spatially continuous setting are then set up and investigated through primal and dual. In this regard, terminologies used by graph-cuts based techniques are revisited and explained under a new variational perspective. New optimization results on the exactness of the proposed convex models are derived and discussed with helps of the continuous max-flow formulations. The proposed continuous max-flow based algorithms are based upon classical convex optimization theories, which provide fast and reliable numerical schemes. In contrast to discrete graph-based methods, the algorithms can be easily speeded up by adopting a multigrid or parallel numerical scheme, e.g. GPU.

The max-flow methods can also be extended to other min-cut problems with multiple phases, e.g. [12]. It also paves the way to understand the classical graph based max-flow / min-cut algorithms in a completely variational manner. To this end, the proposed max-flow algorithmic scheme can also be generalized to solve min-cut problems over a general weighted graph, where the cut information, i.e. labeling function, works as associated multipliers. This is one topic of our future studies.

Parallel to the development of this thesis, the Split-Bregman method has been applied to solve the convexified labeling problem (1.13), and was also shown to significantly outperform the method of Bresson et al [31], see [72]. A detailed comparison with this method will be another interesting work in the future.

Moreover, the GPU implementation of the proposed continuous max-flow algorithm and its application to massive 3-D data are of great interests both in practice and theory.

Multi-Class Image Partition: Potts Model

Multi-class image partition with the minimum total perimeter can be expressed as the well-known Potts model. We essentially study the convex relaxed Potts model through two approaches: the entropy-maximization smoothed method and the continuous max-flow method.

The first study proposes a novel duality-based approach for continuous multi-labeling problems based on a convex relaxation of Potts model. The dual model could be used to give insight into the exactness of the relaxation. Sufficient conditions were derived for when optimal solutions to the Potts model could be obtained from a dual solution to the relaxed model. Close connections between optimal labelings and geometrical clustering of spatial points were also revealed. We then suggested a smoothing method based on the log-sum exponential function, so as to deal with the nonsmooth dual problem, and indicated that such a smoothing approach leads to a novel smoothed primal-dual model and suggests labelings with maximum entropy. A new expectation maximization like algorithm was proposed based on smoothed dual model which was shown to be superior in efficiency compared to earlier approaches. Numerical experiments also showed that this approach could outperform several competitive approaches in various aspects, such as lower energies and better visual quality.

The second approach introduces and investigates the novel continuous max-flow model which is dual to the studied convex relaxed Potts problem, which results in a new variational perspective of flows and cuts in the spatially continuous configuration and properly recovers close connections between flows and cuts. Moreover, in comparison to previous efforts which are trying to compute the optimal labeling functions in a direct way, we propose the new multiplier-based max-flow algorithm. The main advantages of such max-flow based algorithm are: it avoids extra computation load to explicitly explore the pointwise simplex constraint, each flow function is updated in a simple way; in addition, its numerical scheme contains a natural parallel framework, which can be easily accelarated by the modern parallel computation platform, e.g. GPU. Numerical experiments show it outperforms state of art approaches in terms of quality and efficiency.

For both approaches, their GPU implementations are of utmost interests for many practical applications of Potts model. To consider the non-Euclidean distances to the Potts model, as mentioned in [106], provides a useful cue to solve some practical computer vision problems. In addition, to incorporate the shape prior and some global clues, e.g. histogram matching, are also amazing future directions of studies.

Especially, the MDL based Potts model was just proposed by Yuan and Boykov [27], which optimize the smoothness of partition boundaries joint with the total number of appearence models. In [27], the direct convex programming method was considered, which can only work with images of relatively small size and takes a longer time to compute (a couple of hours to proceed an image with 200×200 pixels and about 20 labels). The application of the continuous max-flow to this problem is of great interests and importances. A fast continuous max-flow solver to MDL based image segmentation is expected to significantly speed up computation with high-accuracy!

7 Appendix

7.1 Detailed Matrix Representations of MFD

By the definitions of mimetic finite-difference (MFD) over the 2-D square grid, reshaping the scalar/vector fields columnwise into vectors of appropriate lengths, these operators act on the corresponding vector spaces as matrices which can be described by using the first-order forward difference matrix

$$\boldsymbol{D}_m := \begin{pmatrix} -1 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 & 0 \\ & & \ddots & \ddots & \ddots & & \\ 0 & 0 & 0 & \dots & -1 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & -1 & 1 \end{pmatrix} \in \mathbb{R}^{m-1,m}$$

and the tensor product $\otimes$ of matrices. Then we have

$$\mathrm{G} = \begin{pmatrix} \boldsymbol{I}_n \otimes \boldsymbol{D}_m \\ \boldsymbol{D}_n \otimes \boldsymbol{I}_m \end{pmatrix}, \quad \mathrm{GP} = \begin{pmatrix} -\boldsymbol{D}_n \otimes \boldsymbol{I}_m \\ \boldsymbol{I}_n \otimes \boldsymbol{D}_m \end{pmatrix},$$

which explains the notation $^{\perp}$. The operator $\nabla^{\perp,h}$ is just the restriction of GP to H_P^o. Although ∇^h is not the restriction of G to H_P^o, we will again use the notation $^{\perp}$. More precisely, we define

$$\nabla^h = \begin{pmatrix} \boldsymbol{I}_{n-2} \otimes \boldsymbol{D}_{m-2} \\ \boldsymbol{D}_{n-2} \otimes \boldsymbol{I}_{m-2} \end{pmatrix}, \quad \nabla^{\perp,h} = \begin{pmatrix} \boldsymbol{D}_{n-1}^{\mathsf{T}} \otimes \boldsymbol{I}_{m-2} \\ -\boldsymbol{I}_{n-2} \otimes \boldsymbol{D}_{m-1}^{\mathsf{T}} \end{pmatrix},$$

$$\mathrm{div}^h = \begin{pmatrix} \boldsymbol{I}_{n-1} \otimes \boldsymbol{D}_m, \boldsymbol{D}_n \otimes \boldsymbol{I}_{m-1} \end{pmatrix},$$

where $\boldsymbol{I}_n$ denotes the $n \times n$ identity matrix and the matrices are considered without the zero rows/columns due to embedding. Then we have on the dual grid

$$\overline{\nabla^h} = -(\mathrm{div}^h_{|_{H_S^o}})^*, \quad \overline{\mathrm{div}^h} = -(\nabla^h)^*, \quad \overline{\mathrm{curl}^h} = (\nabla^{\perp,h})^*, \tag{7.1}$$

where the adjoint operator corresponds to the transposed matrices. Using properties of the tensor product of matrices, we obtain that $\mathrm{curl}\,\nabla = 0$ reads as

$$\overline{\mathrm{curl}^h}\,\overline{\nabla^h} = 0, \quad \mathrm{div}^h\ \nabla^{\perp,h} = -\overline{\nabla^h}^{\,*}\,\overline{\mathrm{curl}^h}^{\,*} = 0. \tag{7.2}$$

We extend the operator $\overline{\nabla^h}$ to $H_{V+\partial V}$ by incorporating the boundary elements $H_{\partial V}$ into the forward differences. Since the distance of a boundary point to its neighboring inner point is only 0.5, we have to multiply the difference filter $(-1, 1)$ at the boundary by 2. The matrix of the resulting operator,

$$\overline{\nabla^h}_1 :\ H_{V+\partial V} \to H_S\,.$$

Discrete TV/G Norms

For our optical flow decomposition and estimation we need discrete versions of the TV norm $\|f\|_{\mathrm{TV}} := \int |\nabla f|\,\mathrm{d}x$ and the G norm $\|f\|_{\mathrm{G}} := \inf_{f=\mathrm{div}p} \|\,|p|\,\|_{L_\infty}$, where $|p(x)| := (p_1(x)^2 + p_2(x)^2)^{1/2}$ for a vector field $p(x) = (p_1(x), p_2(x))$, $p_1, p_2 \in L_\infty$. For a more sophisticated treatment of these norms in the continuous setting, see, e.g., [61, 115].

For $\omega \in H_P^o$, we observe the definition (2.2) and define

$$\|\omega\|_{\mathrm{TV}} := \|\nabla^h \omega\|_{L_1, H_E^o} =\,= \frac{1}{\sqrt{2}} \sum_{\Omega_{\alpha,\beta}\in\Omega} \sqrt{(\mathbf{v}_D^2 + \mathbf{v}_T^2 + \mathbf{v}_R^2 + \mathbf{v}_L^2)_{\alpha,\beta}}$$

where $\mathbf{v} = \nabla^h \omega$. This defines a seminorm on H_P^o. As usual we will skip the prefix "semi" in the following.

Now the G norm of ω is defined by

$$\|\omega\|_{\mathrm{G}} = \underline{\min}_{\omega=\mathrm{div}^h \mathbf{p}} \|\,\mathbf{p}\,\|_{L_\infty, H_E}$$

which is computed similarly as (2.3).

We define the TV and the G norm on H_V. For $\rho \in H_V$, let

$$\|\rho\|_{\overline{\mathrm{TV}}} := \|\overline{\nabla^h}\,\rho\|_{L_1, H_S}$$

which is given in (2.2). By (2.3), we also define

$$\|\rho\|_{\overline{\mathrm{G}}} := \min_{\rho=\mathrm{div}^h \mathbf{p}} \|\mathbf{p}\|_{L_\infty, H_S} \tag{7.3}$$

where $\mathbf{p}$ is set to be zero at the boundary edges, i.e. $\mathbf{p} \in H_S^o$.

Discrete integral identities Next we are interested in a discrete version of the *Gaussian integral identity* $\int_\Omega \operatorname{div} u \,\mathrm{d}x = \int_{\partial\Omega} n \cdot u \,\mathrm{d}s$. To this end, we introduce the boundary operator

$$\mathbb{B}^h := \begin{pmatrix} \boldsymbol{I}_{n-1} \otimes \boldsymbol{B}_m & \mathbf{0} \\ \mathbf{0} & \boldsymbol{B}_n \otimes \boldsymbol{I}_{m-1} \end{pmatrix},$$

where $\mathbf{0}$ are zero matrices of appropriate sizes and

$$\boldsymbol{B}_m := \begin{pmatrix} -1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix} \in \mathbb{R}^{2,m}.$$

Then the *mimetic Gaussian integral identity* becomes

$$\mathbf{1}^{\mathsf{T}}_{\dim H_V} \operatorname{div}^h u = \mathbf{1}^{\mathsf{T}}_{\dim \partial H_V} \mathbb{B}^h u. \tag{7.4}$$

7.2 Projected-Gradient Descent Step for Total-Variation Minimization

In this thesis, many algorithms rely on solving the subproblem of the total-variation minimization:

$$\min_{p \in C_\lambda} \|\operatorname{div} p - f\|^2 \,. \tag{7.5}$$

where C_λ is the convex set such that

$$C_\lambda := \{p \,|\, p \in C_c^1(\Omega, \mathbb{R}^2)\,,\ |p(x)| \leq \lambda\,,\ \forall x \in \Omega\,\}\,.$$

With helps of the mimetic finite-difference method, (7.5) can be represented by

$$\min_{p \in C_\lambda \subset H_S^o} \left\|\operatorname{div}^h p - f\right\|^2_{H_V} \tag{7.6}$$

where f is the scalar field defined in H_V and for any vector field $p \in C_\lambda \subset\subset H_S^o$, at each cell $\Omega_{(\alpha,\beta)}$ should satisfy

$$\ell_{\alpha,\beta}(p) := \sqrt{\frac{1}{2}\big((p^b)^2 + (p^r)^2 + (p^t)^2 + (p^l)^2\big)_{(\alpha,\beta)}} \leq \lambda\,.$$

(7.5) can be solved by iterative projected-gradient descent steps as follows:

$$p^{k+1} = \operatorname{Proj}_{\|p\|_\infty \leq \lambda}\big(p^k + \tau \nabla(\operatorname{div} p^k - f)\big)\,.$$

For its discretized version (7.6), the above step can be

$$p^{k+1} = \mathrm{Proj}_{C_\lambda}\big(p^k + \tau\overline{\nabla^h}(\mathrm{div}^h\, p^k - f)\big) \tag{7.7}$$

where the step-size τ is mostly chosen depending on the largest eigen-value of the matrix $\overline{\nabla^h}\,\mathrm{div}^h$.

The projection operation in (7.7) for any vector field $p \in H_S^o$ to the convex set C_λ can be approximated by the following two steps:

- Define $\tilde{\ell}(p) \in H_V$:

$$\tilde{\ell}_{\alpha,\beta}(p) := \begin{cases} \lambda/\ell_{\alpha,\beta}(p) & \text{when } \ell_{\alpha,\beta}(p) \geq \lambda \\ 1 & \text{when } \ell_{\alpha,\beta}(p) < \lambda \end{cases},$$

  and define the discrete vector field $q \in H_S$:

$$q^t_{(\alpha,\beta)} := (\tilde{\ell}_{(\alpha,\beta)} + \tilde{\ell}_{(\alpha,\beta-1)})/2\,, \quad q^b_{(\alpha,\beta)} := (\tilde{\ell}_{(\alpha,\beta+1)} + \tilde{\ell}_{(\alpha,\beta)})/2$$

$$q^l_{(\alpha,\beta)} := (\tilde{\ell}_{(\alpha,\beta)} + \tilde{\ell}_{(\alpha-1,\beta)})/2\,, \quad q^r_{(\alpha,\beta)} := (\tilde{\ell}_{(\alpha+1,\beta)} + \tilde{\ell}_{(\alpha,\beta)})/2\,;$$

- $\tilde{p} := \mathrm{Proj}_{C_\lambda}(p) \in H_S$ is computed by:

$$\tilde{p}^t_{(\alpha,\beta)} := p^t_{(\alpha,\beta)}\, q^t_{(\alpha,\beta)}\,, \quad \tilde{p}^b_{(\alpha,\beta)} := p^b_{(\alpha,\beta)}\, q^b_{(\alpha,\beta)}$$

$$\tilde{p}^l_{(\alpha,\beta)} := p^l_{(\alpha,\beta)}\, q^l_{(\alpha,\beta)}\,, \quad \tilde{p}^r_{(\alpha,\beta)} := p^r_{(\alpha,\beta)}\, q^r_{(\alpha,\beta)}\,.$$

Bibliography

[1] S. Alliney. Digital filters as absolute norm regularizers. *Signal Processing, IEEE Transactions on*, 40(6):1548 –1562, June 1992.

[2] S. Alliney. Recursive median filters of increasing order: a variational approach. *Signal Processing, IEEE Transactions on*, 44(6):1346 –1354, June 1996.

[3] L. Amodei and M. N. Benbourhim. A vector spline approximation. *J. Approx. Theory*, 67(1):51–79, 1991.

[4] Ben Appleton and Hugues Talbot. Globally minimal surfaces by continuous maximal flows. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(1):106–118, 2006.

[5] Douglas N. Arnold. Differential complexes and numerical stability. In *Proceedings of the International Congress of Mathematicians, Vol. I (Beijing, 2002)*, pages 137–157, Beijing, 2002. Higher Ed. Press.

[6] Douglas N. Arnold, Richard S. Falk, and Ragnar Winther. Finite element exterior calculus, homological techniques, and applications. *Acta Numer.*, 15:1–155, 2006.

[7] G. Aubert, R. Deriche, and P. Kornprobst. Computing optical flow via variational techniques. *SIAM J. Appl. Math.*, 60(1):156–182 (electronic), 2000.

[8] Gilles Aubert and Jean-François Aujol. A variational approach to removing multiplicative noise. *SIAM Journal of Applied Mathematics*, 68(4):925–946, 2008.

[9] Jean-François Aujol and Antonin Chambolle. Dual norms and image decomposition models. *International Journal of Computer Vision*, 63(1):85–104, 2005.

[10] Jean-François Aujol, Guy Gilboa, Tony F. Chan, and Stanley Osher. Structure-texture image decomposition - modeling, algorithms, and parameter selection. *International Journal of Computer Vision*, 67(1):111–136, 2006.

[11] E. Bae and X.C. Tai. Graph cut optimization for the piecewise constant level set method applied to multiphase image segmentation. *Scale Space and Variational Methods in Computer Vision*, pages 1–13, 2009.

[12] E. Bae, J. Yuan, X.-C. Tai, and Y. Boykov. A study on continuous max-flow and min-cut approaches part ii: Multiple linearly ordered labels. Technical report CAM-10-62, UCLA, CAM, 2010.

[13] E. Bae, J. Yuan, and X.C. Tai. Global minimization for continuous multiphase partitioning problems using a dual approach. Technical report CAM09-75, UCLA, CAM, September 2009.

[14] E. Bae, J. Yuan, and X.C. Tai. Global minimization for continuous multiphase partitioning problems using a dual approach. *International Journal of Computer Vision*, To appear.

[15] Arindam Banerjee, Srujana Merugu, Inderjit Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. In *Journal of Machine Learning Research*, pages 234–245, 2004.

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

[17] F. Becker, B. Wieneke, J. Yuan, and C. Schnörr. A variational approach to adaptive correlation for motion estimation in particle image velocimetry. In Gerhard Rigoll, editor, *Pattern Recognition – 30th DAGM Symposium*, volume 5096, pages 335–344. Springer Verlag, 2008.

[18] F. Becker, B. Wieneke, J. Yuan, and C. Schnörr. Variational correlation approach to flow measurement with window adaption. In *14th International Symposium on Applications of Laser Techniques to Fluid Mechanics*, page 1.1.3, 2008.

[19] Andrea L Bertozzi, Selim Esedoglu, and Alan Gillette. Inpainting of binary images using the cahn-hilliard equation. *IEEE Trans Image Process*, 16(1):285–91, 2007.

[20] Dimitri P. Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Computer Science and Applied Mathematics. Academic Press Inc. [Harcourt Brace Jovanovich Publishers], New York, 1982.

[21] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, September 1999.

[22] Pavel B. Bochev and James M. Hyman. Principles of mimetic discretizations of differential operators. In *Compatible spatial discretizations*, volume 142 of *IMA Vol. Math. Appl.*, pages 89–119. Springer, New York, 2006.

[23] Endre Boros and Peter L. Hammer. Pseudo-boolean optimization. *Discrete Appl. Math.*, 123(1-3):155–225, 2002.

[24] A. Bossavit. Differential forms and the computation of fields and forces in electromagnetism. *European J. Mech. B Fluids*, 10(5):474–488, 1991.

[25] Alain Bossavit. *Computational electromagnetism*. Electromagnetism. Academic Press Inc., San Diego, CA, 1998. Variational formulations, complementarity, edge elements.

[26] Alain Bossavit. Extrusion, contraction: their discretization via Whitney forms. *COMPEL*, 22(3):470–480, 2003. Selected papers from the 10th International IGTE Symposium on Numerical Field Computation (Graz, 2002).

[27] J. Yuan Y. Boykov. Tv-based image segmentation with label cost prior. In *BMVC 2010*, 2010.

[28] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:359–374, 2001.

[29] Yuri Boykov and Vladimir Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *ICCV*, pages 26–33, 2003.

[30] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:1222 – 1239, 2001.

[31] X. Bresson, S. Esedoglu, P. Vandergheynst, J.P. Thiran, and S. Osher. Fast global minimization of the active contour/snake model. *Journal of Mathematical Imaging and Vision*, 28(2):151–167, 2007.

[32] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estiation based on a theory for warping. *ECCV 2004, Springer LNCS 3024*, 4:25–36, May 2004.

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

[34] M. Burger, S. Osher, J. Xu, and G. Gilboa. Nonlinear inverse scale space methods for image restoration. *Lecture Notes in Computer Science*, 3752:25, 2005.

[35] I. J. Carlier and B. Wieneke. Report 1 on production and diffusion of fluid mechanics images and data. *Fluid Project Deliverable (http://fluid.irisa.fr/)*, Feb. 2005.

[36] Vincent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. In *ICCV*, pages 694–699, 1995.

[37] A. Chambolle, D. Cremers, and T. Pock. A convex approach for computing minimal partitions. Technical report TR-2008-05, Dept. of Computer Science, University of Bonn, Bonn, Germany, November 2008.

[38] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. Technical report, T.U. Graz, 2010.

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

[40] T. Chan and L.A. Vese. *Active contours without edges*. IEEE Image Proc., 10, pp. 266-277, 2001.

[41] Tony F. Chan and Selim Esedo\=glu. Aspects of total variation regularized L^1 function approximation. *SIAM J. Appl. Math.*, 65(5):1817–1837 (electronic), 2005.

[42] Tony F. Chan, Selim Esedo\=glu, and Mila Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM J. Appl. Math.*, 66(5):1632–1648 (electronic), 2006.

[43] Tony F. Chan, Selim Esedoglu, and Frederick E. Park. Image decomposition combining staircase reduction and texture extraction. *J. Visual Communication and Image Representation*, 18(6):464–486, 2007.

[44] Xue cheng Tai and Magne Espedal. Rate of convergence of some space decomposition methods for linear and nonlinear problems. *SIAM J. Numer. Anal*, 35:1558–1570, 1998.

[45] P.L. Combettes and V.R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation*, 4:1168–1200, 2005.

[46] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, Cambridge, MA, second edition, 2001.

[47] T. Corpetti, D. Heitz, G. Arroyo, E. Memin, and A. Santa-Cruz. Fluid experimental flow estimation based on an optical-flow scheme. *Exp. Fluids*, 40(1):80–97, 2010.

[48] T. Corpetti, É. Mémin, and P. Pérez. Dense estimation of fluid flows. 24(3):365 –380, 2002.

[49] T. Corpetti, É. Mémin, and P. Pérez. Extraction of singular points from dense motion fields: an analytic approach. *J. of Math. Imag. Vision*, 19(3):175–198, 2003.

[50] Th. Corpetti, P. Heas, E. MÃľmin, and N. Papadakis. Pressure image assimilation for atmospheric motion estimation. *Tellus Series A: Dynamic Meteorology and Oceanography*, In Press, 2008.

[51] Jérôme Darbon and Marc Sigelle. Image restoration with discrete constrained total variation part i: Fast and exact optimization. *Journal of Mathematical Imaging and Vision*, 26(3):261–276, 2006.

[52] Jérôme Darbon and Marc Sigelle. Image restoration with discrete constrained total variation part ii: Levelable functions, convex priors and non-convex cases. *Journal of Mathematical Imaging and Vision*, 26(3):277–291, 2006.

[53] Romano G. P. Di Florio D., Di Felice F. Windowing, re-shaping and re-orientation interrogation windows in particle image velocimetry for the investigation of shear flows. *Measurement Science and Technology*, 13(1):953–962, January 2002.

[54] Yefim Dinitz. Dinitz' algorithm: The original version and even's version. *Theoretical Computer Science*, pages 218–240, 2006.

[55] Yiqiu Dong, Michael Hintermuller, and Marrick Neri. An Efficient Primal-Dual Method for L1 TV Image Restoration. *SIAM Journal on Imaging Sciences*, 2(4):1168–1189, 2009.

[56] Vincent Duval, Jean F. Aujol, and Yann Gousseau. The tvl1 model: A geometric point of view. *Multiscale Modeling & Simulation*, 8(1):154–189, 2009.

[57] Jack Edmonds and Richard M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264, 1972.

[58] Ivar Ekeland and Roger Téman. *Convex analysis and variational problems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1999.

[59] E. Esser. Applications of lagrangian-based alternating direction methods and connections to split bregman. Technical report CAM-09-31, UCLA, CAM, 2009.

[60] Ernie Esser, Xiaoqun Zhang, and Tony Chan. A general framework for a class of first order primal-dual algorithms for tv minimization. Technical report CAM-09-67, UCLA, 2009.

[61] L. C. Evans and R. F. Gariepy. *Measure Theory and Fine Properties of Functions*. CRC Press, Boca Raton, FL, 1992.

[62] Ky Fan. Minimax theorems. *Proc. Nat. Acad. Sci. U. S. A.*, 39:42–47, 1953.

[63] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient belief propagation for early vision. In *In CVPR*, pages 261–268, 2004.

[64] L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.

[65] William T. Freeman and Egon C. Pasztor. Learning low-level vision. *International Journal of Computer Vision*, 40:2000, 2000.

[66] Brendan J. Frey and David J. C. MacKay. A revolution: Belief propagation in graphs with cycles. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, pages 479–485. MIT Press, 1998.

[67] Vivette Girault and Pierre-Arnaud Raviart. *Finite element methods for Navier-Stokes equations*, volume 5 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 1986. Theory and algorithms.

[68] Enrico Giusti. *Minimal surfaces and functions of bounded variation*. Australian National University, Canberra, 1977.

[69] Andrew V. Goldberg and Robert E. Tarjan. A new approach to the maximum-flow problem. *J. ACM*, 35(4):921–940, October 1988.

[70] Donald Goldfarb and Wotao Yin. Second-order cone programming methods for total variation-based image restoration. *SIAM J. Sci. Comput.*, 27(2):622–645 (electronic), 2005.

[71] Donald Goldfarb and Wotao Yin. Parametric maximum flow algorithms for fast total variation minimization. *SIAM J. Scientific Computing*, 31(5):3712–3743, 2009.

[72] Tom Goldstein, Xavier Bresson, and Stanley Osher. Geometric applications of the split bregman method: Segmentation and surface reconstruction. *J. Sci. Comput.*, 45:272–293, October 2010.

[73] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society, Series B*, pages 271–279, 1989.

[74] F. Guichard and L. Rudin. Accurate estimation of discontinuous optical flow by minimizing divergence related functionals. In *ICIP*, pages I: 497–500, 1996.

[75] Sandeep Gupta, Eep N. Gupta, and Jerry L. Prince. Stochastic models for div-curl optical flow methods. *IEEE Signal Processing Letters*, 3:32–35, 1996.

[76] D. Heitz, E. Mémin, and C. Schnörr. Variational fluid flow measurements from image sequences: synopsis and perspectives. *Exp. Fluids*, 48(3):369–393, 2010.

[77] W. Hinterberger, O. Scherzer, C. Schnörr, and J. Weickert. Analysis of optical flow

models in the framework of the calculus of variations. *Numer. Funct. Anal. Optim.*, 23(1-2):69–89, 2002.

[78] M. Hintermuller, K. Ito, and K. Kunisch. The primal-dual active set strategy as a semismooth newton method. *SIAM Journal on Optimization*, 13:865, 2002.

[79] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Convex analysis and minimization algorithms. I*, volume 305 of *Fundamental Principles of Mathematical Sciences*. Springer-Verlag, Berlin, 1993. Fundamentals.

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

[81] Y. Huang, M.K. Ng, and Y.W. Wen. A fast total variation minimization method for image restoration. *Multiscale Modeling and Simulation*, 7:774, 2008.

[82] J. M. Hyman and M. J. Shashkov. Adjoint operators for the natural discretizations of the divergence, gradient and curl on logically rectangular grids. *Appl. Numer. Math.*, 25(4):413–442, 1997.

[83] J. M. Hyman and M. J. Shashkov. Natural discretizations for the divergence, gradient, and curl on logically rectangular grids. *Comput. Math. Appl.*, 33(4):81–104, 1997.

[84] James M. Hyman and J. C. Scovel. Deriving mimetic difference approximations to differential operators using algebraic topology. Technical report, Los Alamos National Laboratory, Los Alamos, NM 87545, 1988.

[85] James M. Hyman and Mikhail Shashkov. Mimetic discretizations for Maxwell's equations. *J. Comput. Phys.*, 151(2):881–909, 1999.

[86] James M. Hyman and Mikhail Shashkov. The orthogonal decomposition theorems for mimetic finite difference methods. *SIAM J. Numer. Anal.*, 36(3):788–818 (electronic), 1999.

[87] Hiroshi Ishikawa. Exact optimization for markov random fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1333–1336, 2003.

[88] Hiroshi Ishikawa. Higher-order clique reduction in binary graph cut. In *CVPR*, pages 2993–3000, 2009.

[89] M.-C. Jullien, P. Castiglione, and P. Tabeling. Intermittency of a passive tracer in the inverse energy cascade. *Physical Review E*, 64(3), 2001. Art. No. 035301 Part 2.

[90] Y.M. Jung, S.H. Kang, and J. Shen. Multiphase image segmentation via Modica-Mortola phase transition. *SIAM Journal on Applied Mathematics*, 67(5):1213–1232, 2007.

[91] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[92] Krzysztof C. Kiwiel. Proximal minimization methods with generalized bregman functions. *SIAM Journal on Control and Optimization*, 35:1142–1168, 1995.

[93] Maria Klodt, Thomas Schoenemann, Kalin Kolev, Marek Schikora, and Daniel Cremers. An experimental comparison of discrete and continuous shape optimization methods. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 332–345, Berlin, Heidelberg, 2008. Springer-Verlag.

[94] Timo Kohlberger, Etienne Memin, and Christoph Schnörr. Variational dense motion estimation using the helmholtz decomposition. In *Scale Space Methods in Computer Vision, volume 2695 of LNCS*, pages 432–448. Springer, 2003.

[95] Pushmeet Kohli, M. Pawan Kumar, and Philip H.S. Torr. p^3 and beyond: Move making algorithms for solving higher order functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1645–1656, 2009.

[96] Vladimir Kolmogorov. What metrics can be approximated by geo-cuts, or global optimization of length/area and flux. In *In ICCV*, pages 564–571, 2005.

[97] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(10):1568–1583, 2006.

[98] Vladimir Kolmogorov and Carsten Rother. Minimizing nonsubmodular functions with graph cuts-a review. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(7):1274–1279, 2007.

[99] Vladimir Kolmogorov and Ramin Zabih. Multi-camera scene reconstruction via graph cuts. In *in European Conference on Computer Vision*, pages 82–96, 2002.

[100] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:65–81, 2004.

[101] Nikos Komodakis and Georgios Tziritas. Approximate labeling via graph-cuts based on linear programming. In *In Pattern Analysis and Machine Intelligence*, page 2007, 2007.

[102] Nikos Komodakis, Georgios Tziritas, and Nikos Paragios. Fast, approximately optimal solutions for single and dynamic mrfs. In *CVPR*, 2007.

[103] P. R. Kotiuga. Hodge decomposition and computational electromagnetics. *Thesis*, 1984. Department of electrical engineering,McGill University, Montreal.

[104] R. Larsen, K. Conradsen, and B. k. Ersbøll. Estimation of dense image flow fields in fluids. *IEEE Transactions on Geoscience and Remote Sensing*, 36(1):256–264, jan 1998.

[105] Eugene Lawler. *Combinatorial Optimization : Networks and Matroids*. Dover Publications, March 2001.

[106] J. Lellmann, F. Becker, and C. Schnörr. Convex optimization for multi-class image labeling with a novel family of total variation based regularizers. In *IEEE International Conference on Computer Vision (ICCV)*, pages 646 – 653, 2009.

[107] J. Lellmann, D. Breitenreicher, and C. Schnörr. Fast and exact primal-dual iterations for variational problems in computer vision. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *European Conference on Computer Vision (ECCV)*, volume 6312 of *LNCS*, pages 494–505. Springer Berlin / Heidelberg, 2010.

[108] J. Lellmann, J. Kappes, J. Yuan, F. Becker, and C. Schnörr. Convex multi-class image labeling by simplex-constrained total variation. Technical report, HCI, IWR, Uni. Heidelberg, IWR, Uni. Heidelberg, November 2008.

[109] H. Li and X.C. Tai. Piecewise constant level set methods for multiphase motion. *International Journal of Numerical Analysis and Modeling*, 4:274–293, 2007.

[110] Stan Z. Li. *Markov random field modeling in image analysis*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.

[111] J. Lie, M. Lysaker, and X.-C. Tai. *A binary level set model and some applications to Mumford-Shah image segmentation*. UCLA CAM 04-31, 2004.

[112] J. Lie, M. Lysaker, and X.-C. Tai. A variant of the level set method and applications to image segmentation. *Math. Comp.*, 75(255):1155–1174, 2006.

[113] S. B. Lobo, L. Vandenberghe, and H. Lebret. Applications of second order cone programming. *Linear Algebra Appl.*, 284:193–228, 1998.

[114] Claudio Mattiussi. An analysis of finite volume, finite element, and finite difference methods using some concepts from algebraic topology. *J. Comput. Phys.*, 133(2):289–309, 1997.

[115] Yves Meyer. *Oscillating patterns in image processing and nonlinear evolution equations*, volume 22 of *University Lecture Series*. American Mathematical Society, Providence, RI, 2001. The fifteenth Dean Jacqueline B. Lewis memorial lectures.

[116] Pushmeet Kohli Michael Bleyer, Carsten Rother. Surface stereo with soft segmentation. In *CVPR*, 2010.

[117] H. D. Mittelmann. An independent benchmarking of SDP and SOCP solvers. *Math. Program.*, 95(2, Ser. B):407–430, 2003. Computational semidefinite and second order cone programming: the state of the art.

[118] Yu Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, 2005.

[119] R. A. Nicolaides. Direct discretization of planar div-curl problems. *SIAM J. Numer. Anal.*, 29(1):32–56, 1992.

[120] R. A. Nicolaides and K. A. Trapp. Covolume discretization of differential forms. In *Compatible spatial discretizations*, volume 142 of *IMA Vol. Math. Appl.*, pages 161–171. Springer, New York, 2006.

[121] Roy Nicolaides and Da-Qing Wang. A higher-order covolume method for planar div-curl problems. *Internat. J. Numer. Methods Fluids*, 31(1):299–308, 1999. Tenth International Conference on Finite Elements in Fluids (Tucson, AZ, 1998).

[122] Roy A. Nicolaides and Xiaonan Wu. Covolume solutions of three-dimensional div-curl equations. *SIAM J. Numer. Anal.*, 34(6):2195–2203, 1997.

[123] Mila Nikolova. A variational approach to remove outliers and impulse noise. *Journal of Mathematical Imaging and Vision*, 20(1-2):99–120, 2004.

[124] Mila Nikolova, Selim Esedoglu, and Tony F. Chan. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal on Applied Mathematics*, 66(5):1632–1648, 2006.

[125] RodrÃňguez P. A. Nogueira J., Lecuona A. Limits on the resolution of correlation piv iterative methods. fundamentals. *Experiments in Fluids*, 39(2):305–313, 8 2005.

[126] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling and Simulation*, 4(2):460–489, 2006.

[127] S. Osher and J.A. Sethian. Fronts propagating with curvature dependent speed: algorithms based on hamilton-jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, 1988.

[128] Stanley Osher, Andrés Solé, and Luminita Vese. Image decomposition and restoration using total variation minimization and the H^{-1} norm. *Multiscale Model. Simul.*, 1(3):349–370 (electronic), 2003.

[129] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, 1982.

[130] Nikos Paragios, Yunmei Chen, and Olivier Faugeras. *Handbook of Mathematical Models in Computer Vision*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[131] T. Pock, A. Chambolle, H. Bischof, and D. Cremers. A convex relaxation approach for computing minimal partitions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Miami, Florida, 2009.

[132] T. Pock, T. Schoenemann, G. Graber, H. Bischof, and D. Cremers. A convex formulation of continuous multi-label problems. In *European Conference on Computer Vision (ECCV)*, Marseille, France, October 2008.

[133] Thomas Pock, Thomas Schoenemann, Gottfried Graber, Horst Bischof, and Daniel Cremers. A convex formulation of continuous multi-label problems. In *ECCV 2008*, 2008.

[134] S. B. Pope. *Turbulent Flows*. Cambridge University Press, 2000.

[135] Renfrey B. Potts. Some generalized order-disorder transformations. *In Proceedings of the Cambridge Philosophical Society, Vol. 48*, pages 106–109, 1952.

[136] Kompenhans J. Raffel M., Willert C. *Particle Image Velocimetry*, volume 2. Springer-Verlag Berlin, 2001.

[137] R. Tyrrell Rockafellar. *Convex analysis*. Princeton Mathematical Series, No. 28. Princeton University Press, Princeton, N.J., 1970.

[138] R.T. Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Math. Oper. Res.*, 1(2):97–116, 1976.

[139] Kenneth Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. In *Proceedings of the IEEE*, pages 2210–2239, 1998.

[140] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. GrabcutâĂİ: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23:309–314, 2004.

[141] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1-4):259–268, 1992.

[142] P. Ruhnau, A. Stahl, and C. Schnörr. Variational estimation of experimental fluid flows with physics-based spatio-temporal regularization. *Measurement Science and Technology*, 18:755–763, 2007.

[143] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47:7–42, 2001.

[144] Otmar Scherzer, Wotao Yin, and Stanley Osher. Slope and G-set characterization of set-valued functions and applications to non-differentiable optimization problems. *Commun. Math. Sci.*, 3(4):479–492, 2005.

[145] W. Schwalm, B. Moritz, M. Giona, and M. Schwalm. Vector difference calculus for physical lattice models. *Phys. Rev. E (3)*, 59(1, part B):1217–1233, 1999.

[146] S. Setzer, G. Steidl, and T. Teuber. Deblurring Poissonian images by split Bregman techniques. *J. Visual Commun. Image Repres.*, 2009. accepted.

[147] Simon Setzer. Split bregman algorithm, douglas-rachford splitting and frame shrinkage. In *SSVM*, pages 464–476, 2009.

[148] Mikhail Shashkov. *Conservative finite-difference methods on general grids*. Symbolic and Numeric Computation Series. CRC Press, Boca Raton, FL, 1996. With 1 IBM-PC floppy disk (3.5 inch; HD).

[149] Gabriele Steidl, Joachim Weickert, Thomas Brox, Pavel Mrazek, Martin Welk, and Pavel Mr Azek. On the equivalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and sides. *SIAM J. Numer. Anal*, 42:686–713, 2004.

[150] Gil. Strang. Maximal flow through a domain. *Mathematical Programming*, 26:123–143, 1983.

[151] Gil. Strang. Maximum flows and minimum cuts in the plane. *Advances in Mechanics and Mathematics*, III:1–11, 2008.

[152] D. Suter. Motion estimation and vector splines. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 939–942, Los Alamitos, CA, USA, June 1994. IEEE Computer Society Press.

[153] Richard Szeliski, Ramin Zabih, Daniel Scharstein, Olga Veksler, Aseem Agarwala, and Carsten Rother. A comparative study of energy minimization methods for markov random fields. In *In ECCV*, pages 16–29, 2006.

[154] Xue-Cheng Tai and Paul Tseng. Convergence rate analysis of an asynchronous space decomposition method for convex minimization. *Math. Comp.*, 71(239):1105–1135 (electronic), 2002.

[155] Xue-Cheng Tai and Chunlin Wu. Augmented Lagrangian Method, Dual Methods, and Split Bregman Iteration for ROF Model. In *SSVM. 2009*, pages 502–513. Springer, LNCS vol. 5567, 2009.

[156] Xue-Cheng Tai and Jinchao Xu. Subspace correction methods for convex optimization problems. In *Eleventh International Conference on Domain Decomposition Methods (London, 1998)*, pages 130–139 (electronic). DDM.org, Augsburg, 1999.

[157] Xue-Cheng Tai and Jinchao Xu. Global and uniform convergence of subspace correction methods for some convex optimization problems. *Math. Comp.*, 71(237):105–124, 2002.

[158] Hugues Talbot. Globally optimal surfaces by continuous maximal flows. In *Digital Image Computing: Techniques and Applications*, pages 987–996, 2003.

[159] Marc Teboulle. A unified continuous optimization framework for center-based clustering methods. *J. Mach. Learn. Res.*, 8:65–102, 2007.

[160] Theunissen, R., Scarano, F., Riethmuller, and M.L. An adaptive sampling and windowing interrogation method in piv. *Measurement Science and Technology*, 18(1):275–287, January 2007.

[161] Emanuele Trucco and Alessandro Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1998.

[162] Robert J. Vanderbei. *Linear Programming: Foundations and Extensions*. Springer, second edition, 2001.

[163] Olga Veksler. Graph cut based optimization for mrfs with truncated convex priors. In *CVPR*, 2007.

[164] L. A. Vese and T. F. Chan. A new multiphase level set framework for image segmentation via the mumford and shah model. *International Journal of Computer Vision*, 50:271–293, 2002.

[165] Luminita A. Vese and Stanley J. Osher. Modeling textures with total variation minimization and oscillating patterns in image processing. *J. Sci. Comput.*, 19(1-3):553–572, 2003. Special issue in honor of the sixtieth birthday of Stanley Osher.

[166] A. Vlasenko and C. Schnörr. Physically consistent and efficient variational denoising of image fluid flow estimates. *IEEE Trans. Image Proc.*, 19(3):586–595, 2010.

[167] Martin Wainwright, Tommi Jaakkola, and Alan Willsky. Map estimation via agreement on (hyper)trees: Message-passing and linear programming approaches. *IEEE Transactions on Information Theory*, 51:3697–3717, 2002.

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

[169] Chunlin Wu, Juyong Zhang, and Xue-Cheng Tai. Augmented lagrangian method for total variation restoration with non-quadratic fidelity. Technical report CAM-09-82, UCLA, CAM, 2009.

[170] J. Xu. Iterative methods by space decomposition and subspace correction: A unifying approach. *SIAM Review*, 34:581–613, 1992.

[171] J.-C. Xu. *Theory of Multilevel Methods*. PhD thesis, Cornell University, May 1989.

[172] Wotao Yin, Donald Goldfarb, and Stanley Osher. Image cartoon-texture decomposition and feature selection using the total variation regularized l1 functional. In *VLSM*, pages 73–84. Springer, 2005.

[173] Wotao Yin, Donald Goldfarb, and Stanley Osher. The total variation regularized L^1 model for multiscale decomposition. *Multiscale Model. Simul.*, 6(1):190–211 (electronic), 2007.

[174] J. Yuan, E. Bae, and X.C. Tai. A study on continuous max-flow and min-cut approaches. In *CVPR*, USA, San Francisco, 2010.

[175] J. Yuan, E. Bae, X.C. Tai, and Y. Boykov. A continuous max-flow approach to potts model. In *ECCV 2010*, 2010.

[176] J. Yuan, E. Bae, X.C. Tai, and Y. Boykov. A study on continuous max-flow and min-cut approaches part i: Binary labeling. Technical report CAM-10-61, UCLA, CAM, 2010.

[177] J. Yuan, P. Ruhnau, Mémin E., and C. Schnörr. Discrete orthogonal decomposition and variational fluid flow estimation. In *ScaleSpace 2005*, Hofgeismar, Germany, April 2005.

[178] J. Yuan, C. Schnörr, and Memin E. Discrete orthogonal decomposition and variational fluid flow estimation. *Journal of Mathematical Imaging and Vision*, 28(1):67–80, May 2007.

[179] J. Yuan, C. Schnörr, and G. Steidl. Simultaneous optical flow estimation and decomposition. *SIAM J. Scientific Computing*, 29(6):2283–2304, 2007.

[180] J. Yuan, C. Schnörr, and G. Steidl. Total-variation based piecewise affine regularization. In K. A. Lie, M Lysaker, K. Morken, and X.-C. Tai, editors, *Scale Space and Variational Methods*, LNCS. Springer, 2009.

[181] J. Yuan, C. Schnörr, G. Steidl, and F. Becker. A study of non-smooth convex flow decomposition. In *Proc. Variational, Geometric and Level Set Methods in Computer Vision*, volume 3752, pages 1–12. Springer, 2005.

[182] Jing Yuan, Juan Shi, and Xue-Cheng Tai. A convex and exact approach to discrete constrained tv-l1 image approximation. Technical report CAM-10-51, UCLA, CAM, 2010.

[183] Jing Yuan, Juan Shi, and Xue-Cheng Tai. A convex and exact approach to discrete constrained tv-l1 image approximation. *East Asian Journal on Applied Mathematics*, 1(2):172–186, 2011.

[184] C. Zach, D. Gallup, J.-M. Frahm, and M. Niethammer. Fast global labeling for real-time stereo using multiple plane sweeps. In *Vision, Modeling and Visualization Workshop (VMV)*, 2008.

[185] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime TV- L 1 optical flow. In *DAGM*, pages 214–223, 2007.

[186] Xiaoqun Zhang, Martin Burger, and Stanley Osher. A unified primal-dual algorithm framework based on bregman iteration. Technical report CAM-09-99, UCLA, CAM, 2009.

[187] Lin Zhou, Chandra Kambhamettu, and Dmitry B. Goldgof. Fluid structure and motion analysis from multi-spectrum 2d cloud image sequences. In *CVPR*, 2000.

[188] M. Zhu and T. Chan. An efficient primal-dual hybrid gradient algorithm for total variation image restoration. Technical report, 2008.