Quelques modèles mathématiques et algorithmes rapides pour le traitement d'images

Rémy Abergel

UNIVERSITÉ PARIS DESCARTES

École doctorale 386 : Sciences Mathématiques de Paris Centre

Laboratoire MAP5 CNRS UMR 8145

Quelques modèles mathématiques et algorithmes rapides pour le traitement d'images

Several mathematical models and fast algorithms for image processing

Par Rémy ABERGEL

Thèse de doctorat de Mathématiques Appliquées

Dirigée par Lionel Moisan

Présentée et soutenue publiquement le 4 octobre 2016, devant le jury composé de

Andrés Almansa	Centre National de la Recherche Scientifique	Examinateur
Jean-François Aujol	Université de Bordeaux	Rapporteur
Cécile Louchet	Université d'Orléans	Examinatrice
Lionel Moisan	Université Paris Descartes	Directeur de thèse
Bernard Rougé	Centre d'Études Spatiales de la BIOsphère	Examinateur
Florence Tupin	Telecom Paristech	Rapporteur
Pierre Weiss	Centre National de la Recherche Scientifique	Examinateur

Résumé

Dans cette thèse, nous nous intéressons à différents modèles mathématiques de traitement d'images numériques dits de bas niveau. Si l'approche mathématique permet d'établir des modèles innovants pour traiter les images, ainsi que l'étude rigoureuse des propriétés des images qu'ils produisent, ils impliquent parfois l'utilisation d'algorithmes très consommateurs de temps de calcul et de mémoire. Aussi, nous portons un soin particulier au développement d'algorithmes rapides à partir des modèles mathématiques considérés.

Nous commençons par effectuer une présentation synthétique des méthodes mathématiques basées sur la dualité de Legendre-Fenchel permettant la minimisation d'énergies faisant intervenir la variation totale, fonctionnelle convexe non-différentiable, ceci afin d'effectuer divers traitements sur les images numériques.

Nous étudions ensuite un modèle de discrétisation de la variation totale inspiré de la théorie de l'échantillonnage de Shannon. Ce modèle, appelé « variation totale Shannon » permet un contrôle fin de la régularité des images sur une échelle sous-pixellique. Contrairement aux modèles de discrétisation classiques qui font appel à des schémas aux différences finies, nous montrons que l'utilisation de la variation totale Shannon permet de produire des images pouvant être facilement interpolées. Nous montrons également que la variation totale Shannon permet un gain conséquent en matière d'isotropie et ouvre la porte à de nouveaux modèles mathématiques de restauration.

Après cela, nous proposons une adaptation du modèle TV-ICE (Iterated Conditional Expectations, proposé en 2014 par Louchet et Moisan) au cas du débruitage d'images en présence de bruit de Poisson. Nous démontrons d'une part que le schéma numérique issu de ce modèle consiste en un schéma de point fixe dont la convergence est linéaire, d'autre part que les images ainsi produites ne présentent pas d'effet de marche d'escalier (staircasing), contrairement aux images obtenues avec l'approche plus classique dite du maximum a posteriori. Nous montrons également que le modèle Poisson TV-ICE ainsi établi repose sur l'évaluation numérique d'une fonction gamma incomplète généralisée nécessitant une prise en compte fine des erreurs numériques inhérentes au calcul en précision finie et pour laquelle nous proposons un algorithme rapide permettant d'atteindre une précision quasi-optimale pour une large gamme de paramètres.

Enfin, nous reprenons les travaux effectués par Primet et Moisan en 2011 concernant l'algorithme ASTRE (A contrario Smooth TRajectory Extraction) dédié à la détection de trajectoires régulières à partir d'une séquence de nuages de points, ces points étant considérés comme issus d'une détection préalable dans une

séquence d'images. Si l'algorithme ASTRE permet d'effectuer une détection optimale des trajectoires régulières au sens d'un critère a contrario, sa complexité en $\mathscr{O}(K^2)$ (où K désigne le nombre d'images de la séquence) s'avère être rédhibitoire pour les applications nécessitant le traitement de longues séquences. Nous proposons une variante de l'algorithme ASTRE appelée CUTASTRE qui préserve les performances de l'algorithme ASTRE ainsi que certaines de ses propriétés théoriques, tout en présentant une complexité en $\mathscr{O}(K)$.

Abstract

In this thesis, we focus on several mathematical models dedicated to low-level digital image processing tasks. Mathematics can be used to design innovative models and to provide some rigorous studies of properties of the produced images. However, those models sometimes involve some intensive algorithms with high computational complexity. We take a special care in developing fast algorithms from the considered mathematical models.

First, we give a concise description of some fundamental results of convex analysis based on Legendre-Fenchel duality. Those mathematical tools are particularly efficient to perform the minimization of convex and nonsmooth energies, such as those involving the total variation functional which is used in many image processing applications.

Then, we focus on a Fourier-based discretization scheme of the total variation, called Shannon total variation, which provides a subpixellic control of the image regularity. In particular, we show that, contrary to the classically used discretization schemes of the total variation based on finite differences, the use of the Shannon total variation yields images that can be easily interpolated. We also show that this model provides some improvements in terms of isotropy and grid invariance, and propose a new restoration model which transforms an image into a very similar one that can be easily interpolated.

Next, we propose an adaptation of the TV-ICE (Total Variation Iterated Conditional Expectations) model, recently proposed by Louchet and Moisan in 2014, to address the restoration of images corrupted by a Poisson noise. We derive an explicit form of the recursion operator involved by this scheme, and show linear convergence of the algorithm, as well as the absence of staircasing effect for the produced images. We also show that this variant involves the numerical evaluation of a generalized incomplete gamma function which must be carefully handled due to the numerical errors inherent to the finite precision floating-point calculus. Then, we propose an fast algorithm dedicated to the evaluation of this generalized

incomplete gamma function, and show that the accuracy achieved by the proposed procedure is near optimal for a large range of parameters.

Lastly, we focus on the ASTRE (A contrario Smooth TRajectory Extraction) algorithm, proposed by Primet and Moisan in 2011 to perform trajectory detection from a noisy point set sequence. We propose a variant of this algorithm, called CUTASTRE, which manages to break the quadratic complexity of ASTRE with respect to the number of frames of the sequence, while showing similar (and even slightly better) detection performances and preserving some interesting theoretical properties of the original ASTRE algorithm.

Remerciements

Je tiens tout d'abord à adresser mes sincères remerciements à Jean-François Aujol et Florence Tupin pour avoir accepté la charge de rapporteur. Qu'ils soient assurés de ma plus grande reconnaissance, en particulier pour tout le temps qu'ils m'ont consacré, ainsi que pour leurs commentaires bienveillants sur ce travail. Je souhaite également remercier chaleureusement Andrés Almansa, Cécile Louchet, Bernard Rougé et Pierre Weiss pour leur participation en tant qu'examinateurs à ce jury de thèse. Ayant déjà eu le privilège de pouvoir interagir avec chacun d'entre eux, que ce soit en suivant leurs enseignements, en collaborant sur des projets de recherche, ou encore en échangeant lors de passionnants séminaires scientifiques, je suis aujourd'hui très honoré de pouvoir soumettre mes travaux à leur jugement.

Les mots me manquent pour exprimer la profonde gratitude que je porte à l'égard de mon directeur de thèse, Lionel Moisan, qui m'a patiemment guidé et soutenu durant ces dernières années. Je ne saurais oublier la sympathie avec laquelle il m'a accueilli, ni comment il a su captiver mon intérêt en me proposant, dès notre première rencontre, de fascinants sujets de recherche. De par ses incroyables qualités pédagogiques, scientifiques et humaines, il a éclairé ce doctorat avec beaucoup de bienfaisance et m'a transmis sa passion pour une recherche à la fois belle et rigoureuse. Je souhaite ardemment voir mûrir en moi une créativité et une ouverture d'esprit digne de ses enseignements.

Je tiens à remercier Annie Raoult et Fabienne Comte, respectivement ancienne et actuelle directrice du laboratoire MAP5, qui n'ont de cesse d'œuvrer pour le bien-être des membres de ce laboratoire. Je remercie également Christine Graffigne, directrice de l'UFR de Mathématiques et Informatique, ainsi que Thierry Raedersdorff et son équipe, pour l'aide qu'ils m'ont apportée à de multiples occasions. Un grand merci à Marie-Hélène Gbaguidi, gestionnaire du MAP5, dont l'efficacité n'a d'égale que sa bonne humeur, ainsi qu'à Christophe Castellani, Voehni Kheng, Marie Marduel, Clémence Missebouko, et Isabelle Valéro, pour leur précieuse aide à l'occasion de diverses aventures administratives.

J'adresse de très chaleureux remerciements à la pétillante équipe des matheux du département informatique de l'IUT Paris Descartes, Anne Estrade, Mélanie Gobert, Bérénice Grec, Sébastien Martin, Philippe Radi et Michel Sortais, pour l'excellent accueil qu'ils m'ont réservé ainsi que pour leur gentillesse tout au long de ma mission d'enseignement qui, grâce à eux, s'est déroulée dans les meilleures conditions. Je remercie également, sans pouvoir tous les citer, les enseignants-chercheurs que j'ai côtoyés au MAP5, Flora Alarcon, Florent Benaych-Georges,

Étienne Birmelé, Olivier Bouaziz, Charles Bouveyron, Maya de Buhan, Manon Defosseux, Julie Delon, Sylvain Durand, Jonathan El Methni, Céline Duval, Jean-Claude Fort, Bruno Galerne, Servane Gey, Joan Glaunès, Georges Kœpfler, Raphaël Lachieze-Rey, Christophe Pouzat et Edoardo Provenzi. La bonne humeur qui règne au MAP5 se nourrit grandement des nombreuses discussions amicales qu'ils partagent avec simplicité. Je tiens également à remercier Thierry Stœhr, membre de l'équipe SCRIPT de l'université Paris Diderot, pour m'avoir fait découvrir Orgmode.

En marge de cette thèse, j'ai eu l'immense chance de pouvoir suivre en mission à Saint-Malo et Pointe-à-Pitre une joyeuse équipe de biologistes marins et physiciens, tous aussi passionnants que passionnés. En souvenir de ces deux inoubliables missions dédiées à l'étude des gorgones, j'adresse mes sincères remerciements à Claude et Yolande Bouchon, Annemiek Cornelissen, Julien Derr, Stéphane Douady, Jérôme Fournier, Pascal Lopez et Lionel Moisan.

Que de bons moments passés avec plusieurs générations de dynamiques doctorants, post-doctorants, ATER, ingénieurs de recherche, ou stagiaires ! Un grand merci à Ardo Bar, Rebecca Bauer, Andréa Bondesan (promu au titre de responsable du guide d'accueil), Claire Bouchigny, Gaëlle Chagny, Ronan Costaouec, Axel Davy, Christophe Denis, Mariella Dimiccoli, Anne-claire Egloffe, Christèle Etchegaray, Diarra Fall, Noura Faraj, Julie Fournier, Oriel Frigo, Mélina Gallopin, Thierry Guillemot, Maud Kerebel, Kévin Kuoch, Charlotte Laclau, Loïc Lacouture, Gwennaëlle Mabon, Alkeos Michaïl, Mario Micheli, Cambyse Pakzad, Thomas Picchetti, Léo Planche, Jean Rochet, Samuel Ronsin, Sonia Tabti, Fabien Vergnet et Vincent Vidal. J'adresse des remerciements particuliers à Charlotte Dion, Anne-Sophie Macé et Pierre Roussillon pour leur investissement à l'égard de tous, Alasdair Newson (pour sa théorie de l'invariance des pâtes ainsi que sa relecture attentive du second chapitre de ce manuscrit), Fanny Doré (pour ses gâteaux ε-significativement fondants), et Arthur Leclaire (mon irremplaçable frère de thèse).

J'adresse un clin d'œil affectueux à quelques uns de mes proches et membres de ma famille, à David et Tania, Jade et Raja, Rachid et Noémie, Fukiko, à mes parents pour leur affection à toute épreuve, à mes frères et soeurs envers qui je porte beaucoup d'amour et d'admiration, à mes grands parents, à mes oncles, tantes et cousins, ainsi qu'à ma belle famille (française et japonaise), un grand merci à tous pour votre soutien inconditionnel.

Je réserve enfin mes pensées les plus tendres à mon adorable épouse Yui ainsi qu'à Joan qui grandit si vite et illumine mes jours de son merveilleux sourire.

Table des matières

Chapitre 1

Introduction

Contents

Les images sont omniprésentes dans notre quotidien, nous les percevons à l'aide de nos yeux et les analysons à l'aide de notre cerveau qui s'est entraîné depuis notre naissance à en extraire de l'information utile et à la traiter, permettant par exemple la reconnaissance de visages, de formes, ou l'estimation de la distance qui nous sépare d'un objet en mouvement, de sa vitesse, etc. Si la plupart du temps, l'exécution de ces tâches nous semble immédiate et ne conduit pas à un sentiment d'« effort perceptuel » intense, on estime que le cerveau humain y consacre plus de la moitié de sa capacité totale. Depuis l'avènement de l'ère numérique, le traitement d'images numériques s'est imposé comme un domaine à part entière du traitement du signal, en particulier du fait de la nature bien spécifique des signaux qu'il met en jeu. Ces dernières décennies, des modèles mathématiques ont été développés dans le but de représenter, comprendre et manipuler les images numériques afin de les transformer ou d'en extraire de l'information utile. On distingue deux niveaux de traitements, les traitements dits de

« haut niveau » dédiés à la réalisation de tâches complexes proches de la vision humaine (telles que la reconnaissance de visages, la détection et le suivi d'objets, le dénombrement d'individus dans une foule, etc.), ou les traitements dits de « bas niveau » qui se focalisent sur des « briques » de traitement apparaissant de manière plus systématique en traitement d'image (comme par exemple la suppression de bruit, la détection de formes, de contours, etc.) et pouvant être combinées pour effectuer des traitement plus complexes. L'intérêt de l'approche bas niveau est que les phénomènes mis en jeu sont plus simples à modéliser du point de vue physique et mathématique, ouvrant la voie à leur étude rigoureuse et poussée. Les travaux présentés dans cette thèse se situent plutôt du côté « bas niveau », son objectif étant le développement de modèles mathématiques ainsi que le développement d'algorithmes efficaces dédiés au traitement d'images numériques.

Note concernant les licences des images utilisées dans ce document

Sauf mention explicite, les expériences menées dans cette thèse ont été effectuées à partir d'images sous licence libre « Creative Commons Zéro » (CC0 Public Domain) provenant de la bibliothèque *Pixabay* (https://pixabay.com/).

1.1 Qu'est ce qu'une image numérique ?

Les images numériques sont les images dont l'acquisition est faite à l'aide d'un capteur numérique (comme par exemple un scanner, un appareil photo ou caméscope numérique, une carte d'acquisition vidéo, ...), ou encore les images synthétisées directement à l'aide de programmes informatiques (images de synthèse). Les images numériques sont en général stockées sous forme binaire sur un support informatique (carte SD, clé USB, disque dur) et représentées par des tableaux à plusieurs dimensions, ces dernières pouvant être de nature spatiale (longueur, largeur, profondeur pour les images 2D, 3D), mais aussi temporelle (on parle de signaux 2D+t ou 3D+t, où t désigne le temps, pour désigner un flux d'images 2D ou 3D), ou autres (par exemple des longueurs d'ondes dans le cas des images dites hyperspectrales). Nous nous restreindrons au cas des images à deux dimensions spatiales (longueur, largeur), représentées par des tableaux à deux dimensions. Dans ces tableaux, chaque case est appelée *pixel* de l'image et contient une information traduisant l'*intensité lumineuse* de la scène à une position spatiale donnée. Mathématiquement, on les représente comme des fonctions

à deux variables du type

$$u : \begin{pmatrix} \Omega \subset \mathbb{Z}^2 & \to & \mathbb{R}^d \\ (x,y) & \mapsto & u(x,y) \end{pmatrix} \tag{1.1}$$

où $\Omega = \{0,\dots,M-1\} \times \{0,\dots,N-1\}$ est un rectangle de $\mathbb{Z}^2$ de dimensions $M \times N$ appelé *domaine de l'image* et d désigne la dimension de l'espace utilisé pour représenter les intensités lumineuses. En général, on prend $d = 1$ pour représenter les images *en niveaux de gris* (un exemple est proposé en Figure 1.1) et $d = 3$ pour les images couleurs (nous nous restreindrons cependant dans cette thèse au cas des images en niveaux de gris).

Si le processus de numérisation par le biais d'un système d'acquisition (appareil photo numérique, scanner, etc.) permet le stockage et la manipulation des images, il introduit également un certain nombre d'erreurs pouvant conduire à une image numérique ne rendant pas fidèlement compte de la scène réellement observée. Ces erreurs peuvent être modélisées mathématiquement d'autant plus précisément que notre connaissance du système d'acquisition, mais aussi de la scène observée, est grande. Illustrons brièvement comment l'on peut modéliser l'acquisition d'une image numérique en niveaux de gris issue d'un appareil photo numérique classique. Cette description est très simplifiée et n'a pour objectif que de familiariser le lecteur avec le formalisme mathématique qui suivra. Soit Ω_c un ouvert de $\mathbb{R}^2$ et soit $U_* : \Omega_c \to \mathbb{R}$ telle que pour tout $(x,y) \in \Omega_c$, $U_*(x,y)$ représente l'intensité lumineuse de la scène au point (x,y). Nous parlerons d'*image continue* pour parler de U_* (par opposition à l'*image discrète* qui représentera l'image numérique issue de U_*).

Remarque. *Le choix de modéliser la scène par une fonction U_* définie sur $\mathbb{R}^2$ et non $\mathbb{R}^3$ peut sembler surprenant puisque la scène observée est a priori une scène en trois dimensions (longueur, largeur, profondeur). On peut en pratique voir U_* comme la projection dans un plan (typiquement le plan focal du capteur) d'une scène V_* à trois dimensions. Comme nous ne chercherons pas à reconstruire d'image en trois dimensions, nous ne nous intéresserons pas à ce type de modélisation.*

Supposons donc que l'on fasse l'acquisition de l'image U_* en une image numérique de taille $M \times N$, que l'on représente par un signal $u_0 : \Omega \to \mathbb{R}$. On peut alors modéliser l'étape d'acquisition en considérant que u_0 satisfait

$$\forall (k,l) \in \Omega, \quad u_0(k,l) = KU_*(k,l) + \varepsilon(k,l)\,, \tag{1.2}$$

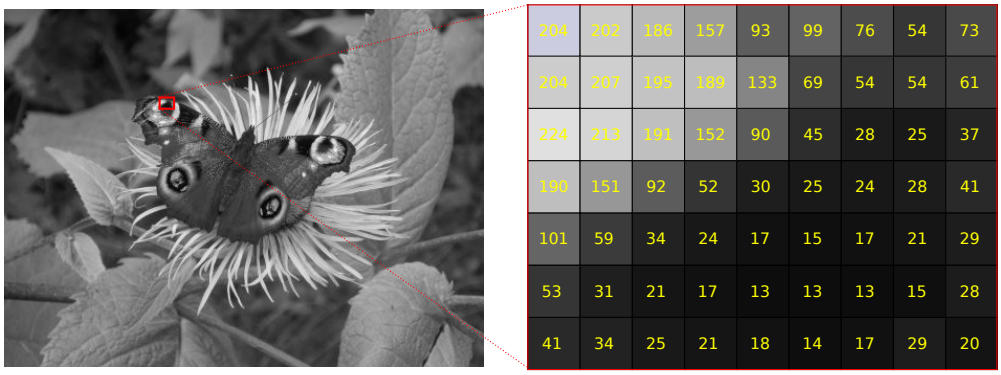


Figure 1.1: Représentation sous forme d'un tableau d'une image en niveaux de gris. Nous affichons dans la partie gauche de cette Figure une image en niveaux de gris. Cette image de largeur M et de hauteur N peut être représentée par un signal $u : \Omega \to \mathbb{R}$ (c'est-à-dire du type (1.1) pour $d = 1$), où Ω désigne un rectangle discret de taille $M \times N$. Par convention, le pixel de coordonnées $(0, 0)$ correspond au coin supérieur gauche de l'image, tandis que le pixel de coordonnées $(M - 1, N - 1)$ correspond au coin inférieur droit de l'image. Dans le cas des images acquises avec un appareil photo standard, les niveaux de gris $(u(k, l))_{(k,l)\in\Omega}$ sont souvent représentés par des entiers positifs codés sur 8-bits, ce qui signifie que les valeurs $u(k, l)$ sont des entiers compris entre 0 et 255 (0 représentant le noir et 255 le blanc). Dans la partie droite de cette Figure, nous indiquons les niveaux de gris de u sur un sous-ensemble de Ω représenté par le rectangle rouge dans l'image.

où $\varepsilon : \Omega \to \mathbb{R}$ représente un bruit de mesure (voir Figure 1.2) et K est un opérateur modélisant des phénomènes de distorsions (souvent propres à la physique du capteur) qui surviennent pendant le processus d'acquisition. Souvent, on considère que les $(\varepsilon(k, l))_{(k,l)\in\Omega}$ sont des variables aléatoires indépendantes et identiquement distribuées selon une loi gaussienne. Ce choix nous amène dans un cadre mathématique standard, mais constitue en pratique une approximation discutable. Dans le cas des images numériques acquises à l'aide de capteurs standards (type CMOS « Complementary Metal-Oxide-Semiconductors » ou CCD « Charge-Coupled Devices »), une étude plus précise du bruit ainsi que des phénomènes physiques liés au capteur conduit à des modèles très différents (voir par exemple [Aguerrebere et al. 2012]). Par ailleurs, on peut décomposer l'opérateur K de manière simplifiée en

$$K = K_{\text{quantif.}} \circ K_{\text{échant.}} \circ K_{\text{dist.}} \circ K_{\text{diffrac.}} \tag{1.3}$$

où

- $K_{\text{diffrac.}} : \mathbb{R}^{\Omega_c} \to \mathbb{R}^{\Omega_c}$ modélise le phénomène physique de diffraction de la lumière qui se produit lorsque les rayons lumineux atteignent le *diaphragme*

Figure 1.2: Bruit de mesure des niveaux de gris d'une image. En regardant de près l'image de gauche (on affiche à droite un grossissement de pixels contenus dans le rectangle rouge de l'image de gauche), il semble que les niveaux de gris mesurés ne traduisent pas exactement la réalité de la scène dont ils sont issus (on observe en effet des variations significatives de niveaux de gris dans des régions manifestement uniformes où l'on s'attend à observer des variations plus régulières des niveaux de gris). En pratique le processus d'acquisition introduit du bruit de mesure qui se traduit par des perturbations aléatoires sur les niveaux de gris mesurés.

de l'appareil photo (ce dernier constitue un obstacle aux rayons lumineux, n'en laissant passer qu'une partie). Du fait de ce phénomène de diffraction, un point lumineux sera vu comme une tâche (appelée tâche de diffraction) par le capteur. Ce terme introduit donc un effet de flou dans l'image acquise par le système. Ce flou est inévitable et constitue une limitation intrinsèque du système optique.

- $K_{\text{dist.}} : \mathbb{R}^{\Omega_c} \to \mathbb{R}^{\Omega_c}$ modélise les phénomènes de distorsions géométriques principalement introduits lors du passage des rayons lumineux dans les lentilles du système optique (aberrations chromatiques, vignettage, etc.)
- $K_{\text{échant.}} : \mathbb{R}^{\Omega_c} \to \mathbb{R}^{\Omega}$ modélise le processus d'échantillonnage de l'image U_* en un nombre fini d'échantillons. Cet opérateur transforme le signal U_* à support continu Ω_c en un signal à support discret Ω. Par exemple dans le cas d'un appareil photo muni d'un capteur à transfert de charge (CCD), chaque valeur du signal $K_{\text{échant.}}U_* : \Omega \to \mathbb{R}$ représente la valeur moyenne du signal U_* sur le support d'une cellule photosensible.
- $K_{\text{quantif.}} : \mathbb{R}^{\Omega} \to X^{\Omega}$ où $X = \{x_0, \dots, x_n\} \subset \mathbb{R}$ désigne un ensemble fini de réels. Cet opérateur modélise l'étape de quantification effectuée lors du codage de l'image en données binaires (seuls un nombre fini de réels peuvent être représentés sur machine).

On peut de plus ajouter d'autres opérateurs, en amont ou en aval de la chaîne (1.3), pour modéliser d'autres phénomènes, comme par exemple un *flou de bougé* permettant de modéliser un mouvement du capteur lors de l'acquisition, mais aussi bien d'autres types de distorsions. Le principal problème qui nous intéressera sera la reconstruction (ou l'estimation) de U_* (ou plus précisément d'une représentation discrète de U_*) à partir de u_0 en considérant des modèles semblables à (1.2), ce type de problème étant qualifié de « problème inverse » (voir Figure 1.3).

Figure 1.3: Quelques exemples de problèmes inverses. Nous illustrons ici quelques exemples d'images synthétisées à l'aide du modèle (1.2) à partir d'une image de bonne qualité. Les dégradations modélisées ici sont les suivantes. L'image de gauche souffre principalement de la présence d'un fort bruit de mesure. L'image du milieu d'un flou de bougé qui modélise un mouvement de la caméra pendant l'acquisition de l'image. L'image de droite souffre de sous-échantillonnage, ce qui modélise un nombre trop faibles de capteurs photosensibles par rapport au niveau de détails de la scène.

1.2 La variation totale

La variation totale a été utilisée en traitement d'image pour la première fois par Rudin, Osher, et Fatemi [1992]. Dans cet article, le problème considéré est formulé dans un cadre continu (les images U sont définies sur un ouvert Ω_c de $\mathbb{R}^2$), et la variation totale (TV) est introduite sous sa forme forte, c'est-à-dire définie par

$$\forall U \in E, \quad \mathrm{TV}(U) = \int_{\Omega_c} \sqrt{(\partial_1 U(x,y))^2 + (\partial_2 U(x,y))^2}\, dx\, dy\,, \tag{1.4}$$

où E désigne l'espace vectoriel composé des images $U : \Omega_c \to \mathbb{R}$ admettant des dérivées partielles $\partial_1 U$ et $\partial_2 U$ sommables sur Ω_c (c'est-à-dire $\partial_1 U, \partial_2 U \in \mathcal{L}^1(\Omega_c)$). Nous verrons au Chapitre 2 que cette définition peut-être étendue sur un espace beaucoup plus grand (et moins régulier) que l'espace E, appelé espace des fonctions à variations bornées (et noté $\mathrm{BV}(\Omega_c)$). On considère que l'on dispose d'une

version imparfaite $U_0 : \Omega_c \to \mathbb{R}$ d'une image parfaite $U_* : \Omega_c \to \mathbb{R}$ satisfaisant

$$\forall (x,y) \in \Omega_c, \quad U_0(x,y) = U_*(x,y) + \varepsilon(x,y)\,, \tag{1.5}$$

où $\varepsilon = U_0 - U_*$ représente un bruit additif (en fait un champ aléatoire gaussien), tel que $\int_{\Omega_c} \varepsilon(x,y)^2\,dx\,dy = \sigma^2$. Nous cherchons à reconstruire (ou estimer) U_* à partir de U_0. L'approche adoptée par Rudin, Osher, et Fatemi [1992] consiste à rechercher une image $\widetilde{U} : \Omega_c \to \mathbb{R}$ de variation totale minimale parmi celles appartenant à l'ensemble $\mathscr{C}_0$ défini par

$$\mathscr{C}_0 = \Big\{ U \in \mathrm{BV}(\Omega_c), \quad \int_{\Omega_c} U(x,y) - U_0(x,y)\,dx\,dy = 0\,, \\ \text{et} \quad \int_{\Omega_c} (U(x,y) - U_0(x,y))^2\,dx\,dy = \sigma^2 \Big\}\,.$$

Plus formellement, cela revient à chercher $\widetilde{U} \in \mathscr{C}_0$ telle que (en admettant qu'une telle image existe)

$$\mathrm{TV}(\widetilde{U}) = \inf_{U \in \mathscr{C}_0} \mathrm{TV}(U)\,. \tag{1.6}$$

Commençons par analyser le problème ainsi formulé. En considérant (1.6), nous cherchons à satisfaire deux critères :

(i) **un critère de « fidélité aux données »** : en se restreignant à l'ensemble $\mathscr{C}_0$, on impose une certaine ressemblance entre $\widetilde{U}$ et U_0. Plus précisément, on ne considère que des images U de même moyenne que U_0, et dont le carré de la distance à U_0 (en norme $\mathcal{L}^2$) est égale à σ^2. On impose ainsi aux images de $\mathscr{C}_0$ de ressembler à U_0, en tenant compte du niveau de bruit présent dans U_0 (quand σ est petit, le bruit qui entache U_0 est faible, et donc U_0 est proche de l'image parfaite U_*, mais quand σ augmente, le bruit qui dégrade U_0 devient important, la distance entre U_0 et U augmente) ;

(ii) **un critère de « régularité »** : parmi les éléments de $\mathscr{C}_0$, nous recherchons une image qui admette une petite variation totale, ce qui permet de discriminer les éléments de $\mathscr{C}_0$ selon leur régularité. Nous expliquerons au chapitre 2 l'intérêt de ce choix, contentons nous à ce stade de remarquer que (1.4) promeut les images U à faible gradient $\nabla U := (\partial_1 U, \partial_2 U)$ puisque (1.4) n'est autre que l'intégrale sur Ω_c de la fonction $(x,y) \mapsto \|\nabla U(x,y)\|_2$ (en notant $\|\cdot\|_2$ la norme euclidienne dans $\mathbb{R}^2$). Ainsi, une image très oscillante (typiquement, une image bruitée) présentera une grande variation totale, alors qu'une image à faible gradients (typiquement une image constante par

morceaux, à condition tout de même d'étendre la définition (1.4) à l'espace $\mathrm{BV}(\Omega_c)$) aura une variation totale faible.

Ainsi, dans le problème (1.6), le critère (i) permet de tenir compte de la connaissance que l'on a sur les données (en particulier, la connaissance du niveau de bruit qui dégrade l'image U_0), tandis que le critère (ii) permet d'introduire un « a priori » sur l'image à reconstruire (ici, on recherche une image avec faible variation totale). Une manière plus moderne de tenir compte de ces deux critères consiste à considérer le problème de minimisation

$$\min_{U\in E} J_\lambda(U) := \frac{\|U-U_0\|_2^2}{2\sigma^2} + \lambda\,\mathrm{TV}(U) \tag{1.7}$$

où $\|U-U_0\|_2^2$ désigne le carré de la distance $\mathcal{L}^2$ entre U et U_0, et λ un réel positif. La recherche d'un minimiseur de J_λ peut également être interprétée comme la recherche d'un compromis entre fidélité aux données (le terme de distance pénalise les images trop éloignées de U_0) et régularité (le terme de variation totale pénalise les images trop irrégulières). Le paramètre λ permet de régler le poids relatif entre les deux termes mis en jeu dans cette énergie. Une des raisons du succès du modèle (1.7) est qu'il peut être interprété de manière élégante comme un problème de Maximum A Posteriori (MAP) en adoptant un point de vue bayesien, comme nous le verrons au Chapitre 2. Notons qu'il est de plus possible de montrer qu'il existe une valeur de λ telle que les problèmes (1.6) et (1.7) soient équivalents (ce qui signifie qu'ils admettent les mêmes solutions), mais en pratique le réglage du paramètre λ dans le modèle (1.7) est laissé à l'appréciation de l'utilisateur, conduisant à une version relaxée du problème (1.6) au sens où la solution obtenue par minimisation de J_λ n'appartient pas forcément à l'ensemble des contraintes $\mathscr{C}_0$.

Revenons au problème (1.6) et remarquons qu'il est formulé dans un cadre continu alors que les images traitées sont en pratique discrètes. La principale raison justifiant ce choix est qu'il nous ramène dans un cadre mathématique usuel, dans lequel nous disposons d'outils permettant la résolution de (1.6). En particulier, la méthode proposée dans [Rudin et al. 1992] consiste à résoudre une équation aux dérivées partielles modélisant l'évolution temporelle d'une copie de l'image initiale U_0 vers une solution de (1.6). Cette équation aux dérivées partielles est alors discrétisée à l'aide de schémas numériques classiques (de type Euler-explicite), en particulier, tous les opérateurs de dérivées partielles ainsi mis en jeu sont remplacés par des schémas aux différences finies. Outre les difficultés inhérentes à l'étape de discrétisation des équations, il est important de souligner que cette approche n'est en pratique pas rigoureusement exacte puisqu'elle débute

dans [Rudin et al. 1992] par l'écriture d'une équation d'Euler-Lagrange associée au problème de minimisation sous contraintes (1.6) qui suppose que la variation totale est une fonctionnelle différentiable, ce qui n'est pas le cas (on peut voir $\mathrm{TV}(U)$ comme une norme $\mathcal{L}^1$ du gradient de U, or la norme $\mathcal{L}^1$ présente une singularité en 0). Néanmoins, ce modèle de minimisation de la variation totale a connu un franc succès dans le domaine de la restauration d'images, en particulier du fait de la capacité de cette fonctionnelle à pénaliser les irrégularités (donc le bruit) tout en préservant les contours de l'image, comme nous l'expliquerons au Chapitre 2.

Depuis Rudin, Osher, et Fatemi [1992], l'utilisation de la variation totale en traitement d'image s'est développée bien au-delà du seul cadre initial de la restauration d'images, elle est désormais couramment utilisée pour effectuer des tâches de restauration diverses, telles que le déflouage [Vogel et Oman 1998], la désocclusion (ou « inpainting ») [Chan et al. 2005], l'interpolation [Guichard et Malgouyres 1998], l'extrapolation de spectre [Rougé et Seghier 1995], la décomposition d'images [Aujol et al. 2005], la décompression [Alter et al. 2005], l'échantillonnage irrégulier [Almansa et al. 2006], la super-résolution [Babacan et al. 2008], la stéréovision [Miled et al. 2009], ou même pour définir un indice de qualité image [Blanchet et Moisan 2012, Leclaire et Moisan 2015]. Par ailleurs, la découverte ces dernières années de nouveaux schémas performants [Chambolle 2004, Beck et Teboulle 2009a, Weiss et al. 2009, Chambolle et Pock 2011], reposant sur des méthodes duales [Rockafellar et Wets 1998, Ekeland et Témam 1999, Boyd et Vandenberghe 2004] et permettant une prise en compte rigoureuse de la non-différentiabilité de la variation totale ainsi que de la nature discrète des données, a considérablement enrichi le domaine et ouvert la voie à des applications où, comme dans [Fadili et Peyré 2011], la projection sur les fonctions à variation totale bornée n'est plus qu'une étape d'un algorithme itératif plus complexe.

Le Chapitre 2 de cette thèse constitue un travail de synthèse quant à l'utilisation de la variation totale en traitement d'image. Nous y expliquerons comment des problèmes de restauration (débruitage, déflouage, désocclusion, zoom, etc.) peuvent être formulés grâce au formalisme bayesien comme des problèmes de minimisations d'énergies impliquant la variation totale (similaires à (1.7)) et nous illustrerons la capacité de la fonctionnelle TV à préserver les contours dans les images. Nous détaillerons quelques outils d'analyse convexe basés sur la dualité de Legendre-Fenchel qui permettent la minimisation de fonctionnelles convexes non-différentiables, comme celles impliquant la variation totale. Nous présenterons entre autres les notions de transformées de Legendre-Fenchel, de sous-différentielle, ainsi que les opérateurs proximaux [Moreau 1965, Rockafellar

1976], enveloppes de Moreau-Yosida [Moreau 1963, Yosida 1980] et leurs principales propriétés. Ces opérateurs sont à la base des schémas modernes d'optimisation convexe non-différentiable, dont nous présenterons l'un des plus connus, l'algorithme de Chambolle et Pock [2011]. Nous utiliserons ensuite ces outils d'analyse convexe pour effectuer différentes tâches de restauration d'images (débruitage, déflouage, désocclusion, zoom), impliquant la minimisation de la variation totale ou l'une de ses variante (la variation totale d'Huber). Nous montrons comment tous ces problèmes peuvent être reformulés sous la forme d'un problème de recherche de point selle (dit problème « primal-dual »), pouvant être efficacement traité en utilisant l'algorithme de Chambolle et Pock [2011]. Nous expliquerons aussi comment formuler le dual d'un problème d'optimisation quelconque, et détaillons le lien existant entre les solutions du problème de départ (le problème primal), avec celles du problème dual ainsi formulé.

1.3 La variation totale « Shannon »

Si depuis Rudin et al. [1992], les méthodes mathématiques ainsi que les algorithmes permettant la minimisation des problèmes du types (1.6) et (1.7) ont connu de remarquables progrès, la manière de discrétiser la variation totale reste encore essentiellement basée sur des schémas aux différences finies. C'est-à-dire qu'étant donnée une image $u \in \mathbb{R}^{\Omega}$ définie sur un domaine discret $\Omega \subset \mathbb{Z}^2$, on considère généralement un schéma du type

$$\forall (k,l) \in \Omega, \quad \begin{cases} \partial_1 u(k,l) & = u(k+1,l) - u(k,l) \\ \partial_2 u(k,l) & = u(k,l+1) - u(k,l) \end{cases} \tag{1.8}$$

en se donnant une convention au niveau des bords de l'image (par exemple $u(k+1,l) = u(k,l)$ si $(k+1,l) \notin \Omega$, et $u(k,l+1) = u(k,l)$ si $(k,l+1) \notin \Omega$, mais d'autres choix sont possibles). Une fois le schéma aux différences finies choisi, on définit, par analogie avec (1.4), la variation total de l'image discrète u par

$$\mathrm{TV}^{\mathrm{d}}(u) = \sum_{(k,l)\in\Omega} \sqrt{\left(\partial_1 u(k,l)\right)^2 + \left(\partial_2 u(k,l)\right)^2}\,, \tag{1.9}$$

ou encore (en remplaçant la norme ℓ^2 du gradient discret $(\partial_1 u, \partial_2 u)$ par une norme ℓ^1),

$$\mathrm{TV}^{\mathrm{d}}_1(u) = \sum_{(k,l)\in\Omega} \left|\partial_1 u(k,l)\right| + \left|\partial_2 u(k,l)\right|, \tag{1.10}$$

comme c'est le cas dans [Chambolle 2005, Darbon et Sigelle 2006] où les problèmes d'optimisation impliquant $\mathrm{TV}_1^{\mathrm{d}}$ sont traités entre autres avec des méthodes de type « graph-cuts », ou encore dans [Louchet et Moisan 2014, Abergel et al. 2015] pour définir un nouveau modèle de restauration appelé TV-ICE (Iterated Conditional Expectation) qui sera développé à la section 1.4 de cette introduction, ainsi qu'au Chapitre 4 de cette thèse. On qualifie souvent le modèle (1.10) d'« anisotrope » du fait de l'utilisation de la norme ℓ^1 du gradient, par opposition au modèle (1.9) qui utilise la norme ℓ^2 (plus isotrope). Il est pourtant amusant de voir qu'aucun de ces deux modèles n'est isotrope dès lors qu'ils sont combinés avec le schéma aux différences finies (1.8), au sens où pour une image quelconque $u \in \mathbb{R}^\Omega$, on a en général

$$\mathrm{TV}^{\mathrm{d}}(u) \neq \mathrm{TV}^{\mathrm{d}}(Ru)\,, \tag{1.11}$$

en notant R un opérateur de rotation d'angle $\pi/2$. Par conséquent, la restauration d'une image u_0 par minimisation d'énergie basée sur TV^{d} sera en générale différente de celle obtenue en traitant de la même manière l'image Ru_0 puis en appliquant la rotation inverse au résultat. Ce manque d'isotropie est identifié dans [Lai et al. 2009, Wang et Lucier 2011, Chambolle et al. 2011, Condat 2016], où de nouveaux schémas d'approximation du gradient sont proposés.

L'anisotropie introduite par l'utilisation par TV^{d} de schémas aux différences finies n'est pas le seul défaut que l'on peut opposer à ce mode de discrétisation de la variation totale. En effet, le principal défaut des méthodes aux différences finies réside dans le fait qu'elles opèrent à l'échelle du pixel : si l'on considère notre image discrète $u : \Omega \subset \mathbb{Z}^2 \to \mathbb{R}$ comme la restriction d'une image continue $U : \mathbb{R}^2 \to \mathbb{R}$ au domaine discret Ω, alors les dérivées partielles de U (en supposant qu'elles existent) satisfont

$$\forall (x,y) \in \mathbb{R}^2, \quad \begin{cases} \partial_1 U(x,y) &= \lim\limits_{h\to 0} \dfrac{U(x+h,y)-U(x,y)}{h} \\ \partial_2 U(x,y) &= \lim\limits_{h\to 0} \dfrac{U(x,y+h)-U(x,y)}{h} \end{cases}$$

et peuvent être effectivement approchées par le schéma

$$\forall (x,y) \in \mathbb{R}^2, \quad \forall h > 0, \quad \begin{cases} \partial_1^h U(x,y) &\approx \dfrac{U(x+h,y)-U(x,y)}{h} \\ \partial_2^h U(x,y) &\approx \dfrac{U(x,y+h)-U(x,y)}{h} \end{cases}$$

qui devient précis dès lors que h est assez petit. Malheureusement en image le pas de discrétisation h des données n'est en général pas contrôlé, on ne peut donc

pas faire tendre h vers 0 et le schéma (1.8) (qui correspond au choix $h = 1$) n'est pas assez précis. Plus grave encore, l'utilisation du schéma (1.8) pour estimer la norme du gradient d'une image introduit des phénomènes de repliement de spectre, comme illustré à la Figure 1.4.

(a) image originale (© CNES) (b) $\|\nabla u\|$, différences finies (c) $\|\nabla u\|$, calcul sous-pixellique

Figure 1.4: Repliement de spectre pour l'estimation de la norme du gradient par différences finies. Note : l'image de gauche utilisée pour cette expérience appartient au Centre National d'Étude Spatiales (CNES). On dispose d'une image (a) dont on cherche à estimer en chaque pixel (k, l) la norme ℓ^2 du gradient $\|\nabla u(k, l)\|$. Cette norme est estimée en (b) à l'aide du schéma aux différences finies (1.8), et en (c) en utilisant le même schéma mais en remplaçant l'image u par une version sur-échantillonnée d'un facteur 2 (zoom par zero-padding). Pour les images (b) et (c), le code couleur va du blanc (faibles valeurs) au noir (valeurs élevées). Les régions de forts gradients étant censées correspondre à des contours de l'image, on s'attend à trouver les contours de l'image u dans l'image $\|\nabla u\|$, on observe cependant que les contours correspondant aux rayures du toit du bâtiment dans l'image (a) ne correspondent pas avec ceux observés dans l'image (b), mais correspondent bien à ceux de l'image (c). Cet exemple surprenant s'explique par un phénomène de repliement de spectre dans l'image (b). Il illustre l'incompatibilité entre la théorie de l'échantillonnage de Shannon avec l'estimation de $\|\nabla u\|$ par le schéma (1.8) (sans sur-échantillonnage), ainsi que la nécessité de manipuler les images à une échelle sous-pixellique afin d'estimer correctement cette quantité.

Comment peut-on manipuler une image discrète à l'échelle sous-pixellique ? La manière usuelle consiste à utiliser une interpolation, c'est à dire qu'à partir des échantillons $(k, l) \in \Omega \mapsto u(k, l)$, on reconstruit une image continue $U_\varphi : \mathbb{R}^2 \mapsto \mathbb{R}$ définie le plus souvent à l'aide d'un noyau d'interpolation $\varphi : \mathbb{R} \to \mathbb{R}$ par

$$\forall (x, y) \in \mathbb{R}^2, \quad U_\varphi(x, y) = \sum_{(k,l)\in\Omega} u(k, l)\, \varphi(x - k)\, \varphi(y - l)\,. \tag{1.12}$$

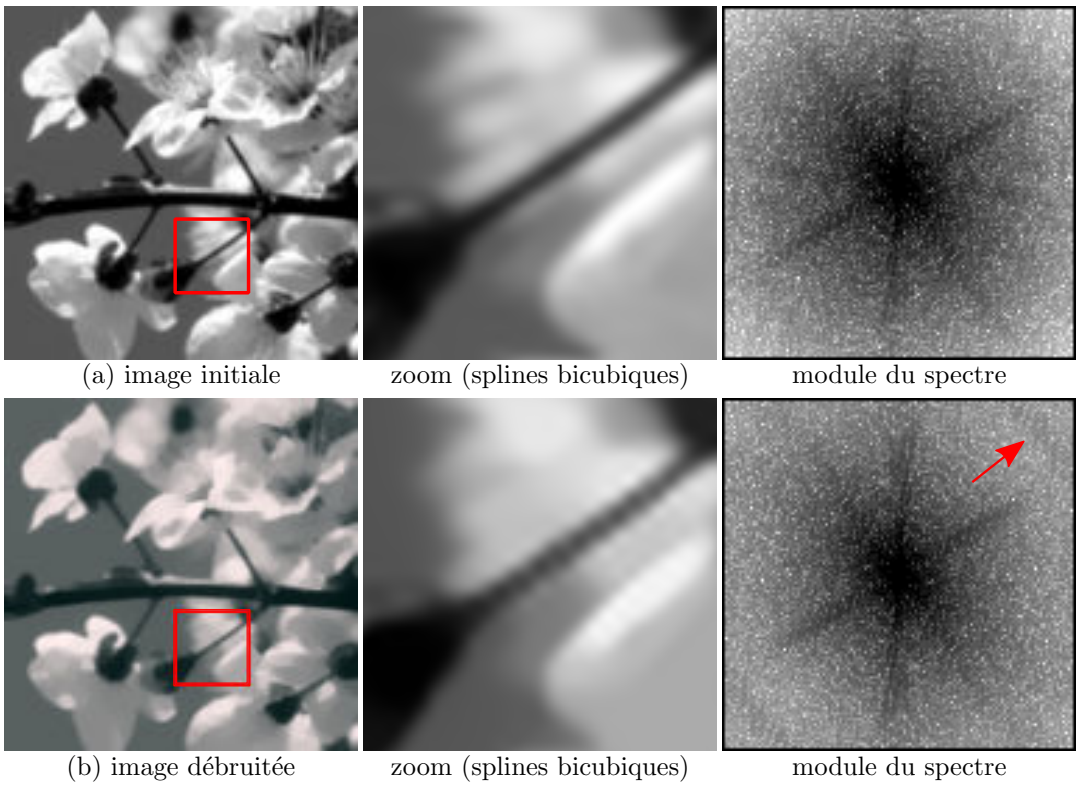
Plusieurs noyaux d'interpolations sont explicités dans la littérature (voir par

exemple [Unser 2000, Thévenaz et al. 2000]), menant chacun à une méthode d'interpolation. Parmi les méthodes d'interpolations les plus connues, on compte l'interpolation par plus proche voisin, l'interpolation bilinéaire, l'interpolation par « bicubic Keys », les interpolations par splines, et l'interpolation par sinus cardinal discret (ou interpolation de Shannon), cette dernière étant héritée de la théorie d'échantillonnage de Shannon (nous la présenterons en détail au Chapitre 3).

Du point de vue du traitement des images numériques, on peut facilement mettre en évidence le fait qu'une image restaurée par minimisation d'énergie impliquant TV^{d} est difficile à interpoler, ou formulé différemment, mal échantillonnée du point de vue de la théorie de Shannon. Une illustration de ce phénomène est proposée à la Figure 1.5, où l'on met en évidence le fait qu'une image débruitée à l'aide de la variation totale discrète ne peut pas être correctement manipulée à l'échelle sous-pixellique au sens où l'interpolation (ici par splines) de cette image ne constitue pas une estimation plausible de l'image à l'échelle sous-pixellique. Ceci limite donc fortement la possibilité d'utiliser les modèles TV^{d} (par exemple le débruitage) comme étape intermédiaire d'une chaîne de traitement plus complexe, impliquant des manipulations sous-pixelliques de l'image telles que des transformations géométriques (translations, rotations, agrandissement), du recalage, de la reconnaissance de formes, etc.

Le Chapitre 3 de cette thèse est consacré à l'étude d'un autre modèle de discrétisation de la variation totale, différent de (1.9). Cette variante, que nous appelons la « variation totale Shannon » (ou STV, pour « Shannon Total Variation »), apparut pour la première fois dans [Malgouyres et Guichard 2001] avant d'être explicitement considérée dans [Moisan 2007] puis utilisée dans [Facciolo et al. 2009, Preciozzi et al. 2014] sous le nom de « Spectral Total Variation » (nous ne retiendrons néanmoins pas ce nom pour éviter les confusions avec les travaux de Gilboa [2013]). Elle consiste en l'estimation par une somme de Riemann de la variation totale exacte (c'est à dire l'intégrale de la norme du gradient) de l'interpolée de Shannon de l'image discrète. Nous montrerons en particulier que l'utilisation de cette variante à la place de TV^{d} permet de produire des images pouvant être interpolées (que se soit à l'aide de splines ou par le sinus cardinal discret) de manière satisfaisante, c'est-à-dire sans artefacts, tout en assurant un niveau de restauration similaire à ceux obtenus avec TV^{d}. En ce sens, le modèle STV ainsi proposé permet de réconcilier la minimisation de la variation totale avec la théorie d'échantillonnage de Shannon. Nous proposerons aussi de nouveaux modèles de restauration d'images utilisant STV et permettant de générer à partir d'une image mal échantillonnée, une image visuellement très proche mais mieux échantillonnée et donc plus facilement interpolable. En ce sens, le modèle

STV rend également possible l'interpolation de Shannon qui est souvent délaissée au profit des splines malgré ses propriétés intéressantes.



(a) image initiale zoom (splines bicubiques) module du spectre

(b) image débruitée zoom (splines bicubiques) module du spectre

Figure 1.5: La variation totale discrète TV^{d} génère du repliement de spectre. Une image (a) est débruitée en utilisant la variation totale discrète TV^{d} définie par (1.9), l'image ainsi produite est affichée en (b). Pour chacune de ces images, on affiche en seconde colonne un agrandissement (interpolation par splines bicubiques) d'une sous partie de l'image (délimitée par le rectangle rouge). On voit que l'agrandissement effectué sur l'image (b) met en évidence la présence d'oscillations parasites, ce qui n'est pas le cas pour l'image (a). Ceci s'explique en regardant la dernière colonne, dans laquelle on affiche le module du spectre (plus précisément, on affiche $\log(1+|\widehat{u}|)$, en notant $|\widehat{u}|$ le module de la transformée de Fourier discrète de l'image u) des image (a) et (b). On observe en effet la présence de repliement fréquentiel (indiqué par une flèche rouge) dans le spectre de l'image (b) mais pas dans celui de l'image (a). Ce repliement fréquentiel a donc été introduit pendant l'étape de débruitage, il s'avère être en pratique responsable des oscillations sous-pixelliques observées dans le domaine spatial. Cette expérience illustre la difficulté rencontrée lorsque l'on souhaite manipuler une image à l'échelle sous-pixellique après un traitement impliquant TV^{d}.

1.4 Restauration en présence de bruit de Poisson

Le bruit de Poisson modélise le phénomène de comptage aléatoire des photons qui frappent le capteur pendant le processus d'acquisition, on parle aussi de « bruit de photon ». Contrairement aux autres sources de bruits évoquées à la section 1.1, ou décrites de manière plus complète dans [Aguerrebere et al. 2012], le bruit de photon est inhérent à la nature quantique de la lumière, et ne peut-être atténué par des améliorations technologiques au niveau du capteur. Étant donné un domaine fini $\Omega \subset \mathbb{Z}^2$, et une image $u : \Omega \to \mathbb{R}_+$ (non directement observable) décrivant l'intensité lumineuse de la scène, on considère que l'on observe une image $u_0 : \Omega \to \mathbb{N}$ de probabilité

$$p(u_0|u) = \prod_{(x,y)\in\Omega} \frac{u(x,y)^{u_0(x,y)}}{u_0(x,y)!} e^{-u(x,y)} \propto \exp\left(-\langle u - u_0 \log u, \mathbb{1}_\Omega \rangle\right) \tag{1.13}$$

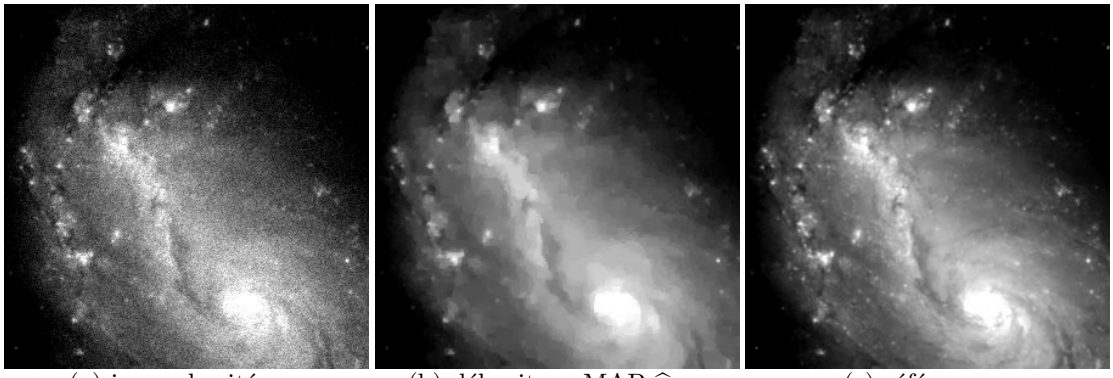
où $u_0 \log u$ désigne l'image obtenue en multipliant terme à terme u_0 avec $\log u$, en adoptant la convention $u_0(x,y)\log u(x,y) = 0$ dès lors que $u_0(x,y) = 0$, $\mathbb{1}_\Omega$ désigne l'image constante prenant la valeur 1 sur Ω, et $\langle \cdot, \cdot \rangle$ désigne le produit scalaire euclidien dans $\mathbb{R}^\Omega$. La notation $\propto$ indique une relation de proportionnalité (une égalité à une constante multiplicative près, pouvant dépendre de u_0 mais pas de u). Le modèle (1.13) consiste simplement à considérer que u_0 est la réalisation d'une image aléatoire $\mathbf{u_0}$ dont les niveaux de gris $\{\mathbf{u_0}(\mathbf{x},\mathbf{y})\}_{(x,y)\in\Omega}$ sont indépendants, et telle que pour tout pixel $(x,y) \in \Omega$, $\mathbf{u_0}(x,y)$ suit une loi de Poisson de paramètre $u(x,y)$. Ce modèle est particulièrement adapté lorsque l'acquisition des images est faite en basse lumière (ce qui signifie qu'une faible quantité de photons est captée par le dispositif d'imagerie), comme c'est par exemple souvent le cas en astronomie ou microscopie. Si la plupart des modèles mathématiques de restauration d'images sont d'abord développés dans le cadre plus usuel du bruit additif blanc gaussien, ils sont presque systématiquement adaptés au cadre du bruit de Poisson (voir par exemple [Setzer et al. 2010, Deledalle et al. 2010, Figueiredo et Bioucas-Dias 2010]) de par le caractère inévitable de ce bruit de photon. Dans le cas d'un bruit de Poisson, nous verrons que le MAP correspond à la recherche de (l'unique) minimiseur de l'énergie J'_λ définie par

$$\forall u \in \mathbb{R}^\Omega, \quad J'_\lambda(u) := \langle u - u_0 \log u, \mathbb{1}_\Omega \rangle + \lambda \, \mathrm{TV}^{\mathrm{d}}(u)\,, \tag{1.14}$$

qui n'est autre que l'image $\widehat{u}_{\text{MAP}}$ qui maximise la densité a posteriori π définie (à un facteur multiplicatif près) par

$$\forall u \in \mathbb{R}^{\Omega}, \quad \pi(u) \propto p(u_0|u)\, p(u)\,, \tag{1.15}$$

dès lors que l'on considère $p(u) \propto e^{-\lambda \mathrm{TV}^{\mathrm{d}}(u)}$ comme a priori sur les images naturelles (un tel choix revient à considérer comme plus vraisemblables les images u ayant une faible variation totale $\mathrm{TV}^{\mathrm{d}}(u)$). Cependant, comme dans le cas du bruit gaussien, le principal défaut de cette approche est que la minimisation de J'_λ (ou de manière équivalente, la maximisation de la densité à posteriori π) a tendance à conduire à des images constantes par morceaux (ce phénomène, appelé « staircasing », est illustré à la Figure 1.6 et est formellement mis en évidence par Nikolova [2000] qui démontre que ce phénomène est dû à la non-différentiabilité en 0 de l'énergie que l'on minimise).



(a) image bruitée u_0 (b) débruitage MAP $\widehat{u}_{\text{MAP}}$ (c) référence u

Figure 1.6: Débruitage TV-MAP d'une image dégradée par un bruit de Poisson. On affiche en (a) une observation bruitée (selon le modèle (1.13)) de l'image de référence (c) supposée sans défauts. On affiche en (b) l'image obtenue par calcul du MAP, c'est à dire l'image qui minimise l'énergie (1.14) (en réglant le paramètre λ de telle sorte à minimiser la I-divergence de Csiszar [1991], assurant ainsi un bon niveau de débruitage par rapport aux métriques usuelles). On observe que l'image (b) présente de large régions constantes et un faible niveau de détails par rapport à (a) et (c). Ce phénomène est caractéristique des approches TV-MAP impliquant la minimisation d'énergies basées sur la variation totale, on parle de « staircasing » (ou effet d'escalier). Note : l'image de référence (c) utilisée dans cette expérience est issue de la bibliothèque `wikimedia.org` (image NGC 1672 spiral galaxy).

Dans le cas d'un bruit gaussien, une alternative au modèle TV-MAP, appelée TV-LSE (Least Square Error), a été proposée par [Louchet et Moisan 2008]. Elle consiste à calculer au lieu du MAP, l'image qui minimise l'erreur quadratique

moyenne

$$\mathrm{EQM}(\widehat{u}) = \mathbb{E}_{u\sim\pi}\big(\|\widehat{u}-u\|_2^2\big) := \int_{\mathbb{R}^\Omega} \|\widehat{u}-u\|_2^2\,\pi(u)\,du\,.$$

Cette image n'est autre que l'espérance a posteriori, c'est à dire l'image $\widehat{u}_{\mathrm{LSE}}$ définie par

$$\widehat{u}_{\mathrm{LSE}} = \mathbb{E}_{u\sim\pi}(u) := \int_{\mathbb{R}^\Omega} u\,\pi(u)\,du\,, \tag{1.16}$$

qui peut être vue comme l'image moyenne suivant π parmi toutes les images de $\mathbb{R}^\Omega$. Malgré la dimension élevée de cet espace (si Ω est de taille 1000×1000, la dimension de l'espace $\mathbb{R}^\Omega$ est 10^6), il est possible d'estimer $\widehat{u}_{\mathrm{LSE}}$ numériquement à l'aide d'un schéma de Monte-Carlo par chaînes de Markov (MCMC) de type Metropolis-Hasting. Il est mis en évidence dans [Louchet et Moisan 2008] puis formellement démontré dans [Louchet et Moisan 2013] que les images générées par le modèle TV-LSE ne présentent pas d'effet de staircasing. On observe de plus que ce modèle mène a des images bien plus naturelles et riches en détails que le modèle TV-MAP. Si le modèle TV-LSE pourrait être facilement adapté au cas du bruit de Poisson, son principal point faible est que l'on ne dispose pas d'algorithme rapide pour estimer $\widehat{u}_{\mathrm{LSE}}$ (le schéma de Metropolis-Hasting présente un taux de convergence en $\mathscr{O}(1/\sqrt{N})$ où N désigne le nombre d'itérations), ce qui peut être problématique pour les applications où le nombre d'images à traiter est important ou lorsque le temps de calcul alloué au traitement des images est limité.

Afin de surmonter cette difficulté computationnelle, Louchet et Moisan [2014] proposèrent une nouvelle variante (toujours dans le cas d'un bruit gaussien), basée sur l'itération d'espérances conditionnelles a posteriori. L'estimateur $\widehat{u}_{\mathrm{ICE}}$ est défini comme la limite du schéma itératif

$$\forall (x,y)\in\Omega,\quad u^{n+1}(x,y) = \mathbb{E}_{u\sim\pi}(u(x,y)\,|\,u((x,y)^c) = u^n((x,y)^c))\,, \tag{1.17}$$

où $(x,y)^c = \Omega\setminus(x,y)$ et $u((x,y)^c)$ désigne la restriction de l'image u à l'ensemble $(x,y)^c$. Cette fois, l'espérance mise en jeu dans (1.17) est celle d'une variable aléatoire réelle et n'implique donc qu'une intégrale sur la droite réelle (au lieu de l'intégrale sur $\mathbb{R}^\Omega$ du modèle (1.16)). Il est alors mis en évidence que l'itération (1.17) peut-être calculée sous forme explicite dès lors que l'on remplace TV^{d} par sa version anisotrope $\mathrm{TV}^{\mathrm{d}}_1$ définie par (1.10). Il est démontré dans [Louchet et Moisan 2014] que le schéma (1.17) revient à construire u^{n+1} en appliquant à u^n une application F contractante, telle qu'il existe un compact $\mathcal{K}_{u_0}$ stable par F qui contienne u_0. Par conséquent, la convergence des itérés $(u^n)_{n\geq 0}$ vers l'image

$\widehat{u}_{\text{ICE}}$ est linéaire dès lors que l'on choisit $u^0 \in \mathcal{K}_{u_0}$ (par exemple en choisissant $u^0 = u_0$). Le modèle TV-ICE possède la remarquable propriété de ne n'introduire aucun paramètre algorithmique (en particulier il n'introduit aucun pas de temps contrairement aux algorithmes primaux-duaux classiques utilisés pour le calcul du MAP), et génère des images visuellement proches de celles délivrées par le modèle TV-LSE.

Au Chapitre 4, nous proposons d'adapter le modèle TV-ICE au cas du bruit de Poisson. Nous y démontrerons que, comme dans le cas gaussien, le schéma TV-ICE Poisson converge linéairement vers une image $\widehat{u}_{\text{ICE}}$ qui ne présente pas d'effet de staircasing. Nous montrerons que l'opérateur de récursion du modèle TV-ICE Poisson consiste à poser $u^0 = 0$ (c'est-à-dire $u^0(x, y) = 0$ pour tout $(x, y) \in \Omega$) puis à itérer pour $n \geq 0$ le schéma

$$\forall (x, y) \in \Omega, \quad u^{n+1}(x, y) = \frac{\sum_{1 \leq k \leq 5} c_k \, I_{a_{k-1}, a_k}^{\mu_k, u_0(x,y)+2}}{\sum_{1 \leq k \leq 5} c_k \, I_{a_{k-1}, a_k}^{\mu_k, u_0(x,y)+1}} \tag{1.18}$$

où les coefficients a_k, c_k, μ_k dépendent explicitement de la restriction de l'image courante u^n au 4-voisinage $\{(x \pm 1, y), (x, y \pm 1)\}$, ainsi que de λ et où l'on a posé

$$I_{a,b}^{\mu,p} = \int_a^b s^{p-1} \, e^{-\mu s} \, ds \,, \quad 0 \leq a \leq b \leq +\infty, \ \mu \in \mathbb{R}^*, \ p \in \mathbb{N}^* \,, \tag{1.19}$$

que nous appellerons fonction gamma incomplète généralisée (noter que le cas $b = +\infty$ n'est autorisé que lorsque $\mu > 0$, l'intégrale n'étant pas définie sinon). Nous expliquerons pourquoi le calcul du ratio de sommes de fonctions gamma incomplètes généralisées impliqué en chaque pixel de l'image par le schéma (1.18) est en pratique délicat à implémenter et détaillerons comment les erreurs peuvent être contrôlées en mettant les termes $I_{a,b}^{\mu,p}$ sous une forme mantisse-exposant du type $\rho \cdot e^{\sigma}$, l'estimation précise et rapide des intégrales $I_{a,b}^{\mu,p}$ sous cette représentation faisant l'objet du Chapitre 5.

1.5 L'importance des algorithmes rapides

Si l'approche du mathématicien face à un problème conduit souvent à la formulation puis l'étude théorique de nouveaux modèles mathématiques, les algorithmes développés pour résoudre ces problèmes de manière rigoureuse peuvent parfois s'avérer peu efficaces et dissuader de leur utilisation sur de vraies données. C'est typiquement le cas de l'approche TV-LSE présentée à la section 1.4 qui

constitue du point de vue mathématique une alternative originale et élégante au modèle MAP pour la restauration d'images conduisant à des résultats théoriques (comme par exemple l'absence formelle de « staircasing » dans l'image débruitée avec TV-LSE) qui traduisent une compréhension avancée des images produites avec ce modèle, mais dont le calcul pratique s'avère peu efficace du fait de la faible vitesse de convergence de la méthode MCMC mise en jeu. Dans le cas du modèle TV-LSE, nous avons vu qu'il était possible de repartir du problème mathématique pour en formuler une nouvelle variante, TV-ICE, menant à un algorithme beaucoup plus rapide (avec une vitesse de convergence linéaire par rapport au nombre d'itérations), tout en conservant des résultats théoriques du modèle TV-LSE (comme l'absence de staircasing).

Un autre exemple où un modèle mathématique mène à un algorithme intéressant mais peu efficace du point de vue computationnel est le modèle de détection de trajectoires régulières à partir d'une séquence de nuages de points, proposé par Primet [2011] durant sa thèse (voir aussi [Primet et Moisan 2012]), sous le nom de ASTRE (pour « A-contrario Smooth TRajectory Extraction »). Cet algorithme est basé sur la méthodologie « a contrario » développée par Desolneux, Moisan, et Morel [2008] (les concepts ont été introduits pour la première fois dans [Desolneux et al. 2000], voir aussi [Desolneux et al. 2003]), qui est basée sur la formulation mathématiques de grands principes de perception visuelle, dont le plus connu est le « principe de Helmoltz » (voir [Lowe 1985]). De manière informelle, ce principe stipule que le système perceptuel humain ne détecte que les structures (ou plus précisément les « gestalt », en référence aux travaux effectués par les « gestaltistes », du début du XX$^{\text{ième}}$ siècle tels que [Wertheimer 1923, Metzger 1975]) qui n'auraient pu apparaître *par hasard* dans un bruit blanc. Une illustration de ce principe est proposée en Figure 1.7. Basée sur le principe de Helmoltz, la méthodologie a contrario consiste à détecter des structures par rejet d'un modèle de hasard (ou modèle naïf) noté $\mathcal{H}_0$, autrement dit, on cherche a détecter les structures *trop rares* pour apparaître *par chance* dans $\mathcal{H}_0$.

Pour construire un détecteur a contrario, il faut dans un premier temps définir un modèle $\mathcal{H}_0$ décrivant ce que pourraient être des données tirées au hasard. Par exemple, si l'on considère le problème de détection d'alignements de points évoqué à la Figure 1.7, les données consistent en une seule image de domaine Ω contenant N points. On peut alors formuler le modèle $\mathcal{H}_0$ suivant : « les N points de l'image ont été tirés indépendamment selon une loi uniforme sur Ω ». Dans un second temps, on doit construire une fonction de mesure permettant de faire ressortir les structures rares dans $\mathcal{H}_0$. Toujours dans le cas du problème de détection des alignements de points, il est proposé dans [Desolneux et al. 2008] de

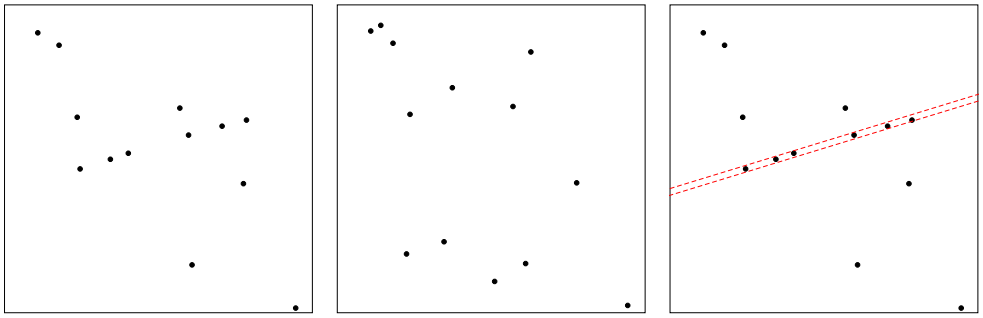


Figure 1.7: Illustration du principe de Hemholtz (cette Figure est partiellement tirée de [Primet 2011]). Pourquoi un alignement de points dans l'image de gauche nous saute-t-il aux yeux ? D'après le principe de Helmoltz, notre système perceptuel détecte cette structure car un tel alignement aurait peu de chances de se produire par hasard si les positions des points avaient été tirées « au hasard » dans cette image. Des alignements de points peuvent être trouvés dans l'image du milieu mais ils ne sautent pas aux yeux, car ils ne constituent pas des évènements rares dans du bruit. L'image de droite illustre l'approche décrite dans [Desolneux et al. 2008] pour formaliser en termes mathématiques l'effet de surprise associée à l'observation d'une structure donnée. En supposant que les positions des 13 points de cette image ont été tirées indépendamment et uniformément sur le domaine rectangulaire (ceci constitue notre modèle $\mathcal{H}_0$), on s'étonne de trouver 6 points dans une bande (délimitée par des pointillés rouges) de faible épaisseur. Ce niveau de surprise peut-être mesuré en calculant la probabilité qu'un tel évènement se produise dans $\mathcal{H}_0$.

découper l'image en bandes $\mathscr{B}$ de faible épaisseur (on discrétise les orientations des bandes et on note $\mathscr{B}_\Omega$ l'ensemble de toutes les bandes ainsi définies). On compte alors dans chaque bande $\mathscr{B}$ le nombre de points $n(\mathscr{B})$ contenus dans $\mathscr{B}$. Plus $n(\mathscr{B})$ est grand, plus on peut considérer comme rare dans $\mathcal{H}_0$ la structure correspondant au groupe de points contenu dans $\mathscr{B}$. Si $n(\mathscr{B})$ constitue une mesure intéressante sur les données, il serait néanmoins délicat d'essayer de la seuiller pour décider si les points contenus dans $\mathscr{B}$ doivent être détectés ou non comme des points alignés dans l'image. En effet, le choix d'un tel seuil promet d'être difficile car la quantité de surprise liée à l'observation d'un nombre de points $n(\mathscr{B})$ dans $\mathscr{B}$ reste très dépendante aux paramètres du problèmes (le nombre total de points N, la surface de la bande $\mathscr{B}$ et la surface du domaine de l'image Ω). Au lieu d'exploiter directement cette mesure, on l'utilise pour définir une famille de fonctions $\{\mathrm{NFA}_{\mathscr{B}}\}_{\mathscr{B}\in\mathscr{B}_\Omega}$ appelée Nombre de Fausses Alarmes (NFA) pour la mesure n. Celle-ci est définie par

$$\forall \mathscr{B} \in \mathscr{B}_\Omega, \quad \forall k \in [0, N], \quad \mathrm{NFA}_{\mathscr{B}}(k) = \#\mathscr{B}_\Omega \cdot \mathbb{P}_{\mathcal{H}_0}(n(\mathscr{B}) = k)\,, \tag{1.20}$$

où $\#\mathscr{B}_\Omega$ désigne le cardinal de l'ensemble $\mathscr{B}_\Omega$, c'est-à-dire le nombre total de

bandes dans l'image. Ce Nombre de Fausses Alarmes satisfait la propriété suivante

$$\mathbb{E}_{\mathcal{H}_0}(\#\{\mathscr{B} \in \mathscr{B}_\Omega,\ \mathrm{NFA}_{\mathscr{B}}(n(\mathscr{B})) \leq \varepsilon\}) \leq \varepsilon\,, \tag{1.21}$$

qui stipule que dans $\mathcal{H}_0$, on trouve en moyenne moins de ε bandes $\mathscr{B}$ satisfaisant $\mathrm{NFA}_{\mathscr{B}}(n(\mathscr{B})) \leq \varepsilon$. On décide de détecter les alignement par seuillage du NFA avec un seuil $\varepsilon > 0$, c'est-à-dire que l'on considère comme alignés les points contenus dans une bande $\mathscr{B}$ dès lors que $\mathrm{NFA}_{\mathscr{B}}(n(\mathscr{B})) \leq \varepsilon$, on dit alors que la bande $\mathscr{B}$ (ou le groupe de points qu'elle contient) est ε-significative. La propriété (1.21) stipule alors qu'en moyenne, moins de ε détections sont faites dans $\mathcal{H}_0$. Cela donne un sens concret au seuil ε, ce dernier représente un majorant du nombre moyen de détections autorisées dans du pur bruit $\mathcal{H}_0$, c'est-à-dire du nombre moyen de fausses détections. Pour construire à partir de (1.20) et (1.21) un algorithme concret de détection d'alignements de points, on adopte une démarche « gloutonne », qui consiste à retirer itérativement des données le groupe de points qui présente le NFA le plus faible (si ce dernier est inférieur à ε) et à recommencer jusqu'à ce qu'il ne reste plus de groupes de points de NFA inférieur à ε dans la séquence.

Depuis [Desolneux et al. 2000], des détecteurs a contrario ont été développés pour une grande quantité d'applications, telles que la détection de contours [Desolneux et al. 2001]), de segments [Von Gioi et al. 2008a,b], de jonctions [Xia et al. 2014], de spots (ou tâches) sur fond texturé [Grosjean et Moisan 2009], de changement sous-pixelliques dans des images radars [Robin et al. 2009, 2010], et bien d'autres encore [Rabin et al. 2009, Akinlar et Topal 2013]. Parmi les principaux atouts de ces détecteurs, on met souvent en avant leur robustesse au bruit, ainsi que le fait qu'ils ne nécessitent le réglage que d'un seul paramètre ε et ce réglage est particulièrement simple à la lumière de la propriété (1.21) puisque le paramètre ε représente un majorant du nombre de fausses détections autorisées. Un autre intérêt majeur de ce modèle est que la formule du NFA qu'il produit (par exemple (1.20) dans notre exemple de détection d'alignements) peut être utilisée pour filtrer les structures détectées à l'aide d'un autre détecteur, afin d'en éliminer les fausses détections. De manière plus générale, le NFA peut être utilisé pour quantifier la détectabilité d'une structure, au sens où, pour un seuil de détection ε donné (par exemple $\varepsilon = 1$ pour fixer les idées), toute structure ayant un NFA supérieur à 1 apparaîtra en moyenne environ 1 fois dans des données aléatoires $\mathcal{H}_0$, on peut alors dire qu'elle n'est pas détectable (au niveau de seuil $\varepsilon = 1$). L'étude du NFA conduit alors à des résultats très intéressants concernant la détectabilité des structures, on peut par exemple dans le cas des alignements de points s'intéresser au nombre minimal de points que doit contenir une bande

donnée $\mathscr{B}$ pour devenir détectable, on peut également étudier comment évolue ce nombre minimal de points en fonction du nombre total de points présents dans les données N, de l'épaisseur des bandes, de la taille du domaine Ω, etc. La méthodologie a contrario fait encore l'objet de recherches passionnantes, voir par exemple [Desolneux 2016, Desolneux et Doré 2016] où l'on s'intéresse à des modèles $\mathcal{H}_0$ plus riches que les modèles classiques, dans lesquelles on est capable d'assurer qu'une structure donnée n'est pas significative (cette problématique est également adressée dans la thèse de Doré [2014]).

Revenons à présent au problème de détection de trajectoires considéré par Primet [2011]. Dans ce travail, on considère comme point de départ la donnée d'une séquence de points préalablement détectés dans une séquence d'images. On considère donc un ensemble $\{f_1, \dots, f_K\}$ contenant K *frames* (le terme de frame désigne ici un ensemble de points issus de la détection effectuée sur une image de domaine Ω), tel que chaque frame f_k contienne N_k points (un point étant représenté par ses coordonnées dans Ω). Les points traduisent la présence d'objets dans la séquence d'image, néanmoins l'étape de détection de ces points étant imparfaite, il faut garder à l'esprit que ces données sont entachées d'erreurs : certains points de la séquence correspondent à de fausses détections, on parle de points aberrants, d'autres sont au contraire manquants, c'est à dire qu'ils n'ont pas été détectés dans certaines frames. On s'intéresse au problème de détection de trajectoires régulières (ou lisses) dans de telles données. Des problématiques similaires sont couramment considérées (incluant ou non l'étape de pré-détection des points) dans la littérature [Reid 1979, Bar-Shalom et al. 1983, Rangarajan et Shah 1991, Chetverikov et Verestoy 1999, Veenman et al. 2001, 2003, Bar-Shalom 2006, Fleuret et al. 2008, Berclaz et al. 2011]. De manière générale, ces méthodes sont souvent conçues dans un cadre assez restreint et nécessitent des modifications ad hoc successives pour traiter le cas général (présence de points aberrants, ou points manquants) que l'on considère ici. L'approche a contrario ASTRE proposée dans Primet [2011] se révèle extrêmement efficace pour traiter ce problème, mais conduit néanmoins à un algorithme dont la complexité en $\mathscr{O}(K^2)$ s'avère rédhibitoire pour traiter de longues séquences (typiquement dès lors que $K \geq 1000$ frames). Dans le Chapitre 6 de cette thèse, nous nous intéressons à une variante de ASTRE qui consiste à découper la séquence $\{f_1, \dots, f_K\}$ en sous-séquences de taille plus petites (avec recouvrement), et à traiter séquentiellement ces sous-séquences en définissant une stratégie de prolongement des trajectoires dans les zones de recouvrements entre les frames. Cette variante, appelée CU-TASTRE, conduit à un algorithme de complexité $\mathscr{O}(K)$, tout en conservant la propriété de NFA qui permet de contrôler le nombre de fausses détections. Cette

variante est d'ores et déjà utilisée par Dimiccoli et al. [2016] pour effectuer le suivi de particules fluorescentes dans de longues séquences d'images.

1.6 Organisation de la thèse

Nous proposons ci-dessous une synthèse de chaque chapitre composant cette thèse.

Chapitre 2

Ce chapitre constitue à la fois une introduction concernant l'utilisation de la variation totale en traitement d'image et un travail de synthèse de la littérature concernant les outils d'analyse convexe, en particulier de la dualité de Legendre-Fenchel, sur lesquels se basent les algorithmes modernes permettant la minimisation des fonctionnelles convexes non-différentiables qui mettent en jeu la variation totale. Après avoir détaillé ces différentes notions, nous les appliquons à divers problèmes classiques de traitement d'images.

Chapitre 3

Dans ce chapitre, nous étudions un modèle de discrétisation de la variation totale appelé variation totale Shannon (STV). Contrairement aux modèles de discrétisation classiques faisant appel à des schémas aux différences finies (TV^{d}), la variation totale Shannon consiste en l'estimation par une somme de Riemman de la variation totale continue de l'interpolée de Shannon de l'image discrète de départ. Nous montrons comment, comme dans le cas de TV^{d}, les méthodes duales modernes peuvent être utilisées pour minimiser les énergies convexes non-différentiables impliquant STV. Nous illustrons sur de nombreux exemples comment ce modèle, grâce à la pénalisation des oscillations à l'échelle sous-pixelliques qu'il impose, permet de réconcilier la minimisation de la variation totale avec la théorie de l'échantillonnage de Shannon, au sens où les images obtenues avec ce modèle peuvent être interpolées sans artefacts. Nous montrons également que STV permet un gain considérable en terme d'isotropie par rapport au modèle TV^{d}. Nous proposons enfin un nouveau modèle de restauration d'image dans lequel l'utilisation de STV permet de construire à partir d'une image u_0 donnée, une image u visuellement très proche de u_0 mais pouvant être interpolée (en utilisant l'interpolée de Shannon) de manière beaucoup plus satisfaisante que u_0. On profite ainsi d'une collaboration réussie entre théorie de Shannon et variation totale,

puisque la théorie de Shannon nous permet d'améliorer l'estimation de la variation totale et la variation totale nous permet d'améliorer la qualité de l'interpolation de Shannon sur les images numériques.

Chapitre 4

Dans ce chapitre, nous adaptons un nouveau modèle de restauration d'images appelé TV-ICE (pour *Iterated Conditional Expectations*) au cas où les images sont dégradées par un bruit de Poisson, menant ainsi au modèle *TV-ICE Poisson*. Nous montrons que, comme avec TV-ICE, le modèle TV-ICE Poisson consiste en la recherche d'un point fixe d'une application contractante, conduisant à un schéma numérique dont le taux de convergence est linéaire. Nous montrons également d'un point de vue formel que les images générées par ce modèle ne présentent pas l'effet dit de *staircasing* (qui correspond à la création dans l'image de régions constantes par morceaux délimitées par des contours artificiels) dont souffrent habituellement les images obtenues en utilisant l'approche classique du maximum a posteriori (TV-MAP). Enfin, nous nous concentrons sur la formulation explicite des itérations du schéma associé au modèle TV-ICE Poisson. Si l'établissement de cette formulation explicite ne pose aucune difficulté du point vue théorique, nous remarquons qu'il conduit à un problème numérique difficile, impliquant l'évaluation d'un ratio de différences de fonctions gamma incomplètes généralisées, nécessitant un contrôle précis des erreurs inhérentes au calcul par ordinateur en précision finie. Nous proposons alors une méthode numériques (qui fait l'objet d'une étude détaillée au Chapitre 5) permettant d'effectuer rapidement et avec une bonne précision les calculs mis en jeux à chaque itération du schéma. Enfin, nous validons expérimentalement les propriétés théorique du modèle TV-ICE Poisson (c'est-à-dire la convergence linéaire du schéma et l'absence de staircasing), et nous comparons les images ainsi obtenues avec celles issue du modèle TV-MAP.

Chapitre 5

Dans ce chapitre, nous détaillons une procédure numérique dédiée à l'évaluation de la fonction gamma incomplète généralisée $\int_x^y s^{p-1}\, e^{-\mu s}\, ds$ où $0 \leq x < y \leq +\infty$, μ désigne un réel non nul et p un entier strictement positif. Notre approche consiste à sélectionner, parmi différentes méthodes d'estimations (impliquant entre autres des développements en séries entières ou en fractions continues, des formules récurrentes d'intégration par parties, ou encore d'approximations par la méthode des trapèzes), celle qui réalise l'estimation la plus rapide et la plus précise en fonc-

tion de la valeur des paramètres x, y, μ, p. Tous les algorithmes mis en jeu dans ce chapitre sont décrits en pseudo-code et nous en proposons une implémentation en langage C. Nous montrons que la précision obtenue avec cet algorithme est quasi-optimale pour une grande gamme de paramètres.

Chapitre 6

Dans ce dernier chapitre, nous nous intéressons au problème de détection de trajectoires régulières à partir d'une séquence (bruitée) de nuage de points. Si cette détection peut-être réalisée de manière optimale, au sens d'un critère a contrario, en utilisant l'algorithme ASTRE (A-contrario Smooth TRajectory Extraction), sa complexité quadratique (en temps et en mémoire) par rapport au nombre de frames contenues dans la séquences le rend en pratique inutilisable pour la plupart des applications. Nous proposons donc une variante de cet algorithme, appelée CUTASTRE, qui consiste à découper la séquence en sous-séquences plus petites (avec recouvrement entre les frames qui composent ces sous-séquences) et à traiter ces sous-séquences séquentiellement (mais non-indépendamment), ce qui mène à un algorithme de complexité linaire par rapport au nombre total de frames composant la séquence. Nous décrivons l'algorithme CUTASTRE à l'aide d'un pseudo-code et nous en proposons également une implémentation en langage C. Nous expliquons comment les deux nouveaux paramètres introduits par CUTASTRE (le nombre de frames contenues dans une sous-séquence ainsi que dans la zone de recouvrement entre deux sous-séquences) peuvent être réglés de manière satisfaisante. Nous comparerons les performances des deux algorithmes en terme de qualité de détection et temps d'exécution, à la fois sur des données synthétiques et réelles. De manière assez surprenante, en plus de l'amélioration drastique de la complexité de CUTASTRE par rapport à celle de ASTRE, les performances en terme de qualité de détection atteintes par CUTASTRE sont en général légèrement meilleures que celles atteintes avec l'algorithme ASTRE initial.

1.7 Liste des publications

Travaux publiés

Une partie du Chapitre 4 a fait l'objet d'une publication dans l'acte de la conférence SSVM (Scale Space and Variational Methods in Computer Vision) en 2015.

> **Total Variation Restoration of Images Corrupted by Poisson Noise with Iterated Conditional Expectations** (Rémy Abergel, Cécile Louchet, Lionel Moisan, Tieyong Zeng), *proceedings of the 5th International Conference on Scale Space and Variational Methods in Computer Vision*, 2015.

Une partie du Chapitre 6 a fait l'objet d'une publication dans l'acte de la conférence EUSIPCO (European Signal Processing Conference) en 2014. Une implémentation en langage C de l'algorithme que nous proposons est disponible à l'adresse http://www.math-info.univ-paris5.fr/~rabergel/cutastre.html.

> **Accelerated A-contrario Detection of Smooth Trajectories** (Rémy Abergel, Lionel Moisan), *proceedings of the 22nd European Signal Processing Conference (EUSIPCO)*, 2014.

Travaux soumis

Le contenu du Chapitre 3 a fait l'objet d'une soumission dans la revue *Journal of Mathematical Imaging and Vision (JMIV)*.

> **The Shannon Total Variation** (Rémy Abergel, Lionel Moisan), submitted to *Journal of Mathematical Imaging and Vision (JMIV)*, July 2016.

Le contenu du Chapitre 5 ainsi qu'une implémentation en langage C de l'algorithme que nous proposons ont fait l'objet d'une soumission dans la revue *ACM Transactions on Mathematical Software (ACM-TOMS)*.

> **Fast and accurate evaluation of a generalized incomplete gamma function** (Rémy Abergel, Lionel Moisan), submitted to *ACM Transactions on Mathematical Software (ACM-TOMS)*, June 2016. Code available at http://www.math-info.univ-paris5.fr/~rabergel/softwares/deltagammainc.zip

Autres contributions

Des tutoriaux concernant principalement les modèles de restauration d'images par minimisation de la variation totale, avec implémentation des algorithmes en langage Scilab, sont disponibles sur ma page personnelle, dont l'adresse actuelle est : http://www.math-info.univ-paris5.fr/~rabergel/.

Chapter 2

The use of the Total Variation in Image Processing

Contents

In this chapter, we will focus on the total variation and its use in image processing applications. Some definitions and properties, in both the continuous and the discrete setting, will be given in Section 2.1. Some convex analysis tools, based on Legendre-Fenchel duality, and helpful to perform the minimization of convex functionals involving the total variation, will be presented in Section 2.2. Those tools will be used to handle various type of image processing tasks in Section 2.3. In Section 2.4, we will present some more advanced duality results, that are useful to handle with another approach, and sometimes more efficiently, the optimization problems considered in Section 2.3. This chapter results from a study of the literature. Although it is far from giving a complete overview about total variation, convex optimization, and duality, it contains all the information necessary to properly manipulate the total variation in a variational context.

2.1 The total variation

2.1.1 Definitions

We will first consider the continuous setting, where the images are represented as real-valued functions

$$U = \begin{pmatrix} \Omega_c & \to & \mathbb{R} \\ (x,y) & \mapsto & u(x,y) \end{pmatrix},$$

defined on an open subset Ω_c of $\mathbb{R}^2$ (for instance $\Omega_c = (0,1)\times(0,1)$). When the image U admits some partial derivatives $\partial_1 U$ and $\partial_2 U$ in the two directions of the canonical base of $\mathbb{R}^2$, and when those partial derivatives are integrable on Ω_c (i.e. $\partial_1 U$ and $\partial_2 U \in L^1(\Omega_c)$), we call $\nabla U := (\partial_1 U, \partial_2 U) \in \mathbb{R}^{\Omega_c} \times \mathbb{R}^{\Omega_c}$ the gradient of U and the total variation of U is defined by

$$\mathrm{TV}(U) = \int_{\Omega_c} |\nabla U(x,y)|_2 \, dx\, dy\,, \tag{2.1}$$

noting $|\cdot|_2$ the ℓ^2 Euclidean norm in $\mathbb{R}^2$. This definition can be naturally extended when U belongs to the Sobolev space $W^{1,1}(\Omega_c)$ defined as

$$W^{1,1}(\Omega_c) = \left\{ U \in L^1(\Omega_c), \quad \forall i \in \{1,2\}, \quad \partial_i U \in L^1(\Omega_c) \right\},$$

where $\partial_1 U$ and $\partial_2 U$ now denote the weak partial derivatives of U in the two directions of the canonical base of $\mathbb{R}^2$ (see for instance [Ziemer 2012]). Unfortunately the elements of $W^{1,1}(\Omega_c)$ are too regular to efficiently represent images since we can show that they cannot contain any discontinuity across a line (such as the edges or boundaries of objects in an image), or any hypersurface in general (see [Chambolle et al. 2010]). Fortunately, the definition of the total variation of an image can again be extended to the space of functions with bounded variation (which this time allows discontinuities for the images), where the total variation can be written in a weaker form.

Definition 1 (weak formulation of the total variation). *For any image $U \in L^1_{loc}(\Omega_c)$, the total variation of U is defined by*

$$\mathrm{TV}(U) = \sup_{\substack{\phi \in \mathcal{C}_c^\infty(\Omega_c) \\ \forall (x,y)\in\Omega,\ |\phi(x,y)|_2 \le 1}} -\int_{\Omega_c} U(x,y)\mathrm{div}\phi(x,y)\, dx\, dy\,,$$

where $\mathcal{C}_c^\infty(\Omega_c)$ denotes the set of indefinitely differentiable test functions with compact support contained in Ω_c, and $\mathrm{div}\phi = \partial_1\phi + \partial_2\phi$ denotes the divergence of the two dimensional vector field ϕ.

Definition 2 (functions with bounded variation). *We say that the image U has bounded variation whenever it has finite total variation, and we note* $\mathrm{BV}(\Omega_c)$ *the set of all images with domain Ω_c and bounded variation,*

$$\mathrm{BV}(\Omega_c) = \left\{ U \in L^1_{loc}(\Omega_c), \quad \mathrm{TV}(U) < +\infty \right\} .$$

Remark 1. *If ∇U exists and is an element of $L^1(\mathbb{R}^\Omega \times \mathbb{R}^\Omega)$, noting $\langle \cdot, \cdot \rangle$ the Euclidean inner product in $\mathbb{R}^2$, we have*

$$-\int_{\Omega_c} U(x,y)\, \mathrm{div}\phi(x,y)\, dx = \int_{\Omega_c} \langle \nabla U(x,y), \phi(x,y) \rangle \, dx\, dy \,, \tag{2.2}$$

since the divergence operator is the opposite of the adjoint of ∇. This gives an intuitive reason why taking the supremum of (2.2) over all functions $\phi \in \mathcal{C}^\infty_c(\Omega_c)$ such as $|\phi(x,y)|_2 \leq 1$ for all $(x,y) \in \Omega_c$ leads back to (2.1).

We now focus on the discrete setting where the considered images are real valued functions $u : \Omega \mapsto \mathbb{R}$ but now Ω is a rectangle of $\mathbb{Z}^2$, for instance,

$$\Omega = \{0, \ldots, M-1\} \times \{0, \ldots, N-1\}, \quad \text{with } M, N \in \mathbb{N} .$$

The classical way to compute the total variation of the discrete image u is to replace in (2.1) the gradient operator ∇ by a finite differences scheme ∇^{d}, and the integral by a discrete sum. Usually, we set $\nabla^{\mathrm{d}} = (\nabla^{\mathrm{d}}_1, \nabla^{\mathrm{d}}_2)$ with

$$\forall (x,y) \in \Omega, \quad \left\{ \begin{array}{ll} \nabla^{\mathrm{d}}_1 u(x,y) & = u(x+1,y) - u(x,y) \\ \nabla^{\mathrm{d}}_2 u(x,y) & = u(x,y+1) - u(x,y) \end{array} \right. \tag{2.3}$$

using the convention that $u(M,y) = u(M-1,y)$ and $u(x,N) = u(x,N-1)$ for all $(x,y) \in \Omega$. By analogy with the continuous setting, we note $\mathrm{div}^{\mathrm{d}}$ the opposite of the adjoint of ∇^{d}, that is, $\mathrm{div}^{\mathrm{d}} = -(\nabla^{\mathrm{d}})^*$. One can easily check that the finite difference scheme (2.3) yields

$$\forall p = (p_1, p_2) \in \mathbb{R}^\Omega \times \mathbb{R}^\Omega, \quad \mathrm{div}^{\mathrm{d}}(p) = \mathrm{div}^{\mathrm{d}}_1(p_1) + \mathrm{div}^{\mathrm{d}}_2(p_2) \,, \tag{2.4}$$

where, for any $(x,y) \in \Omega$,

$$\mathrm{div}^{\mathrm{d}}_1(p_1)(x,y) = \left\{ \begin{array}{cl} p_1(x,y) & \text{if } x = 0 \\ p_1(x,y) - p_1(x-1,y) & \text{if } 1 < x < M-1 \\ -p_1(x-1,y) & \text{if } x = M-1, \end{array} \right.$$

$$\mathrm{div}^{\mathrm{d}}_2(p_2)(x,y) = \left\{ \begin{array}{cl} p_2(x,y) & \text{if } y = 0 \\ p_2(x,y) - p_2(x,y-1) & \text{if } 1 < y < N-1 \\ -p_2(x,y-1) & \text{if } y = N-1. \end{array} \right.$$

Definition 3 (discrete total variation). *Let $u : \Omega \mapsto \mathbb{R}$ be a discrete image with domain Ω. We call discrete total variation of u the quantity*

$$\mathrm{TV}^{\mathrm{d}}(u) = \sum_{(x,y)\in\Omega} |\nabla^{\mathrm{d}} u(x,y)|_2 \,,$$

where, unless explicitly stated, ∇^{d} denotes the finite differences scheme defined in (2.3).

Notice that even if (2.3) is the most frequently used finite difference scheme for the discrete total variation, many others can be used. Actually the scheme (2.3) is far from optimal, for instance we can see that with such a choice of discretization, TV^{d} suffers from a strong lack of isotropy, in the sense that, in general we have

$$\mathrm{TV}^{\mathrm{d}}(u) \neq \mathrm{TV}^{\mathrm{d}}(\mathrm{R}u) \,,$$

where R denotes the $\pi/2$ rotation operator defined by

$$\forall (x,y) \in \Omega, \quad (\mathrm{R}u)(x,y) = u(y, M - x - 1) \,.$$

The consequence is that when we consider some TV^{d}-based imaging problems, the resulting image is in general not the same if a rotation of $\pi/2$ is applied before or after the process. Some more isotropic schemes exist but they show other drawbacks [Lai et al. 2009, Wang and Lucier 2011, Chambolle et al. 2011, Condat 2016].

2.1.2 The Maximum A Posteriori approach to image reconstruction

Let $u \in \mathbb{R}^{\Omega}$ (that is, $u : \Omega \to \mathbb{R}$) be an (unobserved) intensity image defined on the discrete domain $\Omega = \{0, \dots, M_\Omega - 1\} \times \{0, \dots, N_\Omega - 1\}$ with size $M_\Omega \times N_\Omega$. Instead of u, assume that we are only able to observe a noisy version of Au, where $A : \mathbb{R}^{\Omega} \mapsto \mathbb{R}^{\omega}$ denotes a linear operator, which may model for instance the convolution with the point spread function of an acquisition sensor (but also some other linear observation mechanisms such as tomography, downsampling, etc.), and $\omega = \{0, \dots, M_\omega\} \times \{0, \dots, N_\omega\}$ denotes another discrete domain with size $M_\omega \times N_\omega$ (possibly $\omega = \Omega$). Noting $u_0 \in \mathbb{R}^{\omega}$ the observed image, we consider that

$$\forall (x,y) \in \Omega, \quad u_0(x,y) = Au(x,y) + n(x,y) \,, \tag{2.5}$$

where the $n(x, y)$ are realizations of independent Gaussian random variables with zero-mean and variance σ^2. The images u_0 and n considered above are realizations of random variables noted $\mathbf{u_0}$ and $\mathbf{n}$. The probability density function (p.d.f) corresponding to the observation model (2.5) is therefore given by

$$p(u \,|\, u_0) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^{|\omega|} \exp\left(-\frac{\|Au - u_0\|_2^2}{2\sigma^2}\right), \tag{2.6}$$

noting $|\omega| = M_\omega \times N_\omega$ the cardinality of ω, and $\|\cdot\|_2$ the ℓ^2 Euclidean norm over the finite dimensional vector space $\mathbb{R}^\omega$.

Remark 2. *Equation (2.6) is straightforward to derive since we have*

$$p(u_0 \,|\, u) = \prod_{(x,y)\in\omega} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(Au(x,y) - u_0(x,y))^2}{2\sigma^2}\right),$$

thanks to the independence of the random family $\{\mathbf{n}(x, y)\}_{(x,y)\in\omega}$.

The function $\mathcal{L} = u \mapsto p(u \,|\, u_0)$ is known as the *likelihood* associated to the observation model (2.5). The *Maximum Likelihood Estimator* (MLE) consists in computing a maximizer of the likelihood function $\mathcal{L}$, or equivalently to compute a minimizer of $u \mapsto \|Au - u_0\|_2^2$, leading to the well known *least squares problem* related to the inversion of the linear operator A.

We now move a step forward and try to take benefit from prior knowledge about the unobserved image itself (and not only about the observation model for u_0). This knowledge is here modeled (although many other choices are possible) by the improper TV^{d} prior $p(u) \propto e^{-\beta \mathrm{TV}^{\mathrm{d}}(u)}$ (here, β denotes a positive parameter, the notation $\propto$ indicates an equality up to a global multiplicative constant, and the term improper indicates that the integral of p over $\mathbb{R}^\Omega$ is infinite), so that the unobserved image is now seen as the realization of a random variable $\mathbf{u}$ with (improper) probability density function $u \mapsto p(u)$ which promotes images u having a low discrete total variation $\mathrm{TV}^{\mathrm{d}}(u)$. Thanks to the Bayes rule, we get the (improper) posterior density

$$\pi(u) := p(u \,|\, u_0) \propto p(u_0 \,|\, u)\, p(u) \propto \exp\left(-\frac{\|Au - u_0\|_2^2}{2\sigma^2} - \beta \mathrm{TV}^{\mathrm{d}}(u)\right). \tag{2.7}$$

The *Maximum A Posteriori* (MAP) methodology consists in computing a maximizer $u_{\mathrm{MAP}} \in \mathbb{R}^\Omega$ of the (improper) posterior density π, or equivalently a minimizer

of the convex energy $u \mapsto -\log \pi(u)$. Finally, the MAP approach boils down to the variational problem

$$\min_{u \in \mathbb{R}^\Omega} \|Au - u_0\|_2^2 + \lambda \mathrm{TV}^{\mathrm{d}}(u)\,, \tag{2.8}$$

where the parameter $\lambda = 2\beta\sigma^2$, named *regularity parameter*, controls the trade-off between the so-called *data-fidelity* (the quadratic term) and *regularity* (the total variation term) in the minimization process. A variational problem of the kind (2.8) is called an *inverse problem*, since it consists in recovering u from a noisy version of Au (thus in a sense to invert the operator A). When A is the identity operator in $\mathbb{R}^\Omega$, the inverse problem (2.8) boils down to a *pure image denoising* application, which was first introduced by Rudin, Osher and Fatemi (ROF) in [Rudin et al. 1992].

Remark 3 (other observation models, the example of Poisson noise). *The presence of the quadratic term in* (2.8) *is due to the Gaussian nature of the noise which corrupts the observed image* u_0*. Of course other models of noise are possible (an interesting study about the difference sources of noise occurring during the image acquisition process with a digital camera can be found in [Aguerrebere et al. 2012]), typically in a* low light context*, that is, when only a low amounts of photons reach the sensor during the acquisition process, a more realistic observation model consists in considering* u_0 *as a* photon-count *observation of* Au *(note that, in that case,* Au *must be nonnegatively valued), which means that we consider* $\mathbf{u_0}$ *as an integer-valued random image (*$\mathbf{u_0} \in \mathbb{N}^\omega$*) following the Poisson probability density function,*

$$p(u_0 \,|\, u) = \prod_{(x,y)\in\omega} \frac{Au(x,y)^{u_0(x,y)}}{u_0(x,y)!} e^{-Au(x,y)}\,.$$

The MAP approach can be used to derive other optimization problems (with data-fidelity term different from the quadratic term $\|Au - u_0\|_2^2$*), more adapted to the intended image restoration application.*

The mathematical analysis tools necessary to handle problems of the kind (2.8), and more generally many problems involving the total variation, will be presented in section 2.2.

2.1.3 The LSE approach to total variation denoising

The Bayesian point of view behind the MAP approach is an elegant mean to design (or motivate the use of) some TV-regularized optimization problems dedicated to image reconstruction tasks, however, it also presents several weaknesses:

(i) The first one is obviously the imperfect choice of the prior $p(u)$ as a model for natural images (see [Gousseau and Morel 2001]). Since the choice $p(u) \propto e^{-\beta \mathrm{TV}^{\mathrm{d}}(u)}$ favors the images with low total variation (that are typically piecewise constant images, also called *cartoon images*), looking for a maximizer u_{MAP} of the posterior density makes the process very sensitive with respect to the prior model (or from the optimization point of view, very sensitive to the designed energy $u \mapsto -\log \pi(u)$), whose imperfections inevitably yield undesirable artifacts in the computed image. The typical drawback of the total variation based MAP models is the creation in u_{MAP} of constant areas with artificial boundaries, the so-called *staircasing effect*.

(ii) The second weakness is related to the founding idea of the MAP. As remarked in [Chambolle et al. 2010], even if the prior model were perfectly built (i.e. if the ideal unobserved image u were effectively the realization of a random variable with probability density π), from a Bayesian point of view, there would be no reason for a maximizer u_{MAP} of π to be close to u, since u_{MAP} might be "very rare" under π. A hand-designed posterior density function that illustrates this situation is proposed in [Chambolle et al. 2010, Fig. 1], a similar example is reproduced in Figure 2.1. We also refer to [Nikolova 2007], where it is pointed out that the u_{MAP} solution substantially deviates from both the data acquisition model, and the underlying prior model.

From a statistical point of view, the MAP estimator is the one that minimizes the *hit-or-miss* risk function, that is, a Dirac localized on the true solution. In the case of the pure image denoising (ROF) problem (replace A by the identity operator in (2.7) and (2.8)), it was suggested by Louchet and Moisan [2008] (see also [Louchet and Moisan 2013]) to consider, instead of the MAP, the posterior mean

$$u_{\mathrm{LSE}} = \mathbb{E}_{u\sim\pi}(u \,|\, u_0) = \int_{\mathbb{R}^\Omega} u\, \pi(u)\, du\,, \tag{2.9}$$

which is the image that achieves the *Least-Square-Error* (LSE) under π, leading to a new estimate called TV-LSE. Note that this kind of approach is often called MMSE (Minimizer of the Mean Square Error) or CM (Conditional Mean) in the statistical literature. The properties of the image denoising TV-LSE estimator were analyzed in [Louchet and Moisan 2008, 2013], in particular it was proven

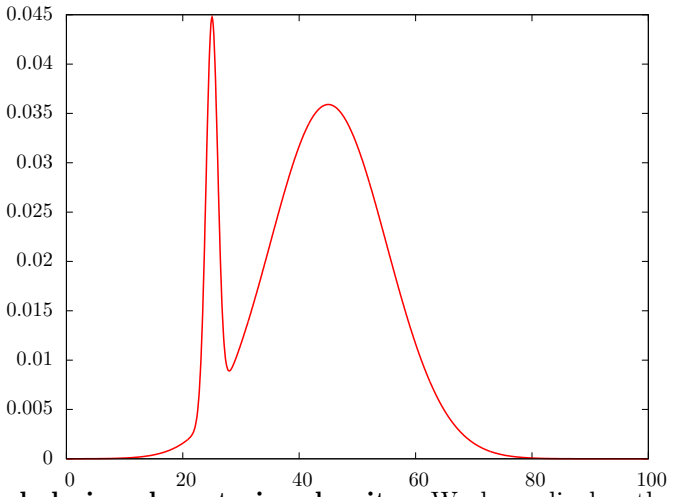


Figure 2.1: Hand designed posterior density. We here display the graph of the one dimensional density $\pi(u) = 0.1/(\sigma_1\sqrt{2\pi})\, e^{-(u-\mu_1)^2/(2\sigma_1^2)} + 0.9/(\sigma_2\sqrt{2\pi})\, e^{-(u-\mu_2)^2/(2\sigma_2^2)}$, where $\mu_1 = 25$, $\sigma_1 = 1$, $\mu_2 = 45$ and $\sigma_2 = 10$. This posterior density reaches its maximum at the point $u_{\text{MAP}} = 25$, while the posterior mean under π is $u_{\text{LSE}} = 43$. We can see that u_{MAP} is quite *rare* under π (the probability of the realization u, sampled under π, to satisfy $u \leq u_{\text{MAP}} + 3$ is less than 5%), while the density π shows more energy in the vicinity of u_{LSE}.

that TV-LSE avoids the constant regions of the staircasing effect while allowing the restoration of sharp edges, leading to more natural images than the MAP estimate. Surprisingly enough, in spite of the high dimension of the space $\mathbb{R}^\Omega$ (for instance the dimension of $\mathbb{R}^\Omega$ is 10^6 when we consider images of size 1000×1000), the integral (2.9) can be numerically computed using a Monte Carlo Markov Chains (MCMC) Metropolis Hasting algorithm (see Algorithm 1 in [Louchet and Moisan 2008]). Besides (this is rare enough to be remarked) an upper bound of the square Euclidean distance between the iterates u_n and the true LSE image u_{LSE} is available at each iteration n of the algorithm. Unfortunately, this algorithm exhibits a slow convergence rate ($\mathcal{O}(n^{-1/2})$ for n iterations), which can be a major inconvenience for many applications. To overcome this computational limitation, a new variant named TV-ICE (*Iterated Conditional Expectations*) and based on the iteration of conditional marginal posterior means, was proposed in [Louchet and Moisan 2014]. This variant yields a numerical scheme which exhibits a linear convergence, and produces images that are visually very close to those obtained using the TV-LSE estimator. The TV-ICE model will be presented and adapted to the Poisson case in chapter 4.

2.1.4 One and two dimensional examples

As discussed in Section 2.1.1, the elements of $\mathrm{BV}(\Omega)$ can assume some discontinuities, making possible the representation of sharp edges into this space. The ability of the total variation regularizer to preserve those discontinuities (in contrast to the classical H^1 models, corresponding to a choice of regularizer of the type $J(u) = \|\nabla^{\mathrm{d}} u\|_2^2$, which tends to promote oversmoothed images, see e.g. [Nikolova 2000]), is usually highlighted in the image processing literature, and some careful studies about the edge-preserving properties of TV can be found in [Chambolle et al. 2010, Caselles et al. 2015]. We will try to here give a simple intuition about this interesting property, by focusing on the one dimensional case.

Let $s : \{0, \dots, M-1\} \to \mathbb{R}$ be a one dimensional discrete signal of size M, we naturally adapt Definition 3 into

$$\mathrm{TV}^{\mathrm{d}}(s) = \sum_{k=0}^{N-2} |s(k+1) - s(k)| .$$

Assuming now that $\{s(k)\}_{0 \leq k < M}$ is a monotone (that is, nonincreasing or nondecreasing) sequence, we easily prove that

$$\mathrm{TV}^{\mathrm{d}}(s) = |s(M-1) - s(0)| ,$$

so that any other monotone signal $\widetilde{s} : \{0, \dots, M-1\} \to \mathbb{R}$ satisfying $\widetilde{s}(0) = s(0)$ and $\widetilde{s}(M-1) = s(M-1)$ has the same total variation as s (this result can be proven in a more general setting, in particular in the continuous setting, as it is done for instance in [Kannan and Krueger 2012]). Consequently, the total variation will not favor any monotone function among those satisfying this constraint, which, in a sense, places the smooth and nonsmooth signals on equal footing, and makes possible the selection of sharp edges (such as a step signal, as in Figure 2.2-left) in the reconstructed signal, when dealing with TV-based models of the type (2.8).

In contrast to the well appreciated allowance of discontinuities provided by the total variation regularizer, this approach also presents some drawbacks and properties with debatable consequences (e.g. the penalization of oscillations discussed below). First, the TV regularizers tends to favor signals signals whose gradient is sparse (that is, often takes the value 0), so that TV based models usually produce signals showing some undesirable *staircase* effects, where one would have expected smooth variations (see Figure 2.2-right). Back to the world of the two dimensional images, the total variation is responsible for the creation of constant areas with artificial edges, as illustrated in Figure 2.3. This undesirable effect

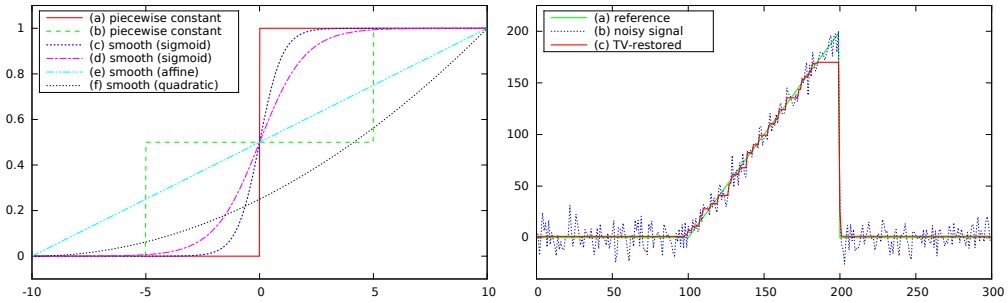


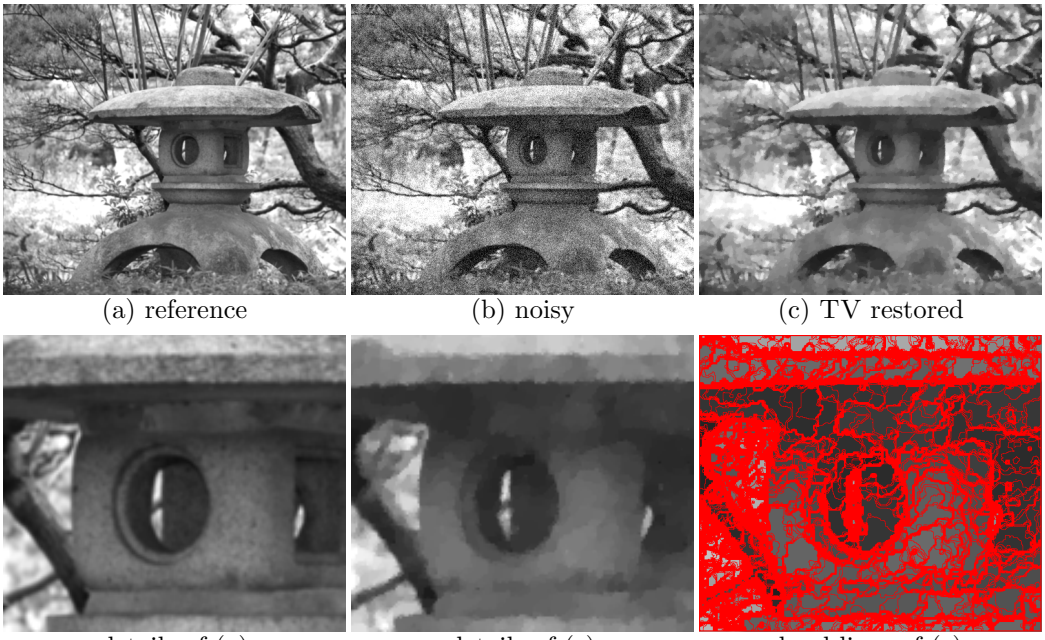
Figure 2.2: Left-side, several signals showing the same total variation. The signals here displayed show different regularity ((a) and (b) present some discontinuities, the others are smooth), but all have the same total variation. Thus, no one is being promoted in terms of TV score in a variational context. This is not the case when we consider the classical H^1 regularizer $J = s \mapsto \|\nabla^{\mathrm{d}} s\|_2^2$, for which all signals yield a different score, the smallest being realized by (e). **Right-side, one dimensional signal reconstruction in presence of noise using ROF.** A reference signal (a) is corrupted by an additive white Gaussian noise (b), then processed using the ROF model (that is, model (2.8) with $A = I$), with $\lambda = 200$, yielding the restored signal (c). The discontinuities of (a) are present in (c), however, the promotion of sparse gradient involved a staircase effect, where we expected a smooth variation.

is known as *staircasing* effect (or staircasing artifact). It was first theoretically studied in [Nikolova 2000], and more recently in [Chambolle et al. 2016]. Second, as a consequence of the promotion of sparse gradients, the TV regularizer penalizes oscillations. On the one hand, oscillating patterns are typically the kind of structures one wants to avoid when dealing with inverse problems, in the sense that, the penalization of those structures is valuable. On the other hand, some oscillatory patterns may correspond to textures that one would like to preserve. Interestingly, some works try to take advantage of this above mentioned behaviour of the total variation (generally presented as a shortcoming), by using the TV term to perform image decomposition into three components, a first one containing the geometrical structure of the image, a second one the texture of the image, and a third one the noise [Vese and Osher 2004, Aujol and Chambolle 2005].

2.1.5 Several well known variants of TV

Isotropic and Anisotropic total variation.

A first variant of the total variation is obtained by replacing the Euclidean ℓ^2 norm of the gradient in Definition 3 by a ℓ^p norm (we will note ℓ^a instead of ℓ^p to avoid confusions when we will introduce, in the next sections, a dual variable traditionally noted p). We define accordingly the $\mathrm{TV}_a^{\mathrm{d}}$ (for $a \geq 1$, possibly



(a) reference (b) noisy (c) TV restored

details of (a) details of (c) level lines of (c)

Figure 2.3: The staircasing effect of TV. A noisy version (undergoing additive white Gaussian noise with zero mean and standard deviation $\sigma = 20$), displayed in (b), of the reference image (a), was denoised using the ROF model (that is, model (2.8), taking $A = I$), with the setting $\lambda = 40$. On the second row, we display some close-up views of (a) and (c), which reveal the presence in (c) of constant areas with artificial boundaries, that were not present into the reference image. This so-called staircasing effect is due to the promotion by the TV term of piecewise constant images. (or *cartoon* images). This effect clearly appears on the level lines of (c), here computed using a bilinear interpolation, which tends to concentrate along the spurious edges of the staircased regions. We also observe that some textures (in particular the microtextures) of (a) are not present in (c), due to the penalization of the oscillations operated by the total variation, showing its inability to deliver well-textured images.

$a = +\infty$) variant of TV^{d},

$$\forall a \in [1, +\infty],\ \forall u \in \mathbb{R}^{\Omega}, \quad \mathrm{TV}^{\mathrm{d}}_{a}(u) = \sum_{(x,y)\in\Omega} |\nabla^{\mathrm{d}} u(x,y)|_{a}\,, \tag{2.10}$$

where $|\cdot|_a$ denotes the ℓ^a norm over the $\mathbb{R}^2$ space, defined by

$$\forall (z_1, z_2) \in \mathbb{R}^2, \quad |(z_1, z_2)|_a = \begin{cases} (|z_1|^a + |z_2|^a)^{1/a} & \text{if } a < +\infty \\ \max(|z_1|, |z_2|) & \text{if } a = +\infty\,. \end{cases}$$

Since the total variation operator in (2.10) is written as a sum of ℓ^a norms, we naturally introduce the norm $\|\cdot\|_{1,a}$ over $\mathbb{R}^{\Omega} \times \mathbb{R}^{\Omega}$, which is a combination between

ℓ^1 and ℓ^a norms, defined by

$$\forall (g_1, g_2) \in \mathbb{R}^\Omega \times \mathbb{R}^\Omega, \quad \|(g_1, g_2)\|_{1,a} = \sum_{(x,y)\in\Omega} |(g_1(x,y), g_2(x,y))|_a \,,$$

so that the total variation operator can be seen as the composition between the norm $\|\cdot\|_{1,a}$ and the linear finite difference operator ∇^{d},

$$\forall a \geq 1,\ \forall u \in \mathbb{R}^\Omega, \quad \mathrm{TV}_a^{\mathrm{d}}(u) = \|\nabla^{\mathrm{d}} u\|_{1,a} \,, \tag{2.11}$$

and this viewpoint will be used in Section 2.3.1 to derive a dual formulation of $\mathrm{TV}_a^{\mathrm{d}}$. Of course (2.11) is a straightforward generalization of TV^{d}, since the two definitions coincide when $a = 2$. In practice we are principally interested in the settings $a = 1$ or $a = 2$.

Definition 4 (isotropic and anisotropic discrete total variation). $\mathrm{TV}_1^{\mathrm{d}}$ *and* $\mathrm{TV}_2^{\mathrm{d}}$ *are respectively called anisotropic and isotropic discrete total variation.*

We will see that from the optimization point of view, the use of $\mathrm{TV}_a^{\mathrm{d}}$ instead of TV^{d} introduces no theoretical nor numerical difficulty. The case $a = 1$ (corresponding to the anisotropic total variation) is sometimes addressed in the literature, mainly because it leads to optimization problems that can be solved by graph-flow techniques (see [Darbon and Sigelle 2006, Chambolle 2005], and references therein). Besides, the anisotropic total variation was also used to derive some numerical algorithms dedicated to the computation of the TV-ICE variants of the TV-LSE model, in the cases of image denoising in presence of Gaussian noise [Louchet and Moisan 2014] or Poisson noise (see Chapter 4, or [Abergel et al. 2015]).

The Huber approximation.

The use of the discrete total variation as a regularizer for image processing applications generates images with piecewise constant regions and artificial boundaries, this is the *staircasing effect* already evoked above. This undesirable effect has been rigorously identified and studied in [Nikolova 2000, Ring 2000], in particular it is proven (in a more general setting than total variation regularization) in [Nikolova 2000] that the non-differentiability at zero of the total variation (or more generally of the cost function to minimize) is responsible for the staircasing artifact. In order to get rid of this artifact, we can replace the ℓ^2 norm of the gradient by a smooth approximation, for instance the so-called Huber-function.

Definition 5 (Huber function). *Let α be a positive parameter, we note $\mathcal{H}_\alpha$ the Huber function with parameter α, defined by*

$$\forall z \in \mathbb{R}^2, \quad \mathcal{H}_\alpha(z) = \begin{cases} \frac{|z|_2^2}{2\alpha} & \text{if } |z|_2 \leq \alpha\,, \\ |z|_2 - \frac{\alpha}{2} & \text{otherwise}\,. \end{cases}$$

From its definition, we can see that the Huber function consists in replacing the (non-differentiable at zero) ℓ^2 norm by a (differentiable at zero) square ℓ^2 norm over the ℓ^2 closed ball with center 0 and radius α. Outside of the ball, $\mathcal{H}_\alpha$ is simply the ℓ^2 norm translated by the quantity $\alpha/2$, so that $\mathcal{H}_\alpha$ is a continuous and differentiable function of $\mathbb{R}^2$ (some one dimensional plots of $\mathcal{H}_\alpha$, for several values of α, are displayed in Figure 2.4).

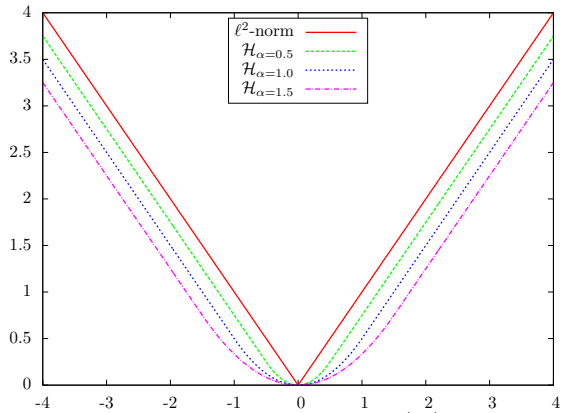


Figure 2.4: One dimensional comparison between $|\cdot|_2$ and $\mathcal{H}_\alpha$. We here display a one dimensional representation of the ℓ^2-norm, and, for several values of α, the graph of the corresponding Huber approximation $\mathcal{H}_\alpha$, which basically consists in replacing the non differentiable term $|x|_2$ by a smooth quadratic term when $x \in [-\alpha, \alpha]$. Otherwise, when $x \notin [-\alpha, \alpha]$, an appropriate offset is used, which makes the $\mathcal{H}_\alpha$ function differentiable.

Definition 6 (Huber total variation). *Given a positive real parameter α, we call Huber total variation with parameter α the operator $\mathrm{HTV}^{\mathrm{d}}_\alpha$ defined by*

$$\forall u \in \mathbb{R}^\Omega, \quad \mathrm{HTV}^{\mathrm{d}}_\alpha(u) = \sum_{(x,y)\in\Omega} \mathcal{H}_\alpha(\nabla^{\mathrm{d}} u(x,y))\,.$$

We will experimentally check that replacing the TV^{d} regularizer by $\mathrm{HTV}^{\mathrm{d}}_\alpha$ in optimization problems of the kind (2.8) leads to images without staircasing artifact, but the removal of artifact is done at the expense of losing the nice theoretical properties of the total variation, such as the formal allowance of discontinuities (and thus sharp edges) into the restored image.

Other variants.

Many variants of TV have been proposed to avoid the staircasing effect, and preserve textures. Some of them consist in smoothing the discontinuity of the TV functional, as it is the case with the Huber variant. Some other consist in adding some higher derivative order terms to the TV functional [Chan et al. 2000], or even in changing the order of the derivative into the TV term itself, yielding a generalized definition of this functional called Total Generalized Variation [Bredies et al. 2010].

2.2 Basis of non-smooth convex analysis and optimization

In this section, we will recall some fundamental results of convex analysis, most of them taken from [Ekeland and Témam 1999]. The aim of this presentation of some already well known results, is to give a minimal, but complete, set of tools that are now widely used to handle convex and non smooth optimization problems, which are particularly efficient to perform total variation based minimization. We refer to [Ekeland and Témam 1999, Rockafellar and Wets 1998, Boyd and Vandenberghe 2004] for much more complete information.

2.2.1 Main definitions and properties

Let us consider a finite-dimensional real vector space E, and let $E^\star$ denote its dual space, which is made of all continuous linear mappings from E to $\mathbb{R}$. Let $\overline{\mathbb{R}}$ denote the set $\mathbb{R} \cup \{-\infty, +\infty\}$ and $\langle \cdot, \cdot \rangle$ denote the bilinear mapping from $E^\star \times E$ to $\mathbb{R}$ defined by $\langle \varphi, u \rangle = \varphi(u)$ for any $\varphi \in E^\star$ and $u \in E$. We first recall the definitions of convex sets and convex functions.

Definition 7 (convex sets and convex functions). *A subset $\mathscr{C}$ of E is convex if and only if for any pair of elements (u, v) of $\mathscr{C}$, the line-segment $[u, v]$ with endpoints u and v,*

$$[u, v] = \{\lambda u + (1 - \lambda)v, \quad 0 \leq \lambda \leq 1\} \ ,$$

is a subset of $\mathscr{C}$. Now given a convex subset $\mathscr{C}$ of E, a mapping F from $\mathscr{C}$ to $\overline{\mathbb{R}}$ is convex if and only if it satisfies

$$\forall u, v \in \mathscr{C}, \quad \forall \lambda \in [0, 1], \quad F(\lambda u + (1 - \lambda)v) \leq \lambda F(u) + (1 - \lambda)F(v) \, ,$$

whenever the right-hand term is defined, i.e. whenever the pair $(F(u), F(v))$ *is different from* $(+\infty, -\infty)$ *and* $(-\infty, +\infty)$*. If moreover the inequality is strict for any* $u \neq v$ *and any* $\lambda \in (0,1)$*, the function is called strictly convex.*

Notice that allowing the functions to assume the value $+\infty$ will be of great importance, since in convex optimization we often need to consider the indicator function of a domain $\mathscr{D} \subset E$, defined by

$$\forall u \in E, \quad \delta_{\mathscr{D}}(u) = \begin{cases} 0 & \text{if } u \in \mathscr{D} \\ +\infty & \text{otherwise,} \end{cases} \tag{2.12}$$

which is a convex function if and only if $\mathscr{D}$ is a convex subset of E. Convex functions that assume the value $-\infty$ are however very special since those functions are infinite everywhere except possibly at a single point where it may take any value (see [Ekeland and Témam 1999, Chap. I, Sec. 2.1]). The underlying reason why we keep the value $-\infty$ (however much we wish to remove it) is that the constants $\pm\infty$ are in a certain sense placed in duality (see [Ekeland and Témam 1999, Chap. I, Def. 4.2]). For instance, we will later consider the Legendre-Fenchel transform which changes the constant function $F = +\infty$ of E into the constant function $F^\star = -\infty$ of the dual space $E^\star$.

In the following we will denote by $\mathrm{dom}F$ the *effective domain of* F, that is the set of vectors $u \in E$ such as $F(u) < +\infty$,

$$\mathrm{dom}F = \{u \in E, \quad F(u) < +\infty\} \ .$$

The effective domain of F is nonempty as soon as F is not identically equal to $+\infty$. In order to get rid of very particular cases, we will say that the function F is *proper* if it never assumes the value $-\infty$ and is different from the constant $+\infty$.

Before focusing on the non-smooth setting, which will retain all our attention later, we briefly recall the notion of Gâteaux-differentiability.

Definition 8 (Gâteaux-differentiability). *Let* F *be a function of* E *into* $\overline{\mathbb{R}}$*, let* u *and* v *be two points of* E*. If the limit as* $\lambda \to 0_+$ *of*

$$\frac{F(u + \lambda v) - F(u)}{\lambda} \tag{2.13}$$

exists, it is called the directional derivative in direction v *of* F *at point* u *and denoted* $\mathrm{D}_v F(u)$*. If furthermore there exists an element* $\varphi_u \in E^\star$ *such that*

$$\forall v \in E, \quad \mathrm{D}_v F(u) = \langle \varphi_u, u \rangle \ ,$$

we say that F is Gâteaux-differentiable at the point u, we call φ_u the Gâteaux-differential of F at the point u, and we note $\varphi_u = \mathrm{D}F(u)$. When moreover E is a Hilbert space, the Riesz representation Theorem states that there exists a unique element of E noted $\nabla F(u)$ such that $\langle \mathrm{D}F(u), u\rangle$ is equal to the inner product between $\nabla F(u)$ and u. In that case $\nabla F(u)$ is called gradient of F a the point u.

Of course a function which is differentiable at point u in the classical sense (that is in the sense of Fréchet) is also Gâteaux-differentiable at u and the two notions of differentials coincide, but the reciprocal is false. The notion of Gâteaux-differentiability is particularly suited to the convex setting since for any convex function F, the expression (2.13) always possesses a limit as $\lambda \to 0_+$, this limit may however be infinite.

We will now focus on the non Gâteaux-differentiable setting (named here the non-smooth setting), and we will show how the notion of Gâteaux-differential shall be generalized for non-smooth convex functions. We start with the notion of lower semi-continuity.

Definition 9 (lower semi-continuity). *A function F from E to $\overline{\mathbb{R}}$ is lower semi-continuous (l.s.c) on E if and only if for any $u_0 \in E$ and for any $\varepsilon > 0$, there exists a neighborhood $\mathcal{V}_0^\varepsilon$ of u_0 such as*

$$\forall u \in \mathcal{V}_0^\varepsilon, \quad F(u) \geq F(u_0) - \varepsilon\,,$$

or equivalently when

$$\forall u_0 \in E, \quad \liminf_{u \to u_0} F(u) \geq F(u_0)\,,$$

noting $\liminf_{u \to u_0} F(u)$ the limit inferior of F at point u_0.

Remark 4. *A useful way to prove the lower semi-continuity (respectively its convexity) of a function $F : E \to \mathbb{R}$ consists in considering its epigraph, which is the set $\mathrm{epi}F = \{(u, \lambda) \in E \times \mathbb{R},\ F(u) \leq \lambda\}$. Indeed F is l.s.c on E if and only if $\mathrm{epi}F$ is closed in $E \times \mathbb{R}$, and F is convex on E if and only if $\mathrm{epi}F$ is convex (see [Ekeland and Témam 1999, Chap. I, Prop. 2.1 and 2.3]). In particular, thanks to this topological characterization, we see that the indicator function $\delta_{\mathscr{D}}$ defined in (2.12), is l.s.c and convex on E if and only if the set $\mathscr{D}$ is a closed and convex subset of E, since in that case, we have $\mathrm{epi}F = \mathscr{D} \times \mathbb{R}_+$.*

Now, we recall the notion of affine continuous function, which is at the same time a very simple and, as we will see in the following, an extremely useful object to efficiently represent convex and lower semi-continuous functions.

Definition 10 (affine continuous fonctions on E). *An affine continuous function on E is a function $\mathcal{A} : E \to \mathbb{R}$ of type $\mathcal{A} : u \mapsto \varphi(u) + \alpha$, where φ is a linear continuous function from E to $\mathbb{R}$ (i.e. an element of the dual space $E^\star$) and α is a real number. We will call φ* the slope *and α* the constant term *of such a function. We note $\mathscr{A}(E)$ the set of all affine continuous functions on E.*

We follow up with the definition of the the spaces $\Gamma(E)$ and $\Gamma_0(E)$.

Definition 11 (the spaces $\Gamma(E)$ and $\Gamma_0(E)$). *We denote by $\Gamma(E)$ the set of functions $F : E \to \overline{\mathbb{R}}$ which are the superior envelope (or pointwise supremum) of a family of continuous affine functions on E. We denote by $\Gamma_0(E)$ the subset of $\Gamma(E)$ composed of the functions other than the constants $+\infty$ and $-\infty$.*

We see immediately from the definition that any element of $\Gamma(E)$ is necessarily convex and lower semi-continuous, since those two properties are stable by passage to the supremum, and are satisfied by any affine continuous function. The next proposition gives a precise characterization of the spaces $\Gamma(E)$ and $\Gamma_0(E)$.

Proposition 1 (characterization of $\Gamma(E)$ and $\Gamma_0(E)$). *A function $F : E \to \overline{\mathbb{R}}$ is an element of $\Gamma(E)$ if and only if F is a convex lower semi-continuous function on E which may assume the value $-\infty$ but in that case F is identically equal to $-\infty$. Consequently the set $\Gamma_0(E)$ is composed of all proper elements of $\Gamma(E)$.*

Let us take a function F in $\Gamma(E)$, by definition, there exists a (possibly empty) subset $\mathcal{S}$ of $\mathscr{A}(E)$ such that

$$\forall u \in E, \quad F(u) = \sup_{\mathcal{A} \in \mathcal{S}} \mathcal{A}(u)\,,$$

and consequently all the affine continuous functions $\mathcal{A} \in \mathcal{S}$ are necessarily less than F everywhere on E,

$$\forall \mathcal{A} \in \mathcal{S}, \quad \forall u \in E, \quad \mathcal{A}(u) \leq F(u)\,.$$

This raises an interesting question: *"What do we get when we consider the superior envelope of all the affine continuous functions lower-bounding F?"* This question is answered by the following proposition that leads us to the notion of Γ-regularization.

Proposition 2 (Γ-regularization). *Let F and G be two functions from E into $\overline{\mathbb{R}}$, the following properties are equivalent to each other:*

(i) G is the pointwise supremum function of the set of all continuous affine functions that lower bound F;

(ii) G is the largest element of $\Gamma(E)$ that lower bounds F.

When (i) or (ii) is satisfied, the function G is then called the Γ-regularization of function F.

Proof (adapted from [Ekeland and Témam 1999, Chap. I, Sec. 3.2]). Let us note G_1 the function satisfying (i) and G_2 the function satisfying (ii). We note $\mathscr{A}_1$ (respectively $\mathscr{A}_2$) the set of all the affine continuous affine functions (respectively all the elements of $\Gamma(E)$) which lower bound F. Since $\mathscr{A}_1 \subset \mathscr{A}_2$ we have $G_1 \leq G_2$. Now since each element $H \in \mathscr{A}_2$ is in $\Gamma(E)$, it can be seen as the pointwise supremum of a family $\mathscr{A}_H$ of affine continuous functions, and necessarily the elements of $\mathscr{A}_H$ lower bound F (since H lower bounds F). Thus, we get that for any $H \in \mathscr{A}_2$ we have $\mathscr{A}_H \subset \mathscr{A}_1$, and we derive the converse inequality $G_2 \leq G_1$. □

2.2.2 Legendre-Fenchel transform

Given a function $F : E \to \overline{\mathbb{R}}$, let us explore further the set of affine continuous functions which lower bound F. We prove below that a continuous affine function $\mathcal{A} : E \to \mathbb{R}$ with slope $\varphi \in E^\star$ and constant term $\alpha \in \mathbb{R}$ (that is, $\mathcal{A} = u \mapsto \langle \varphi, u \rangle + \alpha$) lower bounds F if and only if $\alpha \leq -F^\star(\varphi)$, where

$$F^\star(\varphi) = \sup_{u \in \mathrm{dom} F} \langle \varphi, u \rangle - F(u)\,. \tag{2.14}$$

Notice that when F assumes the value $-\infty$, formula (2.14) yields $F^\star(\varphi) = +\infty$, and when F is the constant $+\infty$ (i.e. when F has empty domain), it yields $F^\star(\varphi) = -\infty$.

Proof. $\mathcal{A}$ remains below F everywhere on E if and only if for any $u \in E$ we have $\langle \varphi, u \rangle + \alpha \leq F(u)$, that is, when $\sup_{u \in E} \langle \varphi, u \rangle - F(u) \leq -\alpha$. The supremum can be restricted to the effective domain of F, since the function $u \mapsto \langle \varphi, u \rangle - F(u)$ is identically equal to $-\infty$ outside of $\mathrm{dom} F$. □

This leads us to the Legendre-Fenchel transformation.

Definition 12 (Legendre-Fenchel transform). *If $F : E \to \overline{\mathbb{R}}$, then formula (2.14) defines a function $F^\star$ from $E^\star$ into $\overline{\mathbb{R}}$ called the Legendre-Fenchel transform of F (also known as the polar or conjugate function of F).*

From its definition, one directly sees that the Legendre-Fenchel transform $F^\star$ of F is an element of $\Gamma(E^\star)$ (in particular it is convex and l.s.c), since it can be seen as the pointwise supremum of the family of continuous affine functions $(\mathcal{A}_u)_{u\in \mathrm{dom} F}$ over the dual space $E^\star$, defined by

$$\forall u \in \mathrm{dom} F, \quad \mathcal{A}_u : \varphi \mapsto \langle \varphi, u\rangle - F(u)\,.$$

Assuming from now that E is a reflexive space (for instance a Hilbert space), writing the Legendre-Fenchel transform of the function $F^\star$ leads to

$$F^{\star\star} : u \mapsto \sup_{\varphi \in \mathrm{dom} F^\star} \langle \varphi, u\rangle - F^\star(\varphi)\,,$$

which is an element of $\Gamma(E^{\star\star})$, and thus an element of $\Gamma(E)$. Since F and $F^{\star\star}$ are defined on the same space, they can be compared to each other, we obtain the following result:

Proposition 3 (bi-Legendre-Fenchel transform). *The bi-Legendre-Fenchel transform $F^{\star\star}$ of any function $F : E \to \overline{\mathbb{R}}$ is none other than the Γ-regularization of F. In particular $F^{\star\star} \leq F$ on E, and we have the equality $F^{\star\star} = F$ if and only if $F \in \Gamma(E)$.*

Proof (adapted from [Ekeland and Témam 1999, Chap. I, Prop. 4.1]). The Γ-regularization of F is the pointwise supremum of the set of all affine continuous functions which lower bound F. This supremum can be restricted to the lower bounding affine continuous functions having maximal constant term, that is, the functions $\mathcal{A}_\varphi : E \to \mathbb{R}$ of the type

$$\forall \varphi \in E^\star, \quad \mathcal{A}_\varphi = u \mapsto \langle \varphi, u\rangle - F^\star(\varphi)\,.$$

Since the pointwise supremum of the family $(\mathcal{A}_\varphi)_{\varphi\in E^\star}$ is exactly the function $F^{\star\star}$, we get the announced result. □

In practice Proposition 3 is of great use to derive a primal-dual reformulation of an optimization problem when the cost function decomposes as the sum of terms where at least one lies in $\Gamma(E)$. For instance, if $F : E \to \overline{\mathbb{R}}$ is an element of $\Gamma(E)$, we have

$$\forall u \in E, \quad F(u) = F^{\star\star}(u) = \sup_{\varphi \in \mathrm{dom} F^\star} \langle \varphi, u\rangle - F^\star(\varphi)\,,$$

so that for any function $G : E \to \overline{\mathbb{R}}$, the problem $\inf_{u\in E} F(u) + G(u)$ is equivalent to the primal-dual problem

$$\inf_{u\in E} \sup_{\varphi\in \mathrm{dom} F^\star} G(u) + \langle \varphi, u\rangle - F^\star(\varphi)\,, \tag{2.15}$$

which may be easier to handle (and we will see later that it is typically the case when we set $F = \mathrm{TV}^{\mathrm{d}}$).

A last important remark is that when E is a Hilbert space, the dual space $E^\star$ identifies to the primal space E, which means that for any element $\varphi \in E^\star$, the terms $\langle \varphi, u\rangle$ in (2.14) or (2.15) can be replaced by the inner product between u and an element $v_\varphi \in E$, thanks again to the *Riesz representation Theorem*. In that case $F^\star$ can be seen as a function of the primal space E (instead of $E^\star$), which is very useful in practical computations.

2.2.3 Subdifferentiability

Now, let us show how the affine continuous functions can be used to recover a notion a differentiability for non-smooth functions. Let F be a function from E to $\overline{\mathbb{R}}$, and $\mathcal{A} : E \to \mathbb{R}$ be an affine continuous function. We say that $\mathcal{A}$ is exact at point $u \in E$ if and only if $\mathcal{A}(u) = F(u)$. When $\mathcal{A}$ is furthermore lower bounding F, we recover a notion of slope (that is called subgradient) for F at the point u.

Definition 13 (subdifferentiability, subdifferential, subgradients). *A function $F : E \to \overline{\mathbb{R}}$ is said subdifferentiable at point $u \in E$ if and only if there exists an affine continuous function $\mathcal{A}$ with slope $\varphi \in E^\star$ which is exact at point u and which lower bounds F on E. In that case the slope φ is called a subgradient of F at point u. The set of all subgradients of F at point u is named the subdifferential of F at point u and noted $\partial F(u)$. By convention we set $\partial F(u) = \emptyset$ when F is not subdifferentiable at point u.*

Notice that if the affine continuous function $\mathcal{A}$ with slope $\varphi \in E^\star$ is exact at the point u, $F(u)$ is necessarily finite and we have

$$\mathcal{A} = v \mapsto \langle \varphi, v - u\rangle + F(u)\,. \tag{2.16}$$

In that case $\mathcal{A}$ lower bounds F (that is, by definition, $\varphi \in \partial F(u)$) if and only if

$$\forall v \in E, \quad \langle \varphi, v - u\rangle + F(u) \leq F(v)\,.$$

Another point of view is that, as soon as the affine continuous function $\mathcal{A}$ with slope φ is exact at point u, it lower bounds F if and only if it has maximal constant term α, that is (recall how was built the Legendre-Fenchel transform in Section 2.2.2), when $\alpha = -F^\star(\varphi)$. Therefore identifying the constant term of (2.16) to the quantity $-F^\star(\varphi)$, yields

$$F(u) + F^\star(\varphi) = \langle \varphi, u \rangle \,.$$

These considerations give two characterizations of the subgradients of F at point u.

Proposition 4 (characterizations of the subgradients). *Let $F : E \to \overline{\mathbb{R}}$ be a function, u a point of E, and φ an element of the dual space $E^\star$. The following properties are equivalent to each other:*

(i) $\varphi \in \partial F(u)$, *i.e.* φ *is a subgradient of* F *at point* u;

(ii) $F(u)$ *is finite and we have* $\langle \varphi, v - u \rangle + F(u) \leq F(v)$ *for any* $v \in E$;

(iii) $F(u) + F^\star(\varphi) = \langle \varphi, u \rangle$.

Proof. We already proved (i) ⇒ (ii). Assuming now that (ii) is satisfied, we get $\sup_{v \in E} \langle \varphi, v \rangle - F(v) \leq \langle \varphi, u \rangle - F(u)$, we recognize in the left-hand term the Legendre-Fenchel transform of F at φ, and the corresponding supremum is attained at $v = u$, thus we get $F^\star(\varphi) = \langle \varphi, u \rangle - F(u)$, which equivalent to (iii), so that (ii) ⇒ (iii). Last when (iii) is satisfied, the terms $F^\star(\varphi)$ and $F(u)$ are necessarily both finite since $\langle \varphi, u \rangle$ is finite, therefore the application $\mathcal{A} : v \mapsto \langle \varphi, v - u \rangle - F(u)$ is affine continuous on E, and since (iii) is satisfied, the supremum $F^\star(\varphi) = \sup_{v \in E} \langle \varphi, v \rangle - F(v)$ is attained at point u. Therefore we have $\langle \varphi, v \rangle - F(v) \leq \langle \varphi, u \rangle - F(u)$ for any $v \in E$, thus, $\mathcal{A}$ lower bounds F on E. Thus (i) is satisfied, so that we proved (iii) ⇒ (i). □

We immediately derive a fundamental result in optimization.

Corollary 1. *A function $F \in \Gamma_0(E)$ admits a minimum at point $u \in E$ if and only if the set of its subgradients at point u contains* 0,

$$F(u) = \min_{v \in E} F(v) \Leftrightarrow 0 \in \partial F(u) \,. \tag{2.17}$$

Proof. Using the equivalence (i) ⇔ (ii) of Proposition 4, we see that

$$0 \in \partial F(u) \Leftrightarrow F(u) \text{ is finite and } \forall v \in E,\ F(u) \leq F(v) \,.$$

We obtain the same result using the geometrical intuition: the function F is minimal at point u if and only if the continuous affine function having null slope and being exact at point u (that is the constant $\mathcal{A} : v \mapsto F(u)$) is less than F everywhere on E. □

The next Proposition shows that the subdifferentiability generalizes the notion of Gâteaux-differential for convex functions (see the proof in [Ekeland and Témam 1999, Chap. I, Prop. 5.3]).

Proposition 5 (relation with Gâteaux-differentiability). *If $F : E \to \overline{\mathbb{R}}$ is a convex function which is Gâteaux-differentiable at a point $u \in E$, it is subdifferentiable at the point u, and $\mathrm{D}F(u)$ is its only subgradient at point u, i.e. $\partial F(u) = \{\mathrm{D}F(u)\}$.*

Notice that the result announced in Proposition 5 does not remain true if we remove the convexity assumption. A trivial counter example is obtained when we consider the function $F = x \mapsto x^3$ from $\mathbb{R}$ into $\mathbb{R}$, which is differentiable anywhere, but subdifferentiable nowhere on $\mathbb{R}$, since it cannot be lower-bounded by any affine continuous function of $\mathbb{R}$.

We will finish this section with two important results about subdifferential calculus in $\Gamma(E)$, which will be largely used in the next sections. Proposition 6 focuses on the subdifferential of a sum $F + G$ of two elements of $\Gamma(E)$, and Proposition 7 on a link existing between ∂F and $\partial F^\star$.

Proposition 6 (subdifferential of a sum of functions). *If F and G are two elements of $\Gamma(E)$, and if there exists a point $v \in \mathrm{dom}F \cap \mathrm{dom}G$ where F (or G) is continuous, then we have*

$$\forall u \in E, \quad \partial(F+G)(u) = \partial F(u) + \partial G(u)\,,$$

where the sum of the two (possibly empty) sets $\partial F(u)$ and $\partial G(u)$ must be understood as the Minkowsky sum between those two sets.

If the inclusion $\partial F(u) + \partial G(u) \subset \partial(F+G)(u)$ is easy to prove using the relation (ii) of Proposition 4, the converse inclusion is much more difficult to show. A proof of Proposition 6 is available in [Ekeland and Témam 1999, Chap. I, Prop. 5.6], and makes use of the *Hahn-Banach separation Theorem*. In the following, we will use Proposition 6 in the case where F is Gâteaux-differentiable at u, so that $\partial(F+G)(u) = \{\mathrm{D}F(u)\} + \partial G(u)$.

Proposition 7. *For every function $F : E \to \overline{\mathbb{R}}$, we have*

$$\varphi \in \partial F(u) \Rightarrow u \in \partial F^\star(\varphi)\,.$$

Besides, when $F \in \Gamma(E)$, we have $\varphi \in \partial F(u) \Leftrightarrow u \in \partial F^\star(\varphi)$.

Proof (adapted from [Ekeland and Témam 1999, Chap. I, Cor. 5.2]). This property is a direct consequence of Proposition 4, indeed when $\varphi \in \partial F(u)$, we have $F(u)+F^\star(\varphi) = \langle \varphi, u\rangle$, and necessary $F(u)$ and $F^\star(\varphi)$ are both finite (in particular $\varphi \in \mathrm{dom} F^\star$). Since $F^{\star\star} \leq F$ (see Proposition 3) we have $F^{\star\star}(u)+F^\star(\varphi) \leq \langle \varphi, u\rangle$. Besides, by definition of the Legendre-Fenchel transform of $F^\star$ at point u, we have $F^{\star\star}(u) = \sup_{\tilde{\varphi}\in \mathrm{dom} F^\star} \langle \tilde{\varphi}, u\rangle - F^\star(\tilde{\varphi})$, thus the inverse inequality $F^{\star\star}(u) + F^\star(\varphi) \geq \langle \varphi, u\rangle$ is also satisfied. Finally we have $F^{\star\star}(u) + F^\star(\varphi) = \langle \varphi, u\rangle$, which is equivalent to $u \in \partial F^\star(\varphi)$, thanks again to proposition 4. Finally we proved the implication $\varphi \in \partial F(u) \Rightarrow u \in \partial F^\star(\varphi)$. If furthermore $F \in \Gamma(E)$, we have $F^{\star\star} = F$, therefore, $u \in \partial F^\star(\varphi) \Rightarrow F^\star(\varphi) + F^{\star\star}(u) = \langle \varphi, u\rangle \Leftrightarrow F^\star(\varphi) + F(u) = \langle \varphi, u\rangle \Rightarrow \varphi \in \partial F(u)$, which proves the converse implication. □

2.2.4 Framework of non-smooth optimization

From now, E denotes a finite dimensional Hilbert space, endowed with an inner product $\langle \cdot, \cdot \rangle$, and the corresponding norm $\|\cdot\| = \sqrt{\langle \cdot, \cdot \rangle}$. We are interested in the minimization of convex functions, i.e. to problems of type

$$\inf_{u\in E} F(u)\,, \tag{2.18}$$

where, unless explicitly mentioned, $F : E \to \overline{\mathbb{R}}$ denotes a proper, convex and lower semi-continuous function, called the *objective function*, or the *cost function*. When the infimum (2.18) is attained somewhere on E, the infimum is a minimum. In that case, any point u^* which realizes this minimum is named *a solution* of problem (2.18), or *a minimizer* of F, and we note

$$u^* \in \operatorname*{argmin}_{u\in E} F(u)\,.$$

If moreover the minimum is uniquely attained, we may note

$$u^* = \operatorname*{argmin}_{u\in E} F(u)\,.$$

Remark that given a non-empty closed and convex subset $\mathscr{C}$ of E, a *constrained* minimization problem of type

$$\inf_{u\in \mathscr{C}} G(u)\,,$$

with proper, convex and lower semi-continuous objective function $G : E \to \mathbb{R}$ is identical to (2.18) when we consider $F = G + \delta_{\mathscr{C}}$ (which is also proper, convex and lower semi-continuous on E, since $\delta_{\mathscr{C}} \in \Gamma(E)$ thanks to Remark 4, and $\delta_{\mathscr{C}}$ is proper since $\mathscr{C} \neq \emptyset$), indeed it is obvious that, in that case, the infimum is the same for both problems, as well as the set of solutions.

Proposition 8 (criterion of existence or uniqueness of solution). *If one of the two following properties is satisfied, the problem (2.18) has at least one solution.*

(i) the set $\mathrm{dom}F$ *is bounded,*

(ii) or the function F is coercive over E, i.e. $\lim_{\|u\| \to +\infty} F(u) = +\infty$.

If (i) or (ii) is satisfied, and if furthermore the function F is strictly convex on $\mathscr{C}$, then the problem (2.18) has exactly one solution.

We will define below the notion of strong convexity that will be useful in the following. It is straightforward to check that this notion extends the notion of strict convexity, in the sense that any strong convex function is necessarily strictly convex.

Definition 14 (strongly convex functions). *A function $F : E \to \overline{\mathbb{R}}$ is called strongly convex with constant η if and only if the function*

$$u \mapsto F(u) - \tfrac{\eta}{2}\|u\|^2$$

is convex on E. We say that F is strongly convex when there exists a constant $\eta > 0$ such that F is strongly convex with constant η.

The following Proposition is due to Rockafellar and Wets [1998, Prop. 12.60] and it links the strong convexity of a function to the regularity of its Legendre-Fenchel conjugate.

Proposition 9 (dualization of strong convexity). *If $F : E \to \overline{\mathbb{R}}$ is an element of $\Gamma_0(E)$ and $\eta > 0$, the following properties are equivalent to each other:*

(i) $F^\star$ is strongly convex with constant η;

(ii) F is Fréchet differentiable on E, and ∇F is Lipschitz continuous with constant $1/\eta$.

Remark 5. *As direct consequence of this proposition, we can see that the Legendre-Fenchel transform $G^\star$ of a η-strongly convex function $G \in \Gamma_0(E)$ is Fréchet differentiable on E, and $\nabla G^\star$ is Lipschitz continuous with constant $1/\eta$. Indeed, we easily obtain this result by applying Proposition 9 to $F = G^\star$ and using $G = G^{\star\star}$.*

2.2.5 Proximity operator and Moreau envelope

We will now present two fundamental and closely related operators: the *Moreau envelope* and the *proximity operator*, which were introduced by Moreau [1965], initially for the purpose of generalizing the operators of ℓ^2 square distance and ℓ^2 projection over a convex set. It turned out that they play a great role in convex optimization.

Proposition 10 (proximity operator and Moreau envelope with parameter σ). *We recall that $F : E \to \overline{\mathbb{R}}$ is a proper, convex and lower semi-continuous function. Given any point $v \in E$ and a real parameter $\sigma > 0$, the function*

$$u \mapsto \tfrac{1}{2\sigma}\|u - v\|^2 + F(u)$$

is proper, lower semi-continuous, strongly convex with constant $\frac{1}{\sigma}$ and coercive on E. Consequently, the problem

$$\inf_{u \in E} \tfrac{1}{2\sigma}\|u - v\|^2 + F(u)\,, \tag{2.19}$$

has a unique solution on E. We define accordingly the proximity operator $\mathrm{Prox}_{\sigma F} : E \to E$ and the Moreau envelope $\mathrm{M}_{\sigma F} : E \to \mathbb{R}$ of F with parameter σ by

$$\begin{cases} \mathrm{Prox}_{\sigma F}(v) = \operatorname*{argmin}_{u \in E} \tfrac{1}{2\sigma}\|u - v\|^2 + F(u) & \text{(2.20a)} \\ \mathrm{M}_{\sigma F}(v) \;\; = \min_{u \in E} \tfrac{1}{2\sigma}\|u - v\|^2 + F(u) & \text{(2.20b)} \end{cases}$$

for any $v \in E$. Note that $\mathrm{M}_{\sigma F}$ is also called the Moreau-Yosida regularization of F with parameter σ [Yosida 1980].

Proof. For any $v \in E$, and any $\sigma > 0$, we see that the function $G : u \mapsto \frac{1}{2\sigma}\|u - v\|^2 + F(u)$ is an element of $\Gamma_0(E)$, since both functions F and $u \mapsto \frac{1}{\sigma}\|u - v\|^2$ are in $\Gamma_0(E)$. Also, it is easy to check that $u \mapsto G(u) - \frac{1}{2\sigma}\|u\|^2$ is convex (since $u \mapsto F(u) - \frac{1}{\sigma}\langle u, v\rangle$ is convex), therefore G is strongly convex with constant $1/\sigma$. It remains to show that G is coercive on E. Since $F \in \Gamma_0(E)$, F is

the pointwise supremum of the set of its lower bounding affine continuous functions, which is not empty (otherwise, F would be the constant $-\infty$, which does not belong to $\Gamma_0(E)$). Therefore, their exists at least one affine continuous function $\mathcal{A}$ such as $F \geq \mathcal{A}$, so that for any $u \in E$, we have $G(u) \geq \frac{1}{2\sigma}\|u-v\|^2 + \mathcal{A}(u)$. Since the right-hand term $u \mapsto \|u-v\|^2 + \mathcal{A}(u)$ of this inequality is coercive (noting φ and α the slope and constant terms of $\mathcal{A}$, we have $\frac{1}{2\sigma}\|u-v\|^2 + \mathcal{A}(u) = \frac{1}{2\sigma}\|u-v\|^2 + \langle \varphi, u\rangle + \alpha = \frac{1}{2\sigma}\|u-v+\sigma\varphi\|^2 - \frac{\sigma}{2}\|\varphi\|^2 + \langle \varphi, v\rangle + \alpha$, whose limit as $\|u\| \to +\infty$, is $+\infty$), it follows that $G(u) \to +\infty$ when $\|u\| \to +\infty$. □

Remark 6. *When F is the indicator function of a non-empty, closed and convex set $\mathscr{C}$, i.e. $F = \delta_{\mathscr{C}}$, $\mathrm{Prox}_{\sigma F}$ is exactly the projection on $\mathscr{C}$, noted $\pi_{\mathscr{C}}$, and $\mathrm{M}_{\sigma F}$ is proportional to the square distance between the point v and the set $\mathscr{C}$, noted $d(v,\mathscr{C})^2$,*

$$\forall v \in E,\ \forall \sigma > 0, \quad \mathrm{Prox}_{\sigma\delta_{\mathscr{C}}}(v) = \pi_{\mathscr{C}}(v) \quad \textit{and} \quad \mathrm{M}_{\sigma\delta_{\mathscr{C}}}(v) = \tfrac{1}{2\sigma}\, d(v,\mathscr{C})^2\,.$$

Now, let us focus on the proximity operators, and show how they can be used to handle problem (2.18). The following results are direct consequences of the definition of $\mathrm{Prox}_{\sigma F}$.

Proposition 11. *For any point $v \in E$, for any parameter $\sigma > 0$, we have the equivalence*

$$v^* = \mathrm{Prox}_{\sigma F} \Leftrightarrow v \in v^* + \sigma \partial F(v^*)\,.$$

The right-hand term can be noted $v \in (I + \sigma\partial F)\,(v^*)$, where I denotes the identity operator, and $v^* \mapsto (I + \sigma\partial F)(v^*)$ is a multivalued operator. For that reason, another common notation for the proximity operator $\mathrm{Prox}_{\sigma F}$ is

$$\mathrm{Prox}_{\sigma F}(v) = (I + \sigma\partial F)^{-1}(v)\,,$$

and $(I + \sigma\partial F)^{-1}$ is single valued, since $\mathrm{Prox}_{\sigma F}(v)$ is uniquely defined (Proposition 10) for any v.

Proof. Let $v \in E$, $\sigma > 0$, and $G \in \Gamma_0(E)$, defined by $G = u \mapsto \frac{1}{2\sigma}\|u-v\|^2$. Since G is differentiable (and thus Gâteaux-differentiable) everywhere on E, it is also subdifferentiable on E, and since E is a Hilbert space, we can identify $E^\star$ to E and write

$$\forall u \in E, \quad \partial G(u) = \{\mathrm{D}G(u)\} = \{\nabla G(u)\} = \{(u-v)/\sigma\}\ ,$$

where the identification $\mathrm{D}G(u) = \nabla G(u)$ must be of course understood as $\mathrm{D}G(u) = v \mapsto \langle \nabla G(u), v \rangle$. Using Corollary 1, and Proposition 6, we get

$$v^* = \mathrm{Prox}_{\sigma F}(v) \Leftrightarrow 0 \in \partial G(v^*) + \partial F(v^*) \Leftrightarrow 0 \in \frac{v^* - v}{\sigma} + \partial F(v^*)\,,$$

which, after basic manipulations performed on the right-hand term, is equivalent to $v \in v^* + \sigma \partial F(v^*)$. □

Proposition 12. *The solutions of* (2.18) *are the fixed points of* $\mathrm{Prox}_{\sigma F}$*, i.e.*

$$\forall \sigma > 0, \quad u^* \in \operatorname*{argmin}_{u \in E} F(u) \Leftrightarrow u^* = \mathrm{Prox}_{\sigma F}(u^*)\,.$$

Proof. We have $u^* \in \operatorname{argmin}_{u \in E} F(u) \Leftrightarrow 0 \in \partial F(u^*) \Leftrightarrow u^* \in u^* + \sigma \partial F(u^*)$, which is equivalent to $u^* = (I + \sigma \partial F)^{-1}(u^*)$, thanks to Proposition 11. □

Another fundamental property of the proximity operator, due to Moreau [1965] (in the case $\sigma = 1$), and that was generalized by Rockafellar [1976] (for any $\sigma > 0$), is now well known as the *Moreau's identity*.

Proposition 13 (Moreau's identity). *Let* $F \in \Gamma_0(E)$ *and* $\sigma > 0$*, we have*

$$\forall v \in E, \quad v = (I + \sigma \partial F)^{-1}(v) + \sigma \big(I + \tfrac{1}{\sigma} \partial F^\star\big)^{-1} \left(\tfrac{v}{\sigma}\right)\,.$$

Proof. For any $F \in \Gamma_0(E)$, $v \in E$, and $\sigma > 0$, we have (Proposition 11)

$$v^* = (I + \sigma \partial F)^{-1}(v) \Leftrightarrow v \in v^* + \sigma \partial F(v^*) \Leftrightarrow \tfrac{v - v^*}{\sigma} \in \partial F(v^*)\,,$$

and using Proposition 7, we get $\frac{v-v^*}{\sigma} \in \partial F(v^*) \Leftrightarrow v^* \in \partial F^\star \left(\frac{v-v^*}{\sigma}\right)$. Besides, we have

$$v^* \in \partial F^\star \left(\tfrac{v-v^*}{\sigma}\right) \Leftrightarrow \tfrac{v}{\sigma} \in \tfrac{v-v^*}{\sigma} + \tfrac{1}{\sigma} \partial F^\star \left(\tfrac{v-v^*}{\sigma}\right) \Leftrightarrow \tfrac{v-v^*}{\sigma} = \big(I + \tfrac{1}{\sigma} \partial F^\star\big)^{-1} \left(\tfrac{v}{\sigma}\right)\,.$$

Finally, replacing v^* by $(I + \sigma \partial F)^{-1}(v)$ in the right-hand term yields the announced result. □

Now, let us comment those results. First, the characterization of the solutions of (2.18) as the fixed points of $\mathrm{Prox}_{\sigma F}$ naturally encourage to consider the following numerical scheme, in order to compute a fixed point of $\mathrm{Prox}_{\sigma F}$ (and thus a minimizer of F),

$$\left\{ \begin{array}{l} \text{choose } \sigma > 0 \text{ and } u^0 \in E\,, \\ \forall k \geq 0, \quad u^{k+1} = \mathrm{Prox}_{\sigma F}(u^k)\,. \end{array} \right. \tag{2.21}$$

It turns out from Proposition 12 that if $\text{Prox}_{\sigma F}$ were a contraction (i.e. Lipschitz continuous with constant less than 1), the numerical scheme (2.21) would converge to a fixed point of $\text{Prox}_{\sigma F}$. In general the $\text{Prox}_{\sigma F}$ operator is not a contraction (unless F is strongly convex), but it satisfies a different property called *firm non-expansiveness* which is sufficient to ensure the convergence of the scheme (2.21) toward a fixed point of $\text{Prox}_{\sigma F}$, provided that such a fixed point exists, i.e. provided that the set of the minimizers of F is non-empty (see more explanations in [Rockafellar 1976, Parikh and Boyd 2013]). Besides, from the iteration $u^{k+1} = \text{Prox}_{\sigma F}(u^k)$, and thanks to Proposition 11, we get $u^k \in u^{k+1} + \sigma \partial F(u^{k+1})$, thus

$$\forall k \geq 0, \quad u^{k+1} \in u^k - \sigma \partial F(u^{k+1}) \,,$$

which can be interpreted as a *semi-implicit subgradient descent* scheme with step σ. Remark that this scheme may be used even in the smooth setting (i.e. when F is Gâteaux-differentiable), in that case its semi-implicitness provides some better conditioning and stability properties than the usual gradient descent scheme (the classical convergence Theorems associated to the steepest gradient descent schemes can be found for instance in [Luenberger and Ye 1984, Weiss 2008]), in particular, and as we already stated before, the convergence of the scheme (2.21) is ensured whatever the choice of σ (as soon as F has minimizers). The obvious limitation of the proximal fixed point scheme, is that each iteration k of (2.21) requires the minimization of the function

$$u \mapsto \tfrac{1}{2\sigma}\|u - u^k\|^2 + F(u) \,,$$

which is usually easier than the direct minimization of F (thanks to the regularity provided by the smooth and strictly convex quadratic term), but may remain non-trivial depending on the nature of F. Besides, Moreau's identity states that the computation of $\text{Prox}_{\sigma F}$ can be done through the computation of $\text{Prox}_{\mu F^\star}$ (taking $\mu = \frac{1}{\sigma}$), opening other possibilities to perform the iterations of the proximal fixed point scheme (2.21).

Remark 7 (Proximal Forward Backward Splitting Algorithm). *An interesting variant of (2.21) is obtained, under the assumption that the function F may be composed as the sum $F = F_1 + F_2$ of two elements of $\Gamma_0(E)$, and F_2 is differentiable. In that case, any minimizer u^* of F satisfies (use Proposition 6 and Corollary 1)*

$$0 \in \partial(F_1 + F_2)(u^*) = \partial F_1(u^*) + \{\nabla F_2(u^*)\}$$

and thus, for any $\sigma > 0$, *we have* $u^* - \sigma\nabla F_2(\bar{x}) \in u^* + \sigma\partial F_1(u^*)$, *yielding*

$$u^* = (I + \sigma\partial F_1)^{-1}(u^* - \sigma\nabla F_2(u^*)) \ ,$$

so that u^* *is a fixed-point of* $u \mapsto (I + \sigma\partial F_1)^{-1}(u - \sigma\nabla F_2(u))$. *The Proximal Forward Backward Splitting algorithm simply consists in the numerical scheme*

$$\left\{ \begin{array}{l} \text{choose } \sigma > 0 \text{ and } u^0 \in E, \\ \forall k \geq 0, \quad u^{k+1} = \mathrm{Prox}_{\sigma F_1}\left(u^k - \sigma\nabla F_2(u^k)\right) \ , \end{array} \right.$$

the forward *term refers to the explicit descent step* $u^{k+1/2} = u^k - \sigma F_2(u^k)$, *while the* backward *term refers to the semi-implicit descent step* $u^{k+1} = \mathrm{Prox}_{\sigma F_1}(u^{k+1/2})$. *A recent mathematical study of this algorithm can be found in [Combettes and Wajs 2005], with a nice proof of convergence provided that* F_2 *has a Lipschitz continuous gradient (see also [Weiss 2008]).*

Now we focus on the Moreau envelope, we describe below a set of properties that are helpful to understand the key role it plays in non-smooth convex optimization. In particular, we will explain why $\mathrm{M}_{\sigma F}$ is essentially a smoothed (or regularized) version of F, which is differentiable, even when F is not.

Proposition 14 (Some properties of the Moreau envelope). *For any element* $F \in \Gamma_0(E)$, *and any parameter* $\sigma > 0$, *the following properties are satisfied.*

(i) $(\mathrm{M}_{\sigma F})^\star = F^\star + \frac{\sigma}{2}\|\cdot\|^2$;

(ii) $\mathrm{M}_{\sigma F} \in \Gamma(E)$ *and* $\mathrm{M}_{\sigma F} = (\mathrm{M}_{\sigma F})^{\star\star} = \left(F^\star + \frac{\sigma}{2}\|\cdot\|^2\right)^\star$;

(iii) $\mathrm{M}_{\sigma F}$ *is Fréchet-differentiable on* E, *and*

$$\forall v \in E, \quad \nabla \mathrm{M}_{\sigma F}(v) = \tfrac{1}{\sigma}(v - \mathrm{Prox}_{\sigma F}(v)) \ ,$$

besides it has same set of minimizer than F,

$$u^* \in \operatorname*{argmin}_{u \in E} F(u) \Leftrightarrow u^* \in \operatorname*{argmin}_{u \in E} \mathrm{M}_{\sigma F}(u) \ ;$$

Proof. Those properties will be only partially proved here, but we will at least comment each one of them at the end of this section.

(i) To prove this property, we use the *infimal convolution* operation that was introduced by Moreau [1963]. Given two functions f and g in $\Gamma_0(E)$, the

infimal convolution between f and g is the function $(f \,\square\, g) : E \to \overline{\mathbb{R}}$ defined by

$$\forall v \in E, \quad (f \,\square\, g)(v) = \inf_{u \in E} f(u) + g(u - v)\,.$$

The infimal convolution is said *dual to the sum* because it satisfies (see [Rockafellar 1970])

$$\forall f, g \in \Gamma_0(E), \quad (f \,\square\, g)^\star = f^\star + g^\star \quad \text{and} \quad (f + g)^\star = (f^\star \,\square\, g^\star)\,.$$

Now, from its definition, we see that the Moreau envelope $\mathrm{M}_{\sigma F}$ is the infimal convolution between F and $G : u \mapsto \frac{1}{2\sigma}\|u\|^2$. Since we can easily show that $G^\star = \varphi \mapsto \frac{\sigma}{2}\|\varphi\|^2$ (the square norm of φ must be understood as the square norm in E of the unique element $v_\varphi \in E$ to which φ identifies), we get the announced result.

(ii) This property is a direct consequence of (i). Indeed, we have seen that $\mathrm{M}_{\sigma F} = (F \,\square\, G)$, where both F and G are elements of $\Gamma_0(E)$, therefore we have $(\mathrm{M}_{\sigma F})^{\star\star} = (F^{\star\star} \,\square\, G^{\star\star}) = (F \,\square\, G) = \mathrm{M}_{\sigma F}$, consequently $\mathrm{M}_{\sigma F} \in \Gamma(E)$. Thus, using $(\mathrm{M}_{\sigma F})^\star = F^\star + \frac{\sigma}{2}\|\cdot\|^2$, we get $\mathrm{M}_{\sigma F} = (\mathrm{M}_{\sigma F})^{\star\star} = \left(F^\star + \frac{\sigma}{2}\|\cdot\|^2\right)^\star$.

(iii) The Fréchet-differentiability of $\mathrm{M}_{\sigma F}$ is a direct consequence of Proposition 9 (simply remark that $(\mathrm{M}_{\sigma F}{}^\star = F^\star + \frac{\sigma}{2}\|\cdot\|^2$ is a strongly convex function), however the computation of its differential is nontrivial and will be admitted here (see the proof in Moreau [1965], in the case $\sigma = 1$, or in [Rockafellar and Wets 1998, Thm. 2.26], for any $\sigma > 0$). Now, remark that thanks to Proposition 12, and using $\nabla \mathrm{M}_{\sigma F} = v \mapsto \frac{1}{\sigma}(v - \mathrm{Prox}_{\sigma F}(v))$, we have the equivalence

$$u^* \in \operatorname*{argmin}_{u \in E} F(u) \Leftrightarrow u^* = \mathrm{Prox}_{\sigma F}(u^*) \Leftrightarrow \nabla \mathrm{M}_{\sigma F}(u^*) = 0\,,$$

and the relation $\nabla \mathrm{M}_{\sigma F}(u^*) = 0$ is a well known necessary and sufficient condition to optimality, for the convex and differentiable function $\mathrm{M}_{\sigma F}$. □

We close this section with comments about each property announced in Proposition 14. Thanks to properties (i) and (ii), we see that $\mathrm{M}_{\sigma F}$ is obtained by taking the Legendre-Fenchel transform $F^\star$ of the objective function F, adding a quadratic regularization, and then taking again the Legendre-Fenchel transform. As pointed out by Parikh and Boyd [2013], without adding the regularization, this would simply give back the initial function F, but thanks to the addition of the quadratic regularization, we obtain $\mathrm{M}_{\sigma F}$, which can be viewed as a smooth approximation of F, differentiable everywhere on E (thanks to Proposition 9 and

Remark 5), since $\mathrm{M}_{\sigma F}$ is the Legendre-Fenchel transform of the strongly convex function $(\mathrm{M}_{\sigma F})^\star = F^\star + \frac{\sigma}{2}\|\cdot\|^2 \in \Gamma_0(E)$. Last, the property (iii) confirms that $\mathrm{M}_{\sigma F}$ is differentiable on E, and states that the set of minimizers of F is the same as the set of the minimizers of its Moreau envelope. From the expression of its gradient $\nabla \mathrm{M}_{\sigma F}(v) = \frac{1}{\sigma}\left(v - \mathrm{Prox}_{\sigma F}(v)\right)$ we get

$$\forall v \in E, \quad \mathrm{Prox}_{\sigma F}(v) = v - \sigma \nabla \mathrm{M}_{\sigma F}(v)\,.$$

Consequently, the proximal fixed point scheme (2.21) can be interpreted as a gradient descent scheme (with step σ) for the minimization of $\mathrm{M}_{\sigma F}$. Of course the Moreau envelope $\mathrm{M}_{\sigma F}$ suffers from the same limitation as the proximity operator $\mathrm{Prox}_{\sigma F}$, in the sense that minimizing $\mathrm{M}_{\sigma F}$ may be as difficult as minimizing F directly, the essential difference being that the strong convexity added by the quadratic regularizer can significantly improve the regularity of the problem.

2.3 Application to total variation based image processing

In this section, we will use the duality tools presented in Section 2.2 to perform several image processing tasks based on total variation minimization. We place ourselves in the discrete setting, let Ω be a bounded subset of $\mathbb{Z}^2$, and $\mathbb{R}^\Omega$ the space of the real valued images with domain Ω, which, endowed with the Euclidean inner product, defined by

$$\forall u \in \mathbb{R}^\Omega,\ \forall v \in \mathbb{R}^\Omega, \quad \langle u, v\rangle_{\mathbb{R}^\Omega} = \sum_{(x,y)\in\Omega} u(x,y)\, v(x,y)\,,$$

is a finite dimensional Hilbert space. We denote by ∇^{d} the finite difference scheme (2.3), and by $\mathrm{div}^{\mathrm{d}}$, the corresponding discrete divergence, given in (2.4), which satisfies $(\nabla^{\mathrm{d}})^* = -\mathrm{div}^{\mathrm{d}}$. Since for any $u \in \mathbb{R}^\Omega$, we note $\nabla^{\mathrm{d}} u = (\nabla_1^{\mathrm{d}} u, \nabla_2^{\mathrm{d}} u)$, where $\nabla_1^{\mathrm{d}} u$ and $\nabla_2^{\mathrm{d}} u$ are two elements of $\mathbb{R}^\Omega$ representing the discrete derivatives of u in the horizontal and vertical directions, we will naturally consider the space $\mathbb{R}^\Omega \times \mathbb{R}^\Omega$, which is also a finite dimensional Hilbert space, once it is endowed with the inner product defined by

$$\langle (p_1, p_2), (q_1, q_2)\rangle_{\mathbb{R}^\Omega \times \mathbb{R}^\Omega} = \langle p_1, q_1\rangle_{\mathbb{R}^\Omega} + \langle p_2, q_2\rangle_{\mathbb{R}^\Omega}\,,$$

for any $p_1, p_2, q_1, q_2 \in \mathbb{R}^\Omega$. More generally, given any finite dimensional Hilbert space E, we will note $\langle\cdot,\cdot\rangle_E$ the Euclidean inner product on E, however, for

commodity, and when no ambiguity is possible, we may drop the index E and simply note $\langle \cdot, \cdot \rangle$. Last, since a Hilbert space E identifies to its dual space $E^\star$, from now, any element φ in $E^\star$, will be replaced by the unique element $p_\varphi \in E$ satisfying $\varphi(q) = \langle p_\varphi, q \rangle_E$ for any $q \in E$. In particular, the Legendre-Fenchel transform of any function defined on E will be viewed as a function of E instead of a function of $E^\star$.

2.3.1 Dual formulation of the total variation and its variants

Isotropic and anisotropic total variation

We recall the definition of $\mathrm{TV}_a^{\mathrm{d}}$ using the norm $\|\cdot\|_{1,a}$ over the space $\mathbb{R}^\Omega \times \mathbb{R}^\Omega$,

$$\forall u \in \mathbb{R}^\Omega, \quad \mathrm{TV}_a^{\mathrm{d}}(u) = \|\nabla^{\mathrm{d}} u\|_{1,a} = \sum_{(x,y)\in\Omega} |\nabla^{\mathrm{d}} u(x,y)|_a \,, \tag{2.22}$$

where $|\cdot|_a$ denotes the ℓ^a norm over $\mathbb{R}^2$ defined by

$$\forall (z_1, z_2) \in \mathbb{R}^2, \quad |(z_1, z_2)|_a = \begin{cases} \left(|z_1|^a + |z_2|^a\right)^{1/a} & \text{if } 1 \leq a < +\infty \\ \max\left(|z_1|, |z_2|\right) & \text{if } a = +\infty \,. \end{cases}$$

The isotropic and anisotropic discrete total variation correspond to the choices $a = 2$ and $a = 1$ respectively. However, within all this section, we will consider a more general setting, where a denotes a (possibly infinite) number higher than one ($1 \leq a \leq +\infty$), since this generalization will not complicate the proofs. We denote by a' the conjugate of a which is the unique number satisfying $\frac{1}{a} + \frac{1}{a'} = 1$ (taking as convention that 1 and $+\infty$ are conjugate to each other).

Proposition 15 (dual formulation of $\mathrm{TV}_a^{\mathrm{d}}$). *For any image $u \in \mathbb{R}^\Omega$, one has*

$$\mathrm{TV}_a^{\mathrm{d}}(u) = \max_{p \in \mathbb{R}^\Omega \times \mathbb{R}^\Omega} \langle \nabla^{\mathrm{d}} u, p \rangle_{\mathbb{R}^\Omega \times \mathbb{R}^\Omega} - \delta_{\|\cdot\|_{\infty,a'} \leq 1}(p) \,,$$

where $\delta_{\|\cdot\|_{\infty,a'}\leq 1}$ denotes the indicator function of the closed unit ball for the norm $\|\cdot\|_{\infty,a'} : p \mapsto \max_{(x,y)\in\Omega} |p(x,y)|_{a'}$, which means

$$\forall p \in \mathbb{R}^\Omega \times \mathbb{R}^\Omega, \quad \delta_{\|\cdot\|_{\infty,a'}\leq 1}(p) = \begin{cases} 0 & \textit{if } \|p\|_{\infty,a'} \leq 1, \\ +\infty & \textit{otherwise.} \end{cases}$$

To prove Proposition 15, we need some intermediate Lemmas.

Lemma 1 (The Legendre-Fenchel transform of a norm). *Let $\|\cdot\|$ be a norm on a finite dimensional real Hilbert space E endowed with the Euclidean inner product $\langle\cdot,\cdot\rangle_E$. Let $\|\cdot\|_*$ be the dual norm of $\|\cdot\|$, which is defined by*

$$\forall v \in E, \quad \|v\|_* = \sup_{u\in E,\ \|u\|\leq 1} \langle v, u\rangle_E \,.$$

Then, the Legendre-Fenchel transform of the $\|\cdot\|$, abusively noted $\|\cdot\|^\star$, is the indicator function of the closed unit ball for the dual norm $\|\cdot\|_*$, defined by

$$\forall v \in E, \quad \delta_{\|\cdot\|_*\leq 1}(v) = \begin{cases} 0 & \text{if } \|v\|_* \leq 1 \\ +\infty & \text{otherwise.} \end{cases}$$

In particular, since the two norms $|\cdot|_a$ and $|\cdot|_{a'}$ defined on $E = \mathbb{R}^2$ are dual to each other, we have $|\cdot|_a^\star = \delta_{|\cdot|_{a'}\leq 1}$.

Proof. Let us compute the Legendre-Fenchel transform of $\delta_{\|\cdot\|_*\leq 1}$. For any $u \in E$, we have

$$\delta^\star_{\|\cdot\|_*\leq 1}(u) = \sup_{v\in E} \langle u, v\rangle_E - \delta_{\|\cdot\|_*\leq 1}(v) = \sup_{v\in E,\ \|v\|_*\leq 1} \langle u, v\rangle_E = \|u\| \,,$$

since the dual norm of $\|\cdot\|_*$ is the norm $\|\cdot\|$, because of the reflexivity of E. Besides, since $\delta_{\|\cdot\|_*\leq 1} \in \Gamma(E)$ (thanks to Remark 4), we have $\delta_{\|\cdot\|_*\leq 1} = \delta^{\star\star}_{\|\cdot\|_*\leq 1}$ (using Proposition 3). Thus, for any $v \in E$, we have $\|v\|^\star = \delta^{\star\star}_{\|\cdot\|_*\leq 1}(v) = \delta_{\|\cdot\|_*\leq 1}(v)$. □

Remark 8. *Another proof of Lemma 1, based on a direct computation of the Legendre-Fenchel transform of $\|\cdot\|$, can be found in [Boyd and Vandenberghe 2004, Example 2.26].*

Lemma 2 (Legendre-Fenchel transform of the norm $\|\cdot\|_{1,a}$). *The Legendre-Fenchel transform of the norm $\|\cdot\|_{1,a}$ defined in (2.22) is the indicator function of the closed unit ball for the norm $\|\cdot\|_{\infty,a'}$ defined in Proposition 15.*

Proof. Let $p \in \mathbb{R}^\Omega \times \mathbb{R}^\Omega$, by definition, the Legendre-Fenchel transform of $\|\cdot\|_{1,a}$

at p, noted $\|p\|_{1,a}^\star$, writes

$$\begin{aligned}\|p\|_{1,a}^\star &= \sup_{g\in\mathbb{R}^\Omega\times\mathbb{R}^\Omega} \langle p,g\rangle_{\mathbb{R}^\Omega\times\mathbb{R}^\Omega} - \|g\|_{1,a} \\ &= \sup_{g\in\mathbb{R}^\Omega\times\mathbb{R}^\Omega} \sum_{(x,y)\in\Omega} \langle p(x,y),g(x,y)\rangle_{\mathbb{R}^2} - |g(x,y)|_a \\ &= \sum_{(x,y)\in\Omega} \sup_{g(x,y)\in\mathbb{R}^2} \langle p(x,y),g(x,y)\rangle_{\mathbb{R}^2} - |g(x,y)|_a \\ &= \sum_{(x,y)\in\Omega} |p(x,y)|_a^\star \\ &= \sum_{(x,y)\in\Omega} \delta_{|\cdot|_{a'}\leq 1}(p(x,y))\end{aligned}$$

since $|\cdot|_a^\star = \delta_{|\cdot|_{a'}\leq 1}$ thanks to Lemma 1. It follows that $\|p\|_{1,a}^\star = 0$ when $\max_{(x,y)\in\Omega} |p(x,y)|_{a'} \leq 1$, and $\|p\|_{1,a}^\star = +\infty$ otherwise, i.e., $\|p\|_{1,a}^\star = \delta_{\|\cdot\|_{\infty,a'}\leq 1}(p)$. □

Remark 9. *Another way to prove Lemma 2 consists in showing that the two norms $\|\cdot\|_{1,a}$ and $\|\cdot\|_{\infty,a'}$ are dual to each other, and then, to use Lemma 1 to get $\|\cdot\|_{1,a}^\star = \delta_{\|\cdot\|_{\infty,a'}\leq 1}$.*

Proof of Proposition 15. Since $\|\cdot\|_{1,a}$ is convex and l.s.c. over $\mathbb{R}^\Omega\times\mathbb{R}^\Omega$, it is an element of $\Gamma(\mathbb{R}^\Omega\times\mathbb{R}^\Omega)$, thereby $\|\cdot\|_{1,a} = \|\cdot\|_{1,a}^{\star\star}$ thanks to Proposition 3. Besides given an image $u\in\mathbb{R}^\Omega$, one as $\mathrm{TV}_a^{\mathrm{d}}(u) = \|\nabla^{\mathrm{d}}u\|_{1,a}$, therefore

$$\mathrm{TV}_a^{\mathrm{d}}(u) = \left\|\nabla^{\mathrm{d}}u\right\|_{1,a}^{\star\star} = \sup_{p\in\mathbb{R}^\Omega\times\mathbb{R}^\Omega} \langle\nabla^{\mathrm{d}}u,p\rangle_{\mathbb{R}^\Omega\times\mathbb{R}^\Omega} - \|p\|_{1,a}^\star\,,$$

and $\|p\|_{1,a}^\star$ is exactly $\delta_{\|\cdot\|_{\infty,a'}\leq 1}(p)$ thanks to Lemma 2. Last, one sees that the supremum is attained, since it is nothing but the maximum of the inner product term over the closed unit ball for the dual norm $\|\cdot\|_{\infty,a'}$. □

The Huber total variation

Let $\alpha > 0$, we recall below the definition of the Huber total variation with parameter α and define at the same time the function $\mathrm{H}_\alpha : \mathbb{R}^\Omega\times\mathbb{R}^\Omega \mapsto \mathbb{R}$,

$$\forall u\in\mathbb{R}^\Omega,\quad \mathrm{HTV}_\alpha^{\mathrm{d}}(u) = \mathrm{H}_\alpha(\nabla^{\mathrm{d}}u) := \sum_{(x,y)\in\Omega} \mathcal{H}_\alpha(\nabla^{\mathrm{d}}u(x,y))\,.$$

We recall also the definition of $\mathcal{H}_\alpha$, the Huber function with parameter α,

$$\forall z\in\mathbb{R}^2,\quad \mathcal{H}_\alpha(z) = \begin{cases} \frac{|z|_2^2}{2\alpha} & \text{if } |z|_2\leq\alpha\,, \\ |z|_2 - \frac{\alpha}{2} & \text{otherwise}\,. \end{cases}$$

The dual formulation of the discrete total variation described in Proposition 15 can be easily adapted to its Huber variant.

Proposition 16 (dual formulation of $\mathrm{HTV}^{\mathrm{d}}_{\alpha}$). *For any $\alpha > 0$, and for any image $u \in \mathbb{R}^{\Omega}$, one has*

$$\mathrm{HTV}^{\mathrm{d}}_{\alpha}(u) = \max_{p \in \mathbb{R}^{\Omega} \times \mathbb{R}^{\Omega}} \langle \nabla^{\mathrm{d}} u, p \rangle_{\mathbb{R}^{\Omega} \times \mathbb{R}^{\Omega}} - \delta_{\|\cdot\|_{\infty,2} \leq 1}(p) - \tfrac{\alpha}{2}\|p\|_2^2 \,.$$

In order to prove this Proposition, we need an intermediate Lemma.

Lemma 3. *$\mathcal{H}_{\alpha}$ is the Moreau envelope with parameter α of the ℓ^2 norm $|\cdot|_2$, it is therefore an element of $\Gamma(\mathbb{R}^2)$. Besides, its Legendre-Fenchel transform satisfies*

$$\forall p \in \mathbb{R}^2, \quad \mathcal{H}^{\star}_{\alpha}(p) = \begin{cases} \frac{\alpha}{2}|p|_2^2 & \text{if } |p|_2 \leq 1 \,, \\ +\infty & \text{otherwise,} \end{cases}$$

or equivalently, $\mathcal{H}^{\star}_{\alpha}(p) = \delta_{|\cdot|_2 \leq 1}(p) + \frac{\alpha}{2}|p|_2^2$, for any $p \in \mathbb{R}^2$.

Proof. Let us show that the Moreau envelope with parameter α of the ℓ^2 norm, noted $\mathrm{M}_{\alpha|\cdot|_2}$, is equal to $\mathcal{H}_{\alpha}$. Let $z \in \mathbb{R}^2$, by definition of the Moreau envelope (given in Proposition 10), we have

$$\mathrm{M}_{\alpha|\cdot|_2}(z) = \min_{y \in \mathbb{R}^2} \tfrac{1}{2\alpha}|y - z|_2^2 + |y|_2 \,,$$

and this minimum is reached at point $\tilde{y} = \mathrm{Prox}_{\alpha|\cdot|_2}(z)$. Thanks to Moreau's identity (Proposition 13) we have

$$\tilde{y} = z - \alpha \cdot \mathrm{Prox}_{\frac{1}{\alpha}|\cdot|_2^{\star}}(z/\alpha) \,, \tag{2.23}$$

and since $|\cdot|_2^{\star} = \delta_{|\cdot|_2 \leq 1}$ (Lemma 1), $\mathrm{Prox}_{\frac{1}{\alpha}|\cdot|_2^{\star}}(z/\alpha)$ is simply the ℓ^2 projection of the quantity z/α on the closed unit ball for the ℓ^2 norm, which is given by

$$\mathrm{Prox}_{\frac{1}{\alpha}|\cdot|_2^{\star}}(z/\alpha) = \begin{cases} z/\alpha & \text{if } |z|_2 \leq \alpha \\ z/|z|_2 & \text{otherwise.} \end{cases}$$

Now we can compute $\tilde{y}$ using (2.23) and derive the value of $\mathrm{M}_{\alpha|\cdot|_2}(z)$ using $\mathrm{M}_{\alpha|\cdot|_2}(z) = \frac{1}{2\alpha}|\tilde{y} - z|_2^2 + |\tilde{y}|_2$. Indeed, when $|z|_2 \leq \alpha$, we have $\tilde{y} = 0$ and $\mathrm{M}_{\alpha|\cdot|_2}(z) = \frac{|z|_2^2}{2\alpha}$. Otherwise, $|z|_2 > \alpha$, and we have $\tilde{y} = \lambda z$ where $\lambda = (1 - \frac{\alpha}{|z|_2}) \in (0, 1)$,

leading to $\mathrm{M}_{\alpha|\cdot|_2}(z) = |z|_2 - \frac{\alpha}{2}$. Finally, we proved that $\mathrm{M}_{\alpha|\cdot|_2}(z) = \mathcal{H}_\alpha(z)$. Consequently, $\mathcal{H}_\alpha \in \Gamma(\mathbb{R}^2)$, and $\mathcal{H}_\alpha^\star$ is equal to the sum between the Legendre-Fenchel transform of $\alpha|\cdot|_2$ (which is $\delta_{|\cdot|_2 \leq 1}$ using Lemma 1) and the quadratic term $\frac{\alpha}{2}|\cdot|_2^2$, thanks to Proposition 14. More precisely, we have $\mathcal{H}_\alpha^\star(p) = \delta_{|\cdot|_2 \leq 1}(p) + \frac{\alpha}{2}|p|_2^2$, for any $p \in \mathbb{R}^2$. □

Proof of Proposition 16. Since $\mathcal{H}_\alpha \in \Gamma(\mathbb{R}^2)$, we have $\mathrm{H}_\alpha \in \Gamma(\mathbb{R}^\Omega \times \mathbb{R}^\Omega)$, therefore $\mathrm{HTV}_\alpha^{\mathrm{d}}(u) = \mathrm{H}_\alpha(\nabla^{\mathrm{d}} u) = \mathrm{H}_\alpha^{\star\star}(\nabla^{\mathrm{d}} u)$ thanks to Proposition 3. We derive that

$$\mathrm{HTV}_\alpha^{\mathrm{d}}(u) = \mathrm{H}_\alpha^{\star\star}(\nabla^{\mathrm{d}} u) = \sup_{p \in \mathbb{R}^\Omega \times \mathbb{R}^\Omega} \langle \nabla^{\mathrm{d}} u, p \rangle_{\mathbb{R}^\Omega \times \mathbb{R}^\Omega} - \mathrm{H}_\alpha^\star(p)\,.$$

Besides, we have $\mathrm{H}_\alpha^\star(p) = \sum_{(x,y)\in\Omega} \mathcal{H}_\alpha^\star(p(x,y)) = \delta_{\|\cdot\|_{\infty,2} \leq 1}(p) + \frac{\alpha}{2}\|p\|_2^2$ (using Lemma 3). Again, the supremum is a maximum for the same reason as in the proof of Proposition 15, which yields the announced result. □

2.3.2 A first order primal-dual resolvant algorithm

Consider X and Y two finite-dimensional real vector spaces, an inner product $\langle \cdot, \cdot \rangle_Y$ over Y, and the generic saddle-point problem

$$\min_{x \in X} \max_{y \in Y} \; G(x) + \langle Kx, y \rangle_Y - F^\star(y)\,, \tag{2.24}$$

where $F \in \Gamma_0(Y)$, $G \in \Gamma_0(X)$ and $K : X \mapsto Y$ denotes a continuous linear operator. We set $H : (x, y) \to G(x) + \langle Kx, y \rangle_Y - F^\star(y)$ and we assume that problem (2.24) has at least a solution $(x^*, y^*) \in X \times Y$ (i.e. a saddle point of H).

Remark 10 (reformulation of (2.24)). *Using again Proposition 3, for any $x \in X$, we have $F(Kx) = F^{\star\star}(Kx) = \sup_{y \in Y} \langle Kx, y \rangle_Y - F^\star(y)$, therefore, one can interpret Equation (2.24) as a primal-dual formulation of the primal problem*

$$\operatorname*{argmin}_{x \in X} \; G(x) + F(Kx)\,, \tag{2.25}$$

as soon as $\sup_{y \in Y} \langle Kx, y \rangle_Y - F^\star(y)$ is indeed a maximum (which will be the case in practice), and in that case, if (x^*, y^*) is a solution of (2.24), x^* is automatically a solution of the primal problem (2.25).

We present in Algorithm 1 the numerical scheme proposed by Chambolle and Pock [2011] for solving the generic primal-dual saddle point problem (2.24).

Algorithm 1: Chambolle-Pock resolvant algorithm for problem (2.24)

Initialization: *Choose* $\tau, \sigma > 0$, $\theta \in [0,1]$, $(x^0, y^0) \in X \times Y$ *and set* $\overline{x}^0 = x^0$ *(the convergence of this algorithm toward a solution of the primal-dual problem* (2.24) *was proven in Chambolle and Pock [2011] for* $\theta = 1$ *when* $\tau\sigma|||K|||^2 < 1$*).*

Iterations ($k \geq 0$)**:** *update* x^k, y^k *and* $\overline{x}^k$ *as follows:*

$$\begin{cases} y^{k+1} = (I + \sigma \partial F^\star)^{-1}(y^k + \sigma K \overline{x}^k) & \text{(2.26a)} \\ x^{k+1} = (I + \tau \partial G)^{-1}(x^k - \tau K^* y^{k+1}) & \text{(2.26b)} \\ \overline{x}^{k+1} = x^{k+1} + \theta \left(x^{k+1} - x^k\right) & \text{(2.26c)} \end{cases}$$

We will now give some explanations about its spirit, and briefly summarize some theoretical results about its convergence (Propositions 17 and 18).

Let us first consider the setting $\theta = 0$, in that case we have $\bar{x}^k = x^k$ at each iteration k of Algorithm 1. For any $y \in Y$ let us set $P_y : x \mapsto H(x,y)$, which is an element of $\Gamma_0(X)$, and $D_x : y \mapsto -H(x,y)$, which is an element of $\Gamma_0(Y)$. We show below that the updates (2.26a) and (2.26b) boil down to the proximal scheme

$$\begin{cases} y^{k+1} &= (I + \sigma \partial D_{x^k})^{-1}(y^k) \\ x^{k+1} &= \left(I + \tau \partial P_{y^{k+1}}\right)^{-1}(x^k) \end{cases} \tag{2.27}$$

so that one iteration of Algorithm 1 can be interpreted as a semi-implicit ascent step of $y \mapsto H(x^k, y)$ followed by semi-implicit descent step of $x \mapsto H(x, y^{k+1})$, since we showed in Section 2.2.5 how the proximal iterations could be viewed as semi-implicit subgradient descent steps. Thereby, in the case $\theta = 0$, Algorithm 1 can be viewed as a semi-implicit variant of the classical Arrow-Hurwicz algorithm [Arrow et al. 1958].

Proof. We prove the result only for the dual update (2.26a), the primal update (2.26b) can be treated with similar arguments. Thanks to Proposition 11, we have

$$y^k + \sigma K x^k \in y^{k+1} + \sigma \partial F^\star(y^{k+1})\,.$$

using Propositions 5 and 6, we have $-Kx^k + \partial F^\star(y^{k+1}) = \partial D_{x^k}(y^{k+1})$, therefore, we have $y^k \in y^{k+1} + \partial D_{x^k}(y^{k+1})$, which is equivalent to $y^{k+1} = (I + \sigma \partial D_{x^k})^{-1}(y^k)$ using again Proposition 11. □

Now, remark how much it would be attractive to replace D_{x^k} by $D_{x^{k+1}}$ into (2.27), since this would make the scheme fully implicit. Unfortunately, this would also complicate too much the problem (in general, this would make the practical computation of (2.27) as complicate as the initial problem). By introducing the θ parameter into their algorithm, Chambolle and Pock simply replace D_{x^k} by $D_{\bar{x}^k}$ into (2.27), where $\bar{x}^k = x^k + \theta\left(x^k - x^{k-1}\right)$ represents a *linear approximation* of the next iterate x^{k+1}, based on the current and previous iterates x^k and x^{k-1} (also referred as *extrapolation*, or *over-relaxation*). Therefore, this operation can be viewed as a way to add more implicitness into (2.27), without complicating the practical computation of the iterates. In particular, in the case $\theta = 1$, Chambolle and Pock prove the convergence of their algorithm and gave an estimate of its convergence rate.

Proposition 17 (convergence of Algorithm 1). *Chambolle and Pock [2011] prove the following results.*

(i) When $\theta = 1$ and $\tau\sigma < |||K|||^2$, the sequence $(x^k, y^k)_{k\geq 0}$ generated by Algorithm 1 converges toward a solution (x^*, y^*) of (2.24), in particular the sequence $(x^k)_{k\geq 0}$ converges toward a solution x^* of the primal problem (2.25).

(ii) The convergence rate is $\mathcal{O}(1/N)$ for N iterations, however this rate does not apply directly to the iterates $(x^k, y^k)_{k\geq 0}$ but to the decrease toward zero of the duality gap (see [Chambolle and Pock 2011] for more details) between the Cesàro means $x_N = \frac{1}{N}\sum_{k=1}^{N} x^k$ and $y_N = \frac{1}{N}\sum_{k=1}^{N} y^k$.

We must emphasize that the convergence results summarized in Proposition 17 are derived under very few assumptions, since we only assumed that F and G were elements of $\Gamma_0(X)$ and $\Gamma_0(Y)$ and that problem (2.24) had solutions. Some accelerated variants of Algorithm 1 were also proposed by the same authors, which under additional regularity assumptions on F and G, achieve better convergence rates.

Proposition 18 (accelerated variants of Algorithm 1). *Provided supplementary regularity assumptions about F and G, Algorithm 1 can be accelerated (see Algorithms 2 and 3 in [Chambolle and Pock 2011]) as described below.*

(i) If G is uniformly convex with parameter $\gamma > 0$ (i.e. if $G^\star$ is differentiable and $\nabla G^\star$ is $1/\gamma$ Lipschitz-continuous, this is for instance the case when G is strongly convex with parameter γ), one can set

$$\theta = 1/\sqrt{1 + 2\gamma\tau}, \quad \tau = \tau\,\theta, \quad \sigma = \sigma/\theta\,,$$

at each iteration between the updates (2.26b) *and* (2.26c) *(i.e. between the updates of variables* x^k *and* $\bar{x}^k$*), in order to achieve a* $\mathscr{O}(1/N^2)$ *convergence rate (this time the convergence rate applies to the convergence of the Cesàro mean* $x_N = \frac{1}{N}\sum_{k=1}^{N} x^k$ *toward a solution of* (2.25)*).*

(ii) A similar acceleration is available when $F^\star$ *(instead of* G*) is uniformly convex.*

(iii) If G *and* $F^\star$ *are both uniformly convex with parameters* γ *and* η*, choosing constant steps with the setting*

$$\theta \in \left[\tfrac{1}{1+\mu}, 1\right], \ \tau = \frac{\mu}{2\gamma}, \ \sigma = \frac{\mu}{2\eta}, \ \text{with } \mu \leq \frac{2\sqrt{\gamma\eta}}{|||\lambda\nabla^{\mathrm{d}}|||},$$

yields a linear convergence rate, that is, $\mathscr{O}\left(e^{-N}\right)$*, of the Cesàro means* $x_N = \frac{1}{N}\sum_{k=1}^{N} x^k$ *and* $y_N = \frac{1}{N}\sum_{k=1}^{N} y^k$ *toward the (here unique) solution of* (2.24)*.*

The convergence rate in $\mathscr{O}(1/N)$ is shown to be optimal for a general class of convex optimization problems [Nesterov 2005], as well as the rates $\mathscr{O}(1/N^2)$ (see [Nesterov 1983, 2005, Beck and Teboulle 2009b]) and $\mathscr{O}(e^{-N})$ (see [Nesterov 2004]), provided the additional regularity assumptions on G and $F^\star$ evoked above.

Many algorithms, based on the Legendre-Fenchel duality and which have shown their efficiency in many imaging problems, can be found in the literature, as for instance, the closely related algorithms named Douglas-Rachford Splitting (originally proposed in [Douglas and Rachford 1956], see also [Eckstein and Bertsekas 1992, Combettes 2009] for further developments), and Alternating Direction Method of Multipliers (ADMM) [Gabay and Mercier 1976, Glowinski and Le Tallec 1989]. A nice review about the modern proximal algorithms, and the relations existing among each other, can be found in [Combettes and Pesquet 2011]. We would like to mention in particular the celebrated Proximal Forward Backward Splitting (PFBS) algorithm, already evoked in Section 2.2.5 (Remark 7), for which a convergence Theorem is derived in [Combettes and Wajs 2005]. A convergence rate in $\mathscr{O}(1/N)$ for the PFBS algorithm, thus identical to that stated in Proposition 17, was established in [Weiss 2008], where an acceleration in $\mathscr{O}(1/N^2)$ of the convergence rate is also proposed, at the cost of introducing additional proximal computation at each iteration of the scheme. Note also the Iterative Shrinkage Thresholding Algorithm (ISTA), which can be viewed as a Proximal Forward Backward Splitting algorithm applied to a particular class of inverse problems (that is, linear inverse problems with quadratic data-fidelity and ℓ^1 regularization, see for instance [Daubechies et al. 2004]), and its famous FISTA (Fast ISTA)

variant [Beck and Teboulle 2009b], which basically combines a Nesterov acceleration (which consists in a particular update of the descent step parameter), and an extrapolation (similar to the step (2.26c) of Algorithm 1), in order to reach the optimal $\mathscr{O}(1/N^2)$ convergence rate, without introducing an additional computational cost to the scheme (the same acceleration is in practice also applied to the more general PFBS algorithm). However, only the convergence in $\mathscr{O}(1/N^2)$ of the sequence $C(x^k)$ (noting C the cost function to minimize, and $\{x^k\}_{k\in\mathbb{N}}$ the sequence generated by FISTA), has been established so far. A convergence Theorem for the iterates of FISTA was recently proposed in [Chambolle and Dossal 2015], but no convergence rate is provided.

For all algorithms evoked above (including the Chambolle-Pock algorithm), the improvement of the convergence rate in $\mathscr{O}(1/N^2)$ is always done at the cost of introducing some additional regularity assumptions about the cost function to minimize (otherwise the rate $\mathscr{O}(1/N)$ is optimal). However, in the case where those regularity assumption are not satisfied, the formal convergence Theorem provided by Chambolle and Pock, relying on very low assumptions (in particular, no differentiability assumption for F and G is necessary), confers to their algorithm a valuable advantage. Besides, an important limitation of ISTA, FISTA and the PFBS algorithm, when used for solving TV regularized inverse problems, is that it involves the computation of proximal operator (more precisely, $\mathrm{Prox}_{\sigma\mathrm{TV}}$) for which no closed form is available, so that practical implementations require the use of an additional nested optimization algorithm dedicated to the computation of the proximal term (which usually cannot be done exactly in finite time, introducing errors into the scheme), while this limitation can be easily bypassed using the Chambolle-Pock algorithm, by increasing the number of dual variables (as it will be done in Section 2.3.4). See also [Aujol and Dossal 2015], where some convergence theorems are recovered for FISTA and PFBS algorithm, in the case when proximal maps are inexactly computed, provided that the corresponding errors can be controlled (which is unfortunately difficult for many applications).

A main limitation of the Chambolle-Pock algorithm, is that it involves the setting of two descent step parameters τ and σ, instead of only one for ISTA, FISTA and the PFBS algorithm (and its accelerated variants). Although the setting of τ and σ in Algorithm 1 is guided by the constraint $\tau\,\sigma < |||K|||^2$ of Proposition 17, the setting of these parameters can be difficult for applications where the operator K has a complicated structure, or, when $|||K|||$ is large, since it forces their setting to low values, which significantly slows down the algorithm. In order to overcome those shortcomings, Chambolle and Pock proposed a preconditioned version of their algorithm [Pock and Chambolle 2011], with the claim

that this variant significantly accelerates the convergence on problems with irregular K while leaving the computational complexity of the iterations basically unchanged. The Chambolle-Pock algorithm has been preferred in the following, for its simplicity, the nice theoretical convergence results evoked above, and its remarkable ability to address various image processing tasks.

2.3.3 Total variation based image denoising

In this section we focus on the pure image denoising problem, introduced by Rudin, Osher and Fatemi (ROF) in [Rudin et al. 1992]. Initially formulated as a constrained minimization of the total variation functional, it boils down to compute, for a given regularity parameter $\lambda \geq 0$, the image

$$u_{\text{ROF}} = \operatorname*{argmin}_{u \in \mathbb{R}^\Omega} \|u - u_0\|_2^2 + \lambda \, \text{TV}^{\text{d}}(u) \,, \tag{2.28}$$

from the noisy observation $u_0 \in \mathbb{R}^\Omega$ (or equivalently, to compute $u_{\text{ROF}} = \text{Prox}_{2\lambda \text{TV}^{\text{d}}}(u_0)$, so that we automatically get the existence and uniqueness of solution for the problem (2.28)). Recall that the same problem as (2.28) can be formulated using the Maximum A Posteriori methodology, as it was done in Section 2.1.2 (notice that here, the linear operator of the observation model (2.5) is simply the identity operator), assuming that the observed image u_0 was undergoing additive Gaussian noise with zero mean and (non-necessary known) variance σ^2. This yielded $\lambda = 2\beta\sigma$, so that we see why in practice, the choice of the regularity parameter must be adapted to the level of noise σ in u_0. In this section will be also considered the anisotropic and Huber variants of the ROF problem, which are obtained by replacing in (2.28) the classical discrete total variation term TV^{d} by TV_1^{d} or $\text{HTV}_\alpha^{\text{d}}$ (for a given parameter $\alpha > 0$).

Using Proposition 15, we immediately get a primal-dual reformulation of problem (2.28),

$$u_{\text{ROF}} = \operatorname*{argmin}_{u \in \mathbb{R}^\Omega} \max_{p \in \mathbb{R}^\Omega \times \mathbb{R}^\Omega} \|u - u_0\|_2^2 + \langle \lambda \nabla^{\text{d}} u, p \rangle_{\mathbb{R}^\Omega \times \mathbb{R}^\Omega} - \delta_{\|\cdot\|_{\infty,2} \leq 1}(p) \,, \tag{2.29}$$

which has exactly the form of problem (2.24) when we replace (x, y) by (u, p), and when we take $K = \lambda \nabla^{\text{d}}$ (with adjoint $K^* = -\lambda \text{div}^{\text{d}}$), $G(u) = \|u - u_0\|_2^2$ and $F^\star(p) = \delta_{\|\cdot\|_{\infty,2} \leq 1}(p)$. If we consider the anisotropic or Huber variants of ROF, we must modify accordingly the primal-dual problem (2.29) using the dual formulation of TV_1^{d} (Proposition 15) or $\text{HTV}_\alpha^{\text{d}}$ (Proposition 16). More precisely, we must replace the term $F^\star(p) = \delta_{\|\cdot\|_{\infty,2} \leq 1}(p)$ by $F^\star(p) = \delta_{\|\cdot\|_{\infty,\infty} \leq 1}(p)$ when

considering the anisotropic $\mathrm{TV}_1^{\mathrm{d}}$ variant of TV^{d}, and by $F^\star(p) = \delta_{\|\cdot\|_{\infty,2}\leq 1}(p) + \frac{\lambda\alpha}{2}\|p\|_2^2$ when considering the Huber-variant $\mathrm{HTV}_\alpha^{\mathrm{d}}$ of TV^{d}. Consequently, the solution of the ROF problem and its two variants can be numerically approached using the Chambolle and Pock algorithm, leading to Algorithm 2. Some numerical experiments are proposed in Figures 2.5, 2.6, and 2.7.

Remark 11 (induced ℓ^2 norm of $K = \lambda\nabla^{\mathrm{d}}$). *The classical discretization scheme (2.3) for ∇^{d} yields the upper bound $|||\lambda\nabla^{\mathrm{d}}||| \leq L := \sqrt{8}\lambda$ (see for instance [Chambolle 2004]). This bound is useful to set the time step τ and σ of the Chambolle-Pock algorithm. In practice, we will set $\tau = \sigma = 0.99/L$ so that the condition $\tau\sigma|||K|||^2 < 1$ is satisfied.*

Algorithm 2: resolvant algorithm for problem (2.28) and its variants.

Initialization: *Set L such as $|||\lambda\nabla^{\mathrm{d}}||| \leq L$ (for instance $L = \lambda\sqrt{8}$), set $\theta = 1$, and choose $\tau, \sigma > 0$ such as $\sigma\tau L^2 < 1$ (for instance $\tau = \sigma = 0.99/L$). Choose $u^0 \in \mathbb{R}^\Omega$, $p^0 \in \mathbb{R}^\Omega \times \mathbb{R}^\Omega$ and set $\overline{u}^0 = u^0$.*

Choice of the regularizer (TV^{d}, $\mathrm{TV}_1^{\mathrm{d}}$ or $\mathrm{HTV}_\alpha^{\mathrm{d}}$): *For solving the classical TV^{d} regularized problem (2.28), set $a' = 2$ and $\nu = 1$. For solving the anisotropic $\mathrm{TV}_1^{\mathrm{d}}$ variant, set $a' = +\infty$ and $\nu = 1$. Otherwise, for solving the $\mathrm{HTV}_\alpha^{\mathrm{d}}$ variant, set $a' = 2$ and $\nu = 1 + \sigma\lambda\alpha$.*

Internal definition(s): *Let $\pi_{\infty,a'} : \mathbb{R}^\Omega \times \mathbb{R}^\Omega \to \mathbb{R}^\Omega \times \mathbb{R}^\Omega$ be the ℓ^2 projection over the closed unit ball for the norm $\|\cdot\|_{\infty,a'}$, i.e., for any $p = (p_1, p_2) \in \mathbb{R}^\Omega \times \mathbb{R}^\Omega$ and any $(x, y) \in \Omega$,*

$$\pi_{\infty,a'}(p)(x,y) = \begin{cases} \frac{p(x,y)}{\max(1, |p(x,y)|_2)} & \text{if } a' = 2 \\ \left(\frac{p_1(x,y)}{\max(1,|p_1(x,y)|)}, \frac{p_2(x,y)}{\max(1,|p_2(x,y)|)}\right) & \text{if } a' = +\infty. \end{cases}$$

Iterations: *For $k \geq 0$, update u^k, p^k and $\overline{u}^k$ as follows,*

$$\begin{aligned} p^{k+1} &= \pi_{\infty,a'}\left(\frac{p^k + \sigma\lambda\nabla^{\mathrm{d}}\overline{u}^k}{\nu}\right) \\ u^{k+1} &= \frac{u^k + \tau\lambda\mathrm{div}^{\mathrm{d}}p^{k+1} + 2\tau u_0}{1+2\tau} \\ \overline{u}^{k+1} &= u^{k+1} + \theta\left(u^{k+1} - u^k\right) \end{aligned}$$

Remark 12 (acceleration for the Huber model). *In the case of the Huber model, the function* $G : u \mapsto \|u - u_0\|_2^2$ *is strongly convex with parameter* $\gamma = 2$*, and the function* $F^\star : p \mapsto \delta_{\|\cdot\|_{\infty,2}\leq 1}(p) + \frac{\lambda\alpha}{2}\|p\|_2^2$ *is strongly convex with parameter* $\eta = \lambda\alpha$*, therefore Algorithm 2 can be accelerated using the setting (iii) of Proposition 18, in order to reach a linear convergence rate, as illustrated in Figure 2.6.*

2.3.4 Total variation based inverse problems with square ℓ^2 data-fidelity

Let us now focus on the more general linear inverse problem with quadratic data-fidelity formulated in Section 2.1.2 using the Maximum A Posteriori methodology,

$$u_{\text{MAP}} \in \operatorname*{argmin}_{u\in\mathbb{R}^\Omega} \|Au - u_0\|_2^2 + \lambda \text{TV}^{\text{d}}(u)\,. \tag{2.30}$$

Recall that Ω and ω denote two bounded rectangular subsets of $\mathbb{Z}^2$, λ denotes a positive regularity parameter, $u_0 \in \mathbb{R}^\omega$ denotes the observed image, and $A : \mathbb{R}^\Omega \mapsto \mathbb{R}^\omega$ denotes the linear operator to be inverted (we will illustrate several classical imaging applications by selecting different choices for A). Notice that the existence of a minimizer of problem (2.30) is guaranteed (the objective function is convex and coercive) but we have not necessarily uniqueness, depending on the choice of A. From the practical viewpoint, the inverse problem (2.30) can be used to perform many image processing tasks, we will detail some of them, but let us first explain how the general problem can be handled using the Chambolle-Pock algorithm.

Proposition 19 (primal-dual saddle-point formulation of (2.30)). *Any solution* u_{MAP} *of problem (2.30) satisfies*

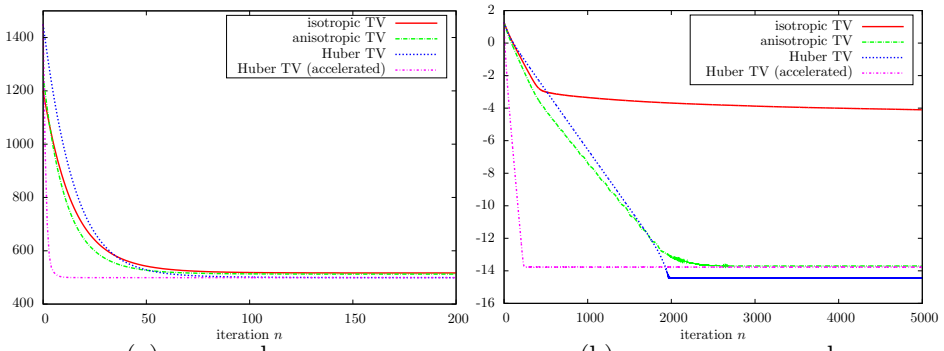
$$u_{\text{MAP}} \in \operatorname*{argmin}_{u\in\mathbb{R}^\Omega} \max_{(p,q)\in(\mathbb{R}^\Omega\times\mathbb{R}^\Omega)\times\mathbb{R}^\omega} G(u) + \langle Ku, (p,q)\rangle - F^\star(p,q)\,, \tag{2.31}$$

where $G : u \to 0$ *is the null function,* $F^\star : (p,q) \to \delta_{\|\cdot\|_{\infty,2}\leq 1}(p) + \|\frac{q}{2} + u_0\|_2^2$ *and* $K : \mathbb{R}^\Omega \to \big(\mathbb{R}^\Omega \times \mathbb{R}^\Omega\big) \times \mathbb{R}^\omega$ *is the linear operator defined by*

$$\forall u \in \mathbb{R}^\Omega, \quad Ku = \big(\lambda\nabla^{\text{d}} u, Au\big)\,.$$

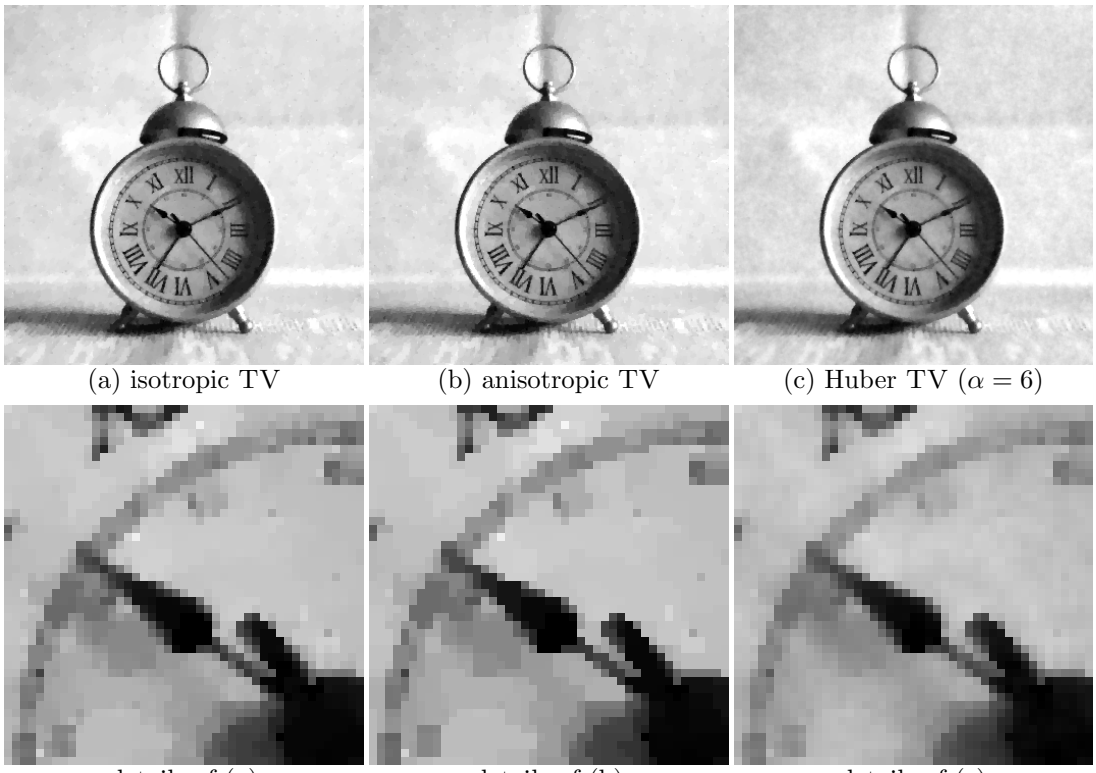
(a) reference (b) noisy

Figure 2.5: Reference image, and its noisy version. We display here a reference image (a), and its noisy version (b), that were used in the next numerical experiments. The dynamic of (a) is $[0, 255]$, and the noisy image (b) is undergoing additive white Gaussian noise with zero mean and standard deviation $\sigma = 20$. Some denoised versions of (b), computed using Algorithm 2, are displayed in Figure 2.7, while the numerical convergence is controlled in Figure 2.6.



(a) energy decrease (b) convergence speed

Figure 2.6: Numerical convergence achieved by Algorithm 2. The noisy image of Figure 2.5 was processed with the ROF model (2.28), using alternatively $J = \mathrm{TV}^{\mathrm{d}}$ (isotropic TV), $J = \mathrm{TV}^{\mathrm{d}}_1$ (anisotropic TV), or $J = \mathrm{HTV}^{\mathrm{d}}_\alpha$ (Huber-TV), as regularizer. The resulting images, as well as the precise setting of the model parameters (λ and α), are available in Figure 2.7. The numerical computation was done using Algorithm 2, using the setting $\tau = \sigma = 0.99/(\lambda\sqrt{8})$, or, in the case of the accelerated Huber TV, using the setting (iii) of Proposition 18 (see also Remark 12, where the strongly convexity constants are explicited). We display in (a) the evolution of the energy $E(u^n) := \|u^n - u_0\|_2^2 + \lambda J(u^n)$, computed at each iteration n of the algorithm. Although no theoretical result ensures the decrease of $E(u^n)$ with respect with n, this decrease is experimentally observed here. We display in (b), the evolution of $\log_{10}(\|u^n - u^\infty\|_2)$, along the iterations of the algorithm (u^∞ denotes the image obtained after 10^5 iterations). We see that a better convergence rate is reached using $\mathrm{TV}^{\mathrm{d}}_1$ and $\mathrm{HTV}^{\mathrm{d}}_\alpha$, in comparison with that reached using TV^{d}. Besides, in the case of the accelerated Huber variant, we observe a linear convergence rate, as predicted in Proposition 18.



(a) isotropic TV (b) anisotropic TV (c) Huber TV ($\alpha = 6$)

details of (a) details of (b) details of (c)

Figure 2.7: Image denoising using ROF model. The noisy image of Figure 2.5 was processed with the ROF model (2.28), using alternatively as a regularizer, the isotropic TV^{d}, or Huber ($\mathrm{HTV}^{\mathrm{d}}_{\alpha}$) variants. For each simulation, the regularity parameter was set in order to deliver images showing the same *method noise* (if we note $\bar{u}$ the restored image, the corresponding *method noise* is the quantity $\|\bar{u} - u_0\|_2^2$, which represents the *amounts of noise* removed from u_0), so that the images can be fairly compared. More precisely, we used respectively $\lambda = 42$, $\lambda = 26$ and $\lambda = 42$, for the TV^{d}, $\mathrm{TV}^{\mathrm{d}}_1$ and $\mathrm{HTV}^{\mathrm{d}}_{\alpha}$ models, the resulting images are respectively displayed in (a), (b), and (c). Some close-up views of the restored images are displayed in the last row, we see that both anisotropic and isotropic TV models yields images suffering of the staircasing effect (somehow more anisotropic in (b), where we observe many horizontal and vertical spurious edges), while this effect is removed when using the Huber variant, which however delivers an images with smoothed edges.

Proof. Writing $f(v) = \|v - u_0\|_2^2$, one easily gets the expression the Legendre-Fenchel transform $f^\star(q) = \|\frac{q}{2} + u_0\|_2^2 - \|u_0\|_2^2$, now since $f \in \Gamma_0(\mathbb{R}^\omega)$ we have

$$\|Au - u_0\|_2^2 = f(Au) = f^{\star\star}(Au) = \sup_{q \in \mathbb{R}^\omega} \langle Au, q\rangle - \|\tfrac{q}{2} + u_0\|_2^2 + \|u_0\|_2^2\,,$$

and the supremum is attained since the cost functional is concave (that is, its opposite is convex), differentiable, and its gradient vanishes at the point $q = 2(Au - u_0)$. Replacing accordingly this term in (2.30), and removing the constant term $\|u_0\|_2^2$ (which does not change the set of minimizers), last replacing as well the TV^{d} term by its dual formulation using Proposition 15, exactly yields (2.31). □

Remark 13 (anisotropic, or Huber variants of (2.30)). *Again, we can formulate the anisotropic or Huber variants of the inverse problem by replacing the* TV^{d} *term by* $\mathrm{TV}_1^{\mathrm{d}}$ *or* $\mathrm{HTV}_\alpha^{\mathrm{d}}$ *into* (2.30)*. Proposition 19 and its proof are straightforward to adapt, leading to slightly different primal-dual problems for each considered variant. More precisely by changing* $F^\star$ *into* $F^\star(p,q) = \delta_{\|\cdot\|_{\infty,\infty}\leq 1}(p) + \|\frac{q}{2} + u_0\|_2^2$ *we get the anisotropic variant of* (2.31)*, and by changing* $F^\star$ *into* $F^\star(p,q) = \delta_{\|\cdot\|_{\infty,2}\leq 1}(p) + \frac{\alpha\lambda}{2}\|p\|_2^2 + \|\frac{q}{2} + u_0\|_2^2$ *we get the Huber variant of* (2.31)*.*

Since the primal-dual problem (2.31) has exactly the form of the generic saddle-point problem (2.24) considered by Chambolle and Pock, it can be numerically solved using Algorithm 1. It is important to remark that the update of the dual variable (here the tuple $y = (p,q)$) in Algorithm 1 could be here split into two independent updates (one for p and one for q) thanks to the additive separability with respect to p and q of $(p,q) \to \langle Ku, (p,q)\rangle - F^\star(p,q)$. Finally all the updates have closed-form expressions, and the Chambolle-Pock algorithm applied to (2.30), as well as its anisotropic or Huber variants, boils down to Algorithm 3. This algorithm can be easily implemented as soon as a closed-form for A, its adjoint A^*, and an (as precise as possible) upper bound of $|||A|||$ are available.

Application to (non-blind) image deconvolution

In the case of image deconvolution, the linear operator A in (2.30) is the convolution with a point spread function (modeling for instance some blurring phenomena such as diffraction, defocus, or motion). Notice that a convolution can only model uniform phenomena, while in practice optical devices suffer from more complicated distortions, such as chromatic aberrations, stigmatism and coma, vignetting, etc. The correction of such non-uniform distortions is therefore out of the scope of the restoration application that we detail here.

Algorithm 3: resolvant algorithm for problem (2.30) and its variants.

Initialization: *Set L such as $|||K||| \leq L$ (for instance $L^2 \leq |||\lambda\nabla^{\mathrm{d}}|||^2 + |||A|||^2$), set $\theta = 1$, and choose $\tau, \sigma > 0$ such as $\sigma\tau L^2 < 1$ (for instance $\tau = \sigma = 0.99/L$). Choose $u^0 \in \mathbb{R}^\Omega$, $p^0 \in \mathbb{R}^\Omega \times \mathbb{R}^\Omega$, $q^0 \in \mathbb{R}^\omega$, and set $\overline{u}^0 = u^0$.*

Choice of the regularizer (TV^{d}, $\mathrm{TV}_1^{\mathrm{d}}$ or $\mathrm{HTV}_\alpha^{\mathrm{d}}$): *For solving the classical TV^{d} regularized inverse problem (2.30), set $a' = 2$ and $\nu = 1$. For solving the anisotropic $\mathrm{TV}_1^{\mathrm{d}}$ variant, set $a' = +\infty$ and $\nu = 1$. Otherwise, for solving the $\mathrm{HTV}_\alpha^{\mathrm{d}}$ variant, set $a' = 2$ and $\nu = 1 + \sigma\lambda\alpha$.*

Requirement(s): *Denote by $\pi_{\infty,a'}$ the ℓ^2 projection over the closed unit ball for the norm $\|\cdot\|_{\infty,a'}$. The explicit expression of $\pi_{\infty,a'}$ is given in Algorithm 2.*

Iterations: *For $k \geq 0$, update u^k, p^k, q^k and $\overline{u}^k$ as follows,*

$$\begin{aligned} p^{k+1} &= \pi_{\infty,a'}\left(\frac{p^k + \sigma\lambda\nabla^{\mathrm{d}}\overline{u}^k}{\nu}\right) \\ q^{k+1} &= \frac{2\,q^k + 2\,\sigma\left(A\overline{u}^k - u_0\right)}{2+\sigma} \\ u^{k+1} &= u^k + \tau\lambda\mathrm{div}^{\mathrm{d}}p^{k+1} - \tau A^* q^{k+1} \\ \overline{u}^{k+1} &= u^{k+1} + \theta\left(u^{k+1} - u^k\right) \end{aligned}$$

Let us consider a discrete convolution kernel $k_A \in \mathbb{R}^{\omega_A}$ with finite domain $\omega_A \subset \mathbb{Z}^2$. We define the associated operator $A : \mathbb{R}^\Omega \mapsto \mathbb{R}^\omega$ by

$$\forall (x,y) \in \omega, \quad Au(x,y) = \sum_{(a,b)\in\omega_A} k_A(a,b)\,u(x-a, y-b)\,, \tag{2.32}$$

where ω denotes the subset of Ω made of all the pixels $(x,y) \in \Omega$ such as $(x,y) - \omega_A \subset \Omega$ (in the following, we assume that ω is nonempty). Remark that it is also possible to define the convolution with kernel k_A as an operator $A : \mathbb{R}^\Omega \mapsto \mathbb{R}^\Omega$ at the cost of an extension of u outside of Ω, usually a *periodic* or *mirroring* condition, or a *zero-extension* of the image u is considered, which is of course nonrealistic in most situations.

Remark 14 (adjoint of A defined in (2.32)). *The adjoint of the operator A defined in (2.32) is the operator $A^* : \mathbb{R}^\omega \mapsto \mathbb{R}^\Omega$ defined by*

$$\forall v \in \mathbb{R}^\omega, \quad \forall (x,y) \in \Omega, \quad A^* v(x,y) = \sum_{(a,b)\in\omega_A} k_A(a,b)\, v(x+a, y+b)\,,$$

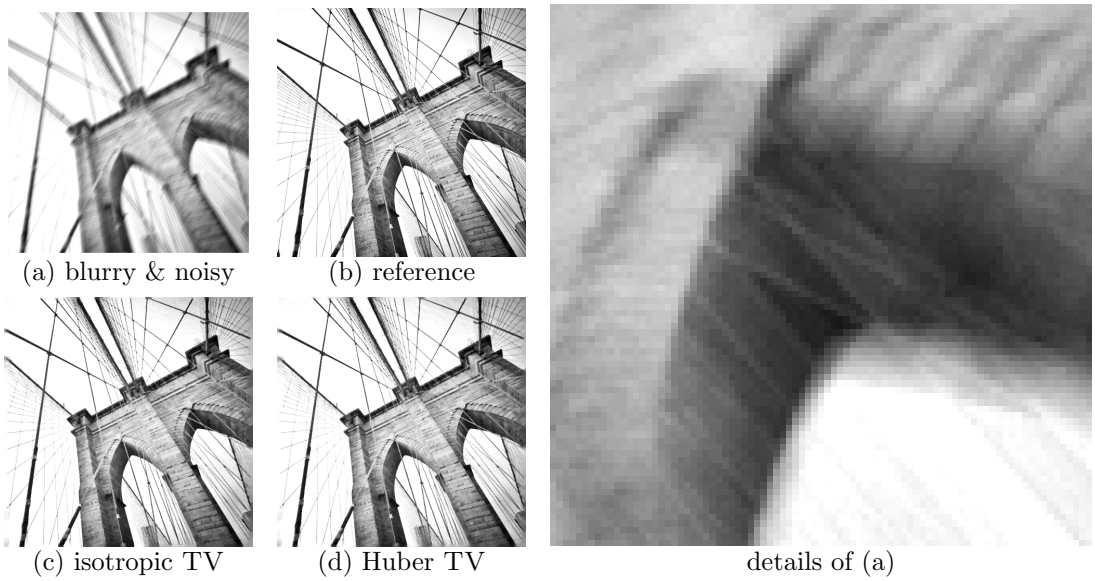
with the convention that $v(x+a, y+b) = 0$ when $(x+a, y+b) \notin \omega$.

Remark 15 (induced ℓ^2 norm of A defined in (2.32)). *Let us show that $|||A||| \leq ||k_A||_1$. Indeed for any image $u \in \mathbb{R}^\Omega$, using (twice) the dual formulation of the ℓ^2 norm over $\mathbb{R}^\omega$, we get*

$$\begin{aligned}
\|Au\|_2 &= \max_{v\in\mathbb{R}^\omega,\ \|v\|_2\leq 1} \langle v, Au\rangle \\
&= \max_{v\in\mathbb{R}^\omega,\ \|v\|_2\leq 1} \sum_{(x,y)\in\omega} \sum_{(a,b)\in\omega_A} v(x,y)\, k_A(a,b)\, u(x-a, y-b) \\
&\leq \sum_{(a,b)\in\omega_A} \max_{v\in\mathbb{R}^\omega,\ \|v\|_2\leq 1} \sum_{(x,y)\in\omega} v(x,y)\, k_A(a,b)\, u(x-a, y-b) \\
&= \sum_{(a,b)\in\omega_A} \max_{v\in\mathbb{R}^\omega,\ \|v\|_2\leq 1} \langle v, k_A(a,b)\, u(\cdot - a, \cdot - b)\rangle \\
&= \sum_{(a,b)\in\omega_A} |k_A(a,b)| \cdot \|u(\cdot - a, \cdot - b)\|_2\,,
\end{aligned}$$

with the obvious notation $u(\cdot - a, \cdot - b) = ((x,y) \in \omega \mapsto u(x-a, y-b))$. As for any $(a,b) \in \omega_A$ we have $\|u(\cdot - a, \cdot - b)\|_2 \leq \|u\|_2$ (in this inequality, the left-hand norm is the ℓ^2 norm over $\mathbb{R}^\omega$ and the right-hand one is the ℓ^2 norm over $\mathbb{R}^\Omega$), we get the upper bound $\|Au\|_2 \leq \|k_A\|_1 \cdot \|u\|_2$ and thus $|||A||| \leq \|k_A\|_1$.

Thanks to Remarks 14 and 15, we now have all the information necessary to use Algorithm 3 and approximate some solutions of the inverse problem (2.30). Some results are displayed in Figures 2.8 and 2.9 (motion and out-of-focus deblurring).

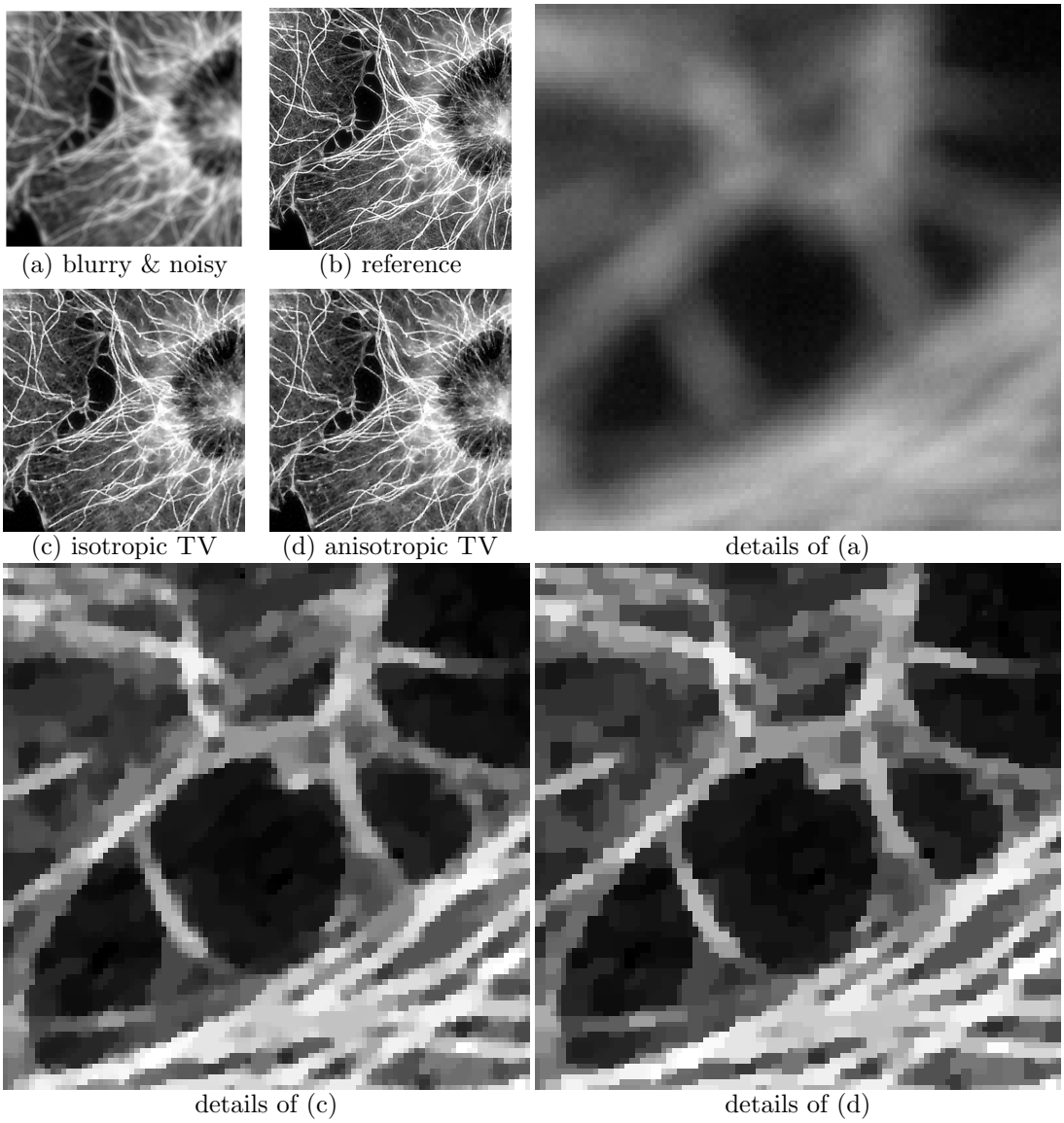


(a) blurry & noisy (b) reference

(c) isotropic TV (d) Huber TV details of (a)

details of (c) details of (d)

Figure 2.8: Motion deblurring using isotropic or Huber discrete total variation. A degraded (blurry and noisy) image (a) is computed by convolving the reference image (b) with a motion blur kernel k_A before adding a white Gaussian noise, with zero-mean and standard deviation equal to 2. The degraded image (a) is then processed by solving the corresponding TV^{d} and $\mathrm{HTV}^{\mathrm{d}}_{\alpha}$ ($\alpha = 10$) regularized inverse problems (2.30). Again the setting of λ was done at fixed method noise (here the quantity $\|A\bar{u} - u_0\|_2^2$, for each restored image $\bar{u}$), yielding $\lambda = 0.3$ for the TV^{d} model, and $\lambda = 0.345$ for the $\mathrm{HTV}^{\mathrm{d}}_{\alpha}$ one. The resulting images, computed using Algorithm 3 (setting $\tau = \sigma = 0.99/L$, and $L^2 = 8\lambda^2 + \|k_A\|_1^2$), are displayed in (c) and (d). Looking at the details of (c) and (d), we check again that the TV^{d} model yields an image with sharp edges, but suffering from staircasing, while this artifact is removed when using the Huber variant which delivers an image that looks more natural, although a bit less sharp.



(a) blurry & noisy (b) reference details of (a)

(c) isotropic TV (d) anisotropic TV

details of (c) details of (d)

Figure 2.9: Out-of-focus deblurring using isotropic or anisotropic discrete total variation. We performed a similar experiment as in Figure 2.8, over a fluorescence microscopy biological image of actin filaments and microtubules in interphase cells (source http://cellimagelibrary.org/images/240, first channel), using as a blur kernel, the indicator of a disk with radius 7 pixels (which models an out-of-focus phenomenon). The blurry and noisy ($\sigma = 2$) image (a) was processed with model (2.30), using the isotropic (TV^{d}) or anisotropic ($\mathrm{TV}_1^{\mathrm{d}}$) discrete total variation as regularizer. We have respectively set $\lambda = 0.2$, or $\lambda = 0.098$, so that the restored images, displayed in (c) and (d), show the same method noise. Both models deliver images showing improved sharpness, in comparison to the initial blurry image (a). However, we observe into these two images, the presence of staircasing effect, that is the presence of constant regions delimited by spurious edges. In particular, image (d) suffers from a strong *blocky* effect (the constant regions are rectangular), which must obviously be related to the use of the ℓ^1 norm in definition 2.10.

Application to image zooming

The inverse problem (2.30) can also be used to perform image zooming (see for instance [Malgouyres and Guichard 2001]). For such application, the operator A is often assumed to be a blurring kernel followed by a subsampling procedure. Many definitions of A are possible, we will here consider $A : \mathbb{R}^\Omega \to \mathbb{R}^\omega$, where $\omega = [0, M) \times [0, N) \cap \mathbb{Z}^2$ denotes a small (more precisely subsampled) discrete domain, and $\Omega = [0, \delta M) \times [0, \delta N) \cap \mathbb{Z}^2$ a bigger (or resampled) one, δ being a positive integer (which represents the zoom factor). We define A by,

$$\forall u \in \mathbb{R}^\Omega,\ \forall (x, y) \in \omega,\ Au(x, y) = \frac{1}{\delta^2} \sum_{(a,b)\in[0,\delta)^2\cap\mathbb{Z}^2} u(\delta x + a, \delta y + b)\ , \tag{2.33}$$

so that Au is exactly the zero order unzoom with factor δ of the image u. We can easily show that the corresponding adjoint $A^* : \mathbb{R}^\omega \to \mathbb{R}^\Omega$, is given by

$$\forall v \in \mathbb{R}^\omega, \quad \forall (x, y) \in \Omega,\ A^*v(x, y) = \frac{1}{\delta^2}\, v\left(\lfloor \tfrac{x}{\delta} \rfloor, \lfloor \tfrac{y}{\delta} \rfloor\right)\ , \tag{2.34}$$

and the induced ℓ^2 norm of A is $|||A||| = \frac{1}{\delta}$.

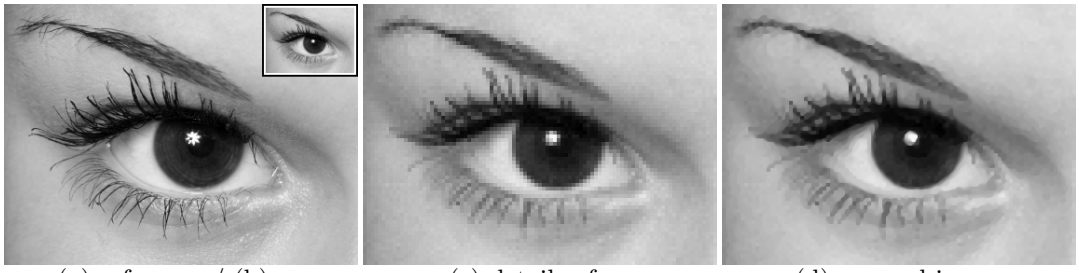
Remark 16. *The operator A defined in (2.33) can also be seen as a discrete approximation of a captor integration procedure, which models the photon count averaging process that is done over the (in reality continuous) square domain $[0, \delta] \times [0, \delta]$ of each captor covering the focal plane of the image acquisition system (this is for instance the case for usual cameras equipped with CCD or CMOS sensors).*

2.3.5 Minimizing the total variation under affine constraints

Let us now consider some constrained minimization problems of the kind

$$u_{\text{constr}} \in \operatorname*{argmin}_{u\in\mathbb{R}^\Omega} \mathrm{TV}^d(u) \quad \text{subject to} \quad Au = u_0\ , \tag{2.35}$$

where u_0 denotes the observed image with discrete domain ω, u_{constr} denotes the reconstructed image with discrete domain Ω, and A denotes again a linear operator from $\mathbb{R}^\Omega$ to $\mathbb{R}^\omega$. In other words, we are interested in the computation of an image u_{constr} having the smallest discrete total variation among those satisfying the constraint $Au = u_0$. Remark that the inverse problem (2.30) is none other



(a) reference / (b) u_0 (c) details of u_0 (d) zoomed image

Figure 2.10: Image zooming using the isotropic discrete total variation. The 380×280 sized reference image (a) was downsampled by a factor $\delta = 4$ (using the operator A given in (2.33)) and corrupted with an additive white Gaussian noise with zero mean and standard deviation $\sigma = 2$, yielding image u_0 displayed in (b). We display in (c) the image obtained by rezooming the image u_0 to its initial resolution, using a zero-order (nearest neighbour) zoom with factor δ, and we display in (d) the image delivered by model (2.30), setting $\lambda = 0.2$. The regularization operated by the TV model yields a more natural image than (c), with attenuated blocky effect. However, the image (d) delivered by this model is far from being perfect, it is in particular very poorly textured in comparison to (a), and as usual, suffers from staircasing.

than a relaxed version of (2.35). Of course, in presence of noise the constraint must be relaxed and one should consider model (2.30) instead of (2.35), however the constrained model may be interesting when the level of noise in u_0 is low, especially because it does not need the setting of any regularity parameter λ.

Using one more time the Proposition 15, we get a primal-dual reformulation of (2.35),

$$u_{\text{constr}} \in \operatorname*{argmin}_{u\in\mathbb{R}^\Omega} \max_{p\in\mathbb{R}^\Omega\times\mathbb{R}^\Omega} \delta_{A^{-1}u_0}(u) + \langle \nabla^{\mathrm{d}} u, p\rangle - \delta_{\|\cdot\|_{\infty,2}\leq 1}(p)\ , \tag{2.36}$$

where $\delta_{A^{-1}u_0}$ denotes the indicator function of the inverse image of u_0 by the operator A, that is the (closed and convex) set $A^{-1}u_0 := \left\{u \in \mathbb{R}^\Omega,\ Au = u_0\right\}$ that we assume to be nonempty. A solution of problem (2.36) can be numerically computed using the Chambolle-Pock algorithm, taking $G = \delta_{A^{-1}u_0}$, $F^\star = \delta_{\|\cdot\|_{\infty,2}\leq 1}$ and $K = \nabla^{\mathrm{d}}$ (with adjoint $K^* = -\mathrm{div}^{\mathrm{d}}$ and induced ℓ^2 norm satisfying $|||K||| \leq \sqrt{8}$), it boils down to Algorithm 4.

Remark 17. *The primal update in Algorithm 4 involves the computation of a projection onto the set $A^{-1}u_0$, which may be nontrivial according to the choice of A. When necessary, one can avoid this computation, at the cost of introducing another dual variable $q \in \mathbb{R}^\omega$ related to the constraint $Au = u_0$, using the relation $\delta_{A^{-1}u0}(u) = \delta_{\{u_0\}}(Au) = \delta^{\star\star}_{\{u_0\}}(Au)$, where $\delta_{\{u_0\}}$ denotes the indicator function of the singleton $\{u_0\}$.*

Algorithm 4: resolvant algorithm for problem (2.35) and its variants.

Initialization: *Set L such as $|||K||| \leq L$ (for instance $L = \sqrt{8}$), set $\theta = 1$, and choose $\tau, \sigma > 0$ such as $\sigma\tau L^2 < 1$ (for instance $\tau = \sigma = 0.99/L$). Choose $u^0 \in \mathbb{R}^\Omega$, $p^0 \in \mathbb{R}^\Omega \times \mathbb{R}^\Omega$, and set $\overline{u}^0 = u^0$.*

Choice of the regularizer (TV^{d}, $\mathrm{TV}_1^{\mathrm{d}}$ or $\mathrm{HTV}_\alpha^{\mathrm{d}}$): *For using the TV^{d} regularizer, as formulated in (2.35), set $a' = 2$ and $\nu = 1$. For solving the anisotropic $\mathrm{TV}_1^{\mathrm{d}}$ variant, set $a' = +\infty$ and $\nu = 1$. Otherwise, for solving the $\mathrm{HTV}_\alpha^{\mathrm{d}}$ variant, set $a' = 2$ and $\nu = 1 + \sigma\lambda\alpha$.*

Requirement(s): *Denote by π_0 the ℓ^2 projection over the set $A^{-1}u_0 = \{u \in \mathbb{R}^\Omega,\ Au = u_0\}$. Denote by $\pi_{\infty,a'}$ the ℓ^2 projection over the closed unit ball for the norm $\|\cdot\|_{\infty,a'}$ (see the explicit expression of $\pi_{\infty,a'}$ in Algorithm 2).*

Iterations: *For $k \geq 0$, update u^k, p^k and $\overline{u}^k$ as follows,*

$$\begin{aligned} p^{k+1} &= \pi_{\infty,a'}\left(\frac{p^k + \sigma\nabla^{\mathrm{d}}\overline{u}^k}{\nu}\right) \\ u^{k+1} &= \pi_0\left(u^k + \tau\mathrm{div}^{\mathrm{d}}p^{k+1}\right) \\ \overline{u}^{k+1} &= u^{k+1} + \theta\left(u^{k+1} - u^k\right) \end{aligned}$$

Application to image zooming

Let us consider again the zero order unzoom (or captor integration) operator $A : \mathbb{R}^\Omega \to \mathbb{R}^\omega$ with integer factor $\delta \geq 1$, defined in (2.33). One can show that in that case, the ℓ^2 projection π_0 over $A^{-1}u_0$ is given by

$$\forall v \in \mathbb{R}^\Omega, \quad \pi_0(v) = v - \delta^2 A^* A v + \delta^2 A^* u_0\,, \tag{2.37}$$

where A^* denotes the adjoint of A, given in (2.34). The result (2.37) can be proven by a direct computation of

$$\pi_0(v) = \underset{u \in \mathbb{R}^\Omega}{\mathrm{argmin}}\ \tfrac{1}{2}\|u - v\|_2^2 \quad \text{subject to} \quad Au = u_0\,, \tag{2.38}$$

or, more elegantly, using some advanced duality results that we will present later, in Section 2.4. Alternatively, we can use the celebrated Lagrangian formalism

and check that $\pi_0(v)$ is the solution of the constrained convex and differentiable problem (2.38), since the tuple $(\bar{u} := \pi_0(v), \bar{p} := \delta^2 Av - \delta^2 u_0)$ satisfies the Karush-Kuhn-Tucker conditions associated to (2.38), which amounts to check that $A\bar{u} = u_0$ and $\nabla_u \mathcal{L}(\bar{u}, \bar{p}) = 0$, noting $\mathcal{L} = (u, p) \mapsto \frac{1}{2}\|Au - v\|_2^2 + \langle Au - u_0, p\rangle$, and $\nabla_u \mathcal{L}(\bar{u}, \bar{p})$ the gradient of $u \mapsto \mathcal{L}(u, \bar{p})$ at the point $\bar{u}$ (see [Boyd and Vandenberghe 2004], and references therein). However, we must emphasize that this value of $\bar{p}$ was here derived using the dual methodology of Section 2.4, we will therefore come back later on this point.

Thanks to (2.37), we are now able to use Algorithm 4 to compute a solution of (2.35), in the case of image zooming. This model does not involve the setting of a regularity parameter λ and delivers images that are very similar to that obtained using the relaxed problem (2.30) with a small λ.

Image reconstruction from partial measurements

The constrained problem (2.35) can also be used to perform other image processing tasks, we briefly address here the problem of image reconstruction from partial measurements. Consider that the gray levels of a reference image $u_{\text{ref}} \in \mathbb{R}^\Omega$ are only observed on a subdomain $\omega_0 \subset \Omega$, and we try to recover the missing measurements. This problem is now usually called *image inpainting*, but was in fact introduced by Masnou and Morel [1998] as *image disocclusion* problem. There exists many approaches to image inpainting (we refer to the nice work [Newson et al. 2014], and references therein, for recent advances in this domain), a very simple one consists in looking for the image showing the smallest total variation among those which coincide with the observed measurements, that is over ω_0 (see for instance [Chan et al. 2005], which uses a similar approach to perform at the same time inpainting and deconvolution). This can be modeled as a problem of kind (2.35), when seeing $A : \mathbb{R}^\Omega \to \mathbb{R}^\Omega$ as a masking operator,

$$\forall u \in \mathbb{R}^\Omega, \quad \forall (x,y) \in \Omega, \quad Au(x,y) = \begin{cases} u(x,y) & \text{if } (x,y) \in \omega_0 \\ 0 & \text{otherwise,} \end{cases}$$

and $u_0 = Au_{\text{ref}}$, the (incomplete) observation u_{ref}. In that case, the ℓ^2 projection π_0 on the constraint set $A^{-1}u_0$ is simply given by

$$\forall u \in \mathbb{R}^\Omega, \quad \forall (x,y) \in \Omega, \quad \pi_0(u)(x,y) = \begin{cases} u_0(x,y) & \text{if } (x,y) \in \omega_0 \\ u(x,y) & \text{otherwise.} \end{cases}$$

Therefore, Algorithm 4 can be implemented with no difficulty, and used to approximate a solution of (2.35), as done in Figure 2.11.

(a) reference (b) observed image (c) restored image

Figure 2.11: Image inpainting. Some pixels of the reference image (a) were masked to model an observation (b) with incomplete measurements (masked pixels are represented in black). The domain of the unmasked pixels, noted ω_0, represents 40% of the full domain Ω of the reference image. The image (b) is then processed using model (2.35), in order to extrapolate the gray levels from ω_0 to the full domain Ω, yielding image (c). This total variation based variational approach is rather simple, and far better results may be obtained using modern (in particular patch-based) approaches. In particular, in the example considered here, the position of the masked pixels were randomly chosen (the pixels of the binary mask were generated according to a Bernoulli law with parameter $p = 40\%$). Actually the problem becomes far more complicated in more realistic situation, where the masking operator contains more structure.

The problem of image reconstruction from partial measurements can also be considered when the available measurements are done in the frequency domain, which typically happens in the case of MRI (Magnetic Resonance Imaging) or tomography reconstruction. This example, termed as spectrum extrapolation, can also be formulated as a problem of the type (2.35), it will be presented in more details in Chapter 3.

2.4 The dual point of view

We will end-up this chapter with the presentation of a general framework devoted to the dualization of optimization problems. In contrast to the methodology used in Section 2.3, which consisted in changing the initial (primal) minimization problem

$$(\mathscr{P}) \quad \inf_{x \in X} C(x)\,, \tag{2.39}$$

into a primal-dual saddle-point problem, we will now entirely leave the primal space (here noted X), and show how we can associate to $(\mathscr{P})$ a dual problem $(\mathscr{P}^\star)$, defined as a supremum over a dual space (that will be noted Y). We will

explain the relationships linking the infimum $(\mathscr{P})$ with the supremum $(\mathscr{P}^\star)$, as well as those linking the solutions of $(\mathscr{P})$ to that of $(\mathscr{P}^\star)$. We will then apply this methodology to several image processing problems. This presentation will be restricted to the case where X and Y are finite dimensional Hilbert spaces (in particular X and Y will identify to $X^\star$ and $Y^\star$, which greatly simplifies the studies and the notations), we refer to [Ekeland and Témam 1999, Chap. III] for more general results.

2.4.1 Generic dual of an optimization problem

Let X and Y denote two finite dimensional Hilbert spaces, endowed with the inner products $\langle\cdot,\cdot\rangle_X$ and $\langle\cdot,\cdot\rangle_Y$. Let us associate to the primal problem $(\mathscr{P})$ with cost C, defined in (2.39), what we call a *perturbation function* $\Phi : X \times Y \to \overline{\mathbb{R}}$, which is a function satisfying

$$\forall x \in X, \quad \Phi(x,0) = C(x)\,. \tag{2.40}$$

For the moment no more assumption is done for Φ. The dual problem of $(\mathscr{P})$ with respect to the perturbation Φ, is defined by

$$(\mathscr{P}^\star) \quad \sup_{y\in Y} -\Phi^\star(0,y)\,, \tag{2.41}$$

where $\Phi^\star$ denotes the Legendre-Fenchel transform of Φ (notice again that the interpretation of $\Phi^\star$ as a function of $X \times Y$ instead of $X^\star \times Y^\star$ was possible thanks to the identification of $X^\star \times Y^\star$ to $X \times Y$). Many relationships between $(\mathscr{P})$ and $(\mathscr{P}^\star)$ are derived in [Ekeland and Témam 1999, Chap. III], we will only state here the main ones.

Proposition 20 (extremality relation). *Under the following assumptions,*

(i) $\Phi \in \Gamma_0(X \times Y)$*;*

(ii) the primal problem $(\mathscr{P})$ *admits at least one solution* $\bar{x}$*;*

(iii) there exists a point $x_0 \in X$*, such as* $y \mapsto \Phi(x_0, y)$ *is finite and continuous at the point* $y = 0$*;*

the dual problem $(\mathscr{P}^\star)$ *admits at least one solution* $\bar{y} \in Y$*. Besides, the solutions of* $(\mathscr{P})$ *and* $(\mathscr{P}^\star)$ *are characterized by the relation*

$$\Phi(\bar{x},0) + \Phi^\star(0,\bar{y}) = 0 \Leftrightarrow \left\{ \begin{array}{l} \bar{x} \text{ is a solution of } (\mathscr{P}) \\ \bar{y} \text{ is a solution of } (\mathscr{P}^\star), \end{array} \right.$$

and this relation is named extremality relation. *In particular,* $\Phi(\bar{x},0) = -\Phi^\star(0,\bar{y})$ *so that the value of the infimum of* $(\mathscr{P})$ *equals that of the supremum* $(\mathscr{P}^\star)$.

Remark 18 (bidual problem). *The dual problem* $(\mathscr{P}^\star)$ *can be viewed as an infimum, (indeed,* $\sup_{y\in Y} -\Phi^\star(0,y)$ *and* $-\inf_{y\in Y}\Phi^\star(0,y)$ *are the same problems), and we can easily reiterate the dualization process. Considering the dual of* $(\mathscr{P}^\star)$ *with respect to the perturbation* $\widetilde{\Phi} = (x,y) \mapsto \Phi^\star(x,y)$*, yields the bidual problem* $(\mathscr{P}^{\star\star})$*,*

$$-\sup_{x\in X} -\widetilde{\Phi}^\star(x,0) \quad \textit{or equivalently,} \quad \inf_{x\in X} \Phi^{\star\star}(x,0)\,,$$

which is equivalent to the primal problem $(\mathscr{P})$ *when* $\Phi \in \Gamma_0(X\times Y)$*, since in that case* $\Phi^{\star\star} = \Phi$*. We intuitively understand here the importance of hypothesis (i) in Proposition 20, which maintains a kind of equivalence between the problems* $(\mathscr{P})$ *and* $(\mathscr{P}^\star)$.

In all the following, we assume satisfied the hypotheses of Proposition 20, in particular the infimum $(\mathscr{P})$ and the supremum $(\mathscr{P}^\star)$ are attained, and will be replaced by the argmin and argmax operators, since we focus now on the solutions of these problems.

2.4.2 Interesting particular cases

First, consider the case where the cost function C of problem $(\mathscr{P})$ is of type

$$\forall x \in X, \quad C(x) = J(x, Kx)$$

where $J \in \Gamma_0(X\times Y)$, and $K : X \to Y$ denotes a linear operator. Let us associate to $(\mathscr{P})$ the perturbation function $\Phi = (x,y) \mapsto J(x, Kx-y)$ (which satisfies (2.40), i.e., $\Phi(x,0) = C(x)$). This choice yields, for any $(\tilde{x},\tilde{y}) \in X\times Y$,

$$\begin{aligned}\Phi^\star(\tilde{x},\tilde{y}) &= \sup_{(x,y)\in X\times Y} \langle(\tilde{x},\tilde{y}),(x,y)\rangle_{X\times Y} - J(x,Kx-y)\\ &= \sup_{(x,y)\in X\times Y} \langle(\tilde{x}+K^*\tilde{y},-\tilde{y}),(x,y)\rangle_{X\times Y} - J(x,y)\,,\end{aligned}$$

using the change of variable $y \mapsto Kx - y$ (and noting, for any a, c in X, and any b, d in Y, $\langle(a,b),(c,d)\rangle_{X\times Y} = \langle a,c\rangle_X + \langle b,d\rangle_Y$). Finally, we recognize $\Phi^\star(\tilde{x},\tilde{y}) = J^\star(\tilde{x}+K^*\tilde{y},-\tilde{y})$, so that the dual of $(\mathscr{P})$ with respect to Φ, is given by

$$(\mathscr{P}^\star) \quad \operatorname*{argmax}_{y\in Y} -J^\star(K^*y,-y)\,. \tag{2.42}$$

Besides, using the extremality relation of Proposition 20, we see that $\bar{x}$ and $\bar{y}$ are respectively solutions of $(\mathscr{P})$ and $(\mathscr{P}^\star)$ if and only if

$$J(\bar{x}, K\bar{x}) + J^\star(K^*\bar{y}, -\bar{y}) = 0\,,$$

and since we have $0 = \langle \bar{x}, K^*\bar{y}\rangle_X - \langle K\bar{x}, \bar{y}\rangle_Y = \langle (\bar{x}, K\bar{x}), (K^*\bar{y}, -\bar{y})\rangle_{X\times Y}$, we get $J(\bar{x}, K\bar{x}) + J^\star(K^*\bar{y}, -\bar{y}) = \langle (\bar{x}, K\bar{x}), (K^*\bar{y}, -\bar{y})\rangle_{X\times Y}$. Therefore, using Proposition 4, we get the characterization

$$(K^*\bar{y}, -\bar{y}) \in \partial J(\bar{x}, K\bar{x}) \Leftrightarrow \left\{ \begin{array}{l} \bar{x} \text{ is a solution of } (\mathscr{P}) \\ \bar{y} \text{ is a solution of } (\mathscr{P}^\star). \end{array} \right. \tag{2.43}$$

Now, let us come back to the primal version (2.25) of the generic primal-dual problem (2.24) addressed by Chambolle and Pock. In that case, the cost function to minimize is given by

$$\forall x \in X, \quad C(x) = J(x, Kx) := G(x) + F(Kx)\,, \tag{2.44}$$

so that the function J considered before is now additively separable with respect to (x, Kx). Thanks to this additional property, we can easily show that $J^\star = (\tilde{x}, \tilde{y}) \mapsto G^\star(\tilde{x}) + F^\star(\tilde{y})$, and the dual problem $(\mathscr{P}^\star)$ derived in (2.42) becomes

$$(\mathscr{P}^\star) \quad \operatorname*{argmax}_{y\in Y} -G^\star(K^*y) - F^\star(-y)\,. \tag{2.45}$$

The extremality relation (2.43) also benefits from the separability of J, however, rather than adapting (2.43) (which involves the knowledge of additional rules about subdifferential calculus), it is simpler to reuse Proposition 20. The points $\bar{x}$ and $\bar{y}$ are respectively solutions of $(\mathscr{P})$ and $(\mathscr{P}^\star)$ if and only if $\Phi(\bar{x}, 0) + \Phi^\star(0, \bar{y}) = 0$, that is, if and only if

$$G(\bar{x}) + F(K\bar{x}) + G^\star(K^*\bar{y}) + F^\star(-\bar{y}) = 0\,,$$

which we can transform into

$$\Big[G(\bar{x}) + G^\star(K^*\bar{y}) - \langle \bar{x}, K^*\bar{y}\rangle_X\Big] + \Big[F(K\bar{x}) + F^\star(-\bar{y}) - \langle K\bar{x}, -\bar{y}\rangle_Y\Big] = 0\,.$$

Remark now that the two terms between brackets are both nonnegative (this can be easily checked by replacing the Legendre-Fenchel transforms by suprema using

Definition 12), it follows that both are null. Thus, thanks to Proposition 4, we get $K^*\bar{y} \in \partial G(\bar{x})$ and $-\bar{y} \in \partial F(K\bar{x})$, and finally, the extremality relation becomes

$$\left\{ \begin{array}{ll} K^*\bar{y} & \in \partial G(\bar{x}) \\ -\bar{y} & \in \partial F(K\bar{x}) \end{array} \right. \Leftrightarrow \left\{ \begin{array}{l} \bar{x} \text{ is a solution of } (\mathscr{P}) \\ \bar{y} \text{ is a solution of } (\mathscr{P}^\star). \end{array} \right. \tag{2.46}$$

In several cases, the dual problem $(\mathscr{P}^\star)$ is easier to solve than $(\mathscr{P})$, and sometimes (unfortunately not always) the extremality relation (2.46) can be used to retrieve a solution $\bar{x}$ of the primal problem $(\mathscr{P})$ from a solution $\bar{y}$ of the dual problem $(\mathscr{P}^\star)$. In the next section, we will show how this methodology can be used to recover Moreau's identity, which is a perfect example where the extremality condition makes possible the explicit computation of a solution of the primal problem $(\mathscr{P})$ given a solution of the dual problem $(\mathscr{P}^\star)$. We will then study some previously considered problems in the light of this dual methodology, such as the ROF denoising problem (2.28), and the inverse problem (2.30). We will also point out some limits of this approach, since in the case of the inverse problem (2.30), we will see that the link between primal and dual problems becomes implicit. But first of all, let us come back on the projection problem (2.37) considered in Section 2.3.5.

2.4.3 Back to several optimization problems

Dual of the projection problem (2.37)

In section 2.3.5, we needed to compute the ℓ^2 projection $\pi_0(v)$ of an element $v \in \mathbb{R}^\Omega$ over the constraint set $A^{-1}u_0 = \{u \in \mathbb{R}^\Omega,\ Au = u_0\}$, where $A : \mathbb{R}^\Omega \to \mathbb{R}^\omega$ denoted the zero order unzoom operator with integer factor δ, that we defined in (2.33). Computing $\pi_0(v)$ amounts to compute a solution $\bar{u}$ of the primal problem

$$(\mathscr{P}) \quad \operatorname*{argmin}_{u \in \mathbb{R}^\Omega} G(u) + F(Au)\,,$$

noting $G(u) = \frac{1}{2}\|u - v\|_2^2$, and $F = \delta_{\{u_0\}}$, the indicator function of the singleton $\{u_0\}$. This problem has the generic form (2.44), when setting $X = \mathbb{R}^\Omega$, and $Y = \mathbb{R}^\omega$. Since for any $p \in \mathbb{R}^\omega$ we have $G^\star(A^*p) = \frac{1}{2}\|A^*p + v\|_2^2 - \frac{1}{2}\|v\|_2^2$ and $\delta^\star_{\{u_0\}}(-p) = -\langle p, u_0\rangle$, we associate to $(\mathscr{P})$ a dual problem, which, after basic manipulations (change the argmax into argmin, remove constant terms, ...), yields

$$(\mathscr{P}^\star) \quad \operatorname*{argmin}_{p \in \mathbb{R}^\omega} \tfrac{1}{2}\|A^*p + v\|_2^2 - \langle p, u_0\rangle\,.$$

Wee see that $(\mathscr{P}^\star)$ is a convex and differentiable problem, the gradient of its cost function is $p \mapsto A(A^*p + v) - u_0$, and thanks to the relation $AA^*p = \frac{p}{\delta^2}$ (which is easy to prove using (2.33) and (2.34)), we see that this gradient vanishes at the point $\bar{p} = \delta^2 u_0 - \delta^2 Av$, which is the solution of $(\mathscr{P}^\star)$. Therefore, using the extremality relation (2.46), we get $A^*\bar{p} \in \partial G(\bar{u})$, that is, $A^*\bar{p} = \bar{u} - v$ (since $\partial G(\bar{u}) = \{\nabla G\bar{u}\} = \{\bar{u} - u_0\}$), and thus, $\bar{u} = v - \delta^2 A^*Av + \delta^2 A^*u_0$, as announced in Section 2.3.5.

Recovering Moreau's identity

Let us set $X = Y$, $K = I$ (the identity operator over the space X), and $G = x \mapsto \frac{1}{2\sigma}\|x - x_0\|_2^2$ (for a given parameter $\sigma > 0$, and a given $x_0 \in X$), so that the primal problem $(\mathscr{P})$ boils down to the computation of $\bar{x} = (I + \sigma\partial F)^{-1}(x_0)$. We can easily show that $G^\star = y \mapsto \frac{\sigma}{2}\|y + \frac{x_0}{\sigma}\|_2^2 - \frac{1}{2\sigma}\|x_0\|_2^2$, so that (2.45) yields the dual problem

$$(\mathscr{P}^\star) \quad \operatorname*{argmax}_{y \in Y} -\tfrac{\sigma}{2}\|y + \tfrac{x_0}{\sigma}\|_2^2 + \tfrac{1}{2\sigma}\|x_0\|_2^2 - F^\star(-y)\,.$$

By removing the constant term $\frac{1}{2\sigma}\|x_0\|_2^2$ (which does not change the set of maximizers of $(\mathscr{P}^\star)$), changing the argmax into the argmin of the opposite cost function, and changing y into $-y$ (beware this also changes the sign of the argmin), we get

$$(\mathscr{P}^\star) \quad -\operatorname*{argmin}_{y \in Y} \tfrac{\sigma}{2}\|y - \tfrac{x_0}{\sigma}\|_2^2 + F^\star(y)\,.$$

which exactly boils down to the computation of $\bar{y} = -\big(I + \frac{1}{\sigma}\partial F^\star\big)^{-1}\big(\frac{x_0}{\sigma}\big)$. Besides, the extremality relation (2.46) states that the solutions $\bar{x}$ and $\bar{y}$ of $(\mathscr{P})$ and $(\mathscr{P}^\star)$ are necessarily linked by the relation $\bar{y} \in \partial G(K\bar{x})$. Since the only subgradient of G at the point $K\bar{x} = \bar{x}$ (recall that $K = I$) is its gradient $\nabla G(\bar{x}) = \frac{\bar{x} - x_0}{\sigma}$, we get $\bar{y} = \frac{\bar{x} - x_0}{\sigma}$, yielding $x_0 = \bar{x} - \sigma\,\bar{y}$. Finally, we recover Moreau's identity,

$$x_0 = (I + \sigma\partial F)^{-1}(x_0) + \sigma\big(I + \tfrac{1}{\sigma}\partial F^\star\big)^{-1}\big(\tfrac{x_0}{\sigma}\big)\,,$$

that was derived in Proposition 13.

A dual formulation of the ROF problem

The ROF image denoising problem (2.28) corresponds to the choice of the cost function

$$C = u \mapsto J(u, Ku) := G(u) + F(Ku)\,,$$

where $G(u) = \|u - u_0\|_2^2$, $Ku = \lambda \nabla^{\mathrm{d}} u$ (with adjoint $K^* = -\lambda \mathrm{div}^{\mathrm{d}}$), and $F(Ku) = \|\lambda \nabla^{\mathrm{d}} u\|_{1,2} = \lambda\, \mathrm{TV}^{\mathrm{d}}(u)$ (notice we changed the variables (x, y) into (u, p), and we set $X = \mathbb{R}^\Omega$, $Y = \mathbb{R}^\Omega \times \mathbb{R}^\Omega$). The Legendre-Fenchel transforms of F and G are given by $F^\star = \delta_{\|\cdot\|_{\infty,2} \leq 1}$ (Proposition 2) and $G^\star = v \mapsto \frac{1}{4}\|v + 2\, u_0\|_2^2 - \|u_0\|_2^2$. Therefore, using (2.45), we get a dual formulation of the ROF problem (2.28), which, after basic manipulations, yields

$$(\mathscr{P}^\star) \quad \operatorname*{argmin}_{p \in \mathbb{R}^\Omega \times \mathbb{R}^\Omega} \|\tfrac{\lambda}{2} \mathrm{div}^{\mathrm{d}} p - u_0\|_2^2 \quad \text{subject to} \quad \|p\|_{\infty,2} \leq 1\,,$$

Besides, using the extremality relation (2.46), we see that the solutions $\bar{u}$, $\bar{p}$ of $(\mathscr{P})$ and $(\mathscr{P}^\star)$ must satisfy $-\lambda \mathrm{div}^{\mathrm{d}} \bar{p} = 2(\bar{u} - u_0)$, in particular, we recover $\bar{u}$ from $\bar{p}$ using

$$\bar{u} = u_0 - \frac{\lambda}{2} \mathrm{div}^{\mathrm{d}} \bar{p}\,.$$

The dual problem $(\mathscr{P}^\star)$ is far easier to handle than the primal problem $(\mathscr{P})$, in particular the dual problem is convex and differentiable, which is not the case for the primal problem. For instance, $(\mathscr{P}^\star)$ can be reformulated as the ℓ^2 projection of the quantity $\frac{2}{\lambda} u_0$ over the convex set

$$\mathscr{D} = \mathrm{div}^{\mathrm{d}} \mathscr{C}, \quad \text{where} \quad \mathscr{C} = \{p \in \mathbb{R}^\Omega \times \mathbb{R}^\Omega,\ \|p\|_{\infty,2} \leq 1\}\,,$$

and a semi-implicit scheme dedicated to this projection was proposed in [Chambolle 2004]. However, problem $(\mathscr{P}^\star)$ is also easy to handle using a simple projected gradient method, as done in [Chambolle 2005], since the projection over $\mathscr{C}$, i.e. the unit ball for the norm $\|\cdot\|_{\infty,2}$, is straightforward to compute (see its closed-form expression $\pi_{\infty,2}$ in Algorithm 2). In both cases, the numerical algorithms dedicated to the computation of the solution of $(\mathscr{P}^\star)$ involve the setting of only one time step parameter, instead of two when using the primal-dual algorithm of Chambolle and Pock. Besides, thanks to the regularity of the dual problem (the cost function is strongly convex with Lipschitz continuous gradient), some efficient convergence rates can be reached using some Nesterov acceleration strategies proposed in [Nesterov 1983, 2005], as it is done in [Weiss et al. 2009].

A dual formulation of the TV regularized inverse problem

We now focus on the inverse problem (2.30), related to the inversion of the operator $A : \mathbb{R}^\Omega \to \mathbb{R}^\omega$. This problem corresponds again to the choice of a cost function of the type

$$C = u \mapsto J(u, Ku) := G(u) + F(Ku)\,,$$

where $G(u) = 0$, $Ku = (\lambda\nabla^{\mathrm{d}}u, Au)$, and $F(Ku) = F_1(\lambda\nabla^{\mathrm{d}}u) + F_2(Au)$, setting $F_1(\lambda\nabla^{\mathrm{d}}u) = \|\lambda\nabla^{\mathrm{d}}u\|_{1,2} = \lambda\,\mathrm{TV}^{\mathrm{d}}(u)$ and $F_2(Au) = \|Au - u_0\|_2^2$. Remark that we have set here $X = \mathbb{R}^\Omega$, and $Y = \left(\mathbb{R}^\Omega \times \mathbb{R}^\Omega\right) \times \mathbb{R}^\omega$. Since we have $G^\star = \delta_{\{0\}}$ (the indicator function of the singleton $\{0\}$, noting 0 the zero of $\mathbb{R}^\Omega$), and $F^\star = (p,q) \mapsto \delta_{\|\cdot\|_{\infty,2}\leq 1}(p) + \frac{1}{4}\|q + 2\,u_0\|_2^2 - \|u_0\|_2^2$, using (2.45), we get a dual formulation of the inverse problem (2.30), which, after basic manipulations, yields

$$(\mathscr{P}^\star) \qquad \operatorname*{argmin}_{(p,q)\in(\mathbb{R}^\Omega\times\mathbb{R}^\Omega)\times\mathbb{R}^\omega} \|q - 2\,u_0\|_2^2 \quad \text{subject to} \quad \begin{cases} -\lambda\mathrm{div}^{\mathrm{d}}(p) + A^*q = 0 \\ \|p\|_{\infty,2} \leq 1 \end{cases}$$

which can be again interpreted as a projection in the dual space, over the intersection between the kernel of the linear operator $(p,q) \mapsto -\lambda\mathrm{div}^{\mathrm{d}}p + A^*q$ and the closed unit ball of the norm $\|\cdot\|_{\infty,2}$. Let $\bar{u}$ and $(\bar{p},\bar{q})$ denote some solutions of the primal and dual problems, a straightforward adaptation of the extremality relation (2.46) taking into account the additive separability of F with respect to (p,q) yields the relation

$$\begin{cases} -\lambda\mathrm{div}^{\mathrm{d}}\bar{p} + A^*\bar{q} = 0 \\ -\bar{p} \in \partial F_1(\lambda\nabla^{\mathrm{d}}\bar{u}) \\ -\bar{q} \in \partial F_2(A\bar{u}) = \{2(A\bar{u} - u_0)\} \end{cases}$$

which unfortunately does not give an explicit way to compute $\bar{u}$ from $\bar{p}$, so that even if we were able to easily compute $\bar{p}$, this would not help to easily solve the primal problem. However, notice that such kind of dual study for inverse problems is very useful to prove the equivalence between two famous numerical schemes (the Alternating Direction Method of Multipliers (ADMM) and the Douglas-Rachford Splitting (DRS) algorithms), in the sense that the iterates of ADMM applied to a primal problem $(\mathscr{P})$ are identical to that of DRS applied to a dual problem $(\mathscr{P}^\star)$, as nicely proven in [Chambolle and Pock 2011].

Chapter 3

The Shannon Total Variation

Contents

The content of this chapter, excepting Appendix 3.8.F, has been submitted to Journal of Mathematical Imaging and Vision (JMIV), in July 2016.

Abstract

Discretization schemes commonly used for total variation regularization lead to images that are difficult to interpolate, which is a real issue for applications requiring subpixel accuracy and aliasing control. In this chapter, we reconciliate total variation with Shannon interpolation and study a Fourier-based estimate that behaves much better in terms of grid invariance, isotropy, artifact removal, and sub-pixel accuracy. We show that this new variant (called Shannon total variation) can be easily handled with classical primal-dual formulations, and illustrate its efficiency on several image processing tasks, including deblurring, spectrum extrapolation, and a new aliasing reduction algorithm.

3.1 Introduction

Since total variation (TV) regularization was proposed by Rudin, Osher and Fatemi for image denoising [Rudin et al. 1992], it has proven extremely useful for many applications (and beyond image data, for that matter) like image deblurring [Vogel and Oman 1998, Chan and Wong 1998], inpainting [Chan et al. 2005], interpolation [Guichard and Malgouyres 1998], spectral extrapolation [Rougé and Seghier 1995], image decomposition [Vese and Osher 2003], super-resolution [Babacan et al. 2008], stereovision [Miled et al. 2009], and much more (see [Chambolle et al. 2010] and references therein for more examples). In the last decade, the development of dual and primal-dual formulations [Chambolle 2004, Beck and Teboulle 2009a, Weiss et al. 2009, Fadili and Peyré 2011, Chambolle and Pock 2011] and graph-cuts methods [Darbon and Sigelle 2006] has provided efficient algorithms for TV-based minimization problems, thus increasing even further the popularity of TV regularization.

A modern way to explain the efficiency of TV is to see it as a sparsity-promoting model: being defined by a L^1 norm (of the gradient), TV minimization tends to favor solutions whose gradient is sparse (that is, often takes the value 0), which corresponds to the so-called cartoon images. Of course, real-life photographs are not cartoons, but outside textured regions (which can be ignored in many image analysis tasks) they are close to that. Another explanation of the usefulness of TV is its ability to penalize oscillations (which is typically the kind of structures one wants to avoid when solving an ill-posed inverse problem) while allowing discontinuities at the same time.

When it comes to implementing an optimization problem involving a TV reg-

ularization term, like, e.g., TV denoising of an image u_0 by

$$\operatorname*{argmin}_u \|u - u_0\|^2 + \lambda \mathrm{TV}(u), \tag{3.1}$$

(where $\lambda > 0$ is a positive parameter selecting the desired amount of regularization), the issue of TV discretization arises. Most algorithms choose to approximate the continuous TV by a sum (over all pixels) of the ℓ^2 norm of a discrete finite-difference estimate of the image gradient, that is,

$$\mathrm{TV}^{\mathrm{d}}(u) = \sum_{(k,l)\in\Omega} \sqrt{(\partial_1 u(k,l))^2 + (\partial_2 u(k,l))^2} \tag{3.2}$$

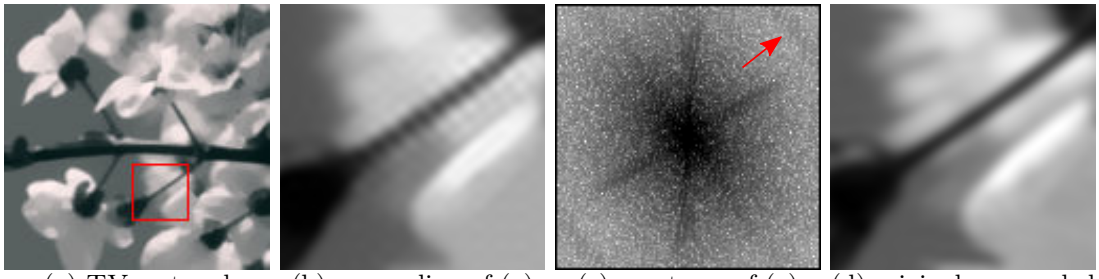
$$\text{where} \quad \begin{cases} \partial_1 u(k,l) = u(k+1,l) - u(k,l), \\ \partial_2 u(k,l) = u(k,l+1) - u(k,l), \end{cases} \tag{3.3}$$

and $u : \Omega \to \mathbb{R}$ is a discrete gray-level image defined on the finite domain $\Omega \subset \mathbb{Z}^2$ (we purposely ignore boundary issues here, as they are not related to our discussion). In the following, we shall refer to (3.2) as the discrete TV. In some situations, an anisotropic scheme (ℓ^1 norm) may be used [Chambolle 2005, Louchet and Moisan 2014, Abergel et al. 2015], leading to the anisotropic discrete TV

$$\mathrm{TV}^{\mathrm{d}}_{\mathrm{ani}}(u) = \sum_{(k,l)\in\Omega} |\partial_1 u(k,l)| + |\partial_2 u(k,l)|.$$

Curiously enough, as popular as they are, these numerical schemes present strong drawbacks in terms of image quality at pixel and subpixel scales. Indeed, an image obtained by minimizing TV^{d}-based energies is very difficult to interpolate, or, said differently, badly sampled according to Shannon theory. In practice, this means that trying to interpolate such an image will result in the appearance of undesired artifacts (see Figure 3.1), generally a mix between blockiness and ringing depending on the interpolation method. This strongly limits the possibility of exploiting an image delivered by a TV^{d}-based scheme, as usual operations like geometric transformations, registration, sub-pixel shape matching, derivative estimates (not to mention others) require well-interpolable images. New discrete schemes have been recently proposed [Chambolle et al. 2011, Condat 2016] to improve the isotropy of the discrete TV, but they do not solve (nor address) the interpolability issue we consider here.

In this chapter, we study a new formulation of the discrete TV, which reconciliates TV minimization and Shannon theory. This variant, which we shall name



(a) TV-restored (b) resampling of (a) (c) spectrum of (a) (d) original, resampled

Figure 3.1: Discrete TV produces aliasing. An image denoised with a classical discrete implementation of TV denoising (a) is improperly sampled, as attested by the aliasing artifact appearing in its Fourier spectrum ((c), red arrow), which is responsible for the undesired oscillating patterns that appear when magnifying the image using Shannon interpolation ((b), red arrows). Note that this artifact is not present on the original image (d). This experiment illustrates the difficulty of manipulating images at a subpixel scale after a processing involving the discrete TV.

Shannon Total Variation (STV), first appeared in [Malgouyres and Guichard 2001], and was later explicitly considered in [Moisan 2007] and then used in [Facciolo et al. 2009, Preciozzi et al. 2014] under the name *Spectral Total Variation* (but we shall not keep this name since it would introduce a confusion with [Gilboa 2013]). The STV variant consists in estimating the true total variation of the exact (continuous) total variation of the Shannon interpolate of u by using a Riemann sum approximation of the associated integral. We show that STV successfully addresses the above-mentioned issues and delivers images on which the discrete sinc and spline interpolations behave nicely, while preserving the desired properties of TV regularization. The lack of isotropy observed with classical finite difference schemes is also naturally avoided with STV. This comes at the expense of a few Fourier Transforms at each iteration of the optimization process, which is, in most applications, an affordable cost considering the strong benefits in terms of image quality.

The chapter is organized as follows. In Section 3.2, we present the discrete sinc interpolation as a consequence of Shannon sampling Theorem, and discuss in particular the (generally overlooked) difficulties encountered with Nyquist frequencies in the case of even image dimensions. We also give an independent justification of discrete sinc interpolation as the unique linear interpolation that defines invertible subpixellic translations, and discuss the link with B-spline interpolation. In Section 3.3, we define STV and discuss the choice of the upsampling factor used to discretize the continuous TV integral into a Riemann sum. We then show in Section 3.4 that STV-based algorithms can be efficiently implemented by deriving

a dual formulation which can be used in the powerful Chambolle-Pock optimization procedure. In Section 3.5, we illustrate the use of STV regularization in the case of several classical applications (denoising and more general inverse problems like deblurring, image magnification with spectrum extrapolation, tomography). We then present a new STV-based image restoration model involving a weight function in Fourier domain, which leads to interesting applications in terms of de-aliasing and can be viewed as an "image Shannonizer" as it provides a way to approximate a given image by a well-sampled one according to Shannon interpolation (Section 3.6). We finally conclude in Section 3.7 and present some perspectives.

3.2 Shannon interpolation

3.2.1 Shannon Sampling Theorem

A classical way to understand the relation between a (d-dimensional) continuous signal and its sampled version is Shannon Sampling Theorem, which can be considered in some way as the foundation of the digital era. In the following, we write $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{d} x_i y_i$ the canonical Euclidean inner product between two vectors $\mathbf{x} = (x_i)$ and $\mathbf{y} = (y_i)$ of $\mathbb{R}^d$.

Theorem 1 (Shannon-Whittaker). *Consider a positive real number δ and an absolutely integrable function $f : \mathbb{R}^d \to \mathbb{R}$ whose Fourier Transform*

$$\widehat{f}(\xi) = \int_{\mathbb{R}^d} f(\mathbf{x})\, e^{-i\langle \mathbf{x}, \xi \rangle}\, d\mathbf{x} \tag{3.4}$$

$$\textit{satisfies} \quad \forall \xi \notin \left[-\frac{\pi}{\delta}, \frac{\pi}{\delta}\right]^d, \quad \widehat{f}(\xi) = 0. \tag{3.5}$$

Then, f is continuous and uniquely determined by its values on $\delta\mathbb{Z}^d$, as for any $\mathbf{x} \in \mathbb{R}^d$,

$$f(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{Z}^d} f(\delta\, \mathbf{k}) \mathrm{sinc}\left(\frac{\mathbf{x}}{\delta} - \mathbf{k}\right) \tag{3.6}$$

where the cardinal sine function is defined on $\mathbb{R}^d$ by

$$\mathrm{sinc}(\mathbf{x}) = \prod_{i=1}^{d} \frac{\sin(\pi x_i)}{\pi x_i} \tag{3.7}$$

with the continuity-preserving convention $\frac{\sin(0)}{0} = 1$.

In this chapter, we will focus on one-dimensional signals ($d = 1$) and two-dimensional images ($d = 2$), but the extension to higher dimensions is straightforward. Apart from establishing a clear correspondence between the support of the Fourier spectrum of the bandlimited function f and the critical sampling step δ permitting its exact reconstruction from discrete samples, Shannon Sampling Theorem provides with Equation 3.6 (for $\delta = 1$) an interpolation formula that extends to $\mathbb{R}^d$ a discrete signal initially defined on $\mathbb{Z}^d$. However, this formula cannot be used as such in practice since it involves an infinite number of samples. We first discuss that issue in the simpler case $d = 1$.

3.2.2 Discrete Shannon interpolation of 1-D signals

Let us consider a discrete signal $s : I_M \to \mathbb{R}$ where $M \in \mathbb{N}^*$ and $I_M = \{0, 1, \dots M-1\}$. In order to define the Shannon interpolate $S : \mathbb{R} \to \mathbb{R}$ of s using (3.6), we first need to extend s into an infinite signal $\tilde{s} : \mathbb{Z} \to \mathbb{R}$, so that

$$S(x) = \sum_{k \in \mathbb{Z}} \tilde{s}(k) \operatorname{sinc}(x - k). \tag{3.8}$$

Extending s with 0 in $\mathbb{Z} \setminus I_M$ would be a poor solution, as it would interpolate a constant discrete signal s by an oscillating function. Instead, the classical solution consists in extending s as a M-periodic function $\tilde{s}(k) = s(k \mod M)$. Using such a periodic extension is not completely straightforward as it does not fit the hypotheses of Shannon Sampling Theorem (a M-periodic $\tilde{s} : \mathbb{Z} \to \mathbb{R}$ cannot be the sampled version of an absolutely integrable bandlimited function), but we can formally write

$$\begin{aligned} S(x) &= \sum_{k \in \mathbb{Z}} \tilde{s}(k) \operatorname{sinc}(x - k) \\ &= \sum_{p \in \mathbb{Z}} \sum_{k \in I_M} s(k) \operatorname{sinc}(x - k - pM) \\ &= \sum_{k \in I_M} s(k) \left(\sum_{p \in \mathbb{Z}} \operatorname{sinc}(x - k - pM) \right), \end{aligned}$$

and the factor of $s(k)$ can be explicitly computed with

Proposition 21 (discrete cardinal sine). *Define the discrete cardinal sine of*

order M as the M-periodization of the cardinal sine function, that is,

$$\mathrm{sincd}_M(x) := \lim_{n\to+\infty} \sum_{p=-n}^{n} \mathrm{sinc}(x - pM). \tag{3.9}$$

Then, one has

$$\mathrm{sincd}_M(x) = \begin{cases} \dfrac{\sin(\pi x)}{M \sin\left(\frac{\pi x}{M}\right)} & \text{if} \quad M \text{ is odd}, \\ \dfrac{\sin(\pi x)}{M \tan\left(\frac{\pi x}{M}\right)} & \text{if} \quad M \text{ is even}, \end{cases} \tag{3.10}$$

where the indeterminate forms $0/0$ are solved by continuity, that is, $\mathrm{sincd}_M(x) = 1$ for any $x \in M\mathbb{Z}$.

The proof is given in Appendix 3.8.A. In view of Proposition 21, we can rewrite the interpolation of s as

$$S(x) = \sum_{k\in I_M} s(k)\, \mathrm{sincd}_M(x - k). \tag{3.11}$$

Note that for small values of $|x|$ (more precisely, when $|x| \ll M$), we have $M \sin \frac{\pi x}{M} \simeq M \tan \frac{\pi x}{M} \simeq \pi x$, so that $\mathrm{sincd}_M(x) \simeq \mathrm{sinc}(x)$, which formally shows the asymptotic equivalence between sinc and sincd_M interpolation as $M \to +\infty$.

In practice, (3.11) is barely used, since there is an equivalent (but numerically more efficient) formulation due to the fact that sincd_M is a trigonometric polynomial.

Proposition 22. *The function sincd_M is a trigonometric polynomial, which can be written*

$$\mathrm{sincd}_M(x) = \mathrm{Re}\left(\frac{1}{M} \sum_{\alpha\in\widehat{I}_M} e^{2i\pi \frac{\alpha x}{M}} \right) \tag{3.12}$$

where $\widehat{I}_M = \left[-\frac{M}{2}, \frac{M}{2}\right) \cap \mathbb{Z}$ and the real part in (3.12) is required only if M is even.

Proof. The set $\widehat{I}_M$ is made of M consecutive integer values, and can thus be written

$$\widehat{I}_M = \{a, a+1, \dots a+M-1\},$$

where $a = -\lfloor \frac{M}{2} \rfloor$ denotes the (lower) integer part of $\frac{M}{2}$. Thus, if $x \notin M\mathbb{Z}$ we have

$$\sum_{\alpha \in \widehat{I}_M} e^{2i\pi \frac{\alpha x}{M}} = \sum_{\alpha=a}^{a+M-1} \left(e^{2i\pi \frac{x}{M}}\right)^{\alpha} = e^{2i\pi \frac{ax}{M}} \cdot \frac{1 - e^{2i\pi x}}{1 - e^{2i\pi \frac{x}{M}}} = e^{i\pi x \frac{2a+M-1}{M}} \cdot \frac{\sin(\pi x)}{\sin \pi \frac{x}{M}}.$$

If M is odd, $2a + M - 1 = 0$ and we get

$$\frac{1}{M} \sum_{\alpha \in \widehat{I}_M} e^{2i\pi \frac{\alpha x}{M}} = \frac{\sin(\pi x)}{M \sin \pi \frac{x}{M}} = \mathrm{sincd}_M(x)$$

as expected. If M is even, $2a + M - 1 = -1$ and we now obtain

$$\mathrm{Re}\left(\frac{1}{M} \sum_{\alpha \in \widehat{I}_M} e^{2i\pi \frac{\alpha x}{M}}\right) = \frac{\sin(\pi x)}{M \sin \pi \frac{x}{M}} \cdot \mathrm{Re}(e^{-i\frac{\pi x}{M}}) = \frac{\sin(\pi x)}{M \tan \pi \frac{x}{M}} = \mathrm{sincd}_M(x)$$

as well. □

A consequence of Proposition 22 is that the Shannon interpolation formula (3.11) can be rewritten using the Discrete Fourier Transform recalled below.

Definition 15. *The discrete Fourier Transform (DFT) of a signal $s : I_M \to \mathbb{R}$ is the M-periodic complex-valued signal $\widehat{s}$ defined by*

$$\forall \alpha \in \mathbb{Z}, \quad \widehat{s}(\alpha) = \sum_{k \in I_M} s(k) e^{-2i\pi \frac{\alpha k}{M}}.$$

Proposition 23. *The discrete Shannon interpolation of a signal $s : I_M \to \mathbb{R}$ can be written*

$$S(x) = \mathrm{Re}\left(\frac{1}{M} \sum_{\alpha \in \widehat{I}_M} \widehat{s}(\alpha)\, e^{2i\pi \frac{\alpha x}{M}}\right), \tag{3.13}$$

and the real part is required only if M is even.

Proof. Thanks to Proposition 22, the Shannon interpolate of s defined by (3.11) can be rewritten

$$\begin{aligned} S(x) &= \sum_{k \in I_M} s(k)\, \mathrm{Re}\left(\frac{1}{M} \sum_{\alpha \in \widehat{I}_M} e^{2i\pi \frac{\alpha(x-k)}{M}}\right) \\ &= \mathrm{Re}\left(\frac{1}{M} \sum_{\alpha \in \widehat{I}_M} \left(\sum_{k \in I_M} s(k) e^{-2i\pi \frac{\alpha k}{M}}\right) e^{2i\pi \frac{\alpha x}{M}}\right) \end{aligned}$$

from which (3.13) directly follows. □

Note that if $x \in I_M$, the function $\alpha \mapsto \widehat{s}(\alpha)\, e^{2i\pi \frac{\alpha x}{M}}$ is M-periodic, and since $\widehat{I}_M$ is an interval of M consecutive values, we have

$$\frac{1}{M} \sum_{\alpha \in \widehat{I}_M} \widehat{s}(\alpha)\, e^{2i\pi \frac{\alpha x}{M}} = \frac{1}{M} \sum_{\alpha \in I_M} \widehat{s}(\alpha)\, e^{2i\pi \frac{\alpha x}{M}} = s(x)$$

as we recognize the inverse DFT of $\widehat{s}$. As expected, the Shannon interpolation defined by (3.13) is exact (that is, the restriction of S to I_M is exactly s).

Also remark that when M is even, we need a real part to cancel the imaginary part of the term $\alpha = -\frac{M}{2}$ in the sum (3.13) since the conjugate term (which would correspond to $\alpha = \frac{M}{2}$) is not present in the sum. The real part can be avoided when $\widehat{s}(-\frac{M}{2}) = 0$, or by considering instead a sum with $M+1$ terms, as stated by

Proposition 24. *Define, for integer M,*

$$\varepsilon_M(\alpha) = \begin{cases} 1/2 & \text{if } |\alpha| = \frac{M}{2}, \\ 1 & \text{otherwise.} \end{cases} \tag{3.14}$$

The discrete Shannon interpolate of a signal $s : I_M \to \mathbb{R}$ can be written

$$S(x) = \frac{1}{M} \sum_{-\frac{M}{2} \leq \alpha \leq \frac{M}{2}} \varepsilon_M(\alpha) \cdot \widehat{s}(\alpha)\, e^{2i\pi \frac{\alpha x}{M}}. \tag{3.15}$$

Note that if M is odd, ε_M is identically equal to 1. This asymmetry between the case M odd and M even can be simply explained. Let us define as T_M the real vector space of real-valued trigonometric polynomials that can be written as complex linear combinations of $(x \mapsto e^{2i\pi \frac{\alpha x}{M}})_{-\frac{M}{2} \leq \alpha \leq \frac{M}{2}}$. If M is odd, $\dim T_M = M$ and there is a unique element S of T_M that exactly interpolates s, and it is given by (3.13). If M is even, $\dim T_M = M+1$ and any element of T_M that exactly interpolates s can be written under the form $S(x) + \lambda \sin(\pi x)$ with $\lambda \in \mathbb{R}$, and the interpolation formula (3.13) corresponds to the implicit (minimal norm) choice $\lambda = 0$.

3.2.3 Shannon interpolation of 2-D images

Let $u : I_M \times I_N \to \mathbb{R}$ be a discrete $M \times N$ image. Its 2-dimensional DFT $\widehat{u} : \mathbb{Z}^2 \to \mathbb{C}$ is defined by

$$\widehat{u}(\alpha, \beta) = \sum_{\substack{k \in I_M \\ l \in I_N}} u(k, l) e^{-2i\pi\left(\frac{\alpha k}{M} + \frac{\beta l}{N}\right)}, \tag{3.16}$$

and the natural extension of (3.11) is

Definition 16. *The discrete Shannon interpolate of an image $u : I_M \times I_N \to \mathbb{R}$ is $U : \mathbb{R}^2 \to \mathbb{R}$ defined by*

$$U(x, y) = \sum_{\substack{k \in I_M \\ l \in I_N}} u(k, l) \operatorname{sincd}_M(x - k) \operatorname{sincd}_N(y - l)\,. \tag{3.17}$$

As in the 1-D case, Definition 16 can be reformulated in the Fourier domain.

Proposition 25. *The discrete Shannon interpolate of an image $u : I_M \times I_N \to \mathbb{R}$ can be written*

$$U(x, y) = \frac{1}{MN} \sum_{\substack{-\frac{M}{2} \leq \alpha \leq \frac{M}{2} \\ -\frac{N}{2} \leq \beta \leq \frac{N}{2}}} \varepsilon_M(\alpha)\varepsilon_N(\beta) \cdot \widehat{u}(\alpha, \beta) e^{2i\pi\left(\frac{\alpha x}{M} + \frac{\beta y}{N}\right)}, \tag{3.18}$$

where ε_M and ε_N are defined in (3.14).

Proof. Simply remark that (3.12) can be rewritten

$$\operatorname{sincd}_M(x) = \frac{1}{M} \sum_{-\frac{M}{2} \leq \alpha \leq \frac{M}{2}} \varepsilon_M(\alpha) e^{2i\pi \frac{\alpha x}{N}} \tag{3.19}$$

and (3.18) follows quite directly from (3.16) and (3.17). □

Note that if both M and N are odd, (3.18) boils down to

$$U(x, y) = \frac{1}{MN} \sum_{\substack{\alpha \in \widehat{I}_M \\ \beta \in \widehat{I}_N}} \widehat{u}(\alpha, \beta) e^{2i\pi\left(\frac{\alpha x}{M} + \frac{\beta y}{N}\right)}, \tag{3.20}$$

which is exactly the definition of the inverse DFT of $\widehat{u}$ for integer values of x and y. Thus, one could wonder whether in the general case (M, N even or odd) the generalization of (3.13), that is,

$$U'(x,y) = \mathrm{Re}\left(\frac{1}{MN} \sum_{\substack{\alpha \in \widehat{I}_M \\ \beta \in \widehat{I}_N}} \widehat{u}(\alpha,\beta)\, e^{2i\pi\left(\frac{\alpha x}{M} + \frac{\beta y}{N}\right)} \right), \tag{3.21}$$

would be an equivalent definition of U as in the 1-D case. In fact, (3.17) and (3.21) both define bivariate trigonometric polynomials of $T_M \otimes T_N$ that exactly interpolate u in $I_M \times I_N$, but they differ when both M and N are even. In that case, $U'(x,y)$ can still be rewritten in a form similar to (3.18), but we have to change the coefficient $\varepsilon_M(\alpha)\varepsilon_N(\beta)$ into

$$\varepsilon'_{M,N}(\alpha,\beta) = \begin{cases} \frac{1}{2} & \text{if } \ (\alpha,\beta) = \pm(\frac{M}{2}, \frac{N}{2}), \\ 0 & \text{if } \ (\alpha,\beta) = \pm(-\frac{M}{2}, \frac{N}{2}), \\ 1 & \text{otherwise.} \end{cases} \tag{3.22}$$

Thus, one easily shows that

$$U'(x,y) = U(x,y) - \widehat{u}\left(\frac{M}{2}, \frac{N}{2}\right) \sin(\pi x) \sin(\pi y). \tag{3.23}$$

Even if this difference is expected to be small for natural images (the Fourier coefficients of a natural image decrease rather quickly as the frequency increases), the true interpolate U is to be preferred to U' as it is separable and more invariant; in particular, the transform $u \mapsto U'$ does not commute with the plane transforms $(x,y) \mapsto (-x,y)$ and $(x,y) \mapsto (x,-y)$.

In the literature, most papers involving 2-D discrete Shannon interpolation either do not mention this issue [Getreuer 2011, Malgouyres and Guichard 2001], or restrict their study to odd dimensions [Simon and Morel 2016], or use the (slightly incorrect) variant U' [Briand and Vacher 2015] (probably because taking the real part is the most simple way to get rid of the imaginary part that naturally appears when Nyquist frequencies are not carefully handled).

3.2.4 Dealing with periodization artifacts

Using discrete Shannon interpolation requires a careful handling of edge effects, as the implicit periodization of the image may produce interpolation artifacts (that

is, undesired oscillations) near the boundary of the image domain if the intensity values on the opposite edges of the image domain do not match well. This issue is discussed in detail in [Moisan 2011], and an efficient solution is proposed that consists in decomposing the original image into the sum of a periodic image and a smooth image. Other solutions exist like symmetrization or apodization using an appropriate weight function (e.g., a Hamming window), but they appear to be less efficient in general. In all the experiments presented throughout this chapter (and in particular in Section 3.5 and 3.6), the periodic plus smooth decomposition of [Moisan 2011] will systematically be used.

3.2.5 Shannon interpolation and reversible transforms

As we saw earlier, Shannon Sampling Theorem provides a nice theoretical framework that establishes a one-to-one correspondence between continuous bandlimited and discrete images, which naturally leads to the discrete Shannon interpolation we just presented. Interestingly, there is another justification for Shannon interpolation, that does not explicitly refer to Shannon Sampling Theorem: basically, it is the only linear interpolation that defines invertible subpixellic translations (in a periodic setting). In the following, we assume for simplicity that M is an odd integer, and write $\mathcal{S}$ the space of M-periodic signals $s : \mathbb{Z} \to \mathbb{R}$.

Theorem 2. *There exists a unique family of linear operators $(T_z)_{z\in\mathbb{R}}$ on $\mathcal{S}$ such that :*

(i) $z \mapsto T_z$ *is continuous,*

(ii) $\forall k, z \in \mathbb{Z},\ T_z s(k) = s(k - z)$,

(iii) $\forall w, z \in \mathbb{R},\ T_{w+z} = T_w \circ T_z$,

(iv) $\lim_{z\to 0} |z|^{-1}\|T_z - id\|_2$ *is minimal.*

It is defined by

$$T_z s(k) = S(k - z), \tag{3.24}$$

where S is the discrete Shannon interpolate of s defined in (3.11) or equivalently in (3.13).

The Proof is given in Appendix 3.8.B. Theorem 2 remains true for M even, provided that we define $\mathcal{S}$ in this case by

$$\mathcal{S} = \left\{ s : I_M \to \mathbb{R},\ \sum_{k\in I_M} (-1)^k s(k) = 0 \right\}. \tag{3.25}$$

(Note that it is equivalent to assume $\widehat{s}(M/2) = 0$). This restriction is needed to exclude from $\mathcal{S}$ the alternated signal $k \mapsto (-1)^k$, which clearly cannot be translated in a way compatible with Hypotheses (ii) and (iii).

Theorem 2 shows that the only minimal continuous semi-group extending the integer (periodic) translations is given by Shannon interpolation. This result is interesting in the sense that it brings another justification to Shannon interpolation without referring to Shannon Sampling Theorem (or to the Fourier Transform, for that matter): among linear interpolation methods, only Shannon interpolation is able to translate images without information loss.

From Equation (3.74), we can see that a subpixellic translation with Shannon interpolation can be implemented with two DFTs, as

$$\widehat{T_z s}(\alpha) = e^{-2i\pi\alpha z/M}\, \widehat{s}(\alpha)\,. \tag{3.26}$$

Moreover, T_z is a linear isometry ($\|T_z s\|_2 = \|s\|_2$), which is another way to explain that no information loss occurs.

Signal and image magnification is also very easy to perform with discrete Shannon interpolation, as it essentially boils down to a *zero-padding* in the Fourier domain (for even dimensions, it is also necessary to split the coefficients corresponding to Nyquist frequencies $\alpha = \pm\frac{M}{2}$ or $\beta = \pm\frac{N}{2}$). More surprisingly, image rotation can also be implemented efficiently with the DFT (see [Yaroslavsky 1996]), thanks to the following factorization of a rotation matrix into a product of shear matrices:

$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} = \begin{pmatrix} 1 & -t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \sin\theta & 1 \end{pmatrix} \begin{pmatrix} 1 & -t \\ 0 & 1 \end{pmatrix} \tag{3.27}$$

with $t = \tan\frac{\theta}{2}$. As a shear transform like

$$v(x, y) = u(x - ty, y) \tag{3.28}$$

consists in applying 1-D translations to each line of u, a 2-D rotation can be decomposed as a combination of 1-D translations, which can be implemented in the Fourier domain. For that reason, image rotation with discrete Shannon interpolation is a linear isometry, and can thus be considered as a lossless transform.

3.2.6 Link with spline interpolation

A popular alternative to Shannon interpolation is spline interpolation. Without going too much into details (see [Unser et al. 1991, Unser 2000] and the

references therein), it is worth mentioning the relation between spline and Shannon interpolation, and to understand how they can be combined to yield what is probably the most accurate and efficient linear interpolation of bandlimited signals.

The spline interpolation of order n ($n \in \mathbb{N}$) of a signal $s \in \ell^2(\mathbb{Z})$ can be written

$$S^n(x) = \sum_{k\in\mathbb{Z}} c(k)\, \beta^n(x-k), \tag{3.29}$$

where $\beta^n : \mathbb{R} \to \mathbb{R}$ is the spline of order n defined by induction by $\beta^0 = \mathbb{1}_{[-\frac{1}{2},\frac{1}{2})}$ and $\beta^{k+1} = \beta^k * \beta^0$ for all $k \in \mathbb{N}$. It can be shown that the signal $c : \mathbb{Z} \to \mathbb{R}$ is uniquely defined by the interpolation constraint $S^n(k) = s(k), k \in \mathbb{Z}$. When $n \in \{0, 1\}$, one has $c = s$ and spline interpolation corresponds to piecewise constant ($n = 0$) or piecewise affine ($n = 1$) interpolation. When $n > 1$, c depends linearly on s and can be efficiently computed using recursive filtering [Unser et al. 1991]. As remarked in [Unser 1997], spline interpolation achieves an optimal trade-off between complexity (the support of β^n is an interval with length $n + 1$) and asymptotic accuracy (rate of convergence towards the unsampled signal as the sampling step tends to 0). How does spline interpolation compare with Shannon interpolation? Indeed, (3.29) can be rewritten as

$$S^n(x) = \sum_{k\in\mathbb{Z}} s(k)\, \beta^n_{\text{card}}(x-k), \tag{3.30}$$

where $\beta^n_{\text{card}} : \mathbb{R} \to \mathbb{R}$ is the cardinal spline of order n defined in the Fourier domain by

$$\widehat{\beta^n_{\text{card}}}(\xi) = \frac{\left(\operatorname{sinc}\frac{\xi}{2\pi}\right)^{n+1}}{\sum_{k\in\mathbb{Z}} \beta^n(k) e^{-ik\xi}}. \tag{3.31}$$

This provides a nice interpretation of spline interpolation in the Fourier domain, as the Fourier transform of (3.30) yields

$$\widehat{S^n}(\alpha) = \widehat{s}(\alpha)\widehat{\beta^n_{\text{card}}}(\alpha), \tag{3.32}$$

where $\widehat{s}(\alpha) = \sum_{k\in\mathbb{Z}} s(k)e^{-ik\alpha}$ is the Fourier Transform of the discrete signal s. Thus, if S is a bandlimited signal ($\operatorname{supp}\widehat{S} \subset [-\pi, \pi]$) and $s(k) = S(k)$ for all $k \in \mathbb{Z}$, the Fourier transform of S_n is deduced from $\widehat{S}$ by periodization and multiplication by $\widehat{\beta^n_{\text{card}}}$. This is to be compared to Shannon interpolation, that recovers the exact signal S since

$$\widehat{S}(\alpha) = \widehat{s}(\alpha)\mathbb{1}_{[-\pi,\pi]}. \tag{3.33}$$

In fact, $\widehat{\beta^n_{\text{card}}} \to \mathbb{1}_{[-\pi,\pi]}$ as $n \to +\infty$ [Aldroubi et al. 1992] (or, equivalently, $\beta^n_{\text{card}} \to$ sinc), which means that spline interpolation can be viewed as an approximation of Shannon interpolation (the equivalence being asymptotically obtained for $n = +\infty$). For finite n however, the effect of spline interpolation in the Fourier domain is questionable: it creates high frequencies aliases (by spectrum periodization), and then attenuates the whole spectrum (the known part $[-\pi, \pi]$ included) by an apodization function that is a smooth approximation of $\mathbb{1}_{[-\pi,\pi]}$. This apodization function (that is, $\widehat{\beta^n_{\text{card}}}$) is represented in Figure 3.2 for various values of n.

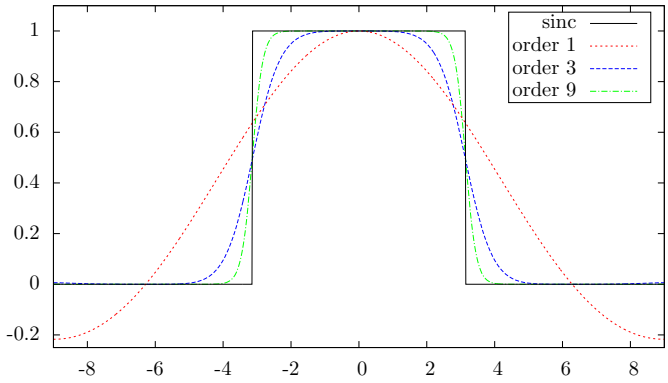
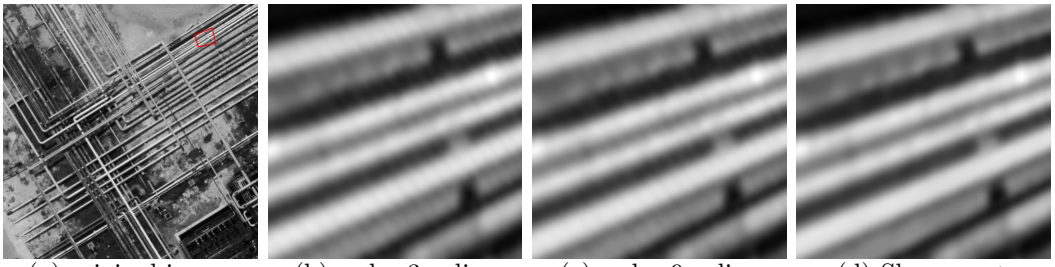


Figure 3.2: Cardinal splines in the Fourier domain. The Fourier transform of the interpolation kernels β^n_{card} are represented for $n = 1, 3, 9$. As n increases, they get closer to the ideal low-pass filter obtained with the sinc kernel. The approximation is responsible for blur (attenuation of known frequencies) and aliasing (creation of high frequencies duplicated from existing low frequencies) on spline-interpolated images.

On the one hand, spline interpolation is computationally efficient, and also versatile: it can be used to magnify an image by an arbitrary factor, or to apply an homography or a non-rigid transform to an image. On the other hand, Shannon interpolation is very accurate, as it does not attenuate known Fourier coefficients or create high-frequency aliases. Getting the best of the two worlds (that is, the accuracy of exact Shannon interpolation and the efficiency of spline interpolation) is easy: magnify the original image by a small factor (e.g. 2), and then use spline interpolation on the magnified image. Figure 3.3 illustrates the interest of such a combination in the case of a homographic transform.

In this section, we gave a precise definition of Shannon interpolation (with a careful treatment of Nyquist frequencies in the case of even dimensions), and saw how it provides a nice framework for interpolating bandlimited images with a high degree of accuracy. It is particularly useful for imaging sciences that require an accurate treatment of subpixel scales and a strict control of artifacts (in particular,



(a) original image (copyright CNES) (b) order 3 spline (c) order 9 spline (d) Shannon + order 3 spline

Figure 3.3: High quality homographic transforms using a combination of Shannon and spline interpolations. Applying an homographic transform to an image (a) requires the use of an interpolation scheme. Spline kernels are interesting but may produce undesired artifacts (the slight superimposed line hatch patterns in b,c) due to the creation of spurious high frequencies. Applying the same transform with Shannon interpolation alone would be computationally very expensive, but a simple $\times 2$ magnification with Shannon interpolation followed by an homographic transform implemented by a spline of order 3 produces an artifact-free image for a computational cost equivalent to spline interpolation.

satellite imaging). As we shall see in the next sections, Shannon interpolation can be made compatible with total variation regularization, provided that we use what we shall call the *Shannon total variation*.

3.3 The Shannon total variation

3.3.1 Definition

Let $|\cdot|$ denotes the ℓ^2 norm over $\mathbb{R}^2$, let $\Omega = I_M \times I_N$ denote a 2-D discrete domain of size $M \times N$ and $u \in \mathbb{R}^\Omega$ a discrete gray-level image with domain Ω. We define the Shannon total variation of u by

$$\mathrm{STV}_\infty(u) = \int_{[0,M]\times[0,N]} |\nabla U(x,y)| \, dxdy \, , \tag{3.34}$$

where U is the Shannon interpolation of u defined in (3.17), and $\nabla U : \mathbb{R}^2 \to \mathbb{R}^2$ denotes the gradient of the trigonometric polynomial U. No closed-form formula exist for (3.34), but we can approximate this continuous integral with the Riemann sum

$$\mathrm{STV}_n(u) = \frac{1}{n^2} \sum_{(k,l)\in\Omega_n} |\nabla_n u(k,l)| \, , \tag{3.35}$$

where $n \in \mathbb{N}^*$, $\Omega_n = I_{nM} \times I_{nN}$ and

$$\forall (k,l) \in \Omega_n, \quad \nabla_n u(k,l) = \nabla U\left(\tfrac{k}{n}, \tfrac{l}{n}\right) .$$

In order to compute $\mathrm{STV}_n(u)$, we need to focus on the practical computation of $\nabla_n u$. By differentiating (3.18), we get the gradient of U, that is, $\forall (x,y) \in \mathbb{R}^2$,

$$\nabla U(x,y) = \frac{1}{MN} \sum_{\substack{-\frac{M}{2} \le \alpha \le \frac{M}{2} \\ -\frac{N}{2} \le \beta \le \frac{N}{2}}} e^{2i\pi\left(\frac{\alpha x}{M} + \frac{\beta y}{N}\right)} g_{\widehat{u}}(\alpha,\beta)\,, \tag{3.36}$$

where

$$g_{\widehat{u}}(\alpha,\beta) = 2i\pi\, \varepsilon_M(\alpha)\varepsilon_N(\beta)\, \widehat{u}(\alpha,\beta) \begin{pmatrix} \alpha/M \\ \beta/N \end{pmatrix} . \tag{3.37}$$

Therefore, $\nabla_n u$ can be efficiently computed in the Fourier domain for $n \ge 2$ with the following

Proposition 26. *For any $n \ge 2$ and any $(\alpha,\beta) \in \widehat{\Omega_n} := \widehat{I}_{nN} \times \widehat{I}_{nM}$, we have*

$$\widehat{\nabla_n u}(\alpha,\beta) = \begin{cases} n^2 g_{\widehat{u}}(\alpha,\beta) & \text{if } |\alpha| \le \frac{M}{2},\ |\beta| \le \frac{N}{2}\,, \\ 0 & \text{otherwise,} \end{cases} \tag{3.38}$$

where $g_{\widehat{u}}$ is given by (3.37).

Proof. The result comes directly when writing (3.36) with $(x,y) = \left(\frac{k}{n}, \frac{l}{n}\right)$, and extending the sum to the frequency domain $\widehat{\Omega_n}$ by adding zero terms. Note that $\widehat{\Omega_n}$ contains all the frequencies (α,β) such that $-\frac{M}{2} \le \alpha \le \frac{M}{2}$ and $-\frac{N}{2} \le \beta \le \frac{N}{2}$ involved in (3.36) since $n > 1$. □

The next Proposition establishes an upper-bound for the induced ℓ^2 norm (noted $|||\cdot|||$) of the ∇_n operator, which will be useful later.

Proposition 27. *For any $n \ge 2$, we have*

$$|||\nabla_n||| \le n\pi\sqrt{2}\,. \tag{3.39}$$

Proof. Let $u \in \mathbb{R}^{\Omega}$, from (3.38) we deduce

$$\|\widehat{\nabla_n u}\|^2 = \left\|n^2 g_{\widehat{u}}\right\|^2 \le 4\pi^2 n^4 \left\|\widehat{u}\right\|^2 \left(\frac{1}{4} + \frac{1}{4}\right) , \tag{3.40}$$

since for any (α,β) such as $|\alpha| \leq \frac{M}{2}$ and $|\beta| \leq \frac{N}{2}$, we have $|\varepsilon_M(\alpha)\varepsilon_N(\beta)\frac{\alpha}{M}|^2 \leq \frac{1}{4}$ and $|\varepsilon_M(\alpha)\varepsilon_N(\beta)\frac{\beta}{N}|^2 \leq \frac{1}{4}$. Then, using the Parseval identity in (3.40), that is,

$$\|\nabla_n u\|^2 = \frac{1}{n^2 MN}\|\widehat{\nabla_n u}\|^2 \quad \text{and} \quad \frac{1}{MN}\|\widehat{u}\|^2 = \|u\|^2\,,$$

yields $\|\nabla_n u\|^2 \leq 2\pi^2 n^2 \|u\|^2$ and consequently (3.39). □

Similarly to Proposition 26, we can compute the adjoint of ∇_n in the Fourier domain (the proof is detailed in Appendix 3.8.C).

Proposition 28. *Let* $\mathrm{div}_n = -\nabla_n^*$, *then for any* $n \geq 2$, $p = (p_x, p_y) \in \mathbb{R}^{\Omega_n} \times \mathbb{R}^{\Omega_n}$, *and* $(\alpha,\beta) \in \widehat{\Omega} := \widehat{I_M} \times \widehat{I_N}$, *we have*

$$\widehat{\mathrm{div}_n(p)}(\alpha,\beta) = 2i\pi\left(\frac{\alpha}{M} h_{\widehat{p_x}}(\alpha,\beta) + \frac{\beta}{N} h_{\widehat{p_y}}(\alpha,\beta)\right),$$

with

$$h_{\widehat{p_x}}(\alpha,\beta) = \begin{cases} \widehat{p_x}(\alpha,\beta) & \text{if } |\alpha| < \frac{M}{2},\ |\beta| < \frac{N}{2} \\ \frac{1}{2}\left(\widehat{p_x}(\alpha,\beta) - \widehat{p_x}(-\alpha,\beta)\right) & \text{if } \alpha = -\frac{M}{2},\ |\beta| < \frac{N}{2} \\ \frac{1}{2}\left(\widehat{p_x}(\alpha,\beta) + \widehat{p_x}(\alpha,-\beta)\right) & \text{if } |\alpha| < \frac{M}{2},\ \beta = -\frac{N}{2} \\ \frac{1}{4}\sum\limits_{\substack{s_1=\pm 1 \\ s_2=\pm 1}} s_1\, \widehat{p_x}(s_1\alpha, s_2\beta) & \text{if } (\alpha,\beta) = (-\frac{M}{2}, -\frac{N}{2})\,, \end{cases}$$

and

$$h_{\widehat{p_y}}(\alpha,\beta) = \begin{cases} \widehat{p_y}\,(\alpha,\beta) & \text{if } |\alpha| < \frac{M}{2},\ |\beta| < \frac{N}{2} \\ \frac{1}{2}\left(\widehat{p_y}\,(\alpha,\beta) + \widehat{p_y}\,(-\alpha,\beta)\right) & \text{if } \alpha = -\frac{M}{2},\ |\beta| < \frac{N}{2} \\ \frac{1}{2}\left(\widehat{p_y}\,(\alpha,\beta) - \widehat{p_y}\,(\alpha,-\beta)\right) & \text{if } |\alpha| < \frac{M}{2},\ \beta = -\frac{N}{2} \\ \frac{1}{4}\sum\limits_{\substack{s_1=\pm 1 \\ s_2=\pm 1}} s_2\, \widehat{p_y}\,(s_1\alpha, s_2\beta) & \text{if } (\alpha,\beta) = (-\frac{M}{2}, -\frac{N}{2})\,. \end{cases}$$

Notice that Propositions 26 to 28 can be easily adapted to the case $n = 1$. However, we shall not need to consider this case as STV_1 happens to be a poor approximation of STV_∞ (see next section). Note also that similar definitions and propositions could be established for the U' variant of Shannon interpolation mentioned in (3.21). This variant yields somewhat simpler formulas (no weights are required to handle Nyquist frequencies in the case of even dimensions) since all

operators can be obtained by taking the real part of complex-valued images. However, in addition to being less invariant (as discussed in the end of Section 3.2.3), U' is also computationally less efficient as it requires the computation of DFTs of complex-valued images.

3.3.2 Choice of the oversampling factor n

When estimating $\mathrm{STV}_\infty(u)$ with $\mathrm{STV}_n(u)$, which value of the oversampling factor n should we choose? We experimentally observed on many images that the convergence with respect to n is extremely fast, so that in practice choosing $n = 2$ or $n = 3$ is enough. Note that an estimate of $\mathrm{STV}_\infty(u)$ could also be obtained by using a finite difference scheme on the image magnified with Shannon interpolation, that is, $n^{-1}\mathrm{TV}^{\mathrm{d}}(Z_n u)$ with

$$\forall (k,l) \in \Omega_n, \quad Z_n u(k,l) = U\left(\frac{k}{n}, \frac{l}{n}\right).$$

Both estimate are consistent in the sense that

$$\lim_{n\to+\infty} \mathrm{STV}_n(u) = \lim_{n\to+\infty} n^{-1}\mathrm{TV}^{\mathrm{d}}(Z_n u) = \mathrm{STV}_\infty(u).$$

However, the convergence speed is much worse for the latter, which comforts us in the choice of STV_n (see Table 3.1).

n	$n^{-1}\mathrm{TV}^{\mathrm{d}}(Z_n u)$	$\mathrm{STV}_n(u)$
1	$1.6 \cdot 10^{-1}$	$\mathbf{1.8 \cdot 10^{-2}}$
2	$4.2 \cdot 10^{-2}$	$\mathbf{1.3 \cdot 10^{-3}}$
3	$2.1 \cdot 10^{-2}$	$\mathbf{1.7 \cdot 10^{-4}}$
5	$8.6 \cdot 10^{-3}$	$\mathbf{7.3 \cdot 10^{-5}}$
10	$2.8 \cdot 10^{-3}$	$\mathbf{3.4 \cdot 10^{-6}}$

Table 3.1: Relative errors of two STV_∞ estimates. We compare two estimates of $\mathrm{STV}_\infty(u)$ when u is the classical "*Lena*" image. As we can observe, the relative errors are much smaller with $\mathrm{STV}_n(u)$ (third column) than with $n^{-1}\mathrm{TV}^{\mathrm{d}}(Z_n u)$ (second column), and the convergence with respect to n is faster. Even for $n = 2$, the STV_2 estimate is very accurate with a relative error of 0.1% or so. This experiment has been repeated on many other images, including pure noise images, and yielded similar conclusions for all of them.

As concerns the idea of estimating $\mathrm{STV}_\infty(u)$ with $\mathrm{STV}_1(u)$, the following result shows that it could lead to incorrect results, as controlling $\mathrm{STV}_1(u)$ is not

sufficient to control $\mathrm{STV}_\infty(u)$. We believe that, on the contrary, such a control is ensured as soon as $n \geq 2$, even though we have no proof of this affirmation yet.

Theorem 3. *There exists no constant C such that*

$$\mathrm{STV}_\infty(u) \leq C \cdot \mathrm{STV}_1(u)$$

for any positive integer M and any discrete image u of size $M \times M$.

The proof is given in Appendix 3.8.D. It consists in building a sequence of discrete images u_M with size $M \times M$ such that $\mathrm{STV}_1(u_M)$ is fixed but $\mathrm{STV}_\infty(u_M)$ increases to $+\infty$ with M.

In all the experiments reported in this chapter, we used STV_n with $n = 3$, but we observed only very slight improvements (and sometimes none) compared to the case $n = 2$, which should probably be preferred when computational issues are important. Note also that one could choose non-integer values of n (only nM and nN have to be integers), which could also be interesting for computational issues.

3.4 Duality tools for handling the STV regularizer in a variational framework

3.4.1 Recall of convex analysis

We here briefly recall some classical convex analysis results needed for non-smooth convex optimization. We refer to [Ekeland and Témam 1999] for a more detailed presentation.

Consider a finite-dimensional real vector space E and let $E^\star$ denotes its dual space, that is, the set of linear mappings from E to $\mathbb{R}$. Let $\overline{\mathbb{R}}$ denotes the set $\mathbb{R} \cup \{-\infty, +\infty\}$ and $\langle \cdot, \cdot \rangle : E^\star \times E \to \mathbb{R}$ the bilinear mapping defined by

$$\forall \varphi \in E^\star,\ \forall u \in E, \quad \langle \varphi, u \rangle = \varphi(u)\,.$$

An affine function on E is a function $\mathcal{A} : u \mapsto \langle \varphi, u \rangle + \alpha$, where $\varphi \in E^\star$ is called the slope of $\mathcal{A}$ and $\alpha \in \mathbb{R}$ the constant term. We denote by $\Gamma(E)$ the set of functions $F : E \to \overline{\mathbb{R}}$ which are the pointwise supremum of a family of affine functions over E. One can show that F is an element of $\Gamma(E)$ if and only if it is convex and lower semi-continuous (l.s.c.) and does not take the value $-\infty$ unless it is constant. In order to dismiss singular cases, we say that F is proper if it

never assumes the value $-\infty$ and is different from the constant $+\infty$. We denote by $\Gamma_0(E)$ the set of proper elements of $\Gamma(E)$.

Given a function $F : E \to \overline{\mathbb{R}}$, the Γ-regularization of F is the largest element of $\Gamma(E)$ which lower bounds F, or, equivalently, the pointwise supremum of all affine functions that lower bound F. Note that an affine function $\mathcal{A}$ with slope $\varphi \in E^\star$ and constant term $\alpha \in \mathbb{R}$ satisfies $\mathcal{A} \leq F$ if and only if $\alpha \leq -F^\star(\varphi)$, where

$$F^\star(\varphi) = \sup_{u \in \mathrm{dom} F} \langle \varphi, u \rangle - F(u)\,, \tag{3.41}$$

and $\mathrm{dom} F = \{u \in E,\ F(u) < +\infty\}$. The function $F^\star : E^\star \to \overline{\mathbb{R}}$ is called the Legendre-Fenchel transform of F (or the polar, or the conjugate of F). It is an element of $\Gamma(E^\star)$, as it can be seen as the pointwise supremum over the dual space $E^\star$ of all affine functions $\{\mathcal{A}_u\}_{u \in \mathrm{dom} F}$ defined by

$$\forall u \in \mathrm{dom} F, \quad \mathcal{A}_u : \varphi \mapsto \langle \varphi, u \rangle - F(u)\ .$$

Since here E has finite dimension, it is a reflexive space and the Legendre-Fenchel transform of $F^\star$ (noted $F^{\star\star}$) is an element of $\Gamma(E^{\star\star})$ (and thus an element of $\Gamma(E)$), which happens to be exactly the Γ-regularization of F. In particular $F^{\star\star} \leq F$ and we have the characterization

$$F \in \Gamma(E) \Leftrightarrow F^{\star\star} = F\ , \tag{3.42}$$

which is very useful to derive a primal-dual reformulation of an optimization problem when the cost function decomposes as a sum with at least one term in $\Gamma(E)$. Besides, since E (endowed with the Euclidean inner product) is a Hilbert space, it is self-dual in the sense that any element of $E^\star$ can be represented as the inner product with an element of E, which is very useful in practical computations.

3.4.2 Chambolle-Pock Algorithm

The recent use in imaging of those powerful convex analysis tools based on duality allowed to properly handle total variation-based variational problems (see e.g. [Chambolle 2004, Zhu and Chan 2008]). This initiated some flourishing theoretical research (see e.g. [Aujol and Chambolle 2005, Fadili and Peyré 2011]) as well as the development of efficient numerical schemes [Chambolle and Pock 2011, Combettes and Wajs 2005, Beck and Teboulle 2009a, Weiss et al. 2009, Ochs et al. 2014, Drori et al. 2015, Raguet et al. 2013] dedicated to nonsmooth optimization. We will here briefly recall the formulation of the celebrated first

order primal-dual algorithm of Chambolle and Pock [Chambolle and Pock 2011], which can be used to address various total variation based image processing tasks and comes with nice convergence theorems.

Consider X and Y two finite-dimensional real vector spaces, an inner product $\langle \cdot, \cdot \rangle$ over Y and the generic saddle-point problem

$$\min_{x \in X} \max_{y \in Y} \; G(x) + \langle Kx, y \rangle - F^\star(y) \; , \tag{3.43}$$

where $F \in \Gamma_0(Y)$, $G \in \Gamma_0(X)$ and $K : X \to Y$ denotes a linear operator. We set $H : (x, y) \mapsto G(x) + \langle Kx, y \rangle - F^\star(y)$ and we assume that problem (3.43) has at least one solution (i.e. a saddle-point of H). Recall that thanks to (3.42), for any $x \in X$ we have

$$F(Kx) = F^{\star\star}(Kx) = \sup_{y \in Y} \langle Kx, y \rangle - F^\star(y) \, , \tag{3.44}$$

therefore one can interpret Equation (3.43) as a primal-dual formulation of the primal problem

$$\min_{x \in X} \; G(x) + F(Kx) \tag{3.45}$$

as soon as the $\sup_{y \in Y}$ is indeed a maximum (which will be the case in practice). The proximal splitting algorithm proposed by Chambolle and Pock in [Chambolle and Pock 2011] (see also [Moreau 1965, Rockafellar 1970], or more recently [Parikh and Boyd 2013, Combettes and Pesquet 2011] for more details about proximity operators and proximal algorithms) for solving problem (3.43) is described in Algorithm 5 below.

In the case $\theta = 0$, one iteration k of Algorithm 5 consists in a proximal ascent of $y \mapsto H(x^k, y)$ followed by a proximal descent of $x \mapsto H(x, y^{k+1})$, yielding a semi-implicit variant of the classical Arrow-Hurwicz algorithm [Arrow et al. 1958]. In the case $\theta > 0$, the iterate $\overline{x}^{k+1} = x^{k+1} + \theta \left(x^{k+1} - x^k \right)$ represents a linear approximation (or extrapolation) of the next iterate x^{k+2} based on the current and the previous iterates x^{k+1} and x^k; it is used to make the scheme more implicit and prove the convergence (in the case $\theta = 1$ and $\tau\sigma |||K|||^2 < 1$) of the sequence $(x^k, y^k)_{k \geq 0}$ towards a saddle-point of H, with an estimate of the convergence rate in $\mathscr{O}(1/N)$ after N iterations (see Theorem 1 in [Chambolle and Pock 2011]). Notice that some accelerated variants of this algorithm were also proposed by the same authors, which under regularity assumptions on $F^\star$ and G achieve better convergence rates, thanks to Nesterov-like acceleration strategies [Nesterov 1983] (see Algorithms 2 and 3 in [Chambolle and Pock 2011]).

Algorithm 5: Chambolle-Pock resolvant algorithm for problem (3.43)

Initialization: *Choose* $\tau, \sigma > 0$, $\theta \in [0,1]$, $x^0 \in X$, $y^0 \in Y$, *and set* $\overline{x}^0 = x^0$ *(note: for* $\theta = 1$, *convergence towards a solution of* (3.43) *was proven in [Chambolle and Pock 2011] when* $\tau\sigma|||K|||^2 < 1$*).*

Iterations: *For* $k \geq 0$, *update* x^k, y^k *and* $\overline{x}^k$ *as follows,*

$$\begin{aligned} y^{k+1} &= \operatorname*{argmin}_{y \in Y} \ \tfrac{1}{2\sigma} \left\| y - (y^k + \sigma K \overline{x}^k) \right\|_2^2 + F^\star(y) \\ x^{k+1} &= \operatorname*{argmin}_{x \in X} \ \tfrac{1}{2\tau} \left\| x - \left(x^k - \tau K^* y^{k+1}\right) \right\|_2^2 + G(x) \\ \overline{x}^{k+1} &= x^{k+1} + \theta \left(x^{k+1} - x^k\right) \end{aligned}$$

3.4.3 Dual formulation of the Shannon total variation

The STV_n operator defined in (3.35) can be rewritten under the form $\mathrm{STV}_n(u) = \frac{1}{n^2}\|\nabla_n u\|_{1,2}$, noting $\|\cdot\|_{1,2}$ the norm over the space $\mathbb{R}^{\Omega_n} \times \mathbb{R}^{\Omega_n}$ defined by

$$\forall g \in \mathbb{R}^{\Omega_n} \times \mathbb{R}^{\Omega_n}, \quad \|g\|_{1,2} = \sum_{(x,y) \in \Omega_n} |g(x,y)| \,.$$

One easily checks that the dual norm of $\|\cdot\|_{1,2}$ is the norm $\|\cdot\|_{\infty,2}$ defined by

$$\forall p \in \mathbb{R}^{\Omega_n} \times \mathbb{R}^{\Omega_n}, \ \|p\|_{\infty,2} = \max_{(x,y) \in \Omega_n} |p(x,y)| \,.$$

Consequently (see e.g. [Boyd and Vandenberghe 2004]), the Legendre-Fenchel transform of $\|\cdot\|_{1,2}$, noted $\|\cdot\|_{1,2}^\star$, is the indicator function of the closed unit ball for the norm $\|\cdot\|_{\infty,2}$, defined by

$$\forall p \in \mathbb{R}^{\Omega_n} \times \mathbb{R}^{\Omega_n}, \ \delta_{\|\cdot\|_{\infty,2} \leq 1}(p) = \begin{cases} 0 & \text{if } \|p\|_{\infty,2} \leq 1 \,, \\ +\infty & \text{otherwise.} \end{cases}$$

We will now use the duality tools described in Section 3.4.1 to derive a dual formulation of the STV_n operator.

Proposition 29 (dual formulation of STV_n). *For any integer* $n \geq 1$ *and for any image* $u \in \mathbb{R}^\Omega$,

$$\mathrm{STV}_n(u) = \max_{p \in \mathbb{R}^{\Omega_n} \times \mathbb{R}^{\Omega_n}} \langle \tfrac{1}{n^2} \nabla_n u, p \rangle - \delta_{\|\cdot\|_{\infty,2} \leq 1}(p) \,.$$

Proof. Since $\|\cdot\|_{1,2}$ is convex and l.s.c. over $\mathbb{R}^{\Omega_n} \times \mathbb{R}^{\Omega_n}$, it is an element of $\Gamma(\mathbb{R}^{\Omega_n} \times \mathbb{R}^{\Omega_n})$, thereby $\|\cdot\|_{1,2} = \|\cdot\|_{1,2}^{\star\star}$ thanks to (3.42). Besides, given any image $u \in \mathbb{R}^{\Omega}$, one has $\mathrm{STV}_n(u) = \frac{1}{n^2}\|\nabla_n u\|_{1,2} = \|\frac{1}{n^2}\nabla_n u\|_{1,2}$. Therefore, $\mathrm{STV}_n(u) = \left\|\frac{1}{n^2}\nabla_n u\right\|_{1,2}^{\star\star}$, i.e.

$$\mathrm{STV}_n(u) = \sup_{p \in \mathbb{R}^{\Omega_n} \times \mathbb{R}^{\Omega_n}} \langle \tfrac{1}{n^2}\nabla_n u, p\rangle_{\mathbb{R}^{\Omega_n} \times \mathbb{R}^{\Omega_n}} - \|p\|_{1,2}^{\star} \,,$$

and $\|p\|_{1,2}^{\star}$ is exactly $\delta_{\|\cdot\|_{\infty,2} \leq 1}(p)$. Last, one can see that the supremum is attained, since it is nothing but the maximum of the inner product term over the closed unit ball for the dual norm $\|\cdot\|_{\infty,2}$. □

3.4.4 The Huber STV

The use of TV^{d} as a regularizer for image processing applications has a well-known drawback, the so-called *staircasing effect*, which is the creation of piecewise constant regions with artificial boundaries where one would have expected smooth intensity variations (see for instance [Nikolova 2000, Chan et al. 2000, Ring 2000] for theoretical results about the staircasing). Several variants of TV^{d} have been proposed in order to avoid this undesirable effect (see for instance [Bredies et al. 2010, Louchet and Moisan 2013, 2014]). In the numerical experiments that will be presented in Section 3.5, we observed that although this staircasing effect is significantly attenuated when using the STV_n variant of TV^{d}, it remains present (at least visually) in the processed images.

In the case of TV^{d}, a classical way to get rid of the staircasing effect consists in replacing the ℓ^2 norm $|\cdot|$ of the gradient in the definition of the TV operator by its smooth Huber approximation with parameter $\alpha > 0$ (coming from the statistical literature [Huber 1964, 1973], and used for instance in [Weiss and Blanc-Féraud 2009, Werlberger et al. 2009, Chambolle and Pock 2011]). It is defined by

$$\forall y \in \mathbb{R}^2, \quad \mathcal{H}_\alpha(y) = \begin{cases} \frac{|y|^2}{2\alpha} & \text{if } |y| \leq \alpha \,, \\ |y| - \frac{\alpha}{2} & \text{otherwise} \,. \end{cases} \tag{3.47}$$

The same adaptation can be easily done in the case of STV by replacing the ℓ^2 norm by the Huber-function $\mathcal{H}_\alpha$ in Equations (3.34) and (3.35), which in the case of the Riemann approximation leads to

$$\mathrm{HSTV}_{\alpha,n}(u) = \frac{1}{n^2} \sum_{(x,y) \in \Omega_n} \mathcal{H}_\alpha\left(\nabla_n u(x,y)\right) \,, \tag{3.48}$$

for any image $u \in \mathbb{R}^{\Omega}$. Next Proposition establishes a dual reformulation of (3.48).

Proposition 30 (dual formulation of $\mathrm{HSTV}_{\alpha,n}$). *Let $\alpha > 0$ and $n \geq 1$. For any image $u \in \mathbb{R}^{\Omega}$, one has*

$$\mathrm{HSTV}_{\alpha,n}(u) = \max_{p \in \mathbb{R}^{\Omega_n} \times \mathbb{R}^{\Omega_n}} \langle \tfrac{1}{n^2} \nabla_n u, p \rangle - \delta_{\|\cdot\|_{\infty,2} \leq 1}(p) - \tfrac{\alpha}{2n^2} \|p\|_2^2 \, .$$

The Proof is given in Appendix 3.8.E. In the following, we shall use the dual formulations of STV_n and $\mathrm{HSTV}_{\alpha,n}$ provided by Propositions 29 and 30 in order to reformulate many optimization problems frequently considered in image restoration in their primal-dual form (3.43).

3.5 Image processing applications

In this section, we illustrate the interest of STV in the case of several TV-based image processing applications. As we shall see, replacing the classical discrete TV by STV does not raise any theoretical nor numerical difficulty, and brings clear improvements regarding subpixellic scales.

3.5.1 Image denoising

The STV variant of the denoising model (3.1) proposed by Rudin, Osher and Fatemi (ROF) in [Rudin et al. 1992] writes

$$\operatorname*{argmin}_{u \in \mathbb{R}^{\Omega}} \|u - u_0\|_2^2 + \lambda \, \mathrm{STV}_n(u) \, , \tag{3.49}$$

where $u_0 \in \mathbb{R}^{\Omega}$ denotes the observed image with (discrete) domain Ω, and $\lambda \geq 0$ is the so-called regularity parameter that controls the trade-off between the data-fidelity term (the square ℓ^2 distance to u_0) and the regularity term $\mathrm{STV}_n(u)$ in the minimization process. Using Proposition 29, we immediately get a primal-dual reformulation of (3.49),

$$\operatorname*{argmin}_{u \in \mathbb{R}^{\Omega}} \max_{p \in \mathbb{R}^{\Omega_n} \times \mathbb{R}^{\Omega_n}} \|u - u_0\|_2^2 + \langle \tfrac{\lambda}{n^2} \nabla_n u, p \rangle - \delta_{\|\cdot\|_{\infty,2} \leq 1}(p) \, , \tag{3.50}$$

which has exactly the form of (3.43) with $(x, y) = (u, p)$, $G(u) = \|u - u_0\|_2^2$, $K = \frac{\lambda}{n^2} \nabla_n$ (with adjoint $K^* = -\frac{\lambda}{n^2} \mathrm{div}_n$), and $F^\star(p) = \delta_{\|\cdot\|_{\infty,2} \leq 1}(p)$.

Notice that replacing $\mathrm{STV}_n(u)$ by $\mathrm{HSTV}_{\alpha,n}(u)$ into (3.49) leads to the Huber STV_n variant of ROF. In view of Proposition 30, it amounts to replace the term $F^\star(p) = \delta_{\|\cdot\|_{\infty,2}\leq 1}(p)$ by $F^\star(p) = \delta_{\|\cdot\|_{\infty,2}\leq 1}(p) + \frac{\lambda\alpha}{2n^2}\|p\|_2^2$ into the primal-dual problem (3.50).

For both STV_n and $\mathrm{HSTV}_{\alpha,n}$ regularizers, the corresponding primal-dual problem can be numerically solved by specializing Algorithm 5, which yields Algorithm 6 below. Notice that (3.39) yields the upper bound $|||K||| \leq \frac{\lambda\pi\sqrt{2}}{n}$, which is useful to set the parameters τ and σ of the algorithm. The images resulting from the different (discrete or Shannon, Huber or usual) TV-based image denoising models are compared in Figure 3.4 and 3.5: we illustrate in Figure 3.4 the improved behavior of STV over the classical discrete TV regarding posterior interpolation, and do the same in Figure 3.5 for the Huber variant.

Algorithm 6: Chambolle-Pock resolvant Algorithm for Problem (3.49)

Initialization: *Choose* $\tau, \sigma > 0$, $\theta \in [0,1]$, $u^0 \in \mathbb{R}^\Omega$, $p^0 \in \mathbb{R}^{\Omega_n} \times \mathbb{R}^{\Omega_n}$, *set* $\overline{u}^0 = u^0$ *and set* $\nu = 1$ *when using the* STV_n *regularizer and* $\nu = 1 + \sigma\frac{\alpha\lambda}{n^2}$ *when using the* $\mathrm{HSTV}_{\alpha,n}$ *regularizer. Denote by* $\pi_{\infty,2}$ *the* ℓ^2 *projection on the closed unit ball for the norm* $\|\cdot\|_{\infty,2}$ *in* $\mathbb{R}^{\Omega_n} \times \mathbb{R}^{\Omega_n}$, *which is defined by*

$$\forall (x,y) \in \Omega_n, \quad \pi_{\infty,2}(p)(x,y) = \frac{p(x,y)}{\max\left(1, |p(x,y)|\right)},$$

for any $p \in \mathbb{R}^{\Omega_n} \times \mathbb{R}^{\Omega_n}$.

Iterations: *For* $k \geq 0$, *update* p^k, u^k *and* $\overline{u}^k$ *with*

$$p^{k+1} = \pi_{\infty,2}\left(\left(p^k + \tfrac{\sigma\lambda}{n^2}\nabla_n \overline{u}^k\right)/\nu\right) \tag{3.51a}$$

$$u^{k+1} = \frac{u^k + \frac{\tau\lambda}{n^2}\mathrm{div}_n p^{k+1} + 2\tau u_0}{1+2\tau} \tag{3.51b}$$

$$\overline{u}^{k+1} = u^{k+1} + \theta\left(u^{k+1} - u^k\right) \tag{3.51c}$$

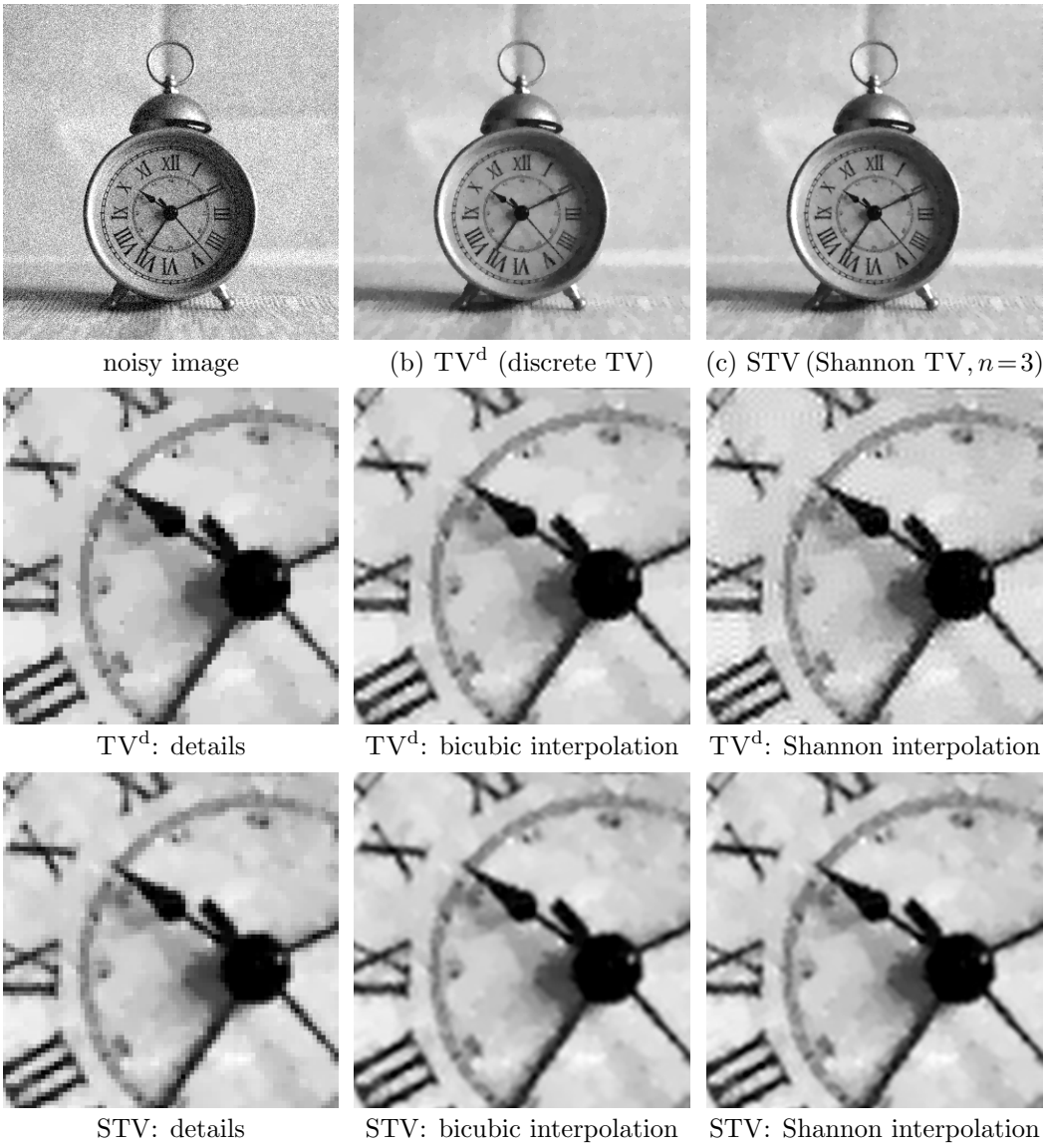


Figure 3.4: Comparison of discrete TV and Shannon TV for image denoising. A noisy image (top, left) undergoing additive white Gaussian noise with zero mean and standard deviation $\sigma = 20$ (see also the reference image in Figure 3.5) was processed with the ROF model using the TV^{d} (top, center) and STV_3 (top, right) discretizations. The regularity parameter λ was set in order to get the same norm of the estimated noise (the difference between the noisy and the restored image) in each simulation. In the second row we display a cropping of the TV^{d}-restored image oversampled with factor 3 using different interpolation methods (from left to right: nearest neighbor, bicubic spline and Shannon interpolation). In the third row, the same operation is realized on the STV-restored image. We can see that images TV^{d} and STV images look globally similar. The details on the left of rows 2 and 3 reveal the presence of staircasing in both cases, but this artifact is significantly attenuated in the case of STV. Looking at the second row, we see that the TV^{d} image cannot be interpolated in a satisfying way, since both bicubic and Shannon interpolation methods yield images with undesirables oscillations (*ringing*) localized near objects contours. This is not the case with the STV image, that can be interpolated without creating artifacts with both bicubic and Shannon interpolations (row 3).

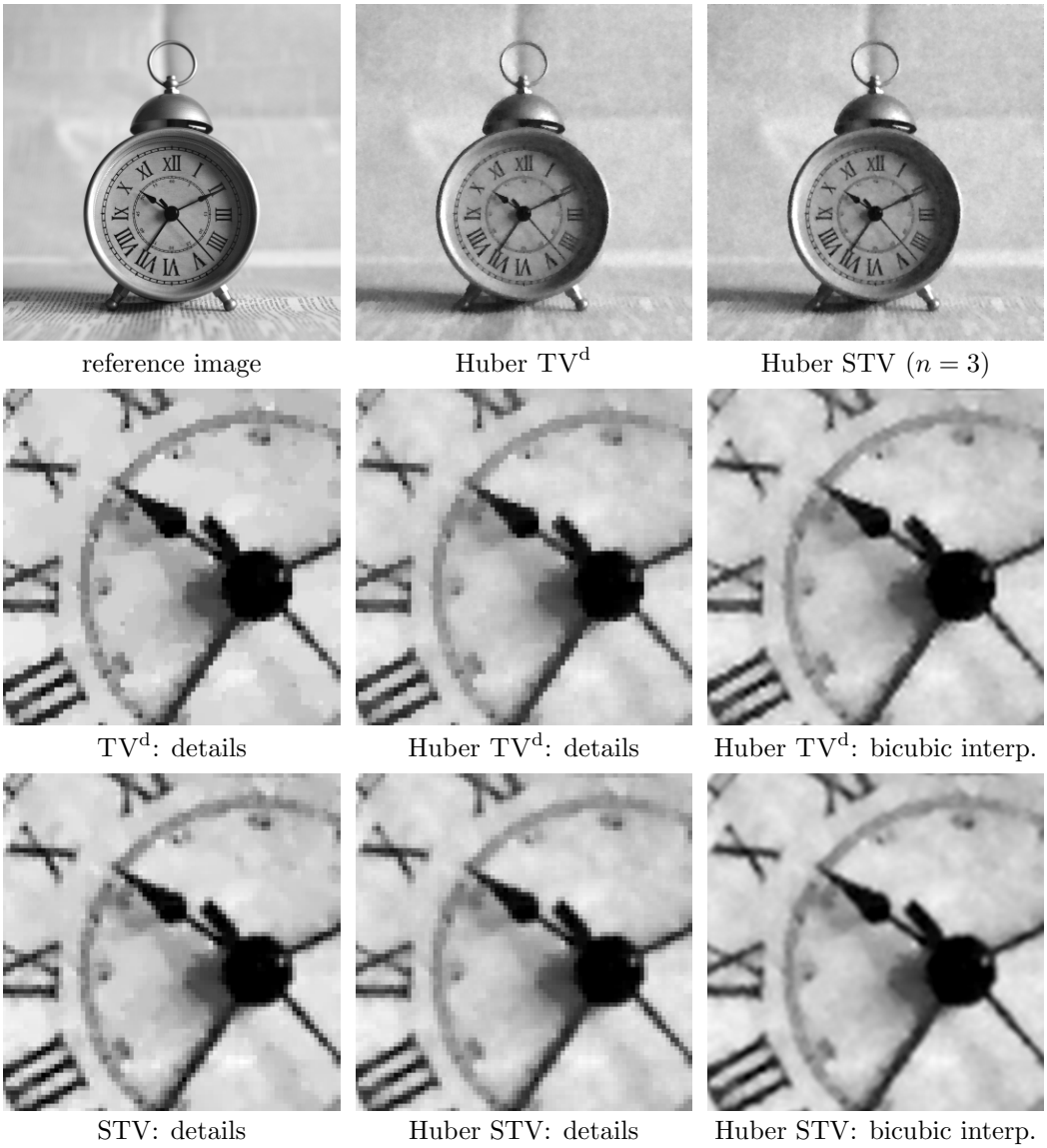


Figure 3.5: Image denoising with Huber-TV and Huber-STV. This experiment is similar to Figure 3.4, except that we here consider the Huber variant (with $\alpha = 5$) of ROF denoising, both for the TV^{d} and STV discretizations. As expected, the Huber variant avoids the staircasing effect for both discretizations (TV^{d} and STV). However, it does not solve the interpolability issue for TV^{d}: the bicubic interpolation (interp.) of Huber TV^{d} presents several ringing artifacts (like the non-Huber TV^{d} displayed in Figure 3.4), and these artifacts are again completely avoided by considering the STV discretization.

3.5.2 Inverse problems

Let us now consider the more general case of a linear inverse problem addressed with quadratic data fidelity and STV regularization. It writes

$$\tilde{u} \in \operatorname*{argmin}_{u \in \mathbb{R}^\Omega} \|Au - u_0\|_2^2 + \lambda \, \mathrm{STV}_n(u) \,, \tag{3.52}$$

where $u_0 \in \mathbb{R}^\omega$ denotes the observed image (ω being a finite subset of $\mathbb{Z}^2$, possibly $\omega = \Omega$) and $A : \mathbb{R}^\Omega \to \mathbb{R}^\omega$ is a linear operator which may model the convolution with the impulse response of an acquisition device (defocus or motion blur for instance) or other linear observation mechanisms such as tomography, downsampling, loss of image regions, etc.

Proposition 31 (primal-dual formulation of (3.52)). *Any solution $\tilde{u}$ of Problem (3.52) satisfies*

$$\tilde{u} \in \operatorname*{argmin}_{u \in \mathbb{R}^\Omega} \max_{\substack{p \in \mathbb{R}^{\Omega_n} \times \mathbb{R}^{\Omega_n} \\ q \in \mathbb{R}^\omega}} G(u) + \langle Ku, (p,q) \rangle - F^\star(p,q) \,,$$

where $G(u) = 0$, $F^\star(p,q) = \delta_{\|\cdot\|_{\infty,2} \leq 1}(p) + \|\frac{q}{2} + u_0\|_2^2$ and $K : \mathbb{R}^\Omega \to \left(\mathbb{R}^{\Omega_n} \times \mathbb{R}^{\Omega_n}\right) \times \mathbb{R}^\omega$ is the linear operator defined by $Ku = \left(\frac{\lambda}{n^2} \nabla_n u, Au\right)$ for any $u \in \mathbb{R}^\Omega$.

Proof. Writing $f(v) = \|v - u_0\|_2^2$, one easily gets the expression of the Legendre-Fenchel transform of f, that is $f^\star(q) = \|\frac{q}{2} + u_0\|_2^2 - \|u_0\|_2^2$. Besides, since $f \in \Gamma_0(\mathbb{R}^\omega)$, we have

$$\begin{aligned} \|Au - u_0\|_2^2 &= f(Au) = f^{\star\star}(Au) \\ &= \sup_{q \in \mathbb{R}^\omega} \langle Au, q \rangle - \|\tfrac{q}{2} + u_0\|_2^2 + \|u_0\|_2^2 \,, \end{aligned} \tag{3.53}$$

and the supremum is attained since the cost functional is concave, differentiable, and its gradient vanishes at the point $q = 2(Au - u_0)$. Replacing the quadratic term accordingly into (3.52), removing the constant $\|u_0\|_2^2$ (which does not change the set of minimizers), and replacing as well the STV_n term by its dual formulation using Proposition 29, we obtain the desired result. □

Again, the Huber version of (3.52) is obtained by replacing the $\mathrm{STV}_n(u)$ term by $\mathrm{HSTV}_{\alpha,n}(u)$, which simply changes $F^\star(p,q) = \delta_{\|\cdot\|_{\infty,2} \leq 1}(p) + \|\frac{q}{2} + u_0\|_2^2$ into $F^\star(p,q) = \delta_{\|\cdot\|_{\infty,2} \leq 1}(p) + \frac{\alpha\lambda}{2n^2}\|p\|_2^2 + \|\frac{q}{2} + u_0\|_2^2$.

Note that the adjoint of K (defined in Proposition 31) is $K^*(p,q) = -\frac{\lambda}{n^2}\mathrm{div}_n p + A^*q$, and its induced ℓ^2 norm satisfies

$$|||K|||^2 \leq |||\tfrac{\lambda}{n^2}\nabla_n|||^2 + |||A|||^2 \leq 2\left(\tfrac{\pi\lambda}{n}\right)^2 + |||A|||^2 .$$

Thus, Chambolle-Pock Algorithm can be rewritten in the present case as Algorithm 7 below. The update of the dual variable (here the tuple (p,q)) in the generic Algorithm 5 was split into two independent updates thanks to the additive separability with respect to p and q of the function $(p,q) \mapsto \langle Ku, (p,q)\rangle - F^\star(p,q)$.

Algorithm 7: Chambolle-Pock resolvant Algorithm for Problem (3.52)

Initialization: *Choose* $\tau, \sigma > 0$, $\theta \in [0,1]$, $u^0 \in \mathbb{R}^\Omega$, $p^0 \in \mathbb{R}^{\Omega_n} \times \mathbb{R}^{\Omega_n}$, $q^0 \in \mathbb{R}^\omega$, *set* $\overline{u}^0 = u^0$ *and set* ν *as in Algorithm 6.*

Iterations: *For* $k \geq 0$, *update* p^k, u^k *and* $\overline{u}^k$ *with*

$$\begin{aligned}
p^{k+1} &= \pi_{\infty,2}\left(\left(p^k + \tfrac{\sigma\lambda}{n^2}\nabla_n \overline{u}^k\right)/\nu\right) \\
q^{k+1} &= \frac{2\,q^k + 2\sigma\left(A\overline{u}^k - u_0\right)}{2+\sigma} \\
u^{k+1} &= u^k + \tfrac{\tau\lambda}{n^2}\mathrm{div}_n p^{k+1} - \tau A^* q^{k+1} \\
\overline{u}^{k+1} &= u^{k+1} + \theta\left(u^{k+1} - u^k\right)
\end{aligned}$$

Application to image deconvolution

In the case of image deconvolution, the linear operator A in (3.52) is the convolution with a point spread function k_A (modeling for instance some blurring phenomenon such as diffraction, defocus, motion blur, ...). Let us consider such a discrete convolution kernel $k_A \in \mathbb{R}^{\omega_A}$ with finite domain $\omega_A \subset \mathbb{Z}^2$, and define the associated operator $A : \mathbb{R}^\Omega \to \mathbb{R}^\omega$ by

$$Au(x,y) = \sum_{(a,b)\in\omega_A} k_A(a,b)\, u(x-a, y-b)\ , \tag{3.55}$$

where ω denotes the subset of Ω made of all the pixels $(x,y) \in \Omega$ such as $(x,y) - \omega_A \subset \Omega$. In order to use Algorithm 7, we need the explicit expression of A^* :

$\mathbb{R}^\omega \to \mathbb{R}^\Omega$, which writes

$$A^*v(x,y) = \sum_{(a,b)\in\omega_A} k_A(a,b)\, v(x+a, y+b)\ , \tag{3.56}$$

for $v \in \mathbb{R}^\omega$ and $(x,y) \in \Omega$, with the convention that $v(x+a, y+b) = 0$ when $(x+a, y+b) \notin \omega$. One easily checks that $|||A||| \leq ||k_A||_1$ as well.

Most authors define the convolution with kernel k_A as an operator $A : \mathbb{R}^\Omega \to \mathbb{R}^\Omega$ at the cost of an extension of u outside of Ω, usually a periodic or a mirroring condition, or a zero-extension. Such a convention simplifies the analysis (and the computations, especially in the periodic case where the convolution can be implemented with the DFT), but we shall not use it here as it is unrealistic and thus of little help to process real data. Experiments illustrating STV deblurring are displayed in Figure 3.6 (motion blur) and 3.7 (out of focus).

Application to image zooming and inpainting

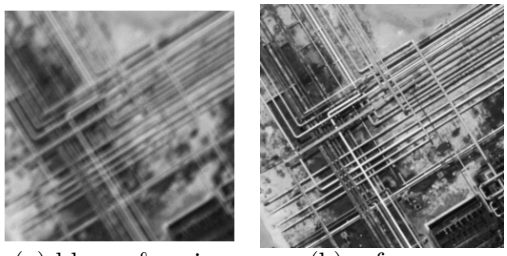
The variational formulation (3.52) can be used to perform many other image processing tasks: as soon as we can derive a closed-form expression for A, its adjoint A^*, and estimate an upper bound for $|||A|||$, Algorithm (7) can be implemented without difficulty. We here mention two more examples of applications (zoom and inpainting), each corresponding to a particular choice of A. We experimentally checked that, in both cases, the use of STV_n instead of TV yields nicely interpolable images.

In the case of image zooming, the operator A is often assumed to be a blurring kernel followed by a subsampling procedure (see [Malgouyres and Guichard 2001, Chambolle and Pock 2011]). A simple particular case is the discrete captor integration model $A : \mathbb{R}^\Omega \to \mathbb{R}^\omega$ defined by

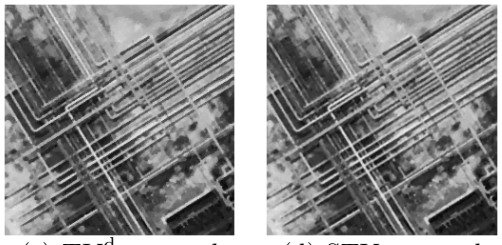
$$Au(x,y) = \frac{1}{\delta^2} \sum_{(a,b)\in I_\delta^2} u(\delta x + a, \delta y + b)\ , \tag{3.57}$$

where $\omega = I_M \times I_N$ denotes a small discrete domain and $\Omega = I_{\delta M} \times I_{\delta N}$ a bigger one, δ (the magnification factor) being an integer at least equal to 2. In that case, we easily obtain the relation $|||A||| = \frac{1}{\delta}$ and the expression of the adjoint operator $A^* : \mathbb{R}^\omega \to \mathbb{R}^\Omega$ as

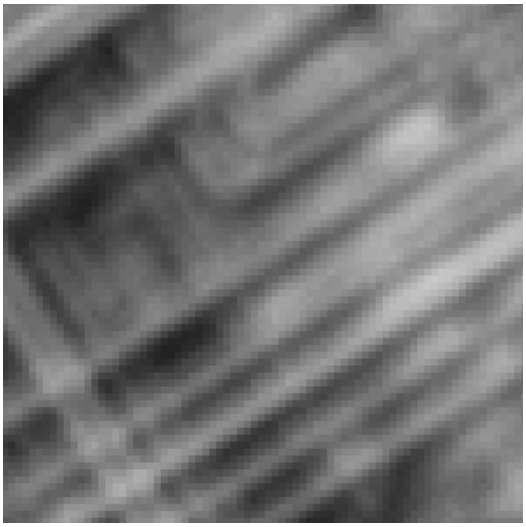
$$A^*v(x,y) = \frac{1}{\delta^2}\, v\left(\lfloor \tfrac{x}{\delta} \rfloor, \lfloor \tfrac{y}{\delta} \rfloor\right)\ . \tag{3.58}$$



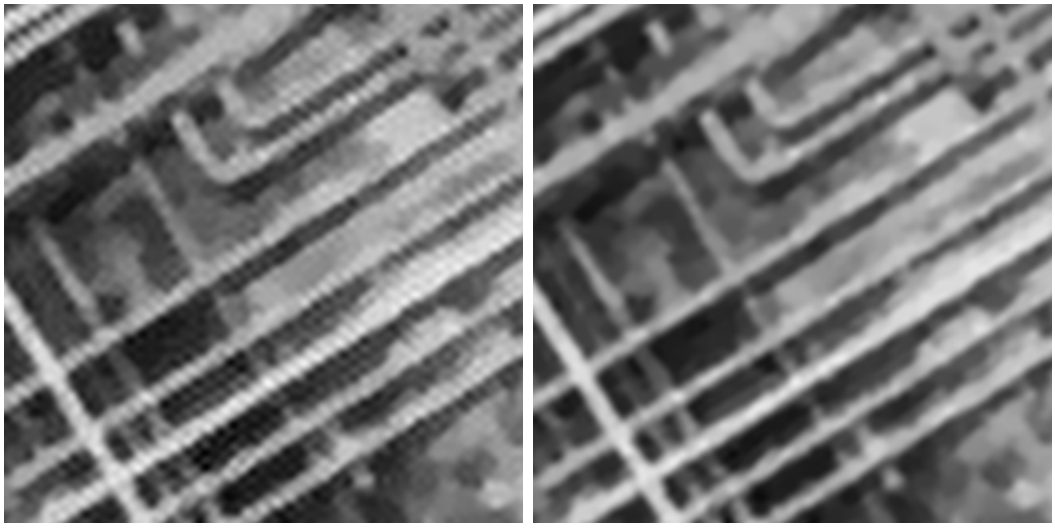
(a) blurry & noisy (b) reference



(c) TV^{d}-restored (d) STV-restored

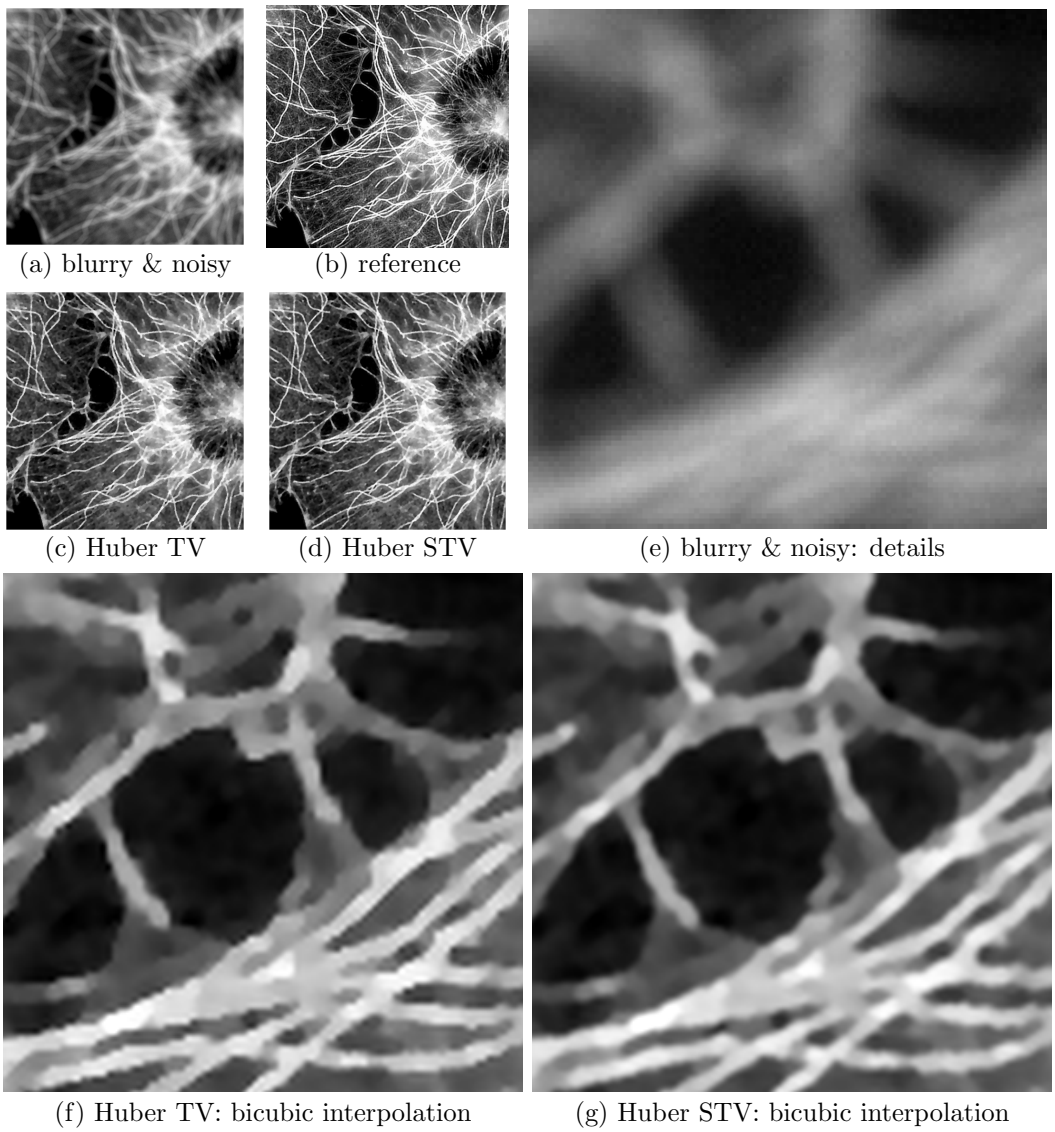


(e) blurry & noisy: details



(f) TV^{d}-restored: Shannon interpolation (g) STV-restored: Shannon interpolation

Figure 3.6: Motion deblurring with discrete TV and Shannon TV. A degraded (blurry and noisy) image (a) is synthesized by convolving the reference image (b) with a real-data motion blur kernel and then adding a white Gaussian noise with zero-mean and standard deviation $\sigma = 2$. The degraded image (a) is then processed by solving the corresponding TV^{d} and STV_3 regularized inverse problems (Equation (3.52)). As in Figure 3.4, the regularization parameter λ is set in such a way that the amount of estimated noise (here the quantity $\|A\tilde{u} - u_0\|_2$, where $\tilde{u}$ is the restored image) is the same for both methods. The resulting images (c) and (d) are quite similar, but the magnified views (f) and (g) (magnification of factor 4 with Shannon interpolation) clearly shows that they strongly differ in terms of interpolability: as in the denoising case, the interpolated TV^{d} image exhibits strong ringing artifacts, whereas the interpolated STV image does not.



(a) blurry & noisy (b) reference (c) Huber TV (d) Huber STV (e) blurry & noisy: details (f) Huber TV: bicubic interpolation (g) Huber STV: bicubic interpolation

Figure 3.7: Out-of-focus deblurring using Huber TV and Huber STV. This experiment is similar to Figure 3.6, except that we used a fluorescence microscopy image of actin filaments and microtubules in interphase cells (source cellimagelibrary.org, CIL number 240, first channel), a synthetic out-of-focus blur kernel defined by the indicator of a disk with radius 7 pixels, and we replaced the TV^{d} and STV_3 regularizers by their Huber versions ($\alpha = 5$). The conclusions are identical.

Another example is image inpainting, which aims at estimating plausible image intensities in a (nonempty) subpart ω_0 of the image domain Ω where the information is missing. In that case, $\omega = \Omega$, the operator $A : \mathbb{R}^\Omega \to \mathbb{R}^\Omega$ is defined by

$$Au(x, y) = \mathbb{1}_{\omega_0}(x, y) \cdot u(x, y) \,,$$

and one easily checks that $A^* = A$ (A is a diagonal operator) and $|||A||| = 1$.

3.5.3 Constrained minimization

In some situations, it is desirable to consider constrained minimization problems of the type

$$\tilde{u} \in \operatorname*{argmin}_{u \in \mathbb{R}^\Omega} \mathrm{STV}_n(u) \quad \text{subject to} \quad Au = u_0 \,, \tag{3.59}$$

where u_0 denotes the observed image with discrete domain ω, $\tilde{u}$ denotes the reconstructed image with discrete domain Ω, and A denotes again a linear operator from $\mathbb{R}^\Omega$ to $\mathbb{R}^\omega$. In other words, we are interested in the computation of an image $\tilde{u}$ having the smallest Shannon TV among those satisfying the constraint $Au = u_0$. Remark that the inverse problem (3.52) is none other than a relaxed version of (3.59). In the presence of noise, it is better to use the relaxed formulation, but the constrained model (3.59) may be interesting when the level of noise in u_0 is low, especially because it does not require the setting of any regularization parameter λ.

Using Proposition 29, we obtain a primal-dual reformulation of (3.59),

$$\tilde{u} \in \operatorname*{argmin}_{u \in \mathbb{R}^\Omega} \max_{p \in \mathbb{R}^{\Omega_n} \times \mathbb{R}^{\Omega_n}} \delta_{A^{-1}(u_0)}(u) + \langle \tfrac{1}{n^2} \nabla_n u, p \rangle - \delta_{\|\cdot\|_{\infty,2} \leq 1}(p) \,, \tag{3.60}$$

where the (closed and convex) set

$$A^{-1}(u_0) := \big\{ u \in \mathbb{R}^\Omega, \ Au = u_0 \big\}$$

is assumed to be nonempty, and $\delta_{\mathcal{P}}$ denotes the indicator function of a set $\mathcal{P}$ (that is, $\delta_{\mathcal{P}}(p) = 0$ if $p \in \mathcal{P}$, $+\infty$ otherwise). A solution of Problem (3.60) can be numerically computed using Algorithm 8, taking $G = \delta_{A^{-1}(u_0)}$, $F^\star = \delta_{\|\cdot\|_{\infty,2} \leq 1}$ and $K = \frac{1}{n^2} \nabla_n$ in Chambolle-Pock Algorithm.

To illustrate the general framework above, we will consider in the next section the problem of reconstructing an image from partial measurements in the Fourier

Algorithm 8: Chambolle-Pock resolvant Algorithm for Problem (3.60)

Initialization: *Choose* $\tau, \sigma > 0$, $\theta \in [0,1]$, $u^0 \in \mathbb{R}^\Omega$, $p^0 \in \mathbb{R}^{\Omega_n} \times \mathbb{R}^{\Omega_n}$, *set* $\overline{u}^0 = u^0$ *and define* ν *and* $\pi_{\infty,2}$ *as in Algorithm 6. Denote by* π_0 *the* ℓ^2 *projection from* $\mathbb{R}^\Omega$ *onto the (closed and convex) set* $A^{-1}(u_0) = \{u \in \mathbb{R}^\Omega,\ Au = u_0\}$.

Iterations: *For* $k \geq 0$, *update* p^k, u^k *and* $\overline{u}^k$ *with*

$$\begin{aligned} p^{k+1} &= \pi_{\infty,2}\left(\left(p^k + \tfrac{\sigma}{n^2}\nabla_n \overline{u}^k\right)/\nu\right) \\ u^{k+1} &= \pi_0\left(u^k + \tfrac{\tau}{n^2}\mathrm{div}_n p^{k+1}\right) \\ \overline{u}^{k+1} &= u^{k+1} + \theta\left(u^{k+1} - u^k\right) \end{aligned}$$

domain. A particular case is image magnification (assuming that the original low-resolution image does not suffer from aliasing), which corresponds to the recovery of high-frequency components only, but other situations (like tomography) require spectrum interpolation in a more complicated domain. Note also that many other applications, such as image inpainting or image zooming presented in Section 3.5.2, can be easily handled as well with the constrained formulation (3.59).

Application to spectrum extrapolation

Given an image $u_0 \in \mathbb{R}^\Omega$ whose spectrum $\widehat{u}_0$ is known on a certain (symmetric) subdomain $\widehat{\omega}_0$ of $\widehat{\Omega}$, how to extend this spectrum to the whole spectral domain $\widehat{\Omega}$? The trivial *zero-padding* approach, which amounts to extending the spectrum with the constant zero, yields a very oscillatory image in general, in reason of the irregularity (missing Fourier coeffients) of the extrapolated spectrum. A more satisfying reconstruction can be obtained with a variational approach: among all possible spectrum extensions, choose the one that minimizes a given energy. This kind of approach was used by Rougé and Seghier [Rougé and Seghier 1995], who considered the Burg entropy, and by Guichard and Malgouyres [Guichard and Malgouyres 1998, Malgouyres and Guichard 2001], who used the discrete TV (but in a slightly different framework, since they take as input a subsampled image which suffers from aliasing). We here consider the energy STV_n; in a constrained formulation, this is a particular case of (3.59), since the frequency constraint ($\widehat{u}$ and $\widehat{u}_0$ are equal on $\widehat{\omega}_0$) can be enforced under the form $Au = u_0$ where $A = \mathcal{F}^{-1} \circ M_{\widehat{\omega}_0} \circ \mathcal{F}$ ($\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the direct and inverse discrete Fourier

transforms respectively, the operator $M_{\widehat{\omega}_0}$ denotes the pointwise multiplication of a element of $\mathbb{C}^{\widehat{\Omega}}$ with $\mathbb{1}_{\widehat{\omega}_0}$, and $\widehat{u}_0$ is implicitly set to zero outside $\widehat{\omega}_0$). Note that the ℓ^2 projection π_0 onto the set $A^{-1}(u_0)$ is simply obtained in the Fourier domain with

$$\forall u \in \mathbb{R}^{\Omega},\ \forall (\alpha,\beta) \in \widehat{\Omega}, \quad \widehat{\pi_0(u)}(\alpha,\beta) = \begin{cases} \widehat{u}_0(\alpha,\beta) & \text{if } (\alpha,\beta) \in \widehat{\omega}_0 \\ \widehat{u}(\alpha,\beta) & \text{otherwise}. \end{cases}$$

Some examples of spectrum extrapolations are proposed in Figure 3.8 and 3.9.

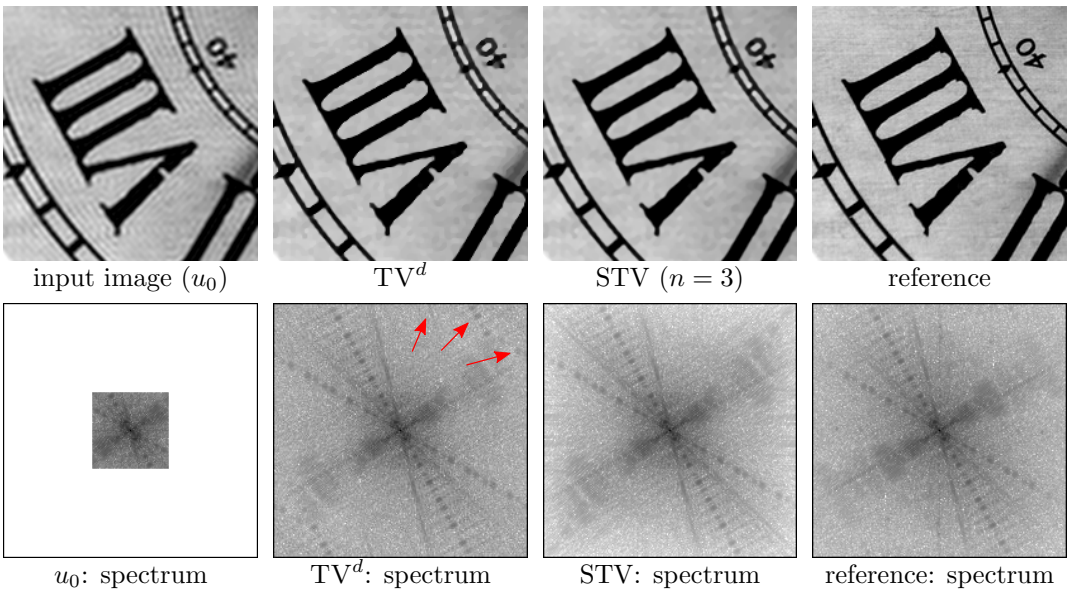


Figure 3.8: Image zooming with spectrum extrapolation. An input image (1st column) is synthesized by setting to 0 the high frequency components (that is, outside a square $\widehat{\omega}_0$) of a reference image (4th column). Spectrum extrapolation is then realized using either the discrete TV (2nd colum) or the STV (3rd column). For each image of the first row, the spectrum (Fourier modulus, in log scale) is displayed below on the second row. As we can observe, the constrained TV minimization framework (3.59) is efficient for spectrum extrapolation: both discretizations manage to reconstruct part of the missing high frequencies and remove the ringing patterns observed in the input image. However, STV is to be prefered to discrete TV as it manages to avoid the aliasing artifacts of the latter (red arrows), and delivers nicely interpolable images.

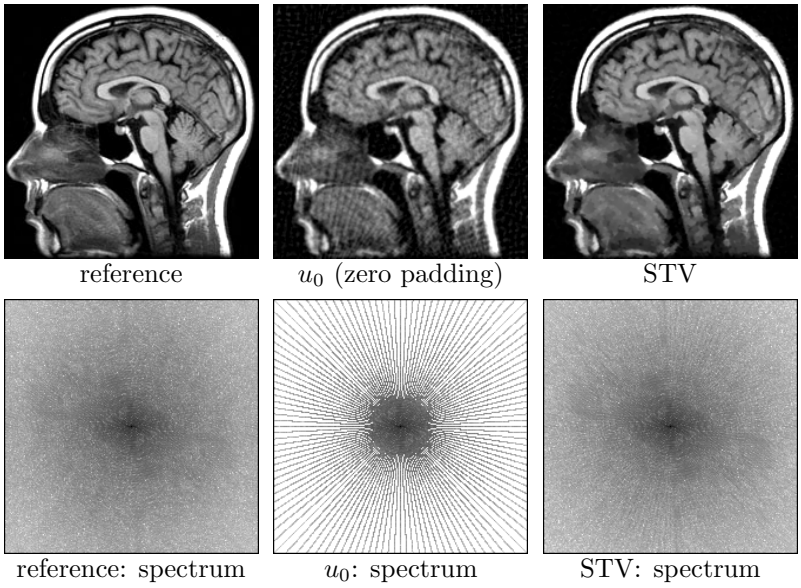


Figure 3.9: Image reconstruction from partial measurements in the Fourier domain. We here reproduce a simplified tomography inversion experiment: a reference image (1st column) is sampled in the Fourier domain along several discrete rays (covering around 35% of the whole frequency domain), and two image reconstruction methods are compared. The first one consists in setting the missing Fourier coefficients to 0 (2nd column), which produces severe ringing artifacts. Extrapolating the missing Fourier coefficients with the constrained STV minimization framework (3.59) yields a much nicer image (3rd column) which can be easily interpolated. As in Figure 3.8, the spectrum of each image of the first row is displayed on the second row.

3.6 Regularization with weighted frequencies

Using STV as a regularizer leads to iterative algorithms that operate in the Fourier domain. This has a non-negligible computational cost, even though this kind of algorithms is common nowadays and there exist very efficient implementations of the Fourier Transform, like FFTW [Frigo and Johnson 2005]. We now consider an image restoration model that benefits from the availability of the Fourier transform of the current image at each iteration.

3.6.1 Model

Given an input image $u_0 : \Omega \to \mathbb{R}$ (with $\Omega = I_M \times I_N$) and a symmetric non-negative map $\gamma : \widehat{\Omega} \to \mathbb{R}_+$, we consider the minimization problem

$$\operatorname*{argmin}_{u \in \mathbb{R}^\Omega} \|\widehat{u} - \widehat{u_0}\|_\gamma^2 + \lambda \, \mathrm{STV}_n(u) \,, \tag{3.62}$$

where $\lambda > 0$ is a regularization parameter and

$$\|\widehat{u} - \widehat{u_0}\|_\gamma^2 = \frac{1}{|\Omega|} \sum_{(\alpha,\beta)\in\widehat{\Omega}} \gamma(\alpha,\beta) \cdot |\widehat{u}(\alpha,\beta) - \widehat{u_0}(\alpha,\beta)|^2$$

is a weighted squared distance between u and u_0 (strictly speaking, it defines a distance only if γ does not vanish). Model (3.62) generalizes two other models considered above. Indeed, STV image denoising (3.49) is obtained with $\gamma \equiv 1$, while the choice $\gamma = \mathbb{1}_{\widehat{\omega_0}}$ leads to a relaxed version of spectrum extrapolation considered in Section 3.5.3. Choosing a more general (non-binary) weight map γ provides a way to selectively regularize the Fourier coefficients of the input image u_0: when $\gamma(\alpha,\beta)$ is large, one expects to obtain $\widehat{u}(\alpha,\beta) \approx \widehat{u_0}(\alpha,\beta)$; on the contrary, the coefficients $\widehat{u}(\alpha,\beta)$ corresponding to small (or zero) values of $\gamma(\alpha,\beta)$ are essentially driven by STV regularization.

3.6.2 Algorithm

Replacing the STV_n term by its dual formulation (Proposition 29) into (3.62) yields the primal-dual problem

$$\operatorname*{argmin}_{u\in\mathbb{R}^\Omega} \max_{p\in\mathbb{R}^{\Omega_n}\times\mathbb{R}^{\Omega_n}} \|\widehat{u} - \widehat{u_0}\|_\gamma^2 + \langle \tfrac{\lambda}{n^2}\nabla_n u, p\rangle - \delta_{\|\cdot\|_{\infty,2}\leq 1}(p)\,. \tag{3.63}$$

In order to apply Algorithm 5 to (3.63), one needs to perform at each iteration k the primal update

$$u^{k+1} = \operatorname*{argmin}_{u\in\mathbb{R}^\Omega} \tfrac{1}{2\tau}\left\|u - u^{k+1/2}\right\|_2^2 + \|\widehat{u} - \widehat{u_0}\|_\gamma^2\,, \tag{3.64}$$

where $u^{k+1/2} = u^k + \frac{\tau\lambda}{n^2}\mathrm{div}_n p^{k+1}$. Thanks to Parseval Identity, this can be rewritten

$$\widehat{u^{k+1}} = \operatorname*{argmin}_{u\in\mathbb{R}^\Omega} \tfrac{1}{2\tau|\Omega|}\left\|\widehat{u} - \widehat{u^{k+1/2}}\right\|_2^2 + \|\widehat{u} - \widehat{u_0}\|_\gamma^2\,, \tag{3.65}$$

from which we easily obtain the explicit formula for the update given in Algorithm 9.

3.6.3 Image Shannonization

One interesting application of Model (3.62) is its ability to (partly or fully) remove aliasing from a given image, thus providing what we could call an "Image

Algorithm 9: Chambolle-Pock resolvant algorithm for problem (3.62)

Initialization: *Choose* $\tau, \sigma > 0$, $\theta \in [0,1]$, $u^0 \in \mathbb{R}^{\Omega}$, $p^0 \in \mathbb{R}^{\Omega_n} \times \mathbb{R}^{\Omega_n}$, *set* $\overline{u}^0 = u^0$ *and define* ν *and* $\pi_{\infty,2}$ *as in Algorithm 6.*

Iterations: *For* $k \geq 0$, *update* p^k, u^k *and* $\overline{u}^k$ *with*

$$\begin{aligned} p^{k+1} &= \pi_{\infty,2}\left(\left(p^k + \tfrac{\sigma\lambda}{n^2}\nabla_n \overline{u}^k\right)/\nu\right) \\ u^{k+1/2} &= u^k + \tfrac{\tau\lambda}{n^2}\mathrm{div}_n p^{k+1} \\ u^{k+1} &= \mathcal{F}^{-1}\left(\frac{\widehat{u^{k+1/2}} + 2\tau\gamma \cdot \widehat{u_0}}{1 + 2\tau\gamma}\right) \\ \overline{u}^{k+1} &= u^{k+1} + \theta\left(u^{k+1} - u^k\right) \end{aligned}$$

Shannonizer". We did not thoroughly investigate this phenomenon yet but the first results we obtained using the simple Gaussian weight function

$$\gamma(\alpha, \beta) = e^{-2\pi^2\sigma^2\left(\frac{\alpha^2}{M^2} + \frac{\beta^2}{N^2}\right)} \tag{3.66}$$

seem interesting enough to be mentioned here.

Aliasing arises when a continuous image is not sampled in accordance with Shannon Theorem, that is, when the sampling step is too large compared to the highest frequency component that the image contains. In that case, the sampled image will be *aliased*, which means that its discrete Fourier coefficients will be the sum of one correct value and several incorrect values arising from higher frequencies that cannot be represented in the available discrete Fourier domain. In practice, since the power spectrum of natural images tends to exhibit a power-law decrease (see [Ruderman 1994]), aliasing mostly impacts the highest frequencies of the discrete image in general; it is thus logical to choose for γ a decreasing function of the distance to the origin. The isotropic map (3.66) is a possibility, but it would certainly be worth exploring other choices.

The Shannon interpolate of an aliased image is very oscillatory in general, because the aliased component define a trigonometric polynomial with improper aliased frequencies. Therefore, we can expect Model (3.62) to show interesting aliasing removal performances, as STV is strongly affected by oscillations. Indeed, we can observe in Figure 3.10 and 3.11 that the aliasing of the input image u_0

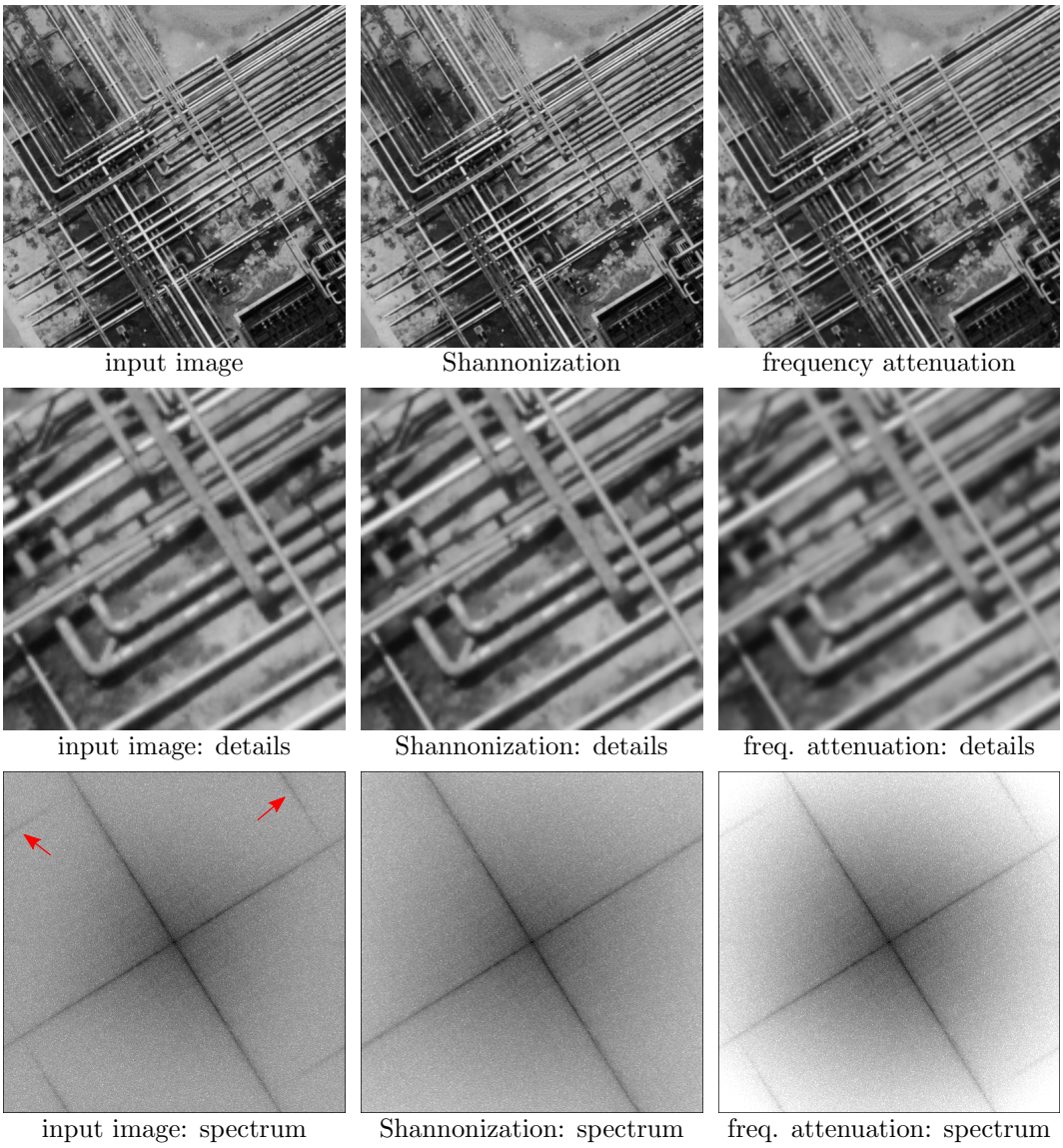


Figure 3.10: Image "Shannonization". The input image (left column) is slightly aliased, as indicated by the periodic continuation patterns (see red arrows) that appear in its Fourier spectrum (3rd row). Processing this image with the "Image Shannonizer" (3.62) results in a visually similar image (middle column) that seems aliasing-free (the patterns are not visible any more on the 3rd row). In comparison, a generic frequency attenuation process (on the right column, with a Gaussian attenuation map) produces a large amount of blur while being less efficient in terms of aliasing removal.

input image Shannonization frequency attenuation

Figure 3.11: Details of Figure 3.10 with Shannon resampling. Different Parts of the three images of the first row of Figure 3.10 are shown after Shannon interpolation. As expected, the output of the "Image Shannonizer" (middle) is well interpolable, contrary to the input image (left) on which oscillations appear. A simple frequency attenutation (right) is not efficient, since it introduces a large amount of undesired blur.

(which is clearly visible on its spectrum) is completely removed after processing through the Image Shannonizer, without introducing noticeable blur on the image.

3.7 Conclusion

In this chapter we showed that images delivered by variational TV-based models could not be easily interpolated when the TV is discretized with a classical finite difference scheme. However, we demonstrated on several examples that a variant called STV (for Shannon TV) successfully addresses this issue, and can be efficiently handled using Legendre-Fenchel duality and Chambolle-Pock Algorithm. We easily adapted the STV variant to Huber-TV regularization, which let us believe that STV could be easily applied to other variants of the discrete TV as well; for example, the Total Generalized Variation (TGV) proposed in [Bredies et al. 2010] involves higher order derivatives that could be computed exactly as in the STV approach.

The choice of the upsampling factor n used to estimate STV with a Riemann sum was discussed and it was shown that $n = 1$ was inadequate. However, it would be interesting to further investigate this issue and prove that $n = 2$ (or intermediate values between 1 and 2) guarantees a close correspondence between the true STV and its estimate STV_n.

We also presented a new STV-based restoration model relying on a weight map in the Fourier domain, and showed that in certain cases it could be used as an "Image Shannonizer", which transforms an image into a very similar one that can be easily interpolated (with Shannon interpolation or spline interpolation for example). This seems particularly interesting, as most images are not perfectly sampled (and hence difficult to interpolate) and would hence benefit a lot from this process. This opens new perspectives on aliasing removal (and thus super-resolution from a single image), but several questions are still to be answered, in particular concerning the choice of the weight map.

3.8 Appendix

3.8.A Proof of Proposition 21

Let us consider, for $x \in \mathbb{R} \setminus \mathbb{Z}$,

$$\begin{aligned} S_n(x) &= \sum_{p=-n}^{n} \operatorname{sinc}(x - pN) \\ &= \operatorname{sinc}(x) + \sum_{p=1}^{n} \operatorname{sinc}(x - pN) + \operatorname{sinc}(x + pN) \\ &= \frac{\sin \pi x}{\pi x} + \sum_{p=1}^{n} (-1)^{pN} \left(\frac{\sin \pi x}{\pi(x - pN)} + \frac{\sin \pi x}{\pi(x + pN)} \right). \end{aligned}$$

Writing $x = \frac{Nt}{\pi}$, we obtain

$$S_n(x) = \frac{\sin Nt}{N} \left(\frac{1}{t} + \sum_{p=1}^{n} (-1)^{pN} \left(\frac{1}{t - p\pi} + \frac{1}{t + p\pi} \right) \right)$$

and the limit $\operatorname{sincd}_N(x) = \lim_{n \to \infty} S_n(x)$ can be computed explicitly using classical series expansions (due to Euler):

$$\begin{aligned} \forall t \in \mathbb{R} \setminus \pi\mathbb{Z}, \quad \frac{1}{\tan t} &= \frac{1}{t} + \sum_{p=1}^{\infty} \frac{1}{t - p\pi} + \frac{1}{t + p\pi}, \\ \frac{1}{\sin t} &= \frac{1}{t} + \sum_{p=1}^{\infty} (-1)^p \left(\frac{1}{t - p\pi} + \frac{1}{t + p\pi} \right). \end{aligned}$$

If N is odd, $(-1)^{pN} = (-1)^p$ and we obtain

$$\operatorname{sincd}_N(x) = \frac{\sin Nt}{N \sin t} = \frac{\sin \pi x}{N \sin \frac{\pi x}{N}},$$

and if N is even, $(-1)^{pN} = 1$ and the other series yields

$$\operatorname{sincd}_N(x) = \frac{\sin Nt}{N \tan t} = \frac{\sin \pi x}{N \tan \frac{\pi x}{N}}$$

as announced. □

3.8.B Proof of Theorem 2

Since each operator T_z is linear and translation-invariant (Hypothesis (ii)), it can be written as a convolution, that is,

$$T_z s(k) = (\psi_z \star s)(k) := \sum_{l \in I_M} \psi_z(k-l)s(l), \tag{3.67}$$

where ψ_z is an element of $\mathcal{S}$. Taking the DFT of (3.67), we obtain

$$\forall \alpha \in \mathbb{Z}, \qquad \widehat{T_z s}(\alpha) = \widehat{\psi}_z(\alpha)\widehat{s}(\alpha). \tag{3.68}$$

Now, from Hypothesis (iii) we immediately get

$$\forall z, w \in \mathbb{R},\ \forall \alpha \in \mathbb{Z}, \qquad \widehat{\psi}_{z+w}(\alpha) = \widehat{\psi}_z(\alpha)\widehat{\psi}_w(\alpha), \tag{3.69}$$

and by continuity of $z \mapsto \widehat{\psi}_z(\alpha)$ (deduced from Hypothesis (i)) we obtain

$$\forall \alpha \in \mathbb{Z}, \qquad \widehat{\psi}_z(\alpha) = e^{\gamma(\alpha)z} \tag{3.70}$$

for some $\gamma(\alpha) \in \mathbb{C}$. Since $\widehat{\psi}_1(\alpha) = e^{\frac{-2i\pi\alpha}{M}}$, we have

$$\gamma(\alpha) = -2i\pi\left(\frac{\alpha}{M} + p(\alpha)\right), \tag{3.71}$$

where $p(\alpha) \in \mathbb{Z}$ and $p(-\alpha) = -p(\alpha)$ (the fact that $T_z u$ is real-valued implies that $\widehat{\psi}_z(-\alpha) = \widehat{\psi}_z(\alpha)^*$).

Last, we compute

$$\begin{aligned}
\|T_z - id\|_2^2 &= \sup_{\|s\|_2=1} \|T_z s - s\|_2^2 \\
&= \frac{1}{M} \sup_{\|s\|_2=1} \|\widehat{T_z s} - \hat{s}\|_2^2 \\
&= \frac{1}{M} \sup_{\|\hat{s}\|_2^2=M} \sum_{\alpha \in \widehat{I}_M} |e^{-2i\pi\left(\frac{\alpha}{M}+p(\alpha)\right)z} - 1|^2 \cdot |\hat{s}(\alpha)|^2 \\
&= 4 \max_{\alpha \in \widehat{I}_M} \sin^2\left(\pi\left(\frac{\alpha}{M} + p(\alpha)\right)z\right) \\
&= 4\pi^2 z^2 \max_{\alpha \in \widehat{I}_M} \left(\frac{\alpha}{M} + p(\alpha)\right)^2 + \underset{z\to 0}{o}(z^2).
\end{aligned}$$

Hence,

$$\lim_{z\to 0} |z|^{-1}\|T_z - id\|_2 = 2\pi \max_{\alpha\in\widehat{I}_M} \left|\frac{\alpha}{M} + p(\alpha)\right| \tag{3.72}$$

and since $\frac{\alpha}{M} \in (-\frac{1}{2}, \frac{1}{2})$ and $p(\alpha) \in \mathbb{Z}$ for any $\alpha \in \widehat{I}_M$, the right-hand term of (3.72) is minimal if and only if $p(\alpha) = 0$ for all $\alpha \in \widehat{I}_M$. We conclude from (3.71) and (3.70) that

$$\forall \alpha \in \widehat{I}_M, \qquad \widehat{\psi}_z(\alpha) = e^{-2i\pi\alpha z/M}, \tag{3.73}$$

and thus (3.68) can be rewritten as

$$T_z s(k) = \frac{1}{M} \sum_{\alpha\in\widehat{I}_M} \widehat{s}(\alpha) e^{-2i\pi\alpha z/M} e^{-2i\pi\alpha k/M}, \tag{3.74}$$

which is exactly $S(k-z)$ thanks to (3.13) (recall that the real part is not needed because M is odd). Therefore, (3.24) is a necessary form for a set of operators (T_z) satisfying Hypotheses (i) to (iv).

Conversely, one easily checks that the operators (T_z) defined by (3.24) satisfy the Hypotheses (i) to (iv). □

3.8.C Proof of Proposition 28

Let us denote by $\nabla_{n,x}u$ and $\nabla_{n,y}u$ the two elements of $\mathbb{R}^{\Omega_n}$ such that $\nabla_n u = (\nabla_{n,x}u, \nabla_{n,y}u)$. In the following, the notation $\langle\cdot,\cdot\rangle_X$ stands for the usual Euclidean (respectively Hermitian) inner product over the real (respectively complex) Hilbert space X. We have

$$\langle \nabla_n u, p\rangle_{\mathbb{R}^{\Omega_n}\times\mathbb{R}^{\Omega_n}} = \langle \nabla_{n,x}u, p_x\rangle_{\mathbb{R}^{\Omega_n}} + \langle \nabla_{n,y}u, p_y\rangle_{\mathbb{R}^{\Omega_n}}\,.$$

Recall that we defined $\mathrm{div}_n = -\nabla_n^*$, the opposite of the adjoint of ∇_n. Noting $\mathrm{div}_{n,x} = -\nabla_{n,x}^*$ and $\mathrm{div}_{n,y} = -\nabla_{n,y}^*$, we have

$$\langle \nabla_n u, p\rangle_{\mathbb{R}^{\Omega_n}\times\mathbb{R}^{\Omega_n}} = \langle u, -\mathrm{div}_{n,x}(p_x) - \mathrm{div}_{n,y}(p_y)\rangle_{\mathbb{R}^{\Omega}}\,.$$

so that we identify $\mathrm{div}_n(p) = \mathrm{div}_{n,x}(p_x) + \mathrm{div}_{n,y}(p_y)$. Let us focus on the computation of $\mathrm{div}_{n,x}(p_x)$. Let $\widehat{\Omega_1}$, $\widehat{\Omega_2}$, $\widehat{\Omega_3}$, $\widehat{\Omega_4}$ be the sets defined by

$$\begin{aligned}
\widehat{\Omega_1} &= \left\{(\alpha,\beta) \in \mathbb{R}^2,\ |\alpha| < \tfrac{M}{2},\ |\beta| < \tfrac{N}{2}\right\} \cap \mathbb{Z}^2 \\
\widehat{\Omega_2} &= \left\{\left(\pm\tfrac{M}{2},\beta\right) \in \mathbb{R}^2,\ |\beta| < \tfrac{N}{2}\right\} \cap \mathbb{Z}^2 \\
\widehat{\Omega_3} &= \left\{\left(\alpha,\pm\tfrac{N}{2}\right) \in \mathbb{R}^2,\ |\alpha| < \tfrac{M}{2}\right\} \cap \mathbb{Z}^2 \\
\widehat{\Omega_4} &= \left\{\left(\pm\tfrac{M}{2},\pm\tfrac{N}{2}\right)\right\} \cap \mathbb{Z}^2\,.
\end{aligned}$$

Notice that some sets among $\widehat{\Omega_2}$, $\widehat{\Omega_3}$ and $\widehat{\Omega_4}$ may be empty according to the parity of M and N. Now, let $h_{\widehat{p_x}}$ be the function defined in Proposition 28 and let us show that

$$\forall (\alpha,\beta) \in \widehat{\Omega}, \quad \widehat{\mathrm{div}_{n,x}(p_x)}(\alpha,\beta) = 2i\pi \frac{\alpha}{M} h_{\widehat{p_x}}(\alpha,\beta). \tag{3.75}$$

Given $z \in \mathbb{C}$, we denote as usual by z^* the conjugate of z. Thanks to Parseval identity, and using Proposition 26 (because we assumed $n \geq 2$), we have

$$\begin{aligned}
\langle \nabla_{n,x} u, p_x \rangle_{\mathbb{R}^{\Omega_n}} &= \frac{1}{n^2 MN} \langle \widehat{\nabla_{n,x} u}, \widehat{p_x} \rangle_{\mathbb{C}^{\Omega_n}} \\
&= \frac{1}{n^2 MN} \sum_{(\alpha,\beta)\in\widehat{\Omega}_n} \widehat{\nabla_{n,x} u}(\alpha,\beta)\ (\widehat{p_x}(\alpha,\beta))^* \\
&= \frac{1}{MN} \sum_{\substack{-\frac{M}{2}\leq\alpha\leq\frac{M}{2} \\ -\frac{N}{2}\leq\beta\leq\frac{N}{2}}} -\widehat{u}(\alpha,\beta) \left(2i\pi\varepsilon_M(\alpha)\varepsilon_N(\beta)\frac{\alpha}{M}\widehat{p_x}(\alpha,\beta)\right)^*.
\end{aligned}$$

It follows that

$$\langle \nabla_{n,x} u, p_x \rangle_{\mathbb{R}^{\Omega_n}} = S_1 + S_2 + S_3 + S_4\,,$$

where for all $k \in \{1,2,3,4\}$, we have set

$$S_k = \frac{1}{MN} \sum_{(\alpha,\beta)\in\widehat{\Omega}_k} -\widehat{u}(\alpha,\beta) \left(2i\pi\varepsilon_M(\alpha)\varepsilon_N(\beta)\frac{\alpha}{M}\widehat{p_x}(\alpha,\beta)\right)^*.$$

Consider S_1 first. Since we have $\varepsilon_M(\alpha) = \varepsilon_N(\beta) = 1$ and $h_{\widehat{p_x}}(\alpha,\beta) = \widehat{p_x}(\alpha,\beta)$ for all $(\alpha,\beta) \in \widehat{\Omega_1}$, we recognize

$$S_1 = \frac{1}{MN} \sum_{|\alpha|<\frac{M}{2},\ |\beta|<\frac{N}{2}} -\widehat{u}(\alpha,\beta) \left(2i\pi\frac{\alpha}{M} h_{\widehat{p_x}}(\alpha,\beta)\right)^*.$$

Now consider S_2. If M is odd, $\widehat{\Omega}_2$ is empty and $S_2 = 0$. Otherwise, since $\varepsilon_M(\alpha)\varepsilon_N(\beta) = 1/2$ for all $(\alpha,\beta) \in \widehat{\Omega_2}$, by grouping together the terms $\left(-\frac{M}{2},\beta\right)$ and $\left(\frac{M}{2},\beta\right)$, we get

$$\begin{aligned}
S_2 &= \frac{1}{MN} \sum_{\alpha=-\frac{M}{2},|\beta|<\frac{N}{2}} -\widehat{u}(\alpha,\beta) \left(2i\pi\frac{1}{2}\frac{\alpha}{M}\widehat{p_x}(\alpha,\beta) - 2i\pi\frac{1}{2}\frac{\alpha}{M}\widehat{p_x}(-\alpha,\beta)\right)^* \\
&= \frac{1}{MN} \sum_{\alpha=-\frac{M}{2},|\beta|<\frac{N}{2}} -\widehat{u}(\alpha,\beta) \left(2i\pi\frac{\alpha}{M} h_{\widehat{p_x}}(\alpha,\beta)\right)^*,
\end{aligned}$$

since we have set $h_{\widehat{p_x}}(-\frac{M}{2},\beta) = \frac{1}{2}\left(\widehat{p_x}(-\frac{M}{2},\beta) - \widehat{p_x}(\frac{M}{2},\beta)\right)$ for $|\beta| < N/2$.

Similarly for the term S_3. When N is odd, $\widehat{\Omega_3} = \emptyset$ and $S_3 = 0$. Otherwise, when N is even, we have $\varepsilon_M(\alpha)\varepsilon_N(\beta) = 1/2$ for all $(\alpha,\beta) \in \widehat{\Omega_3}$, thus, by grouping together the terms $\left(\alpha, -\frac{N}{2}\right)$ and $\left(\alpha, \frac{N}{2}\right)$, we get

$$\begin{aligned} S_3 &= \frac{1}{MN} \sum_{|\alpha|<\frac{M}{2},\beta=-\frac{N}{2}} -\widehat{u}(\alpha,\beta)\left(2i\pi\frac{1}{2}\frac{\alpha}{M}\widehat{p_x}(\alpha,\beta) + 2i\pi\frac{1}{2}\frac{\alpha}{M}\widehat{p_x}(\alpha,-\beta)\right)^* \\ &= \frac{1}{MN} \sum_{|\alpha|<\frac{M}{2},\beta=-\frac{N}{2}} -\widehat{u}(\alpha,\beta)\left(2i\pi\frac{\alpha}{M}h_{\widehat{p_x}}(\alpha,\beta)\right)^*, \end{aligned}$$

since we have set $h_{\widehat{p_x}}(\alpha,-\frac{N}{2}) = \frac{1}{2}\left(\widehat{p_x}(\alpha,-\frac{N}{2}) + \widehat{p_x}(\alpha,\frac{N}{2})\right)$ for $|\alpha| < M/2$.

Lastly, let us consider S_4. When M and N are both even (otherwise $\widehat{\Omega_4} = \emptyset$ and $S_4 = 0$), set $\alpha = -\frac{M}{2}$ and $\beta = -\frac{N}{2}$, we immediately get

$$\begin{aligned} S_4 &= -\widehat{u}(\alpha,\beta)\left(\sum_{s_1=\pm1,s2=\pm1} 2i\pi\frac{1}{4}s_1\frac{\alpha}{M}\widehat{p_x}(s_1\alpha,s_2\beta)\right)^* \\ &= -\widehat{u}(\alpha,\beta)\left(2i\pi\frac{\alpha}{M}h_{\widehat{p_x}}(\alpha,\beta)\right)^*, \end{aligned}$$

since for all $(\alpha,\beta) \in \widehat{\Omega_4}$, we have $\varepsilon_M(\alpha)\varepsilon_N(\beta) = 1/4$ and we have set $h_{\widehat{p_x}}(\alpha,\beta) = \frac{1}{4}\sum_{s_1=\pm1,s2=\pm1} s_1\widehat{p_x}(s_1\alpha,s_2\beta)$ when $\alpha = -\frac{M}{2}$ and $\beta = -\frac{N}{2}$.

Finally, we can write $S_1 + S_2 + S_3 + S_4$ as a sum over $\widehat{\Omega}$, indeed,

$$\begin{aligned} \langle \nabla_{n,x} u, p_x\rangle_{\mathbb{R}^\Omega} &= S_1 + S_2 + S_3 + S_4 \\ &= \frac{1}{MN}\sum_{(\alpha,\beta)\in\widehat{\Omega}} -\widehat{u}(\alpha,\beta)\left(2i\pi\frac{\alpha}{M}h_{\widehat{p_x}}(\alpha,\beta)\right)^*, \end{aligned}$$

and using again the Parseval identity, we get (3.75). With a similar approach, one can check that

$$\forall(\alpha,\beta)\in\widehat{\Omega},\quad \widehat{\mathrm{div}_{n,y}(p_y)}(\alpha,\beta) = 2i\pi\frac{\beta}{N}h_{\widehat{p_y}}(\alpha,\beta),$$

where $h_{\widehat{p_y}}$ is defined in Proposition 28. Consequently, for any $(\alpha,\beta)\in\widehat{\Omega}$, we have

$$\widehat{\mathrm{div}_n(p)}(\alpha,\beta) = 2i\pi\left(\frac{\alpha}{M}h_{\widehat{p_x}}(\alpha,\beta) + \frac{\beta}{N}h_{\widehat{p_y}}(\alpha,\beta)\right),$$

which ends the proof of Proposition 28. □

3.8.D Proof of Theorem 3

Recall that for any integer M, we denote by T_M the real vector space of real-valued trigonometric polynomials that can be written as complex linear combination of the family $(x \mapsto e^{2i\pi\frac{\alpha x}{M}})_{-\frac{M}{2}\leq\alpha\leq\frac{M}{2}}$. In order to prove Theorem 3 we need the following Lemma.

Lemma 4. *Let $M = 2m+1$ be an odd positive integer. The functions F and G defined by,*

$$\forall x \in \mathbb{R}, \quad F(x) = \frac{1}{M}\sum_{\alpha=-m}^{m} e^{\frac{2i\pi\alpha x}{M}},\ G(x) = F(x) - F(x-1)\,,$$

are both in T_M and G satisfies

$$\sum_{k=0}^{M-1}|G(k)| = 2\,, \quad \int_1^M |G(x)|\,dx \geq \frac{8}{\pi^2}\log\left(\frac{2M}{\pi}\right) - 2\,.$$

Proof. F is in T_M by construction, and so is G as the difference of two elements of T_M. Writing $\omega = \frac{\pi}{M}$, we can notice that $F(0) = 1$ and

$$\forall x \in (0, M), \quad F(x) = \frac{e^{2i\omega(-m)x}}{M}\cdot\frac{1-e^{2i\pi x}}{1-e^{2i\omega x}} = \frac{\sin(\pi x)}{M\sin(\omega x)}\,,$$

so that $F(k) = 0$ for all integers $k \in [1, M-1]$. Consequently, $G(0) = 1$, $G(1) = -1$ and $G(k) = 0$ for all integers $k \in [2, M-1]$, thus

$$\sum_{k=0}^{M-1}|G(k)| = |G(0)| + |G(1)| = 2\,,$$

yielding the first announced result of the Lemma. Now, remark that the sign changes of G in $(0, 2m+1)$ occur at integer points $2, 3, \ldots 2m$ and in $\frac{1}{2}$ (by symmetry). Thus, we have

$$J := \int_1^M |G(x)|\,dx = \sum_{k=1}^{2m}(-1)^k\int_k^{k+1} G(x)\,dx = 2\sum_{k=0}^{2m-1}(-1)^k\int_k^{k+1} F(x)\,dx\,,$$

since for all $x \in [0, M]$, we have $G(x) = F(x) - F(x-1)$ and (because M is odd) $F(x) = F(M-x)$. It follows that

$$J \geq 2\left(\sum_{k=0}^{2m}(-1)^k\int_k^{k+1} F(x)\,dx\right) - 2\,,$$

since $|F| \leq 1$ everywhere.

Consequently, by isolating the index $\alpha = 0$ in the definition of F, we get $J \geq 2\left(J' + \frac{1}{M}\right) - 2$, with

$$J' = \sum_{k=0}^{2m} \frac{(-1)^k}{M} \sum_{\substack{-m \leq \alpha \leq m \\ \alpha \neq 0}} \int_k^{k+1} e^{2i\omega\alpha x}\, dx\,.$$

By exchanging the sums and grouping identical terms, we obtain

$$\begin{aligned} J' &= \frac{1}{M} \sum_{\substack{-m \leq \alpha \leq m \\ \alpha \neq 0}} \sum_{k=0}^{2m} (-1)^k \cdot \frac{e^{2i\omega\alpha(k+1)} - e^{2i\omega\alpha k}}{2i\omega\alpha} \\ &= \sum_{\substack{-m \leq \alpha \leq m \\ \alpha \neq 0}} \frac{-1}{i\pi\alpha} \sum_{k=1}^{2m} \left(-e^{2i\omega\alpha}\right)^k . \end{aligned} \tag{3.76}$$

After summation of the geometric progression

$$\begin{aligned} \sum_{k=1}^{2m} \left(-e^{2i\omega\alpha}\right)^k &= -e^{2i\omega\alpha} \cdot \frac{1 - e^{2i\omega\alpha(2m)}}{1 + e^{2i\omega\alpha}} = e^{i\pi\alpha} \frac{i \sin(2\omega m\alpha)}{\cos(\omega\alpha)} = \frac{i \sin(2\omega m\alpha - \pi\alpha)}{\cos(\omega\alpha)} \\ &= -i \tan(\omega\alpha)\,, \end{aligned}$$

Equation (3.76) finally leads to

$$J' = \sum_{\substack{-m \leq \alpha \leq m \\ \alpha \neq 0}} \frac{1}{\pi\alpha} \cdot \tan(\omega\alpha) = \frac{2}{M} \sum_{\alpha=1}^{m} g(\omega\alpha)$$

where $g = t \mapsto \frac{\tan t}{t}$. Now since g is positive and increasing on $(0, \frac{\pi}{2})$, we have

$$\sum_{\alpha=1}^{m} g(\omega\alpha) \geq \int_0^m g(\omega x)\, dx = \frac{1}{\omega} \int_0^{\omega m} g(t)\, dt\,.$$

Using the lower bound $g(t) \geq \frac{2}{\pi} \tan t$ for $t \in (0, \frac{\pi}{2})$, we finally get

$$J' \geq \frac{4}{\pi^2} \int_0^{\omega m} \tan t\, dt = -\frac{4}{\pi^2} \log \cos(\omega m) = -\frac{4}{\pi^2} \log \sin\left(\frac{\omega}{2}\right)$$

and thus $J' \geq \frac{4}{\pi^2} \log\left(\frac{2}{\omega}\right)$, from which the inequality announced in Lemma 4 follows. □

Now, let us prove the Theorem 3 by building a discrete image u such that $\mathrm{STV}_1(u)$ is fixed but $\mathrm{STV}_\infty(u)$ increases with the image size. We consider the function H defined by

$$\forall x \in \mathbb{R}, \quad H(x) = \int_0^x G(t)\,dt\,,$$

where $G \in T_M$ is the real-valued M-periodic trigonometric polynomial defined in Lemma 4 ($M = 2m+1$). Since the integral of G over one period is zero ($\int_0^M G(t)\,dt = 0$), H is also an element of T_M. Consequently, the bivariate trigonometric polynomial defined by

$$\forall (x,y) \in \mathbb{R}^2, \quad U(x,y) = \frac{1}{M} H(x)\,,$$

beongs to $T_M \otimes T_M$, and since M is odd it is exactly the Shannon interpolate of the discrete image defined by

$$\forall (k,l) \in I_M \times I_M, \quad u(k,l) = U(k,l). \tag{3.77}$$

In particular, by definition of STV_1 and STV_∞, we have

$$\mathrm{STV}_1(u) = \sum_{(k,l)\in\Omega} |\nabla U(k,l)|\,, \quad \text{and} \quad \mathrm{STV}_\infty(u) = \int_{[0,M]^2} |\nabla U(x,y)|\,dxdy\,.$$

From Lemma 4, we have on the one hand,

$$\mathrm{STV}_1(u) = \sum_{(k,l)\in\Omega} |\nabla U(k,l)| = \sum_{k=0}^{2m} |H'(k)| = \sum_{k=0}^{2m} |G(k)| = 2\,,$$

and on the other hand,

$$\begin{aligned} \mathrm{STV}_\infty(u) &= \int_{[0,M]^2} |\nabla U(x,y)|\,dxdy = \int_0^M |H'(x)|\,dx \\ &= \int_0^M |G(x)|\,dx \geq \frac{8}{\pi^2} \log\left(\frac{2M}{\pi}\right) - 2\,. \end{aligned}$$

which cannot be bounded from above by a constant independent of M. □

3.8.E Proof of Proposition 30

Let $u \in \mathbb{R}^{\Omega}$, $n \in \mathbb{N}$ and $\alpha \in \mathbb{R}$ such that $n \geq 1$ and $\alpha > 0$. One can rewrite $\mathrm{HSTV}_{\alpha,n}(u) = \frac{1}{n^2} H_\alpha(\nabla_n u)$, where

$$\forall g \in \mathbb{R}^{\Omega_n} \times \mathbb{R}^{\Omega_n}, \quad H_\alpha(g) = \sum_{(x,y)\in\Omega_n} \mathcal{H}_\alpha(g(x,y)) \,.$$

Let us show that the Legendre-Fenchel transform of H_α is

$$H_\alpha^\star(p) = \imath_{\|\cdot\|_{\infty,2}\leq 1}(p) + \tfrac{\alpha}{2}\|p\|_2^2 \,.$$

One easily checks that $\mathcal{H}_\alpha \in \Gamma(\mathbb{R}^2)$, and it follows that $H_\alpha \in \Gamma(\mathbb{R}^{\Omega_n} \times \mathbb{R}^{\Omega_n})$. Thus, for any image $u \in \mathbb{R}^{\Omega}$, we have $H_\alpha(\nabla_n u) = H_\alpha^{\star\star}(\nabla_n u)$ and

$$H_\alpha^{\star\star}(\nabla_n u) = \sup_{p\in\mathbb{R}^{\Omega_n}\times\mathbb{R}^{\Omega_n}} \langle \nabla_n u, p\rangle - H_\alpha^\star(p) \,. \tag{3.78}$$

Besides, we have $H_\alpha^\star(p) = \sum_{(x,y)\in\Omega_n} \mathcal{H}_\alpha^\star(p(x,y))$, and the Legendre-Fenchel transform of $\mathcal{H}_\alpha$ is the function $\mathcal{H}_\alpha^\star(z) = \imath_{|\cdot|\leq 1}(z) + \frac{\alpha}{2}|z|^2$, where $\imath_{|\cdot|\leq 1}$ denotes the indicator function of the unit ball for the ℓ^2 norm in $\mathbb{R}^2$. Indeed, it is proven in [Parikh and Boyd 2013] that $\mathcal{H}_\alpha$ is the Moreau envelope (or Moreau-Yosida regularization) [Moreau 1965, Yosida 1980] with parameter α of the ℓ^2 norm $|\cdot|$, or equivalently the infimal convolution (see [Rockafellar 1970]) between the two proper, convex and l.s.c functions $f_1(x) = |x|$ and $f_2(x) = \frac{1}{2\alpha}|x|^2$, that is

$$\forall y \in \mathbb{R}^2, \quad \mathcal{H}_\alpha(y) = (f_1 \square f_2)(y) := \inf_{x\in\mathbb{R}^2} f_1(x) + f_2(y-x) \,.$$

Thus, we have $\mathcal{H}_\alpha^\star = (f_1 \square f_2)^\star = f_1^\star + f_2^\star$ (see [Rockafellar 1970, Parikh and Boyd 2013]), leading exactly to $\mathcal{H}_\alpha^\star(z) = \imath_{|\cdot|\leq 1}(z) + \frac{\alpha}{2}|z|^2$ for any $z \in \mathbb{R}^2$, since we have $f_1^\star = z \mapsto \imath_{|\cdot|\leq 1}(z)$ and $f_2^\star = z \mapsto \frac{\alpha}{2}|z|^2$. It follows that for any $p \in \mathbb{R}^{\Omega_n} \times \mathbb{R}^{\Omega_n}$, we have

$$H_\alpha^\star(p) = \sum_{(x,y)\in\Omega_n} \mathcal{H}_\alpha^\star(p(x,y)) = \imath_{\|\cdot\|_{\infty,2}\leq 1}(p) + \tfrac{\alpha}{2}\|p\|_2^2 \,, \tag{3.79}$$

and the supremum (3.78) is a maximum for the same reason as in the proof of Proposition 29. Finally, writing $\mathrm{HSTV}_{\alpha,n}(u) = \frac{1}{n^2} H_\alpha(\nabla_n u) = \frac{1}{n^2} H_\alpha^{\star\star}(\nabla_n u)$ using (3.78) and (3.79) leads to the announced result. □

3.8.F Preliminary results about isotropy

In addition to the difficulties encountered when trying to manipulate at subpixellic scales some images that were processed using the discrete total variation, its discretization by means of a finite differences scheme is also responsible for the introduction of anisotropy in those images. For instance, with the choice of discretization done in (3.2)-(3.3), we have in general

$$\mathrm{TV}(u) \neq \mathrm{TV}(Ru)\,,$$

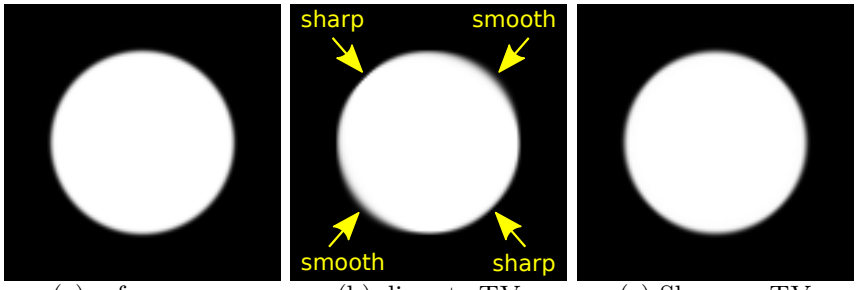
where Ru denotes the $\pi/2$ rotation operator defined by $Ru(x,y) = u(y, M-x-1)$ for any $(x,y) \in \Omega$ and any $u \in \mathbb{R}^\Omega$. Consequently, the image produced by means of discrete TV minimization in classical imaging problems is in general not the same if a rotation of $\pi/2$ is applied before or after the process. We could recently observe in the literature many attempts to define a more isotropic discrete total variation, such as the upwind-TV defined in [Chambolle et al. 2011], which makes use of a more isotropic finite differences scheme than (3.3). Note also the modified discrete TV regularizer very recently proposed by Condat [2016], which is defined by duality and makes use of dual variables oversampled with factor two using bilinear interpolation. This latter variant of TV is claimed to outperform the upwind TV in terms of isotropy, which is confirmed by many experiments.

We show in Figure 3.12 that the use of STV_n instead of the discrete total variation in the ROF model (3.49) yields an improved level of isotropy in the produced image. This improvement is a direct consequence of the use by STV_n of the exact gradient of U, instead of a finite differences scheme (which inevitably suffers from the lack of isotropy of the sampling grid Ω). However, a more careful study of the isotropy of the denoised images, presented in Figure 3.13, shows that even if the use of the STV_n regularizer improves the isotropy, the STV-denoised version of a synthetic rotationally invariant image is not completely isotropic. The reason of this loss of isotropy is that the frequency domain $\widehat{\Omega}$ of those images is rectangular and not circular, thus all frequencies lying outside of

$$\mathscr{D}_{\widehat{\Omega}} = \left\{ (\alpha,\beta) \in \widehat{\Omega},\ \left(\frac{\alpha}{M/2}\right)^2 + \left(\frac{\beta}{N/2}\right)^2 \leq 1 \right\},$$

are not equally represented in all directions. The setting of (in general nonzero) frequencies outside of $\mathscr{D}_{\widehat{\Omega}}$ performed by the minimization of (3.49) is responsible for the introduction of anisotropy into the restoration process.

We can easily avoid the creation of nonzero frequencies outside of $\mathscr{D}_{\widehat{\Omega}}$ by adding the constraint for the produced image to have its frequency support included



(a) reference u_0 (b) discrete TV (c) Shannon TV

Figure 3.12: Denoising an isotropic disk. Image (a) represents an isotropic smoothed disk, noted u_0, whose gray levels are given by $u_0(x,y) = R_{0.01}(0.015 \cdot \sqrt{(x-50)^2+(y-50)^2})$ for $(x,y) \in I_{101} \times I_{101}$, where the radial profile $R_\sigma(\rho) := \int_{-0.5}^{0.5} \frac{1}{\sigma\sqrt{2\pi}} e^{-(t-\rho)^2/2\sigma^2}\, dt$ is the convolution at the point ρ between a Gaussian signal and the characteristic function of the set $[-0.5, 0.5]$. The image (a) is then processed using the TV^{d} and STV_3 regularized ROF problem (with $\lambda = 100$), leading to images (b) and (c). We see that in the case of the image (b) the rotational symmetry of the original image is broken: diagonal edges oriented upright are kept sharp, but diagonal edges oriented upleft become blurry. In the case of image (b), the rotational symmetry is better preserved, although we show in Figure 3.13 that the isotropy can be even further improved by constraining the spectrum support to be included into a disk.

into $\mathscr{D}_{\widehat{\Omega}}$. The addition of such a constraint formally consists in forcing u to lie into the vector space $\mathscr{C}$ defined by

$$\mathscr{C} = \left\{ u \in \mathbb{R}^\Omega,\ (\alpha,\beta) \notin \mathscr{D}_{\widehat{\Omega}} \Rightarrow \widehat{u}(\alpha,\beta) = 0 \right\},$$

in the minimization (3.49). This leads to the following constrained problem

$$\operatorname*{argmin}_{u \in \mathscr{C}} \|u - u_0\|_2^2 + \lambda\, \mathrm{STV}_n(u)\,, \tag{3.80}$$

for which a numerical solution can be computed using Algorithm 6 provided that we change the primal update step (3.51b) into

$$u^{k+1} = \pi_{\mathscr{C}}\left(\frac{u^k + \frac{\tau\lambda}{n^2} \mathrm{div}_n p^{k+1} + 2\tau u_0}{1 + 2\tau} \right),$$

noting $\pi_{\mathscr{C}}(u)$ the projection of u over the constraint set $\mathscr{C}$, which simply consists in setting to zero all frequencies of $\widehat{u}$ outside of $\mathscr{D}_{\widehat{\Omega}}$. More precisely, this projection is explicitly given in the Fourier domain by

$$\forall u \in \mathbb{R}^\Omega, \quad \widehat{\pi_{\mathscr{C}}(u)}(\alpha,\beta) = \begin{cases} \widehat{u}(\alpha,\beta) & \text{if } (\alpha,\beta) \in \mathscr{D}_{\widehat{\Omega}}\,, \\ 0 & \text{otherwise.} \end{cases}$$

In Figure 3.13, we show that denoising a rotationally invariant image using (3.80) yields an almost perfectly rotationally invariant image, which is not the case when we use the unconstrained model (3.49). We also observed on this experiment that the level of isotropy reached by model (3.80) is even better than that reached when using the modified TV regularizer proposed by Condat, which, as a preliminary result, offers interesting perspectives for future works.

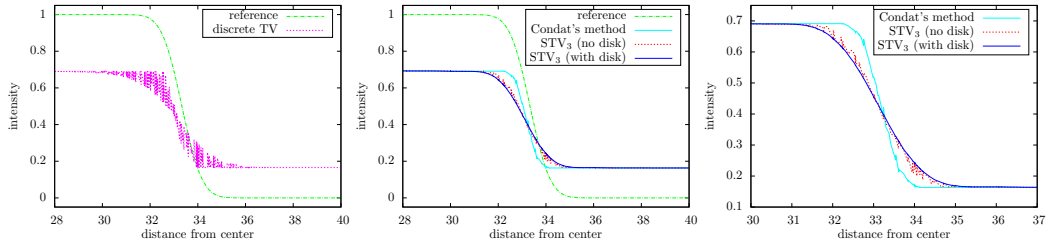


Figure 3.13: Comparison of different image denoising models in terms of isotropy. The rotationally invariant reference image of Figure 3.12-(a) was denoised using the discrete TV, its modified version proposed by Condat [2016], and the Shannon total variation (with or without the constraint of circular frequency domain as indicated using the keywords "with disk" or "no disk") as a regularizer. For each considered image, we display the evolution of its intensity according to the distance from the image center (that is, given the image $u \in \mathbb{R}^{\Omega}$ and for all $(x, y) \in \Omega$, we draw in a graph a point at the coordinate $(\rho(x, y), u(x, y))$, noting $\rho(x, y)$ the distance between the pixel (x, y) and the center of the image). In the case of the perfectly rotationally invariant image, we observe a monotone curve. The strong irregularity of the signal corresponding to the TV^{d}-processed image (left graph) indicates that this image is far from being rotationally invariant (two pixels located at the same distance from the center can have very different gray levels). We can see (middle and right graphs) that the use of Condat's TV model or STV_3 (no disk) significantly improves the level of isotropy, although some small irregularities remain. By adding the frequency constraint to the STV_3 regularized model, we obtain an almost perfectly monotonous curve.

Chapter 4

Total Variation Restoration of Images Corrupted by Poisson Noise with Iterated Conditional Expectations

Contents

The content of this chapter has been partially presented in the conference paper [Abergel et al. 2015], at the occasion of the fifth International Conference on Scale Space and Variational Methods in Computer Vision (SSVM 2015). The section 4.2 is mainly due to Cécile Louchet and Lionel Moisan. The section 4.3.2 was not included in the conference paper and is an original contribution to this work.

Abstract

Interpreting the celebrated Rudin-Osher-Fatemi (ROF) model in a Bayesian framework has led to interesting new variants for Total Variation image denoising in the last decade. The Posterior Mean variant avoids the so-called staircasing artifact of the ROF model but is computationally very expensive. Another recent variant, called TV-ICE (for Iterated Conditional Expectation), delivers very similar images but uses a much faster fixed-point algorithm. In this chapter, we consider the TV-ICE approach in the case of a Poisson noise model. We derive an explicit form of the recursion operator, and show linear convergence of the algorithm, as well as the absence of staircasing effect. We also provide a numerical algorithm that carefully handles precision and numerical overflow issues, and show experiments that illustrate the interest of this Poisson TV-ICE variant.

4.1 Introduction

Since the seminal paper of Rudin, Osher and Fatemi [Rudin et al. 1992], total variation (TV) regularization has been used in numerous image processing applications (see, e.g., [Caselles et al. 2015] and references therein). Reasons for this popularity are multiple. First, TV regularization allows discontinuities (contrary to the L^2 norm of the gradient), which is essential in the world of natural images, dominated by occlusions. Second, its continuous counterpart is part of a fruitful mathematical theory (the space of functions with bounded variation) which results in strong possibilities of theoretical interpretations [Chambolle et al. 2010]. Third, in the last decade several very efficient algorithms have been designed to handle the non-smooth convex optimization problems occurring with TV regularization (e.g., [Darbon and Sigelle 2006, Chambolle and Pock 2011]). In terms of pure denoising performances, TV denoising is less efficient than modern patch-based approaches like NL-means [Buades et al. 2005] or BM3D [Dabov et al. 2007] for example, but remains useful as the simplest possible framework for the study of TV regularization. Understanding the strengths and weaknesses of TV denoising (and variants) certainly helps a lot apprehending more complex inverse problems involving TV regularization.

One weakness of TV regularization is the so-called staircasing effect: where one would have expected a smoothly varying image, the L^1 norm promotes a sparse gradient that results in piecewise constant zones with artificial boundaries. This undesirable effect can be avoided by using a smoother functional, but at the expense of loosing the nice theoretical properties of TV. Other solutions have

been proposed that keep the true definition of TV but change the minimization framework. Indeed, when considering the TV as the Gibbs energy of an image prior in a Bayesian framework, the ROF model can be reinterpreted as finding the image that maximizes the associated posterior density. Replacing this maximum a posteriori (MAP) estimate with the posterior mean leads to a variant of the ROF model, called TV-LSE, that delivers images without staircasing artifacts [Louchet and Moisan 2008, 2013]. More recently, a new variant called TV-ICE [Louchet and Moisan 2014] was proposed to overcome the slow convergence rate of the TV-LSE Monte-Carlo algorithm. It is based on the repeated estimation of conditional marginal posterior means, which boils down to iterating an explicit local operator. In practice, TV-ICE produces images very similar to TV-LSE results, but at a much smaller computational expense.

In the present chapter, we propose to adapt to the case of Poisson noise this TV-ICE method, derived in [Louchet and Moisan 2014] in the case of Gaussian noise. Contrary to most noise sources (electronic noise, dark current, thermal noise) whose effects can be reduced by the improvement of captors, Poisson noise is inherent to the quantum nature of light and thus unavoidable for images acquired in low-light conditions, which is very common in astronomy or in microscopy for example. Even if image restoration models are generally first designed in the simpler case of a white Gaussian additive noise, they need to be adapted to the specific case of Poisson noise. Due to the importance and the inevitability of Poisson noise, this adaptation is almost systematic, as shows for example the case of TV-based image deblurring [Setzer et al. 2010] or NL-means denoising [Deledalle et al. 2010].

In the case of the TV prior, the posterior distribution obtained with Poisson noise strongly differs from the Gaussian case, but the conditional marginal posterior means can be explicitly computed using the incomplete Gamma function. In Section 4.2, we show that the associated iterative algorithm converges linearly and that no staircasing occurs, thanks in particular to the log-concavity of the Poisson distribution. We then give the explicit form of the recursion operator defining our Poisson-TV-ICE model (Section 4.3) and discuss numerical issues, in particular the handling of machine over/under-flow and the efficient computation of the (slightly generalized) incomplete Gamma function. We then numerically check the theoretical properties of the method (convergence rate, absence of staircasing) in Section 4.4, and compare the obtained results with the Poisson noise variant of the ROF model, before we conclude in Section 4.5.

4.2 The Poisson TV-ICE model

4.2.1 Definition

Let $u : \Omega \to \mathbb{R}_+$ be an (unobserved) intensity image defined on a discrete domain Ω (a rectangular subset of $\mathbb{Z}^2$). A photon-count observation of the ideal image u is an integer valued random image $v : \Omega \to \mathbb{N}$ following the Poisson probability density function (p.d.f.)

$$p(v \,|\, u) = \prod_{x \in \Omega} \frac{u(x)^{v(x)}}{v(x)!} e^{-u(x)} \propto \exp\left(-\langle u - v \log u \,,\, \mathbb{1}_\Omega \rangle\right), \tag{4.1}$$

where $\mathbb{1}_\Omega$ denotes the constant image equal to 1 on Ω and $\langle \cdot, \cdot \rangle$ is the usual Euclidean inner product on $\mathbb{R}^\Omega$. The notation $\propto$ here indicates an equality up to a global multiplicative constant (which depends on v). Note that we have to take the convention that $v(x) \log u(x) = 0$ as soon as $v(x) = 0$ in (4.1). We consider here the discrete anisotropic total variation of u, defined in Chapter 2 (Definition 4), that we reformulate in

$$\mathrm{TV}_1^{\mathrm{d}}(u) = \frac{1}{2} \sum_{x \in \Omega} \sum_{y \in \mathcal{N}_x} |u(y) - u(x)| \,, \tag{4.2}$$

where $\mathcal{N}_x$ denotes the 4-neighborhood of a pixel x, with a mirror boundary condition. Using the improper $\mathrm{TV}_1^{\mathrm{d}}$ prior $p(u) \propto e^{-\lambda \mathrm{TV}_1^{\mathrm{d}}(u)}$ (where λ is a positive regularization parameter) and Equation (4.1), we get, thanks to the Bayes rule, the posterior density

$$\pi(u) = p(u \,|\, v) = \frac{p(v \,|\, u)\, p(u)}{\int_{\mathbb{R}_+^\Omega} p(v \,|\, w)\, p(w)\, dw} = \frac{e^{-\langle u - v \log u \,,\, \mathbb{1}_\Omega \rangle - \lambda \mathrm{TV}_1^{\mathrm{d}}(u)}}{\int_{\mathbb{R}_+^\Omega} e^{-\langle w - v \log w \,,\, \mathbb{1}_\Omega \rangle - \lambda \mathrm{TV}_1^{\mathrm{d}}(w)}\, dw} \,. \tag{4.3}$$

As discussed in Chapter 2 (Remark 3), the equivalent of the classical model of Rudin et al. [1992] in the case of a Poisson noise model corresponds to the unique maximizer $\widehat{u}_{\mathrm{MAP}}$ of π, or equivalently the minimizer of the convex energy $E = u \mapsto \langle u - v \log u \,,\, \mathbb{1}_\Omega \rangle + \lambda \mathrm{TV}_1^{\mathrm{d}}(u)$. We explained in Section 2.3 how this minimizer can be efficiently computed using the primal-dual algorithm recently proposed in [Chambolle and Pock 2011]. As mentioned in Introduction, and observed in many numerical examples of Section 2.3, a main drawback of this approach is that $\widehat{u}_{\mathrm{MAP}}$ generally suffers from the staircasing effect, which results in the appearance of flat regions separated by artificial boundaries, and which is particularly strong in the case of the anisotropic total variation.

In the case of a Gaussian noise model with isotropic discrete TV (that is, when $\pi(u) \propto e^{-\|u-v\|_2^2/(2\sigma^2)-\lambda \mathrm{TV}^{\mathrm{d}}(u)}$), this can be avoided by considering, instead of $\widehat{u}_{\mathrm{MAP}}$, the posterior mean

$$\widehat{u}_{\mathrm{LSE}} = \mathbb{E}_{u\sim\pi}(u) = \int_{\mathbb{R}^\Omega} u\,\pi(u)\,du\,, \tag{4.4}$$

which is the image that reaches the Least Square Error under π (see [Louchet and Moisan 2008, 2013]). The numerical computation of $\widehat{u}_{\mathrm{LSE}}$ proposed in [Louchet and Moisan 2008] is based on a Markov Chain Monte Carlo Metropolis-Hastings algorithm, which exhibits a slow convergence rate ($\mathscr{O}(n^{-1/2})$ for n iterations). To overcome this computational limitation, it was proposed in [Louchet and Moisan 2014] a new variant based on the iteration of conditional marginal posterior means. More precisely, the estimate $\widehat{u}_{\mathrm{ICE}}$ is defined as the limit (for an appropriate initialization) of the iterative scheme

$$u^{n+1}(x) = \mathbb{E}_{u\sim\pi}\left(u(x) \;\middle|\; u(x^c) = u^n(x^c)\right) = \int_{\mathbb{R}} u^n(x)\,\pi(u^n)\,du^n(x)\,, \tag{4.5}$$

where $u(x^c)$ denotes the restriction of u to $\Omega \setminus \{x\}$. In the case of the Poisson noise model (4.3), we obtain the following:

Definition 17 (Poisson TV-ICE). *The Poisson TV-ICE recursion is*

$$\forall n \in \mathbb{N},\ \forall x \in \Omega, \quad u^{n+1}(x) = \frac{\int_{\mathbb{R}_+} s^{v(x)+1} e^{-\left(s+\lambda\sum_{y\in\mathcal{N}_x}|u^n(y)-s|\right)} ds}{\int_{\mathbb{R}_+} s^{v(x)} e^{-\left(s+\lambda\sum_{y\in\mathcal{N}_x}|u^n(y)-s|\right)} ds}\,. \tag{4.6}$$

4.2.2 Convergence

Theorem 4. *Given an image $v : \Omega \to \mathbb{N}$, the sequence of images $(u^n)_{n\geq 0}$ defined by $u^0 = 0$ and the recursion (4.6) converges linearly to an image $\widehat{u}_{\mathrm{ICE}}$.*

In the following, we denote by $P_p(s)$ the pointwise Poisson noise p.d.f. with integer parameter $p \geq 0$, that is,

$$\forall s \in \mathbb{R}, \quad P_p(s) = \frac{s^p\, e^{-s}}{p!}\mathbb{1}_{\mathbb{R}_+}(s)\,, \quad \text{where} \quad \mathbb{1}_{\mathbb{R}_+}(s) = \begin{cases} 1 & \text{if } s \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

If $a = (a_i)_{1\leq i\leq 4}$ denotes a 4-uple, we write

$$f_p(a) = \frac{\int_{-\infty}^{+\infty} s\,P_p(s)\,e^{-\lambda\sum_{i=1}^4|s-a_i|}\,ds}{\int_{-\infty}^{+\infty} P_p(s)\,e^{-\lambda\sum_{i=1}^4|s-a_i|}\,ds} = \frac{\int_0^{+\infty} s^{p+1}\,e^{-s}\,e^{-\lambda\sum_{i=1}^4|s-a_i|}\,ds}{\int_0^{+\infty} s^{p}\,e^{-s}\,e^{-\lambda\sum_{i=1}^4|s-a_i|}\,ds}\,, \tag{4.7}$$

$$\text{and} \qquad F : u \mapsto \left(x \mapsto f_{v(x)}(u(\mathcal{N}_x))\right), \tag{4.8}$$

so that the recursion (4.6) can be simply rewritten $u^{n+1} = F(u^n)$.

To prove Theorem 4, we need some intermediate Lemmas.

Lemma 5 (Schmidt [2003]). *Assume that X, a random variable defined on $\mathbb{R}^\Omega$, has a finite second order moment. Then the inequality*

$$\operatorname{cov}(X, g(X)) \geq 0$$

holds for every nondecreasing function $g : \mathbb{R}^\Omega \to \mathbb{R}$ for which $g(X)$ has a finite second order moment. If, moreover, X is not deterministic and g is strictly increasing, then $\operatorname{cov}(X, g(X)) > 0$.

Proof. The proof for the first part is already given in [Schmidt 2003], but we reproduce it here for its shortness, and because the second part of the Lemma derives from it.

$$\begin{aligned}\operatorname{cov}(X, g(X)) &= \mathbb{E}[(X - \mathbb{E}[X])(g(X) - \mathbb{E}[g(X)])] \\ &= \mathbb{E}[(X - \mathbb{E}[X])(g(X) - g(\mathbb{E}[X]))]\,.\end{aligned}$$

The assertion follows because g is increasing. If X is not deterministic, then there exists a Borel set A such that $P(X \in A) > 0$ with $\mathbb{E}[X] \notin A$. Hence the covariance is a sum of nonnegative terms, some of which (those for $X \in A$) are positive. Finally $\operatorname{cov}(X, g(X))$ is positive. □

Lemma 6. *F is monotone: for all images u_0 and u_1,*

$$u_0 \leq u_1 \Rightarrow F(u_0) \leq F(u_1).$$

Proof. Using Lebesgue dominated convergence theorem, one can prove the differentiability of f_p with respect to each a_i and obtain

$$\begin{aligned}\frac{\partial f_p}{\partial a_i}(a) = &\frac{\int_{-\infty}^{+\infty} \lambda \operatorname{sign}(s - a_i)\, s\, P_p(s)\, e^{-\lambda \sum_{j=1}^4 |s-a_j|}\, ds}{\int_{-\infty}^{+\infty} P_p(s)\, e^{-\lambda \sum_{j=1}^4 |s-a_j|}\, ds} \\ &- \frac{\int_{-\infty}^{+\infty} s\, P_p(s)\, e^{-\lambda \sum_{j=1}^4 |s-a_j|}\, ds}{\int_{-\infty}^{+\infty} P_p(s)\, e^{-\lambda \sum_{j=1}^4 |s-a_j|}\, ds} \cdot \frac{\int_{-\infty}^{+\infty} \lambda \operatorname{sign}(s - a_i)\, P_p(s)\, e^{-\lambda \sum_{j=1}^4 |s-a_j|}\, ds}{\int_{-\infty}^{+\infty} P_p(s)\, e^{-\lambda \sum_{j=1}^4 |s-a_j|}\, ds}\,.\end{aligned}$$

Hence $\frac{\partial f_p}{\partial a_i}(a)$ can be seen as the covariance of S and $\lambda \operatorname{sign}(S - a_i)$, where S is a random variable with p.d.f. $s \mapsto \frac{1}{Z} P_p(s)\, e^{-\lambda \sum_{j=1}^4 |s-a_j|}$ (Z denotes a normalization

constant), which has a finite second order moment. Using Lemma 5, the quantity $\frac{\partial f_p}{\partial a_i}(a)$, as the covariance of S with a nondecreasing function of S, is nonnegative. Now if $u_0 \leq u_1$, then as f_p is $\mathcal{C}^1$ we can write

$$(F(u_1) - F(u_0))(x) = \int_0^1 \nabla f_{v(x)}(u_t(\mathcal{N}_x)) \cdot (u_1(\mathcal{N}_x) - u_0(\mathcal{N}_x))\, dt,$$

where $u_t(\mathcal{N}_x) = (1-t)\, u_0(\mathcal{N}_x) + t\, u_1(\mathcal{N}_x)$. Since for any $y \in \mathcal{N}_x$, $\frac{\partial f_{v(x)}}{\partial u(y)}$ and $u_1(y) - u_0(y)$ are both nonnegative, so is $(F(u_1) - F(u_0))(x)$ as the integral of a nonnegative function. □

Lemma 7. *F is strictly nonexpansive for the ℓ^∞ norm: for any images $u \neq u'$,*

$$\|F(u') - F(u)\|_\infty < \|u' - u\|_\infty.$$

Proof. For fixed values of p and $a = (a_i)_{1 \leq i \leq 4}$, let us define the real mapping

$$g : c \mapsto f_p(a + c) - c\,,$$

where $a + c$ is a shorthand for $(a_i + c)_{1 \leq i \leq 4}$. We first prove that the strict decrease of g on $\mathbb{R}$ for all p and a implies the strict nonexpansiveness of F. We must prove that $F(u') < F(u) + c$ and that $F(u') > F(u) - c$ for $c = \|u' - u\|_\infty$. As $u' \leq u + c$ and as F is monotone, we have $F(u') \leq F(u + c)$. It remains to prove that $F(u + c) < F(u) + c$, i.e. that

$$\forall p \in \mathbb{N},\ \forall a \in \mathbb{R}^4,\ \forall c > 0, \quad f_p(a + c) < f_p(a) + c,$$

which is true as soon as g is strictly decreasing on $\mathbb{R}_+$. For the other inequality, we have $F(u') \geq F(u - c)$, so that it remains to prove that $F(u - c) > F(u) - c$, i.e. that

$$\forall p \in \mathbb{N},\ \forall a \in \mathbb{R}^4,\ \forall c > 0, \quad f_p(a - c) > f_p(a) - c,$$

which is true as soon as g is strictly decreasing on $\mathbb{R}_-$.

Second, we prove that g is strictly decreasing. One can prove that

$$g'(c) = \operatorname{cov}\left(S, \frac{P_p'(S + c)}{P_p(S + c)}\right) = \operatorname{cov}(S, (\log P_p)'(S + c)),$$

where S follows a distribution with p.d.f. $s \mapsto \frac{1}{Z} P_p(s + c) e^{-\lambda \sum_{i=1}^4 |s - a_i|}\, ds$ Now, P_p is positive and differentiable and $(\log P_p)'(s) = p/s - 1$, so for all c, the mapping $s \mapsto (\log P_p)'(s + c)$ is strictly decreasing on $(-c, +\infty)$. Again thanks to Lemma 5, as the distribution on S is not deterministic, we get that $g'(c)$ is negative. Hence g is strictly decreasing and the proof is complete. □

Lemma 8. *There exists a subset K of $\mathbb{R}^\Omega$ containing 0 such that $F(K) \subset K$.*

Proof. We set $G(p,c) = f_p(c\mathbb{1}_\mathcal{N}) - c$ and proceed in 4 steps:

(i) For every $p \in \mathbb{N}$, the function $c \mapsto G(p,c)$ is continuous and decreasing. Indeed, $G(p,c)$ is exactly $g(c)$, defined in the proof of Lemma 7, with $a = 0$. So it is differentiable and decreasing.

(ii) For each $p \in \mathbb{N}$, the limit of $G(p,c)$, when c goes to $+\infty$, is negative. Indeed, we have

$$G(p,c) = \frac{\int_{-c}^{+\infty} s\,(s+c)^p\, e^{-s}\, e^{-n\lambda|s|}\, ds}{\int_{-c}^{+\infty} (s+c)^p\, e^{-s}\, e^{-n\lambda|s|}\, ds} = \frac{\int_{-c}^{+\infty} s\,(1+\frac{s}{c})^p\, e^{-(s+n\lambda|s|)}\, ds}{\int_{-c}^{+\infty} (1+\frac{s}{c})^p\, e^{-(s+n\lambda|s|)}\, ds}.$$

We can apply the dominated convergence theorem on both integrals: for $g = s \mapsto s$, or $g = s \mapsto 1$, we have

$$g(s)\,\left(1+\frac{s}{c}\right)^p\, e^{-(s+n\lambda|s|)}\,\mathbb{1}_{(-c,\infty)} \xrightarrow[c\to\infty]{} g(s)\, e^{-(s+n\lambda|s|)} \text{ almost everywhere}$$

and

$$\left|g(s)\,\left(1+\frac{s}{c}\right)^p\, e^{-(s+n\lambda|s|)}\,\mathbb{1}_{(-c,\infty)}\right| \le |g(s)|\, e^{-n\lambda|s|} \quad \text{when} \quad c > p$$

because $\log\left(1+\frac{s}{c}\right) \le \frac{ps}{c} \le s$ when $c > p$, and is integrable because $n\lambda > 0$. Hence,

$$G(p,c) \xrightarrow[c\to\infty]{} \frac{\int_\mathbb{R} s\, e^{-(s+n\lambda|s|)}\, ds}{\int_\mathbb{R} e^{-(s+n\lambda|s|)}\, ds}$$

whose right-hand side is negative because its numerator equals

$$\int_0^{+\infty} s\,(e^{-(1+n\lambda)s} - e^{-(n\lambda-1)s})\, ds$$

and the function inside the integral is negative on $(0,\infty)$.

(iii) We deduce from (i) and (ii) that

$$\forall p \in \mathbb{N},\ \exists\, \mathbf{c}(p) \in \mathbb{R}, \quad c \ge \mathbf{c}(p) \Rightarrow G(p,c) \le 0.$$

(iv) With the latter definition for $p \mapsto \mathbf{c}(p)$, we define $c = \max_{x\in\Omega} \mathbf{c}(v(x))$ and $K = [0,c]^\Omega$. If $u \in K$, then $u \le c$, and as F is monotone, $F(u) \le F(c\mathbb{1}_\Omega)$. Now, as $c \ge \mathbf{c}(v(x))$, by definition of $\mathbf{c}$, $f_{v(x)}(c) \le c$ holds for each $x \in \Omega$, which exactly means that $F(u) \le F(c\mathbb{1}_\Omega) \le c$. Secondly, as $F(u)(x)$ is a ratio of nonnegative quantities, it is nonnegative and $F(u) \ge 0$. In conclusion, $F(u) \in K$.

□

Proof of Theorem 4. Since the map F is strictly non-expansive (Lemma 7) and continuous on the compact set K, there exists a real number $\alpha \in (0,1)$ such that $\|F(w_1) - F(w_2)\|_\infty \leq \alpha \|w_1 - w_2\|_\infty$ for all images $w_1, w_2 \in K$. Moreover, K is stable by F (Lemma 8), so the Banach fixed-point theorem applies and the sequence (u^n) defined in Theorem 4 converges to a fixed point of F, which is unique. The convergence is linear as $\|u^{n+1} - \widehat{u}_{\text{ICE}}\|_\infty \leq \alpha \|u^n - \widehat{u}_{\text{ICE}}\|_\infty$, or in other terms, $\|u^n - \widehat{u}_{\text{ICE}}\|_\infty = \mathscr{O}(\alpha^n)$ as $n \to \infty$. □

4.2.3 No staircasing for Poisson TV-ICE

We here prove that Poisson TV-ICE cannot produce large constant regions that were not at least partially present in the initial data.

Theorem 5. *Let $v : \Omega \to \mathbb{N}$ be a noisy image, and $\widehat{u}_{\text{ICE}}$ its denoised version. Let x and y be two pixels in Ω. Then if $\widehat{u}_{\text{ICE}}$ is constant on $\mathcal{N}_x \cup \mathcal{N}_y \cup \{x, y\}$, necessarily $v(x) = v(y)$.*

To establish the proof, we need the following:

Lemma 9. *For any constant c, the mapping $p \mapsto f_p(c\,\mathbb{1}_{\mathcal{N}})$ is strictly increasing.*

Proof. The mapping $p \mapsto f_p(c\,\mathbb{1}_{\mathcal{N}})$ can be naturally extended to real positive values of p using the right-hand part of Equation (4.7). Using the dominated convergence theorem, we can assess the differentiability of $p \mapsto f_p(c\,\mathbb{1}_{\mathcal{N}})$ and obtain

$$\frac{\partial f_p}{\partial p}(c\,\mathbb{1}_{\mathcal{N}}) = \operatorname{cov}\big(S, \log S\big)\,,$$

where S is a random variable with p.d.f. $s \mapsto \frac{1}{Z} P_p(s)\, e^{-4\lambda|s-c|}$. But as the log function is strictly increasing, using Lemma 5, we have that $\frac{\partial f_p}{\partial p}(c\,\mathbb{1}_{\mathcal{N}})$ is positive. Considering only integer values of p, we obtain the desired result. □

Proof of Theorem 5. Assume that $\widehat{u}_{\text{ICE}}$ takes a constant value $c \in \mathbb{R}$ for every pixel of the set $\mathcal{N}_x \cup \mathcal{N}_y \cup \{x, y\}$. Then taking the limit in (4.6) tells us that $c = \widehat{u}_{\text{ICE}}(x) = f_{v(x)}(\widehat{u}_{\text{ICE}}(\mathcal{N}_x)) = f_{v(x)}(c\,\mathbb{1}_{\mathcal{N}})$, and similarly $c = \widehat{u}_{\text{ICE}}(y) = f_{v(y)}(c\,\mathbb{1}_{\mathcal{N}})$. But using Lemma 9, $p \mapsto f_p(c\,\mathbb{1}_{\mathcal{N}})$ is strictly increasing, so there exists at most one value p such that $f_p(c\,\mathbb{1}_{\mathcal{N}}) = c$. We conclude that necessarily $v(x) = p = v(y)$, which finishes the proof. □

4.3 Numerical computation of Poisson TV-ICE

4.3.1 Explicit form of the Poisson TV-ICE recursion operator

Proposition 32. *The Poisson TV-ICE recursion* $u^{n+1}(x) = f_{v(x)}(u^n(\mathcal{N}_x))$ *can be written*

$$u^{n+1}(x) = \frac{\sum_{1\leq k\leq 5} c_k \, I_{a_{k-1},a_k}^{\mu_k, v(x)+2}}{\sum_{1\leq k\leq 5} c_k \, I_{a_{k-1},a_k}^{\mu_k, v(x)+1}}\,, \tag{4.9}$$

where a_1, a_2, a_3, a_4 *are the values of* $u^n(\mathcal{N}_x)$ *sorted in nondecreasing order (that is,* $0 = a_0 \leq a_1 \leq a_2 \leq a_3 \leq a_4 < a_5 = +\infty$*), and*

$$\forall k \in \{1,\ldots,5\}, \quad \mu_k = 1-(6-2k)\lambda, \quad \log c_k = \lambda\left(\sum_{j=1}^{k-1} a_j - \sum_{j=k}^{4} a_j\right)\,, \tag{4.10}$$

$$\text{and,} \quad I_{x,y}^{\mu,p} = \int_x^y s^{p-1}\, e^{-\mu s}\, ds\,, \quad \textit{for} \quad 0 \leq x \leq y \leq +\infty\,,\ \mu \in \mathbb{R}\,,\ p \geq 1\,. \tag{4.11}$$

Proof. This result is directly obtained after breaking the integration domain in Equation (4.6) so as to get rid of all absolute values. □

4.3.2 Numerical issues

In this section, we will discuss the difficulties raised by the practical evaluation of the integrals $I_{x,y}^{\mu,p}$ involved in (4.9), as well as that raised by the evaluation of the Poisson TV-ICE recursion (4.9) itself. We will then detail how these numerical obstacles can be overcome, yielding a practical Poisson TV-ICE algorithm. The Poisson TV-ICE recursion (4.9) consists in computing the ratio of sums of integrals $I_{x,y}^{\mu,p}$. Those integrals can be viewed as differences of the following generalized lower (γ_μ) or upper (Γ_μ) incomplete gamma functions,

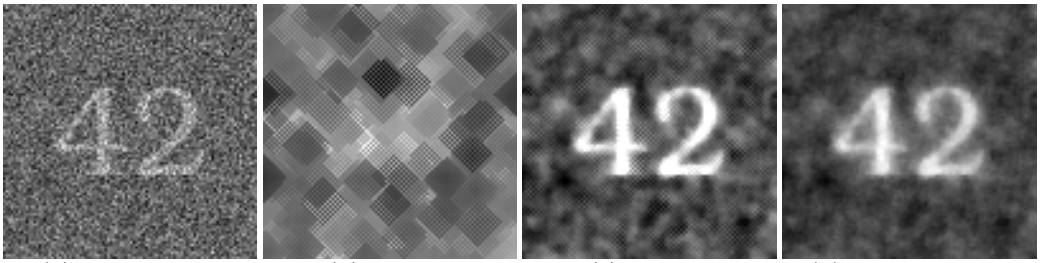
$$\gamma_\mu(p,x) = \int_0^x s^{p-1} e^{-\mu s} ds\,, \quad \Gamma_\mu(p,x) = \int_x^{+\infty} s^{p-1} e^{-\mu s} ds\,, \tag{4.12}$$

$$\text{so that} \quad I_{x,y}^{\mu,p} = \gamma_\mu(p,y) - \gamma_\mu(p,x)\,, \tag{4.13}$$

and, for $\mu > 0$,

$$I_{x,y}^{\mu,p} = \Gamma_\mu(p,x) - \Gamma_\mu(p,y) = \frac{(p-1)!}{\mu^p} - \gamma_\mu(p,x) - \Gamma_\mu(p,y)\,. \tag{4.14}$$

The computation of those quantities is far from being trivial from the practical viewpoint, as illustrated in Figure 4.1, where we show how some careless implementations of 4.9 yield numerical instabilities in the delivered image.



(a) input image (b) naive 1 (c) naive 2 (d) our algorithm

Figure 4.1: Numerical instabilities caused by some naive implementations of the recursion (4.9). The photon-count observation (a) of a synthetic image was processed using the Poisson TV-ICE scheme (with $\lambda = 0.2$), that is by setting $u^0 = 0$ and iterating several times the recursion (4.9), which was implemented in three different ways. We display in (b) the image obtained using a straightforward numerical implementation of (4.9): each integral $I^{\mu,p}_{x,y}$ is computed as the difference $\gamma_\mu(p,x) - \gamma_\mu(p,y)$, where each term $\gamma_\mu(p,z) = \mu^{-p}\gamma_1(p,\mu z)$ is computed using the algorithm proposed in the Numerical Recipes [Press et al. 1992]. Then, we compute the numerator and the denominator of (4.9), before taking the ratio. This approach is rather simple but unfortunately very unstable, since it generates some infinite (overflow) values in the image (the gray levels were saturated for the display). The image (c) was obtained with a similar approach, with a slightly improved implementation that consisted in computing all terms $\gamma_\mu(p,x)$, $\gamma_\mu(p,y)$, $I^{\mu,p}_{x,y}$ with a mantissa-exponent representation, we will explain later how such a representation is useful to get rid of the underflow and overflow errors. The image obtained with this implementation exhibits a checkerboard effect, which is due to cancellation errors occurring in the computation of the terms $I^{\mu,p}_{x,y}$. We display in (d) the image delivered by the algorithm that we propose for the evaluation of (4.9), which carefully handles the different kind of numerical errors presented above.

In order to understand the numerical difficulties inherent to the computation of $I^{\mu,p}_{x,y}$, we first focus on the particular case $\mu = 0$. This case looks rather simple, since for $\mu = 0$ we have a closed-form

$$I^{0,p}_{x,y} = \int_x^y s^{p-1}\, ds = \frac{y^p - x^p}{p}\,. \tag{4.15}$$

However, the practical computation of (4.15) raises two main difficulties:

(i) For some values of x, y, p, the quantity $I^{0,p}_{x,y}$ cannot be represented in the computer floating-point arithmetic, for instance, when it is out of the range defined by the smallest and largest representable floating-point numbers, yielding the so-called *underflow* and *overflow* numerical errors. This limitation complicates the evaluation of the ratio (4.9) which can exhibit a

non-representable numerator and denominator, although the actual ratio is representable in the floating-point arithmetic.

(ii) The numerical computation of the difference $y^p - x^p$, involved in (4.15), becomes very inaccurate when x and y are too close to each other, especially when p is high. Indeed, as two positive numbers get close to each other, the number of identical significant digits they have in common increases, so that the subtraction of one number to the other results in a loss of accuracy, called *loss of significance*, or *cancellation error*.

Instead of computing $I_{x,y}^{0,p}$ using (4.15), we can use the identity

$$y^p - x^p = y^{p-1}(y-x)\sum_{k=0}^{p-1}\left(\frac{x}{y}\right)^k ,$$

which yields a mantissa-exponent representation

$$I_{x,y}^{0,p} = \rho \cdot e^{\sigma} , \quad \text{where} \quad \rho = \frac{(y-x)}{p}\sum_{k=0}^{p-1}\left(\frac{x}{y}\right)^k , \quad \sigma = (p-1)\log y . \tag{4.16}$$

First, remark that ρ and σ in (4.16) can be both computed in double floating-point precision, which greatly extends the range of values for which the quantity $I_{x,y}^{0,p}$ can be represented. Besides, we will explain later (Proposition 33) how such a mantissa-exponent representation for each term $I_{x,y}^{\mu,p}$ involved in the Poisson TV-ICE recursion (4.9) can be used to avoid underflow and overflow in the computation of the image u^{n+1}, under a very low assumption (more precisely, under the assumption that u^{n+1} has its gray levels between $5 \cdot \mathrm{FP}_{\min}$ and $\frac{1}{5} \cdot \mathrm{FP}_{\max}$, noting $\mathrm{FP}_{\min}$ and $\mathrm{FP}_{\max}$ the smallest and largest floating-point numbers).

Second, we remark that the computation of ρ is free of cancellation errors, since $\frac{1}{p}\sum_{k=0}^{p-1}(x/y)^k$ is a sum of positive terms, and the error related to the evaluation of the difference $y - x$ is the machine precision as soon as we assume that x and y are exactly representable with the available floating-point precision (of course, we cannot avoid this assumption, since any algorithm taking x and y as input parameters in order to compute $I_{x,y}^{0,p}$ will in practice replace x and y by the nearest representable numbers $\tilde{x}$ and $\tilde{y}$, so that we are in practice interested in the accurate computation of $I_{\tilde{x},\tilde{y}}^{0,p}$, where $\tilde{x}$ and $\tilde{y}$ are exactly representable numbers).

Finally, we explained how the numerical difficulties (i) and (ii) could be overcome in the case $\mu = 0$. We will now focus on the more complicated case $\mu \neq 0$, however, since Chapter 5 will be entirely dedicated to the numerical evaluation of the quantities $\gamma_\mu(p,x)$, $\Gamma_\mu(p,y)$, and $I_{x,y}^{\mu,p}$ given (μ,p,x,y), we will here only briefly detail the main ideas of the methodology that we propose.

Numerical computation of $\gamma_\mu(p,x)$ and $\Gamma_\mu(p,x)$ with a mantissa-exponent representation

We reviewed the literature to find the available methods for the computation of $\gamma_\mu(p,x)$ and $\Gamma_\mu(p,x)$, and found that for the explored domain $|\mu x| \leq 1000$, $1 \leq p \leq 1000$ (and even far beyond in fact), the selection of the three following algorithms was satisfactory:

(i) A continued fraction for the computation of

$$\gamma_\mu(p,x) = \gamma_\mu^{\text{cfrac}}(p,x) := m^{\text{cfrac}}(\mu x, p) \cdot e^{-\mu x + p \log x},$$

where

$$m^{\text{cfrac}}(\mu x, p) = \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + \dots}}},$$

denotes a continued fraction that will be explicited in Chapter 5.

(ii) A simple recursive integration by parts formula, only valid when $\mu < 0$, yielding

$$\gamma_\mu(p,x) = \gamma_\mu^{\text{ibp}}(p,x) := m^{\text{ibp}}(\mu x, p) \cdot e^{-\mu x + p \log x},$$

where

$$m^{\text{ibp}}(\mu x, p) = \frac{1}{\mu x}\left(\frac{(p-1)!\, e^{\mu x}}{(\mu x)^{p-1}} - \sum_{k=0}^{p-1} \frac{(-1)^k\,(p-1)!\,|\mu x|^{-k}}{(p-1-k)!}\right).$$

(iii) A continued fraction for the computation of

$$\Gamma_\mu(p,x) = \Gamma_\mu^{\text{cfrac}}(p,x) := M^{\text{cfrac}}(\mu x, p) \cdot e^{-\mu x + p \log x},$$

where $M^{\text{cfrac}}(\mu x, p)$ denotes another continued fraction, that will be given in Chapter 5.

By measuring the computation time for each method for a large range of parameters μ, x, p, and thanks to a precise estimation of the relative error provided by Maple$^{\text{TM}}$, we derived a partition of the plan $(\mu x, p)$ into three domains, which makes possible the selection of at least one quantity between $\gamma_\mu(p,x)$ and $\Gamma_\mu(p,x)$, being at the same time rapidly and accurately computable using one of the formulas displayed above. This partition is displayed in Figure 4.2, and can

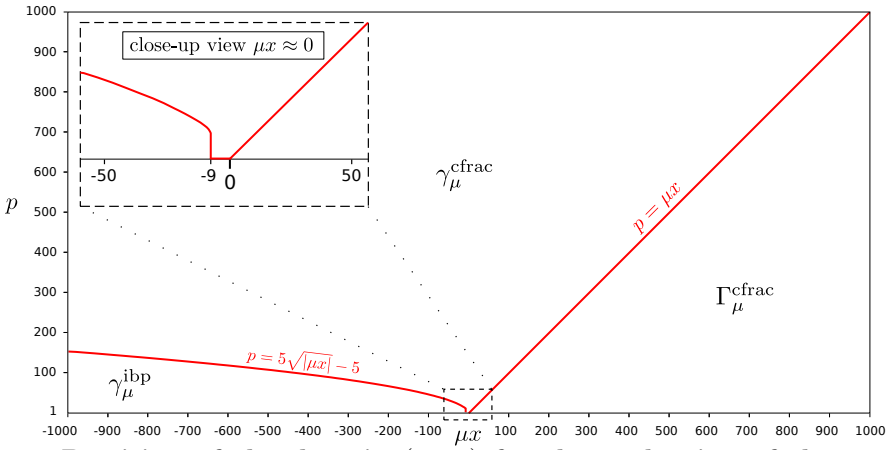


Figure 4.2: Partition of the domain $(\mu x, p)$ for the evaluation of the generalized incomplete gamma function. The rectangular domain of the plane $(\mu x, p)$ above is cut into three regions delimited by the red curves. On each region, one of the three selected algorithm is used to compute numerically either $\gamma_\mu(p,x)$ or $\Gamma_\mu(p,x)$, using the continued fraction $\gamma_\mu^{\text{cfrac}}$, the recursive integration by parts γ_μ^{ibp}, or the other continued fraction $\Gamma_\mu^{\text{cfrac}}$. The numerical experiments used to derive this partition will be presented in detail in Chapter 5.

be summarized as follow: compute $\gamma_\mu^{\text{cfrac}}(p,x)$ when $p \geq p_{\text{lim}}(\mu x)$, otherwise, compute $\Gamma_\mu^{\text{cfrac}}(p,x)$ when $\mu > 0$, or compute $\gamma_\mu^{\text{ibp}}(p,x)$ when $\mu < 0$. The parametric equation of the frontier $z \mapsto p_{\text{lim}}(z)$ being given by

$$\forall z \in \mathbb{R} \cup \{+\infty\}, \quad p_{\text{lim}}(z) = \begin{cases} 5\sqrt{|z|} - 5 & \text{if } z < -9 \\ 0 & \text{if } -9 \leq z \leq 0 \\ z & \text{otherwise.} \end{cases}$$

Of course, once one of the two quantities $\gamma_\mu(p,x)$ or $\Gamma_\mu(p,x)$ is computed, if necessary, one can recover the value of the other using

$$\Gamma_\mu(p,x) + \gamma_\mu(p,x) = \tfrac{(p-1)!}{\mu^p},$$

which is valid as soon as $\mu > 0$ (otherwise the integral $\Gamma_\mu(p,x)$ is indefinite). We will explain in Chapter 5 how we can recover a mantissa-exponent representation of $\gamma_\mu(p,x)$ from a mantissa-exponent representation of $\Gamma_\mu(p,x)$ (and vice-versa).

Numerical computation of $I_{x,y}^{\mu,p}$ with a mantissa-exponent representation

As stated before, the integral $I_{x,y}^{\mu,p}$ can be computed as a difference

$$I_{x,y}^{\mu,p} = I_{\text{diff}} := A - B,$$

using (4.13) or (4.14). Thanks to the partition derived in Figure 4.2, we are now able to select which one should be computed according to the value of (x, y, μ, p). This choice, as well as the corresponding mantissa-exponent representation of the selected difference $A - B$, will be explicited in Chapter 5. However, as also remarked before, even when A and B are accurately evaluated, the numerical computation of I_{diff} may suffer from cancellation errors if A and B are too close to each others, which in practice happens when x and y are too close to each other. In that case, we propose to approximate $I_{x,y}^{\mu,p}$ using a trapezoidal rule, and we will in Chapter 5 derive a good criterion to decide when this approximation should be used.

Numerical computation of the TV-ICE Poisson recursion (4.9)

We briefly explained above the numerical approach we propose to compute the integrals $I_{x,y}^{\mu,p}$ involved in (4.9) using a mantissa-exponent representation $I_{x,y}^{\mu,p} \approx \rho \cdot e^{\sigma}$. We refer again to Chapter 5 for the details and the numerical validation of the effective algorithm that we proposed.

We now consider that we can use this algorithm to compute a mantissa-exponent representation (ρ_k^+, σ_k^+) for each integral $I_{a_{k-1},ak}^{\mu_k,v(x)+2}$ involved in the numerator of the TV-ICE recursion (4.9), and similarly, compute a mantissa-exponent representation (ρ_k, σ_k) for each integral $I_{a_{k-1},ak}^{\mu_k,v(x)+1}$ involved in its denominator, yielding

$$\forall k \in \{1, \dots, 5\}, \quad I_{a_{k-1},ak}^{\mu_k,v(x)+2} = \rho_k^+ \cdot \exp\left(\sigma_k^+\right), \quad I_{a_{k-1},ak}^{\mu_k,v(x)+1} = \rho_k \cdot \exp\left(\sigma_k\right),$$

and thus,

$$\forall k \in \{1, \dots, 5\}, \quad \begin{cases} c_k\, I_{a_{k-1},ak}^{\mu_k,v(x)+2} &= \exp\left(\log\left(c_k\, \rho_k^+\right) + \sigma_k^+\right), \\ c_k\, I_{a_{k-1},ak}^{\mu_k,v(x)+1} &= \exp\left(\log\left(c_k\, \rho_k\right) + \sigma_k\right), \end{cases}$$

where the quantities μ_k and $\log c_k$ are those given in closed-form in (4.10). Then, the recursion (4.9) can be computed using

$$u^{n+1}(x) = \frac{\sum_{1 \le k \le 5} \exp\left(\log\left(c_k\, \rho_k^+\right) + \sigma_k^+ - M\right)}{\sum_{1 \le k \le 5} \exp\left(\log\left(c_k\, \rho_k\right) + \sigma_k - N\right)} \cdot \exp\left(M - N\right), \tag{4.17}$$

where

$$M = \max_{1 \le k \le 5} \left(\log\left(c_k \rho_k^+\right) + \sigma_k^+\right), \quad N = \max_{1 \le k \le 5} \left(\log\left(c_k \rho_k\right) + \sigma_k\right).$$

The advantage of (4.17) (compared to the sequential computation of the numerator A and the denominator B of (4.9), followed by the computation of the ratio $u^{n+1}(x) = A/B$), is its robustness to numerical underflow and overflow.

Proposition 33 (robustness of (4.17) regarding numerical underflow and overflow errors). *The numerical computation of (4.17) is free of numerical underflow and overflow errors, as soon as the effective value of $u^{n+1}(x)$ satisfies*

$$5 \cdot \mathrm{FP}_{\min} < u^{n+1}(x) < \frac{1}{5} \cdot \mathrm{FP}_{\max}\,, \tag{4.18}$$

where $\mathrm{FP}_{\min}$ and $\mathrm{FP}_{\max}$ denote respectively the smallest and highest double floating-point numbers.

Proof. Noting

$$S_1 = \sum_{1\le k\le 5} \exp\left(\sigma_k^+ + \log\left(c_k\,\rho_k^+\right) - M\right), \quad S_2 = \sum_{1\le k\le 5} \exp\left(\sigma_k + \log\left(c_k\,\rho_k\right) - N\right),$$

we have, by construction, $1 < S_1 < 5$, and $1 < S_2 < 5$. Therefore, noting $S = S_1/S_2$, we have

$$\frac{1}{5} < S < 5\,,$$

so that the numerical computation of S is free of underflow and overflow errors. Besides, using the assumption (4.18), we have

$$\mathrm{FP}_{\min} < \frac{u^{n+1}(x)}{S} < \mathrm{FP}_{\max}\,,$$

and since $\exp(M-N) = u^{n+1}(x)/S$, the computation of $\exp(M-N)$ is also free of underflow and overflow errors. Finally, both quantities S and $\exp(M-N)$ are representable with the floating-point arithmetic, so as the product $S\cdot\exp(M-N)$ which equals $u^{n+1}(x)$, and thus automatically satisfies (4.18). □

4.4 Experiments

We first checked the convergence of the proposed Poisson TV-ICE algorithm obtained by iterating the recursion (4.9) (numerically computed using (4.17)) using the initialization $u^0 = 0$. As can be seen in Figure 4.3, numerical convergence is attained for Poisson TV-ICE after a few hundred iterations, and the convergence rate is linear, as announced in Theorem 4.

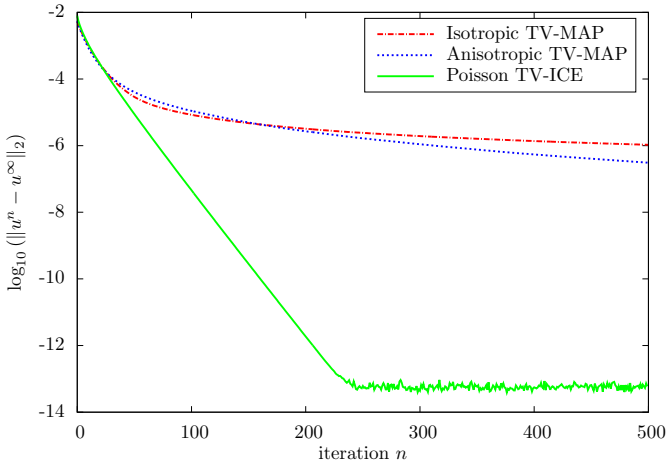


Figure 4.3: Convergence rates for TV-MAP and TV-ICE. We display in logarithmic scale the convergence rates obtained for the proposed implementation of the Poisson TV-ICE algorithm (green plain curve), and for the Chambolle-Pock implementation of the Poisson TV-MAP (using the anisotropic or isotropic discrete total variation regularizer, $\mathrm{TV}_1^{\mathrm{d}}$ or $\mathrm{TV}_2^{\mathrm{d}}$) algorithms (blue/red dashed curves). As announced in Theorem 4, the Poisson TV-ICE scheme achieves a linear convergence rate.

We then chose three images taken from areas concerned with Poisson noise (two from microscopy, and one from astronomy), and simulated a low-light observation (that is, a Poisson noise process) for each of them. Then, we restored the noisy images with both the Poisson TV-MAP and the proposed Poisson TV-ICE methods (see Figure 4.4). As predicted by the theory (Theorem 5), TV-ICE results do not exhibit staircasing effects, contrary to TV-MAP images which provide less details, in particular in the areas where the staircasing artifact causes an important loss of contrast (see, for instance, the bottom-right part of the images of the first row of Figure 4.4). This visual effect was confirmed by the systematically smaller I-divergence values obtained with TV-ICE.

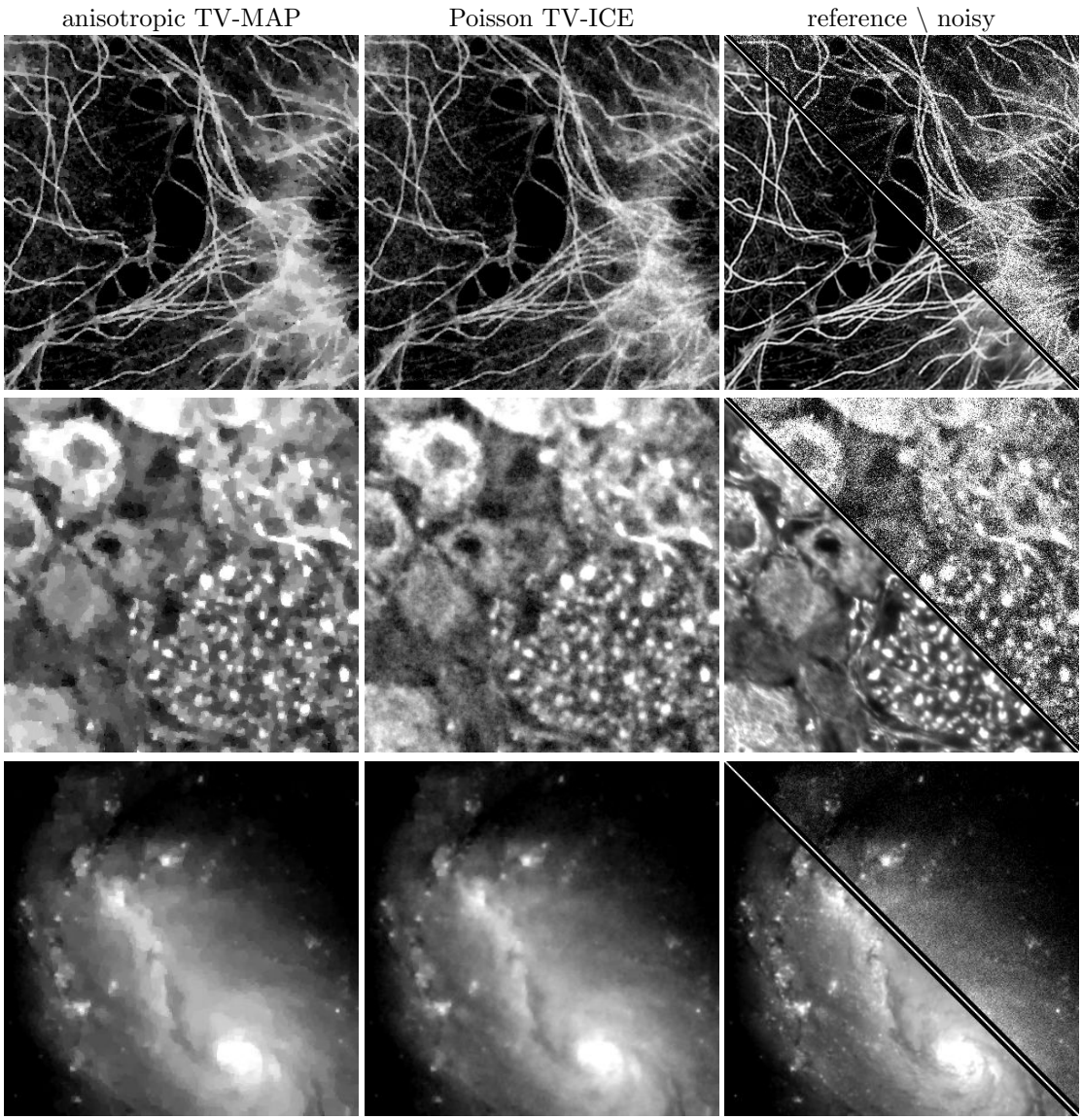


Figure 4.4: Comparison of Poisson TV-MAP and Poisson TV-ICE. Three images (first row: actin filaments and microtubules in interphase cells, second row: mouse dorsal root ganglion, third row: NGC 1672 spiral galaxy) were corrupted with Poisson noise, then denoised with the Poisson TV-MAP algorithm (left column) and the proposed Poisson TV-ICE method (middle column). For each algorithm, we selected the value of the λ parameter that achieved the smallest Csiszar I-divergence [Csiszar 1991] (a measure of distance adapted to the case of Poisson noise) between the reference image u_{ref} (bottom-left part of the images in the right column) and the denoised image $\widehat{u}$, which is defined by $\text{I-div}(u_{\text{ref}}, \widehat{u}) = \langle u_{\text{ref}} \log(u_{\text{ref}}/\widehat{u}) - (u_{\text{ref}} - \widehat{u}) \,,\, \mathbb{1}_\Omega \rangle$. One can clearly see that TV-MAP results exhibit staircasing effects and an associated loss of details in the corresponding flat regions; on the contrary, the TV-ICE images are more natural and more faithful to the fine details of the reference, especially in the regions where TV-MAP produces staircasing. Note that in order to increase the readability of the figure, the dynamic of the images has been linearly amplified, causing some (limited) saturation in dark and white areas. Image sources: www.cellimagelibrary.org and www.wikimedia.org.

4.5 Conclusion and perspectives

We proposed a variant of the recent TV-ICE denoising method adapted to the special case of Poisson noise. The absence of staircasing and the better-quality restored images attested by experiments make Poisson TV-ICE a good alternative to Poisson TV-MAP, and suggests that it could be interesting to derive Poisson TV-ICE variants for more complex inverse problems involving TV terms.

The linear convergence rate of the method is appealing but is not sufficient to compensate for the heavy computations required by the form of the recursion operator (several evaluations of the exponential and logarithm functions are required for each pixel). In our current (non-optimized) implementation, one iteration of TV-ICE is approximately 100 times slower than one iteration of TV-MAP. However, further work could focus on the fast approximation of TV-ICE, and the precise implementation we here proposed would be useful in that context to check the quality of the approximation.

As in the Gaussian case, the generalization of the proposed algorithm to three-dimensional images (or more), or to larger neighborhood systems, is straightforward. However, the comparison with the Poisson TV-LSE variant is, both from a theoretical or practical point of view, still open.

Chapter 5

Fast and Accurate Evaluation of a Generalized Incomplete Gamma Function

Contents

The content of this chapter has been submitted to the journal ACM Transactions on Mathematical Softwares (ACM-TOMS) in June 2016. A software implemented in C language was also submitted and is currently under review. This code is available at http://www.math-info.univ-paris5.fr/~rabergel/softwares/deltagammainc.zip.

Abstract

We propose a computational procedure to evaluate the generalized incomplete gamma function $\int_x^y s^{p-1}\,e^{-\mu s}\,ds$ for $0 \leq x < y \leq +\infty$, a real number $\mu \neq 0$ and a positive integer p. Our approach consists in selecting, according to the value of the parameters x, y, μ, p, the fastest and most accurate estimate among series expansions, continued fractions, recursive integration by parts, or, when $x \approx y$, a first order trapezoidal rule. We show that the accuracy reached by our algorithm is nearly optimal for a large range of parameters.

5.1 Introduction

In this chapter, we focus on the computation of a generalized incomplete gamma function that will be defined below. Let us first recall the definition of the gamma function,

$$\forall a > 0, \quad \Gamma(a) = \int_0^{+\infty} s^{a-1}\,e^{-s}\,ds\,. \tag{5.1}$$

The lower and upper incomplete gamma functions are respectively obtained by allowing the integration domain to vary in (5.1),

$$\forall a > 0, \quad \forall x \geq 0, \quad \gamma(a,x) = \int_0^{x} s^{a-1}\,e^{-s}\,ds \quad \text{and} \quad \Gamma(a,x) = \int_x^{+\infty} s^{a-1}\,e^{-s}\,ds\,. \tag{5.2}$$

The gamma function is usually viewed as an extension of the factorial function since it satisfies $\Gamma(a) = (a-1)!$ for any positive integer a. Note that the gamma function can also be defined for all complex numbers a with positive real part, using the same convergent improper integral as in (5.1), and can even be extended by analytic continuation to all complex numbers except the nonpositive integers, that is, to $a \in \mathbb{C} \setminus \{0, -1, -2, -3, \ldots\}$. Some similar extensions are also available for their incomplete variants $\gamma_\mu(a,x)$ and $\Gamma_\mu(a,x)$. These special functions arise in many areas, such as astronomy and astrophysics [Cannon and Vardavas 1974, Hills 1975], Rayleigh scattering [Kissel et al. 1980], quantum gravity [Bleicher and Nicolini 2010], networks [Moreno et al. 2002], financial mathematics [Linetsky 2006], image analysis [Robin et al. 2010], etc. (see [Chaudhry and Zubair 2001] for more examples). From the mathematical viewpoint, the computation of incomplete gamma functions is typically required in applications involving the

evaluation of χ^2 distribution functions, exponential integrals, error functions (erf), cumulative Poisson or Erlang distributions, etc. Their practical numerical evaluation is still subject to some flourishing research in the modern litterature. The first practical algorithm dedicated to the numerical evaluation of the incomplete gamma functions was, to the best of our knowledge, proposed in [Bhattacharjee 1970]. It consists in evaluating the ratio $\gamma(a,x)/\Gamma(a)$ using a series expansion when $0 < a \leq x < 1$ or $0 \leq x < a$, or the ratio $\Gamma(a,x)/\Gamma(a)$ using a continued fraction in the remaining part of the domain $\{x \geq 0, a > 0\}$. The same strategy is also used in [Press et al. 1992]. Gautschi [1979] proposed another computational procedure, based on Taylor's series and continued fractions, to evaluate those two functions in the region $\{x \geq 0, a \in \mathbb{R}\}$ (in fact, for $a \leq 0$, Tricomi's version [Tricomi 1950, Gautschi 1998] of the lower incomplete gamma function, which remains real for any real numbers x, a, is considered). The criterion proposed in [Bhattacharjee 1970] to decide which one of the two integrals should be computed according to the value of (x,a) is refined, and a more suitable normalization is employed, which extends the range over which those two functions can be represented within standard double precision arithmetics. More recently, Winitzki [2003] focused on the computation of the upper incomplete gamma function and used some series expansions, a continued fraction (due to Legendre), some recurrence relations, or, for large values of x, an asymptotic series. The precision of the approximation is controlled by estimating the number of terms required to reach a given absolute precision according to the values of x and a. However, the study is not considered from the practical point of view, and no algorithm or experimental validation are provided to assess the numerical stability of the proposed method. In [Guseinov and Mamedov 2004], the lower and upper incomplete gamma functions are computed using backward and forward recurrence relations. The experimental validation is done for the range $0.001 \leq x \leq 100$ and $0 < a \leq 100$, which is relatively large in comparison to the numerical validations usually proposed in the litterature. We will also use, for a particular region of the quarter plane $\{(a,x),\ a > 0,\ x < 0\}$, a recurrence relation to compute the lower incomplete gamma function. However, we shall see that in the region $x > 0$, we experimentally achieve a faster convergence by using some continued fractions.

In the present chapter, we consider the more general case of the generalized incomplete gamma function, defined by

$$I_{x,y}^{\mu,p} = \int_x^y s^{p-1}\, e^{-\mu s}\, ds, \quad \text{for} \quad 0 \leq x < y \leq +\infty,\ p > 0,\ \mu \in \mathbb{R} \setminus \{0\}, \tag{5.3}$$

and we restrict the study to integer values of p (even though all the algorithms we

propose also work for non-integer values of p when $\mu x > 0$). Notice that $y = +\infty$ is only allowed when $\mu > 0$, otherwise the integral is equal to $+\infty$. Note also that thanks to the rescaling relation

$$I^{\mu,p}_{x,y} = |\mu|^{-p}\, I^{\varepsilon,p}_{|\mu|x,|\mu|y}, \quad \text{where} \quad \varepsilon = \frac{\mu}{|\mu|}\,, \tag{5.4}$$

we could restrict the study, without any loss of generality, to $\mu \in \{-1, 1\}$. We will however not adopt this restriction since it does not simplify the study, even though the numerical evaluations that we propose are limited to $\mu = \pm 1$, which simplifies the experimental validation. The computation of $I^{\mu,p}_{x,y}$ will be closely related to that of the generalized lower (γ_μ) and upper (Γ_μ) incomplete gamma functions, which we naturally define by

$$\forall \mu \in \mathbb{R}, \quad \gamma_\mu(p,x) = \int_0^x s^{p-1}\, e^{-\mu s}\, ds\,,$$
$$\text{and} \quad \forall \mu > 0, \quad \Gamma_\mu(p,x) = \int_x^{+\infty} s^{p-1}\, e^{-\mu s}\, ds\,. \tag{5.5}$$

Note that when $\mu > 0$ in (5.3) or (5.5), the change of variable $t = \mu s$ would lead us back to the standard definitions of the incomplete gamma functions (up to the multiplicative factor μ^{-p}), but this is not the case when $\mu < 0$. The possibility to evaluate the lower incomplete gamma function $\gamma(p,x)$ with a negative argument x (which amounts to compute $\gamma_\mu(p,|x|)$ with $\mu = -1$) is explored in [Thompson 2013], but in another situation than ours, since he focused on the case $p = n + \frac{1}{2}$, $n \in \mathbb{Z}$.

The generalized incomplete gamma function (5.5) was actually previously introduced in [Fullerton 1972], under the slightly different form

$$J^a_{x_1,x_2} = e^{x_1} \int_{x_1}^{x_2} |s|^{a-1}\, e^{-s}\, ds\,, \quad \text{for any } (x_1,x_2) \in \mathbb{R}^2, \text{ and } a > 0\,. \tag{5.6}$$

The integrals I and J are closely related since one easily checks that

$$\forall x, y,\ 0 < x < y, \quad \forall p > 0, \quad I^{\mu,p}_{x,y} = \begin{cases} \mu^{-p}\, e^{-\mu x}\, J^p_{\mu x,\mu y} & \text{if } \mu > 0\,, \\ |\mu|^{-p}\, e^{-\mu y}\, J^p_{\mu y,\mu x} & \text{if } \mu < 0\,. \end{cases} \tag{5.7}$$

Our parametrization of the integral I using the scale parameter μ will be helpful to avoid the absolute values in the integral, which would inevitably have involved the distinction of the cases $x_1 > x_2$ and $x_1 \le x_2$ in our study. The numerical

evaluation of the generalized incomplete gamma function $I_{x,y}^{\mu,p}$ has found some applications in the field of astronomy, for instance in [Hills 1975], where its computation was needed to model the dynamical evolution of stellar clusters. It was more recently needed in the field of image processing, in [Abergel et al. 2015], where the accurate computation of $I_{x,y}^{\mu,p}$ for a large range of parameters was at the heart of a denoising algorithm for the restoration of images corrupted with Poisson noise. Unfortunately, Fullerton's algorithm, which was not validated for a large range of parameters, presents several weaknesses. As pointed out in [Schoene 1978], for some values of the parameters, the algorithm suffers from numerical instabilities, yielding for instance, a computed integral with incorrect sign, or zero digit of precision. We also observed some overflow issues when we tested the algorithm on a higher range of parameters (typically when $p \approx 10^2$ or higher, but also for many other parameter settings).

Note also that a numerical procedure specific to the evaluation of $I_{x,y}^{\mu,p}$ is available into the scientific computing software *Mathematica* (see [Wolfram Research Inc 1988], but also [Wolfram Research Inc 1998] for the online evaluation of $I_{x,y}^{\mu,p}$). Unfortunately, *Mathematica*'s algorithms are not currently disclosed to the public.

Let us now consider the numerical evaluation of $I_{x,y}^{\mu,p}$. This integral can be computed as a difference of generalized lower (γ_μ) and upper (Γ_μ) incomplete gamma functions, since for any $\mu \in \mathbb{R}$, we have

$$I_{x,y}^{\mu,p} = \gamma_\mu(p,y) - \gamma_\mu(p,x)\,, \tag{5.8}$$

and for $\mu > 0$, we have

$$I_{x,y}^{\mu,p} = \Gamma_\mu(p,x) - \Gamma_\mu(p,y) = \frac{\Gamma(p)}{\mu^p} - \gamma_\mu(p,x) - \Gamma_\mu(p,y)\,. \tag{5.9}$$

The effective computation of $I_{x,y}^{\mu,p}$ using (5.8) or (5.9) raises several numerical issues:

1. For some values of the parameters, the generalized incomplete gamma functions γ_μ and Γ_μ cannot be represented in the computer floating point arithmetic (for example when they exceed $1.9 \cdot 10^{308}$, the largest double precision number). To solve that issue, we will represent all integrals in (5.5) under the form $\rho \cdot e^\sigma$, where ρ and σ are floating point numbers with double precision;

2. The possibility to efficiently compute $\gamma_\mu(p,x)$ and $\Gamma_\mu(p,x)$ depends on the values of the parameters μ, p, x, or more precisely of μx and p because of the scaling relation (5.4). We derived a division of the plane $(\mu x, p)$ allowing an efficient computation of these two functions for each parameter set (μ, p, x);

3. When $I_{x,y}^{\mu,p}$ is computed as the difference $A-B$, the result may be inaccurate if A and B are close to each other (the well-known *cancellation* effect in floating-point arithmetic), which typically happens in (5.8)-(5.9) when x and y are very close to each other. In that case, the integral $I_{x,y}^{\mu,p}$ is well approximated by a first order approximation of the integral. We found a good criterion, to decide when this approximation should be used.

In particular, the issue (1) detailed above is of great importance when some integrals of the kind $I_{x,y}^{\mu,p}$ appear into more complicated mathematical expressions, such as in [Abergel et al. 2015], where the computation of a ratio of sums of generalized incomplete gamma functions is involved, with a numerator and a denominator that may both exceed the highest representable double floating point number, although the ratio itself is representable in the standard computer floating-point arithmetic.

This chapter is organized as follows. In Section 5.2, we recall some mathematical methods based on series expansion, fraction continuation, or recursive integration by parts, that can be used for the numerical evaluation with a mantissa-exponent representation of the generalized lower (Section 5.2.1) and upper (Section 5.2.2) incomplete gamma functions γ_μ and Γ_μ. In Section 5.2.3, we derive theoritical accuracy bounds achievable with such a mantissa-exponent representation, and check experimentally in Section 5.2.4 that we can achieve these bounds by selecting the appropriate method (continued fraction or integration by parts) depending on the values of the parameters μ, p, x. In Section 5.3, we focus on the practical evaluation of the generalized incomplete gamma function $I_{x,y}^{\mu,p}$ (that is, with arbitrary finite values of x and y). The numerical evaluation of this integral is done by means of a difference (5.8)-(5.9), or, when $x \approx y$, using the first order trapezoidal rule. In Section 5.5, our algorithm is compared with the one of Fullerton, and is shown to exhibit a much greater accuracy for a large range of parameters. We finally conclude in Section 5.6 and discuss some perspectives.

5.2 Numerical computation of the generalized lower and upper incomplete gamma functions

5.2.1 Evaluation of the generalized lower incomplete gamma function

Given $p \geq 1$, $x > 0$ and $\mu \neq 0$, we detail below how the generalized lower incomplete gamma $\gamma_\mu(p,x)$ can be evaluated with a mantissa-exponent representation of the kind

$$\gamma_\mu(p,x) = m(\mu x, p) \cdot e^{n(\mu,x,p)}, \quad \text{where} \quad n(\mu,x,p) = -\mu x + p \log x \,. \tag{5.10}$$

The mantissa $m(\mu x, p)$ will be determined using either a series expansion, or a continued fraction, or, in the case $\mu < 0$, a recursive integration by parts. This yields three different computational methods for the evaluation of $\gamma_\mu(p,x)$. Notice that in the following, we will need to extend the representation (5.10) to the particular cases $x = 0$ and (only when $\mu > 0$) $x = +\infty$. For that purpose we set $m(0,p) = 0$, $n(\mu,0,p) = -\infty$ (taking the obvious convention that $0 \cdot e^{-\infty} = 0$), and in the case $\mu > 0$, we set $m(+\infty,p) = 1$ and $n(\mu,+\infty,p) = \log\Gamma(p) - p\log\mu$. The practical computation of $\log\Gamma(p)$ will be discussed in Section 5.4.

Series expansion

Writing the Taylor series expansion with order $p-1$ and integral remainder of the exponential function near zero we get

$$\begin{aligned} e^{\mu x} &= \sum_{k=0}^{p-1} \frac{(\mu x)^k}{k!} + \int_0^{\mu x} \frac{(\mu x - t)^{p-1}}{(p-1)!}\, e^t \, dt \\ &\underset{s = x - t/\mu}{=} e^{\mu x} - \sum_{k=p}^{+\infty} \frac{(\mu x)^k}{k!} + \frac{\mu^p\, e^{\mu x}}{(p-1)!} \int_0^x s^{p-1}\, e^{-\mu s}\, ds \,. \end{aligned}$$

This yields a series expansion of the generalized lower incomplete gamma function under the form $\gamma_\mu(p,x) = \gamma_\mu^{\text{ser}}(p,x)$ where

$$\gamma_\mu^{\text{ser}}(p,x) = m^{\text{ser}}(\mu x, p) \cdot e^{-\mu x + p\log x}\,,$$

$$\text{and} \quad m^{\text{ser}}(\mu x, p) = \sum_{k=0}^{+\infty} \frac{(p-1)!}{(k+p)!} (\mu x)^k \,. \tag{5.11}$$

Although the power series $m^{\text{ser}}(\mu x, p)$ defined above has an infinite radius of convergence, its convergence can be quite slow and numerically unstable according to the values of p and μx. It is suggested in [Press et al. 1992] to evaluate $\gamma_\mu(p,x)$ using (5.11) as soon as $\frac{|\mu x|}{p+1} < 1$; however, according to our experiments, a better convergence rate can be obtained by using a continued fraction development. Thus, we shall not use (5.11) in the algorithm we propose.

Continued fraction

Let us consider the confluent hypergeometric function M, defined by

$$M(a,b,z) = \sum_{n=0}^{+\infty} \frac{a^{(n)}}{b^{(n)}} \frac{z^n}{n!}, \quad \text{where} \quad \forall \alpha,\ \alpha^{(0)} = 1 \text{ and } \alpha^{(n)} = \alpha(\alpha+1)\cdots(\alpha+n-1).$$

Since for any (b,z) we have $M(0,b,z) = 1$, Equation (5.11) rewrites as

$$\gamma_\mu(p,x) = \frac{M(1,p+1,\mu x)}{p \cdot M(0,p,\mu x)} \cdot e^{-\mu x + p\log x}. \tag{5.12}$$

As detailed in [Olver et al. 2010, DLMF, Cuyt et al. 2008, Jones and Thron 1980], the ratio $\frac{M(a,b,z)}{M(a+1,b+1,z)}$ can be continued for any $z \in \mathbb{C}$, as soon as $a \notin \mathbb{Z} \setminus \mathbb{N}$ and $a - b \notin \mathbb{N}$. Under this assumption (which will be satisfied here, since we will consider the setting $a = 0$, $b = p$), and using the usual notation for continued fractions,

$$\frac{\alpha_1}{\beta_1+}\ \frac{\alpha_2}{\beta_2+}\ \frac{\alpha_3}{\beta_3+}\ \cdots = \frac{\alpha_1}{\beta_1 + \frac{\alpha_2}{\beta_2 + \frac{\alpha_3}{\beta_3 + \ldots}}},$$

we get

$$\frac{M(a,b,z)}{M(a+1,b+1,z)} = 1 + \frac{u_1}{1+}\ \frac{u_2}{1+}\ \frac{u_3}{1+}\ \cdots,$$

where $\forall n \geq 0$, $u_{2n+1} = \frac{(a-b-n)z}{(b+2n)(b+2n+1)}$, $u_{2n} = \frac{(a+n)z}{(b+2n-1)(b+2n)}$. Writing the inverse ratio (with $a = 0$ and $b = p$), and after basic manipulations of the continued fraction, we obtain

$$\frac{M(1,p+1,\mu x)}{p \cdot M(0,p,\mu x)} = \frac{a_1}{b_1+}\ \frac{a_2}{b_2+}\ \frac{a_3}{b_3+}\ \cdots,$$

where $a_1 = 1$ and $\forall n \geq 1$, $a_{2n} = -(p-1+n)\cdot\mu x$, $a_{2n+1} = n \cdot \mu x$ and $b_n = p-1+n$. Therefore, Equation (5.12) rewrites as $\gamma_\mu(p,x) = \gamma_\mu^{\text{cfrac}}(p,x)$, where

$$\gamma_\mu^{\text{cfrac}}(p,x) = m^{\text{cfrac}}(\mu x, p) \cdot e^{-\mu x + p\log x},$$

$$\text{and} \quad m^{\text{cfrac}}(\mu x, p) = \frac{a_1}{b_1+}\ \frac{a_2}{b_2+}\ \frac{a_3}{b_3+}\ \cdots \tag{5.13}$$

The above defined continued fraction $m^{\text{cfrac}}(\mu x, p)$ can be evaluated thanks to the modified Lentz's method [Lentz 1976, Thompson and Barnett 1986] which is also described in [Press et al. 1992] and that we recall in Algorithm 10 for the reader's convenience, with however a slight adaptation of the initialization process since we observed some instabilities when using that described in [Press et al. 1992] (see comment in Algorithm 10). This continued fraction converges for any value of μx and the convergence is fast as it requires in general less than 20 approximants to converge, except when $\mu > 0$ and $\mu x \approx p$ (where it takes around p approximants) or when $\mu < 0$ and p is small (several hundred of approximants needed for $p \leq 20$ and $|\mu x| \leq 1000$). Note that $m^{\text{cfrac}}(\mu, px)$ becomes huge when μx is chosen too large compared to p, and numerical instabilities can appear. For that reason, we will restrict the use of (5.13) to a subdomain of the plane $(\mu x, p)$, as discussed in Section 5.3.

Algorithm 10: Modified Lentz's method for continued fractions evaluation.

Input: Two real-valued sequences $\{a_n\}_{n\geq 1}$ and $\{b_n\}_{n\geq 1}$, with $b_1 \neq 0$.

Output: Accurate estimate f of the continued fraction $\frac{a_1}{b_1+}\,\frac{a_2}{b_2+}\,\frac{a_3}{b_3+}\,\cdots$

Initialization:

$d_m \leftarrow 10^{-300}$ `// Number near the minimal floating-point value`

$f \leftarrow \frac{a_1}{b_1}$; $C \leftarrow \frac{a_1}{d_m}$; $D \leftarrow \frac{1}{b_1}$; $n \leftarrow 2$ `// see the algorithm footnote`

repeat

- $D \leftarrow D \cdot a_n + b_n$
- **if** $D = 0$ **then** $D \leftarrow d_m$ $C \leftarrow b_n + \frac{a_n}{C}$
- **if** $C = 0$ **then** $C \leftarrow d_m D \leftarrow \frac{1}{D}$
- $\Delta \leftarrow C \cdot D$
- $f \leftarrow f \cdot \Delta$
- $n \leftarrow n + 1$

until $|\Delta - 1| < \varepsilon_{machine}$

return f

In the initialization step, we manually performed the first pass $n = 1$ of the modified Lentz's algorithm, since we observed some instabilities with the initialization $f = C = d_m$, $D = 0$, presented in [Press et al. 1992]. Indeed, the setting $C = d_m$ may yield $C = +\infty$ after the pass $n = 1$ (when a_1/d_m exceed the highest representable number), and then $\Delta = f = +\infty$, which propagates through the next iterations. By computing manually the first pass, even when the initialization $C = a_1/d_m$ yields $C = +\infty$, the pass $n = 2$ yields $C = b_2 + a_2/C = b_2$, which has a finite value.

Integration by parts

Since we only consider integer values of the parameter p, the generalized lower incomplete gamma function $\gamma_\mu(p,x)$ can be written as a closed-form formula using a recursive integration by parts. Considering the case $\mu < 0$, one gets

$$\gamma_\mu(p,x) = \gamma_\mu^{\mathrm{ibp}}(p,x) := m^{\mathrm{ibp}}(\mu x, p) \cdot e^{-\mu x + p \log x}\,,$$
$$\text{where} \quad m^{\mathrm{ibp}}(\mu x, p) = \frac{1}{\mu x}\left(\tfrac{(p-1)!\, e^{\mu x}}{(\mu x)^{p-1}} - \sum_{k=0}^{p-1} \tfrac{(p-1)!(\mu x)^{-k}}{(p-1-k)!} \right). \quad (5.14)$$

Although the computation of $\gamma_\mu^{\mathrm{ibp}}(p,x)$ is not efficient in general, it happens to be faster than $\gamma_\mu^{\mathrm{cfrac}}(p,x)$ for small values of p. We must however be carefull when computing the alternating sum $m^{\mathrm{ibp}}(\mu x, p)$ since, as usual with alternating sums, it may suffer from dramatic cancellation errors.

Let $t = |\mu x| > 0$, we rewrite (5.14) into

$$m^{\mathrm{ibp}}(-t,p) = \frac{1}{t}\left(\tfrac{(-1)^p (p-1)!\, e^{-t}}{t^{p-1}} + s(t) \right),$$
$$\text{where} \quad s(t) = \sum_{k=0}^{p-1} (-1)^k \frac{(p-1)!\, t^{-k}}{(p-1-k)!}\,. \quad (5.15)$$

By grouping by two the consecutive terms with indexes $k = 2l$ and $k = 2l+1$ of the alternating sum $s(t)$, we get

$$s(t) = \tilde{s}(t) := \sum_{l=0}^{\lfloor \frac{p-2}{2} \rfloor} \frac{(p-1)!\, t^{-(2l+1)}}{(p-1-2l)!}\,(t - (p-1-2l)) + \varepsilon_p(t)\,, \quad (5.16)$$

where $\lfloor z \rfloor$ denotes the integer part of z, and the residual term $\varepsilon_p(t)$ is defined by

$$\varepsilon_p(t) = \begin{cases} (p-1)!\, t^{-(p-1)} & \text{if } p \text{ is odd} \\ 0 & \text{otherwise.} \end{cases}$$

Let us now assume that $t \geq \max(1, p-1)$. First, using $t \geq p-1$, we see that all terms in the sum $\tilde{s}(t)$ are nonnegative, so that we can evaluate $\tilde{s}(t)$, which has exactly the same value as the alternating sum $s(t)$, without any cancellation error using (5.16). It follows that, when p is even, we have

$$m^{\mathrm{ibp}}(-t,p) = \frac{1}{t}\left(\frac{(p-1)! e^{-t}}{t^{p-1}} + \tilde{s}(t) \right),$$

which is a sum of positive terms, so that it does not suffer from cancellation error. When p is odd, (5.15) yields

$$m^{\text{ibp}}(-t,p) = \frac{1}{t}\left(-\frac{(p-1)!e^{-t}}{t^{p-1}} + \tilde{s}(t)\right). \tag{5.17}$$

Noting $\alpha(t) = \frac{(p-1)!e^{-t}}{t^{p-1}}$ and using the fact that $t \geq 1$, we get

$$\frac{\tilde{s}(t)}{\alpha(t)} \geq \frac{\varepsilon_p(t)}{\alpha(t)} = \exp(t) \geq \exp(1),$$

which ensures that no cancellation error occurs when computing the difference between $\tilde{s}(t)$ and $\alpha(t)$, involved in (5.17). Finally, we are able to evaluate (5.14) without cancellation in the region $t \geq \max(1, p-1)$.

Last, from $t > p-1$, we infer that the sequence $\{a_k(t)\}_{k\geq 0}$ defined by

$$\forall k \geq 0, \quad a_k(t) = \begin{cases} \frac{(p-1)!\,t^{-k}}{(p-1-k)!} & \text{if } k \leq p-1 \\ 0 & \text{otherwise,} \end{cases}$$

is nonincreasing, with limit 0. It follows that the remainder $r_n(t) = \sum_{k=n+1}^{+\infty}(-1)^k a_k(t)$ of the alternating series $s(t) = \sum_{k=0}^{+\infty}(-1)^k a_k(t)$ satisfies $|r_n(t)| \leq a_{n+1}(t)$, so that we can numerically estimate $s(t)$ with the partial sum $s_n(t) = \sum_{k=0}^{n}(-1)^k a_k(t)$ as soon as

$$a_{n+1}(t) \leq |s_n(t)| \cdot \varepsilon_{\text{machine}},$$

which may occur for $n < p-1$, making possible in that case to save some computation time. In practice, we compute $s(t) = \tilde{s}(t)$ with (5.16) instead of (5.15), but this stopping criterion can be easily evaluated at each iteration of the summation procedure. Indeed, remarking that the sequence $\{a_{2l}(t) - a_{2l+1}(t)\}_{l\geq 0}$ is positive and nonincreasing (because $t > p-1$), we get

$$\forall l \in \mathbb{N}, \quad a_{2l+2}(t) \leq a_{2l}(t) - a_{2l+1}(t) + a_{2l+3}(t) \leq a_{2l}(t) - a_{2l+1}(t),$$

so that

$$\forall l \in \mathbb{N}, \quad |r_{2l+1}(t)| \leq a_{2l+2}(t) \leq |a_{2l}(t) - a_{2l+1}(t)|.$$

This yields Algorithm 11.

There is a more elegant way to avoid the cancellation errors in the computation of $s(t)$, inspired from Horner's algorithm for polynomial evaluation. It consists in computing

$$s(t) = 1 - \frac{p-1}{t}\cdot\left(1 - \frac{p-2}{t}\cdot\left(1 - \frac{p-3}{t}\cdot\left(\dots\left(1 - \frac{1}{t}\right)\right)\right)\dots\right),$$

Algorithm 11: Compute $m^{\text{ibp}}(\mu x, p) = \frac{1}{\mu x}\left(\frac{(p-1)!\, e^{\mu x}}{(\mu x)^{p-1}} - \sum_{k=0}^{p-1} \frac{(p-1)!(\mu x)^{-k}}{(p-1-k)!}\right)$.

Input: Two real numbers $x \in \mathbb{R}_+$, $\mu < 0$, and a positive integer p, satisfying $|\mu x| > \max(1, p-1)$.

Output: An accurate estimate of $m^{\text{ibp}}(\mu x, p)$.

Initialization: $t \leftarrow |\mu x|$; $c \leftarrow \frac{1}{t}$; $d \leftarrow p-1$; $s \leftarrow c \cdot (t-d)$; $l \leftarrow 1$; *stop* $\leftarrow$ *false*

repeat

  $c \leftarrow \frac{d(d-1)}{t^2}$

  $d \leftarrow d-2$

  $\Delta \leftarrow c(t-d)$ // Now $\Delta = a_{2l}(t) - a_{2l+1}(t)$

  $s \leftarrow s + \Delta$ // Now $s = s_{2l+1}(t) = \sum_{k=0}^{2l+1}(-1)^k a_k(t)$

  if $\Delta < s \cdot \varepsilon_{machine}$ **then** *stop* $\leftarrow$ *true* $l \leftarrow l+1$

until $l > \lfloor \frac{p-2}{2} \rfloor$ *or stop*

if *(not stop) and (p is odd)* **then** $s \leftarrow s + \frac{d\,c}{t}$ // add the term $\varepsilon_p(t) = (p-1)!\, t^{-(p-1)}$

return $\frac{1}{t}\left((-1)^p \cdot e^{-t+\log(p-1)!-(p-1)\log(t)} + s\right)$

or more precisely, $s(t) = v_{p-1}(t)$, where $\{v_n(t)\}_{n\geq 1}$ is the sequence defined recursively by

$$\forall n \geq 1, \quad v_n(t) = \begin{cases} 1 - \frac{1}{t} & \text{if } n = 1\,, \\ 1 - \frac{n}{t} \cdot v_{n-1}(t) & \text{if } n \geq 2\,. \end{cases}$$

Assuming $t \geq 2p$, one can show that the terms of $\{v_n\}_{1\leq n\leq p-1}$ remain in $(\frac{1}{2}, 1)$, so that they can be evaluated without cancellation errors. However, a drawback of this approach is the absence of a simple stopping criterion making possible to end up the computation of $s(t)$ before computing all the first $p-1$ terms of the sequence $\{v_n\}_{n\geq 1}$.

Algorithm for the evaluation of the generalized lower incomplete gamma function

The evaluation of $\gamma_\mu(x, p)$ using one of the computation methods presented above can be done using Algorithm 12. This algorithm returns a mantissa-exponent representation (m, n) of $\gamma_\mu(x, p)$, such as $\gamma_\mu(x, p) = m \cdot e^n$, and returns

$m = 0, n = -\infty$, when $\gamma_\mu(x,p) = 0$ (this will be more generally the case for the mantissa-exponent representations returned by all algorithms we propose).

Algorithm 12: Evaluation of $\gamma_\mu(x,p) = \int_0^x s^{p-1}\, e^{-\mu s}\, ds$ using a series expansion, a continued fraction, or a recursive integration by parts.

Input: Two numbers $x \in \mathbb{R}_+ \cup \{+\infty\}$, $\mu \in \mathbb{R} \setminus \{0\}$, and a positive integer p. Notice that the value $x = +\infty$ is allowed only when $\mu > 0$.

Output: Two numbers $m \in \mathbb{R}$ and $n \in \mathbb{R} \cup \{-\infty\}$ such as $\gamma_\mu(p,x) = m \cdot e^n$.

if $x = 0$ **then** $(m,n) \leftarrow (0, -\infty)$ **else if** $x = +\infty$ *and* $\mu > 0$ **then** $(m,n) \leftarrow (1, \log\Gamma(p) - p\log\mu)$ **else**
  switch *choice of the evaluation method for the mantissa* **do**
    case *series expansion*
      $m \leftarrow m^{\text{ser}}(\mu x, p)$; // using Equation (5.11)
    case *continued fraction*
      $m \leftarrow m^{\text{cfrac}}(\mu x, p)$; // using Equation (5.13) and Algorithm 10
    case *recursive integration by parts (only when* $\mu < 0$ *and* $|\mu x| > \max(1, p-1)$*)*
      $m \leftarrow m^{\text{ibp}}(\mu x, p)$; // using Equation (5.14)
  $n \leftarrow -\mu x + p\log x$

return (m,n)

5.2.2 Evaluation of the generalized upper incomplete gamma function

Let $p \geq 1$, $x > 0$ and $\mu > 0$. The evaluation of $\Gamma_\mu(p,x)$ can be done thanks to another fraction continuation as detailed in [Abramowitz and Stegun 1964, Press et al. 1992]. We accordingly set $\Gamma_\mu(p,x) = \Gamma_\mu^{\text{cfrac}}(p,x)$, where

$$\Gamma_\mu^{\text{cfrac}}(p,x) = M^{\text{cfrac}}(\mu x, p) \cdot e^{-\mu x + p\log x}\,, \quad \text{and} \quad M^{\text{cfrac}}(\mu x, p) = \frac{\alpha_1}{\beta_1+}\,\frac{\alpha_2}{\beta_2+}\,\frac{\alpha_3}{\beta_3+}\,\cdots\,, \tag{5.18}$$

with $\alpha_1 = 1$, $\alpha_n = -(n-1)\cdot(n-p-1)$ for any $n > 1$, and $\beta_n = \mu x + 2n - 1 - p$ for any $n \geq 1$. The continued fraction $M^{\text{cfrac}}(\mu x, p)$ can be numerically evaluated

using again Algorithm 10, except when $\beta_1 = 1$ (i.e. when $\mu x = p - 1$), in which case we must use

$$M^{\text{cfrac}}(\mu x, p) = \frac{\alpha_1}{M}, \quad \text{where} \quad M = \frac{\alpha_2}{\beta_2 +} \, \frac{\alpha_3}{\beta_3 +} \, \frac{\alpha_2}{\beta_2 +} \, \cdots \quad \text{and} \quad \beta_2 \neq 0\,. \tag{5.19}$$

We extend the computation of $\Gamma_\mu(p, x)$ to the cases $x = 0$ and $x = +\infty$ using a similar approach as for $\gamma_\mu(p, x)$. This yields Algorithm 13.

Algorithm 13: Evaluation of $\Gamma_\mu(x, p) = \int_x^{+\infty} s^{p-1}\, e^{-\mu s}\, ds$ using a continued fraction.

Input: Two numbers $x \in \mathbb{R}_+ \cup \{+\infty\}$, $\mu > 0$, and a positive integer p.

Output: Two numbers $M \in \mathbb{R}$ and $N \in \mathbb{R} \cup \{-\infty\}$ such that $\Gamma_\mu(p, x) = M \cdot e^N$.

if $x = 0$ **then** $(M, N) \leftarrow (1, \log\Gamma(p) - p \log \mu)$ **else if** $x = +\infty$ **then** $(M, N) \leftarrow (0, -\infty)$ **else**
  if $\mu x \neq p - 1$ **then**
    $M \leftarrow M^{\text{cfrac}}(\mu x, p)$; // using Equation (5.18) and Algorithm 10
  else
    $M \leftarrow M^{\text{cfrac}}(\mu x, p)$; // using Equation (5.19) and Algorithm 10
  $N \leftarrow -\mu x + p \log x$

return (M, N)

5.2.3 Accuracy of the mantissa-exponent representation and its conversion into scientific notation

Thanks to Algorithms 12 and 13, we are now able to evaluate the integrals $\gamma_\mu(p, x)$ and $\Gamma_\mu(p, x)$ with a mantissa-exponent representation of type $\rho \cdot e^\sigma$, where, in absence of additional multiprecision library, the quantities ρ and σ are evaluated in standard double floating-point precision. Alhough this representation considerably extends the range over which the integrals $\gamma_\mu(p, x)$ and $\Gamma_\mu(p, x)$ can be represented (in comparison with a direct evaluation of those integrals in double precision), the evaluation of the term $\rho \cdot e^\sigma$ may suffer from important loss of precision, according to the values of ρ and σ. Indeed, using a first order approximation of the relative error associated to the term $\rho \cdot e^\sigma$, we get

$$\left|\frac{\Delta(\rho \cdot e^\sigma)}{\rho \cdot e^\sigma}\right| \approx \left|\frac{\Delta\rho}{\rho}\right| + \left|\frac{\Delta(e^\sigma)}{e^\sigma}\right| = \left|\frac{\Delta\rho}{\rho}\right| + |\Delta\sigma| = \left|\frac{\Delta\rho}{\rho}\right| + |\sigma| \cdot \left|\frac{\Delta\sigma}{\sigma}\right| := E \tag{5.20}$$

where $|\Delta X|$ and $|\Delta X/X|$ respectively denote the absolute and relative errors between the actual value of X and its computed value. Unfortunately, we see that E gets large as $|\sigma|$ increases, and since the quantity ρ and σ are in the best case estimated at the machine precision (i.e. $|\Delta\rho/\rho| = |\Delta\sigma/\sigma| = \varepsilon_{\text{machine}}$), we have the lower bound

$$E \geq E_{\min} := 1 + |\sigma| \cdot \varepsilon_{\text{machine}} \,. \tag{5.21}$$

For instance, when $\sigma \approx 4503.5$, the best relative accuracy that can be expected is $E_{\min} \approx 4504.5 \times 2.22 \cdot 10^{-16} \approx 10^{-12}$ using the IEEE 754 Standard for Floating-Point Arithmetic on a 64-bits computer (which yields $\varepsilon_{\text{machine}} = 2.22 \cdot 10^{-16}$). Since in Algorithms 12 and 13, the exponent σ associated to the computation of $\gamma_\mu(p,x)$ or $\Gamma_\mu(p,x)$ is given by $\sigma = -\mu x + p\log x$, we can already establish some theoritical bounds for the relative error E reachable by these algorithms with respect to μ, p, x. This is done in Figure 5.1, and our numerical experiments performed in Section 5.2.4 (Figures 5.2 and 5.3) will show that this theoritical bound is in practice attained by our algorithms.

Unsurprisingly, the same limitation arises when we format the quantity $\rho \cdot e^\sigma$ in scientific notation (that is $\rho \cdot e^\sigma = a \cdot 10^b$, where $a \in [1, 10)$ and $b \in \mathbb{Z}$). This operation can be done using

$$a = \rho \cdot e^{c - \lfloor c \rfloor} \,, \quad b = \lfloor c \rfloor \,, \quad \text{where} \quad c = \frac{\sigma}{\log(10)} + \log_{10}(\rho) \,. \tag{5.22}$$

This time, the evaluation of a suffers from the loss of precision occuring in the evalution of $c - \lfloor c \rfloor$, the fractional part of c, simply because all digits used to represent the integer part of c are as many digits which are lost in the evaluation of its fractional part. Assuming that $\Delta b = 0$ (i.e. that the quantity c is estimated with at least one digit of precision), we get

$$\left|\frac{\Delta(a \cdot 10^b)}{a \cdot 10^b}\right| = \left|\frac{\Delta a}{a}\right| \approx \left|\frac{\Delta \rho}{\rho}\right| + |\Delta c| \leq (1 + |c|) \cdot \varepsilon_{\text{machine}} \,, \tag{5.23}$$

which is similar to (5.21). Although some numerical strategies to retrieve several significant digits may be developed, the most straightforward way to compensate the loss of precision of those two representations would be to evaluate σ and ρ with a more generous floating point precision, which can be easily done using the *x86 Extended Precision Format* (which corresponds to the long double datatype in C langage, and yields $\varepsilon_{\text{machine}} = 1.08 \cdot 10^{-19}$), or using some multiprecision library (such as the *GNU MPFR* C-library, which provides an exact control of the number of significant number of bits used for each variable).

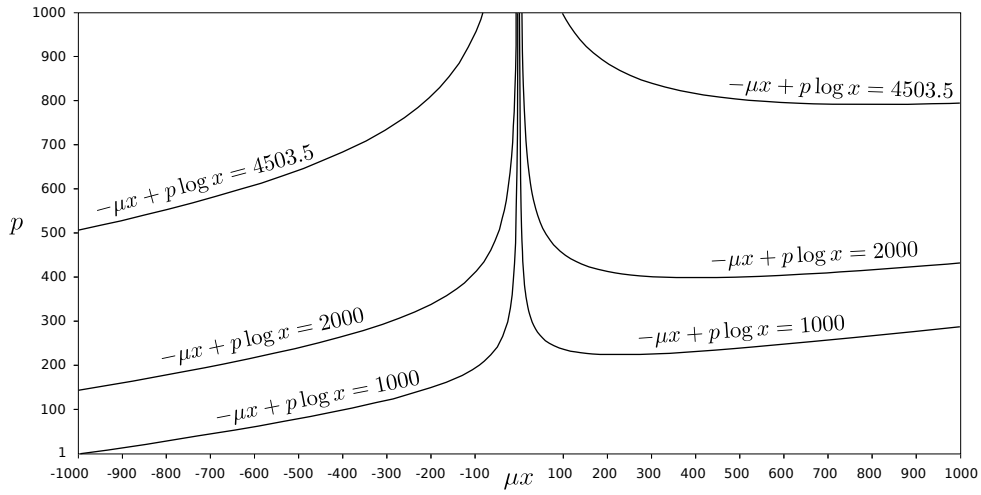


Figure 5.1: Isovalues of the exponent $\sigma = (\mu, x, p) \mapsto -\mu x + p\log x$ ($\mu = -1$, left) and ($\mu = 1$, right). In this figure, we display some isovalues of the exponent part of γ_μ and Γ_μ computed with Algorithm 12 and 13, and whose parametric equation is recalled above. As discussed in Section 5.2.3, the relative precision related to the evaluation of γ_μ and Γ_μ using a mantissa-exponent representation $\rho \cdot e^\sigma$ deteriorates as σ increases. Indeed, even when ρ and σ are estimated with the best available precision ($\Delta\rho/\rho = \Delta\sigma/\sigma = \varepsilon_{\text{machine}}$), the best relative precision that we can expect for the evaluation of $\rho \cdot e^\sigma$ is $E_{\min} = (1 + |\sigma|) \cdot \varepsilon_{\text{machine}}$, as stated in (5.21). The standard precision (on a 64-bits computer) is $\varepsilon_{\text{machine}} = 2.22 \cdot 10^{-16}$, so that $E_{\min} \geq 10^{-12}$ as soon as $\sigma \geq 4503.5$. Interestingly enough, the curve corresponding to the isovalue $\sigma(\mu, x, p) = 4503.5$ fits particularly well with the frontier of the domain where Algorithm 12 and 13 yield a relative accuracy more than 10^{-12} (see Figure 5.2). When using the extended double floating-point precision (corresponding to the long double datatype in C langage), we have $\varepsilon_{\text{machine}} = 1.08 \cdot 10^{-19}$, so that we get $1 \cdot 10^{-16} \leq E_{\min} \leq 2 \cdot 10^{-16}$, in the region $1000 \leq \sigma(\mu, x, p) \leq 2000$ (delimited by the two other isovalues represented above). Again, these two frontiers were experimentally observed in Figure 5.3, where we measure the relative error reached by our algorithms using extended double precision.

5.2.4 Selection of a fast and accurate computational method according to the parameters

We detailed in (5.11), (5.13), (5.14), (5.18), several methods for the numerical evaluation of γ_μ and Γ_μ, with a mantissa-exponent representation. Let us now focus on the accuracy and the computation time of these methods, according to the value of the parameters μ, p, x. For that purpose, we evaluated $\gamma_\mu^{\text{ser}}(x, p)$, $\gamma_\mu^{\text{cfrac}}(x, p)$, $\gamma_\mu^{\text{ibp}}(x, p)$ and $\Gamma_\mu^{\text{cfrac}}(x, p)$ for a large range of parameters:

$$\mu = \pm 1, \quad x \in [0, 1000] \cap \mathbb{N}, \quad p \in [1, 1000] \cap \mathbb{N},$$

more precisely, $\gamma_\mu^{\text{cfrac}}(x,p)$ and $\gamma_\mu^{\text{ser}}(x,p)$ were computed for all theses values of $(\mu x, p)$, but $\gamma_\mu^{\text{ibp}}(x,p)$ was computed only in the case $\mu < 0$, $|\mu x| > \max(1, p-1)$, in accordance to the discussion made in Section 5.2.1, and $\Gamma_\mu^{\text{cfrac}}(x,p)$ was computed only in the case $\mu x \geq 0$.

For each tested value of (μ, x, p) and each evaluation method, we compared the computed values of $\gamma_\mu(p,x)$ and $\Gamma_\mu(p,x)$ (formatted in scientific notation using (5.22)) to those computed with Maple$^{\text{TM}}$ (version 17), with 30 significant decimal digits (which requires large amounts of memory and a long computation time), using the instructions

```
evalf(Int(s^(p-1)*exp(-mu*s),s=0..x,digits=30));
evalf(Int(s^(p-1)*exp(-mu*s),s=x..infinity,digits=30));
```

The values of the integrals estimated with Maple were used as references to evaluate the relative accuracy reached for each method, and each tested value of μ, x, p. The results are displayed in Figure 5.2 and 5.3. We observed from these experiments that for each μ, x, p, at least one computation method yields a relative error less than $2 \cdot 10^{-12}$ when using the standard double floating-point precision in C langage, and only this many in the region (μ, x, p) where the exponent part is above 4503.5 (this region is represented in Figure 5.1). Outside of this region, at least one method yields a relative error less than 10^{-12} (more precisely close to 10^{-13}). Interestingly enough, the observed relative errors perfectly match with the bound (5.21) predicted in Section 5.2.3, showing that, in practice, the accuracy of Algorithms 12 and 13 is only limited by the mantissa-exponent representation. When using the extended double precision (see Figure 5.3), we improve the precision of three orders of magnitude, and again, the selection of the most accurate method yields a relative error which is very close to that predicted in Section 5.2.3, so that we could expect even more accuracy with higher precision computer arithmetics.

By measuring the computation time for each method and each value of (μ, x, p), and thanks to the control of the relative error presented in Figure 5.2, we derived a partition into three domains of the plan $(\mu x, p)$ which makes possible the fast and accurate computation of at least one quantity between $\gamma_\mu(p,x)$ and $\Gamma_\mu(p,x)$. We accordingly propose a parametric equation for the boundary of those three domains, given by

$$\forall \mu x \in \mathbb{R} \cup \{+\infty\}, \quad p_{\lim}(\mu x) = \begin{cases} 5\sqrt{|\mu x|} - 5 & \text{if } \mu x < -9\,, \\ 0 & \text{if } -9 \leq \mu x \leq 0\,, \\ \mu x & \text{otherwise.} \end{cases} \tag{5.24}$$

This equation can be used in the following way:

— when $p \geq p_{\lim}(\mu x)$: compute $\gamma_\mu^{\text{cfrac}}(p,x)$;

— otherwise: compute $\gamma_\mu^{\text{ibp}}(p,x)$ when $\mu < 0$, or $\Gamma_\mu^{\text{cfrac}}(p,x)$ when $\mu > 0$;

as illustrated in Figure 5.4.

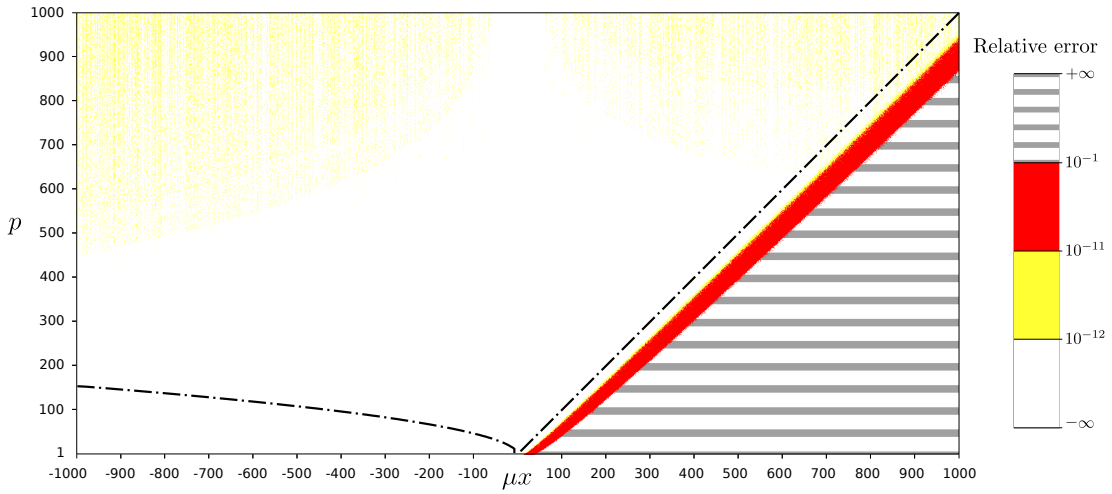
5.3 Evaluation of the generalized incomplete gamma function

As stated before, the accurate evaluation of $I_{x,y}^{\mu,p}$ raises two different issues. First, this integral can be approximated as the difference $A-B$ between two terms $A \geq B \geq 0$ involving the evaluation of the generalized upper and lower incomplete gamma functions γ_μ and Γ_μ, thanks to the relations (5.8)-(5.9). Therefore, we must select which difference can be accurately and efficiently computed according to the parameters μ, x, y, p; this selection is discussed in Section 5.3.1. Second, we must be careful that the accurate evaluation of A and B is not sufficient to garantee an accurate evalutation of the difference $A - B$, because cancellation errors arise when A and B are too close to each other, which happens in practice when $x \approx y$. In that case, we propose to approximate the integral $I_{x,y}^{\mu,p}$ using a first order trapezoidal approximation, as discussed in Section 5.3.2. In order to decide which approximation must be used (between the computation by means of a difference $A - B$, or a first order approximation), we propose in Section 5.3.3 a simple criterion based on the absolute errors. This study results in Algorithm 15 for the evaluation of $I_{x,y}^{\mu,p}$.

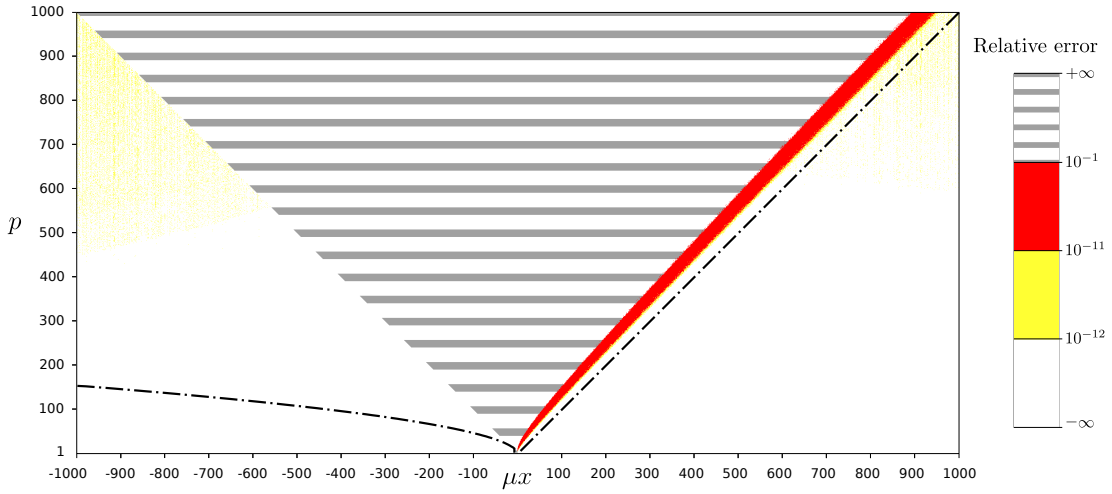
5.3.1 Computing $I_{x,y}^{\mu,p}$ as a difference of generalized incomplete gamma functions

According to the numerical experiments presented in Figures 5.2-5.4, we are now able to decide which integral between $\gamma_\mu(p,x)$ and $\Gamma_\mu(p,x)$ can be computed and how it must be evaluated, according to the value of (μ, p, x), to reach at the same time a good accuracy and a small computation time. We used these results to derive which difference $I_{\text{diff}} = A - B$ should be considered to approximate $I_{x,y}^{\mu,p}$, according to x, y, μ, p. The results are gathered in Table 5.1. A mantissa-exponent representation of I_{diff} is obtained from the mantissa-exponent representations (m_A, n_A) and (m_B, n_B) of A and B (returned by Algorithm 12 or 13):

$$I_{\text{diff}} = \rho_{\text{diff}} \cdot e^{\sigma_{\text{diff}}}, \quad \text{where} \quad \rho_{\text{diff}} = m_A - m_B\, e^{n_B - n_A}, \quad \sigma_{\text{diff}} = n_A\,. \tag{5.25}$$

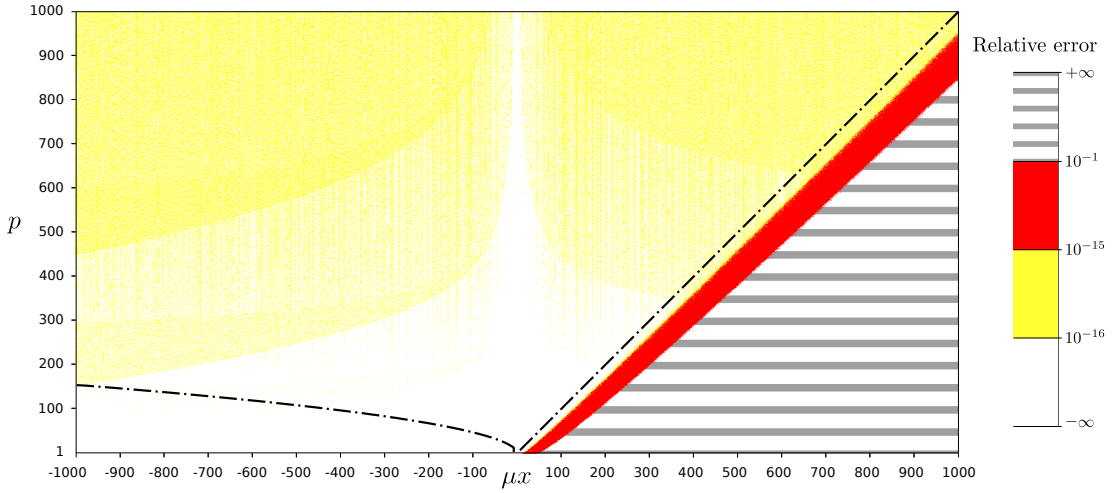


(a) relative error for $\gamma_\mu^{\mathbf{cfrac}}$.

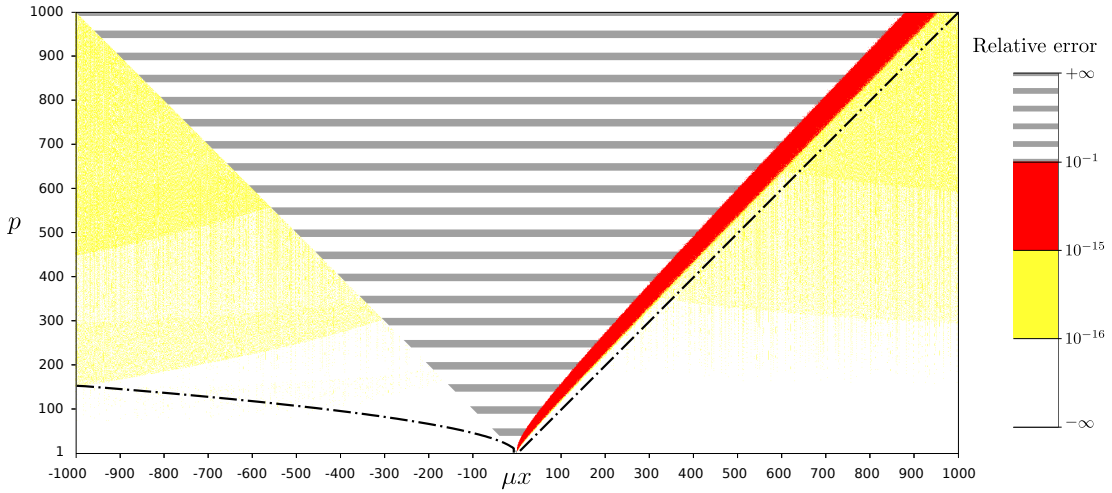


(b) relative error for $\gamma_\mu^{\mathbf{ibp}}$ ($\mu x < 0$, left) and $\Gamma_\mu^{\mathbf{cfrac}}$ ($\mu x > 0$, right).

Figure 5.2: Control of the relative error associated to the computation of $\gamma_\mu(p,x)$ and $\Gamma_\mu(p,x)$ using different methods. We used Algorithms 12 and 13 (implemented in C langage, using the standard double datatype on a 64-bits computer) to compute $\gamma_\mu^{\text{cfrac}}(p,x)$, $\gamma_\mu^{\text{ibp}}(p,x)$, $\Gamma_\mu^{\text{cfrac}}(p,x)$ for $\mu = \pm 1$, x integer in $[0, 10^3]$, and p integer in $[1, 10^3]$. Using Maple, we measured the relative errors reached by each method. The error reached by $\gamma_\mu^{\text{cfrac}}$ is displayed in (a), the error reached by γ_μ^{ibp} (computed only for $\mu < 0$ and $|\mu x| \geq \max(1, p-1)$) is displayed in the left-side of (b), and the error reached by $\Gamma_\mu^{\text{cfrac}}$ (computed only for $\mu > 0$) is displayed in the right-side of (b). The dashed curve (see its parametric equation in (5.24)) splits the plan $(\mu x, p)$ into three domains, each one is associated to one of the three computation methods (γ_μ^{ibp} : left, $\gamma_\mu^{\text{cfrac}}$: middle, $\Gamma_\mu^{\text{cfrac}}$: right), and corresponds to the region where the method is at the same time fast and accurate, compared to the others (see also Figure 5.4). We can see that inside each one of the three domains, the corresponding computation method reaches a relative error always less than 10^{-11} (the actual maximal observed error is in fact close to $2 \cdot 10^{-12}$), and most of the time less than 10^{-12} (in practice close to 10^{-13}). We also observe that the boundary of the region where the relative error (of the selected algorithm) is greater than 10^{-12} coincides almost perfectly with the isovalue $\sigma = 4503.5$ displayed in Figure 5.1, showing that the relative error is in practice only limited by the mantissa-exponent representation (see discussion in Section 5.2.3). This limitation can be compensated by using a more precise floating-point representation, as shown in Figure 5.3.



(a) relative error for $\gamma_\mu^{\mathbf{cfrac}}$.



(b) relative error for $\gamma_\mu^{\mathbf{ibp}}$ ($\mu x < 0$, left) and $\Gamma_\mu^{\mathbf{cfrac}}$ ($\mu x > 0$, right).

Figure 5.3: Improving the accuracy of $\gamma_\mu(p,x)$ and $\Gamma_\mu(p,x)$ using extended double precision. We performed here the same experiment as in Figure 5.2, using a C implementation of Algorithms 12 and 13 with extended double precision (corresponding to the long double datatype in C langage, with machine epsilon $\varepsilon_{\text{machine}} = 1.08 \cdot 10^{-19}$, which is around three orders of magnitude better than the standard double precision). We see that our algorithm fully benefits from this additional precision, since the observed relative error is decreased of around three magnitude orders as well. Besides, we observe again that the main limitation to the precision remains that involved by the mantissa-exponent representation, since, within each domain, the level lines of the relative error of the selected algorithm match very well to the isovalues of σ, and the value of the relative error is in practice very similar to that predicted in (5.21). This suggests that the error bounds we obtain could be reduced even further by simply using a more precise floating-point arithmetic (for instance the GNU MPFR C library).

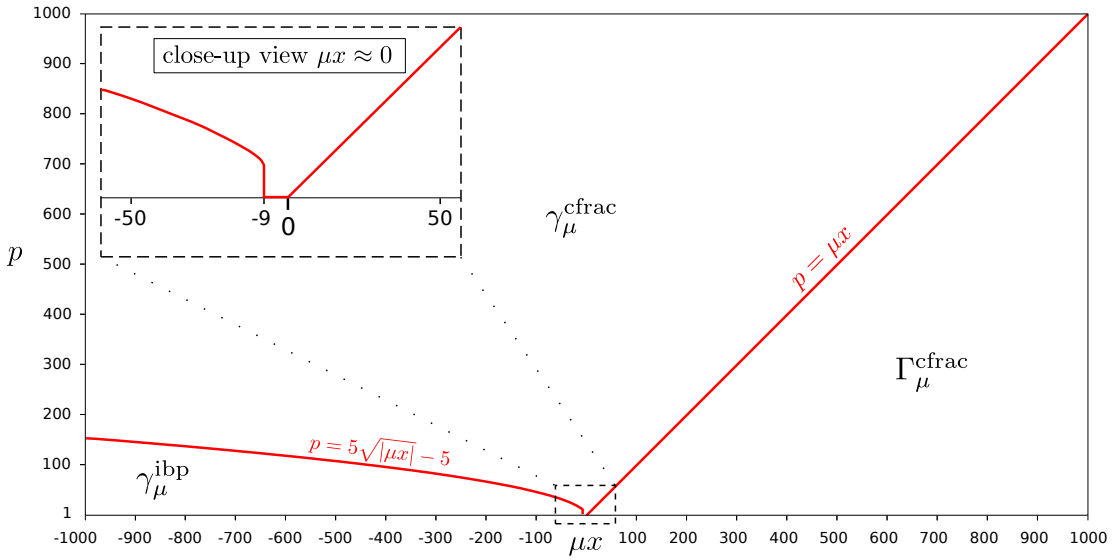


Figure 5.4: Numerical evaluation of the generalized lower or upper incomplete gamma functions. In this figure, we display the graph (red curve) of the frontier p_{lim} defined in (5.24). This curve delimits the plan $(\mu x, p)$ into three regions, each corresponding to the region where one of the three computation methods $\gamma_\mu^{\text{cfrac}}$, γ_μ^{ibp} and $\Gamma_\mu^{\text{cfrac}}$ is optimal (in the sense that its computation is fast and reaches a good relative error). According to our partition, and as indicated on the figure, γ_μ^{ibp} must be computed in the bottom-left region, $\gamma_\mu^{\text{cfrac}}$ in the middle region, and $\Gamma_\mu^{\text{cfrac}}$ in the bottom-right region. More precisely, we select $\gamma_\mu(p,x)$ as soon as $p \geq p_{\text{lim}}(\mu x)$, otherwise we select $\gamma_\mu^{\text{ibp}}(p,x)$ when $\mu < 0$, or $\Gamma_\mu^{\text{cfrac}}(p,x)$ when $\mu > 0$. A close-up view near $\mu x = 0$ shows that $\gamma_\mu^{\text{cfrac}}$ is automatically selected near $\mu x = 0$ since $p \geq 1 \geq p_{\text{lim}}(\mu x) = 0$ when $-9 \leq \mu x \leq 0$ (this avoids the computation of $\gamma_\mu^{\text{ibp}}(p,x)$ for $|\mu x| < \max(1, p-1)$, which is not allowed according to the discussion of Section 5.2.1).

If I_{diff} was computed directely as the difference $A - B$ (which is in practice difficult because A and B may not be representable in the floating-point arithmetic), the absolute error $|\Delta I_{\text{diff}}| = |I_{x,y}^{\mu,p} - I_{\text{diff}}|$ would satisfy $|\Delta I_{\text{diff}}| \approx A\,\varepsilon_{\text{machine}}$ (since $A \geq B \geq 0$), but this is not the case here, due to the mantissa-exponent representation used for A, B, and I_{diff}. A more precise estimation of $|\Delta I_{\text{diff}}|$ is obtained by using a first order approximation

$$|\Delta I_{\text{diff}}| \approx (|\Delta \rho_{\text{diff}}| + |\rho_{\text{diff}}\,\Delta \sigma_{\text{diff}}|)\, e^{\sigma_{\text{diff}}}\,.$$

Besides, as discussed in Figures 5.2 and 5.3, we can reasonably consider that the relative precision reached for the quantities m_A, n_A, m_B and n_B is close to the machine epsilon (which is of course not the case for $m_A \cdot e^{n_A}$ and $m_B \cdot e^{n_B}$, as discussed in Section 5.2.3). It follows that the quantity σ_{diff} is also evaluated at the machine precision, and thus, the corresponding absolute error is $|\Delta \sigma_{\text{diff}}| =$

Computation of $I_{x,y}^{\mu,p}$:	$\mu < 0$	$\mu > 0$
$p < p_{\lim}(\mu x)$	$\gamma_\mu^{\text{ibp}}(p,y) - \gamma_\mu^{\text{ibp}}(p,x)$	$\Gamma_\mu^{\text{cfrac}}(p,x) - \Gamma_\mu^{\text{cfrac}}(p,y)$
$p_{\lim}(\mu x) \le p < p_{\lim}(\mu y)$	$\gamma_\mu^{\text{ibp}}(p,y) - \gamma_\mu^{\text{cfrac}}(p,x)$	$\frac{\Gamma(p)}{\mu^p} - \left(\gamma_\mu^{\text{cfrac}}(p,x) + \Gamma_\mu^{\text{cfrac}}(p,y)\right)$
$p_{\lim}(\mu y) \le p$	$\gamma_\mu^{\text{cfrac}}(p,y) - \gamma_\mu^{\text{cfrac}}(p,x)$	$\gamma_\mu^{\text{cfrac}}(p,y) - \gamma_\mu^{\text{cfrac}}(p,x)$

Table 5.1: Computing $I_{x,y}^{\mu,p}$ as a difference of generalized incomplete gamma functions. We propose here a practical computational method for the evaluation of $I_{x,y}^{\mu,p}$ by means of a difference of type $I_{x,y}^{\mu,p} = A - B$, where $A = \gamma_\mu(p,y)$, $B = \gamma_\mu(p,x)$, or, when $\mu > 0$, $A = \Gamma_\mu(p,x)$, $B = \Gamma_\mu(p,y)$, or $A = \Gamma(p)/\mu^p$, $B = \gamma_\mu(p,x) + \Gamma_\mu(p,y)$. Thanks to Figure 5.4, we derive which difference $A - B$, and which numerical method must be used for the efficient evaluation of A and B, according to the value of x, y, μ, p. It is important to notice that the evaluation of $I_{x,y}^{\mu,p}$ by means of difference $A - B$ is inaccurate when $A \approx B$, which happens when $x \approx y$. In that case, the integral $I_{x,y}^{\mu,p}$ must be approximated differently, as discussed in Section 5.3.2.

$|\sigma_{\text{diff}}| \cdot \varepsilon_{\text{machine}}$. The same kind of equality does not holds for the mantissa ρ_{diff}, whose numerical evaluation suffers from an additional loss of precision due to the exponential term. Indeed, using again a first order approximation, we get $|\Delta\rho_{\text{diff}}| \approx |\Delta m_A| + (|\Delta m_B| + |m_B\,\Delta(n_B - n_A)|)\, e^{n_B - n_A}$, therefore

$$|\Delta\rho_{\text{diff}}| \approx \left(|m_A| + |m_B| \cdot (1 + |n_B| + |n_A|)\; e^{n_B - n_A}\right) \varepsilon_{\text{machine}}\,,$$

and we can drop the absolute values around m_A, m_B and ρ_{diff} (which are nonnegative) to get the approximation

$$|\Delta I_{\text{diff}}| \approx |\widehat{\Delta} I_{\text{diff}}| := \left(m_A + m_B \cdot (1 + |n_B| + |n_A|)\, e^{n_B - n_A} + \rho_{\text{diff}}\, |\sigma_{\text{diff}}|\right) \varepsilon_{\text{machine}}\, e^{\sigma_{\text{diff}}}\,.$$

We will use $|\widehat{\Delta} I_{\text{diff}}|$ as an estimate of the actual absolute error $|\Delta I_{\text{diff}}|$.

5.3.2 Computing $I_{x,y}^{\mu,p}$ using a trapezoidal rule

A simple first order trapezoidal approximation of $I_{x,y}^{\mu,p}$ yields

$$I_{x,y}^{\mu,p} \approx I_{\text{trapezoid}} := (y - x)\, \frac{f_{\mu,p}(x) + f_{\mu,p}(y)}{2}\,, \tag{5.26}$$

where $f_{\mu,p}(s) = s^{p-1}\, e^{-\mu s}$. For the practical implementation, we will compute $I_{\text{trapezoid}}$ using the mantissa-exponent representation $I_{\text{trapezoid}} = \rho_{\text{trapezoid}} \cdot e^{\sigma_{\text{trapezoid}}}$, where

$$\sigma_{\text{trapezoid}} = \max(n_x, n_y), \quad \rho_{\text{trapezoid}} = \frac{y - x}{2\,x} e^{n_x - \sigma_{\text{trapezoid}}} + \frac{y - x}{2\,y} e^{n_y - \sigma_{\text{trapezoid}}}\,,$$

noting $n_x = -\mu x + p \log x$ and $n_y = -\mu y + p \log y$. The following proposition gives an upper bound of the absolute error $|\Delta I_{\text{trapezoid}}| = |I_{x,y}^{\mu,p} - I_{\text{trapezoid}}|$ associated to the approximation of $I_{x,y}^{\mu,p}$ by $I_{\text{trapezoid}}$. Remark that this upper bound is not interesting for all values of μ, x, y, p, but it gets precise as the distance betwen x and y gets small.

Proposition 34. *For any $\mu \in \mathbb{R} \setminus \{0\}$, for any positive integer p, and any non-negative real numbers x, y, such as $x \leq y$, we have the upper bound*

$$|\Delta I_{trapezoid}| \leq |\widehat{\Delta} I_{trapezoid}| := \frac{(y-x)^3}{12} D_{x,y}^{\mu,p}\, y^{\max(0,p-3)}\, e^{\max(-\mu x, -\mu y)}\,,$$

where

$$D_{x,y}^{\mu,p} = \begin{cases} \mu^2 & \text{if } p = 1\,, \\ \max\left(|\mu^2 x - 2\mu|, |\mu^2 y - 2\mu|\right) & \text{if } p = 2\,, \\ C_{x,y}^{\mu,p} & \text{if } p \geq 3\,, \end{cases}$$

and

$$C_{x,y}^{\mu,p} = \begin{cases} |P_\mu(y)| & \text{if } \mu < 0\,, \\ \max\left(|P_\mu(x)|, p-1, |P_\mu(y)|\right) & \text{if } \mu > 0 \text{ and } x \leq \frac{p-1}{\mu} \leq y\,, \\ \max\left(|P_\mu(x)|, |P_\mu(y)|\right) & \text{otherwise,} \end{cases}$$

with $P_\mu(s) = (\mu s)^2 - 2(p-1)\mu s + (p-1)(p-2)$.

Proof (abridged). The first order trapezoidal rule yields the upper bound

$$|\Delta I_{\text{trapezoid}}| \leq \frac{(y-x)^3}{12} \sup_{s \in [x,y]} |f''_{\mu,p}(s)|\,. \tag{5.27}$$

In the case $p \geq 3$, for any $s \in [x, y]$, we have $f''_{\mu,p}(s) = P_\mu(s)\, s^{p-3}\, e^{-\mu s}$, and a straightforward study of the second degree polynomial P_μ yields $C_{x,y}^{\mu,p} = \sup_{s \in [x,y]} |P_\mu(s)|$. It follows that

$$\sup_{s \in [x,y]} |f''_\mu(s)| \leq C_{x,y}^{\mu,p}\, y^{p-3}\, e^{\max(-\mu x, -\mu y)}\,.$$

In the cases $p = 1$ and $p = 2$, a similar study can be led without difficulty. Finally, for any $p \geq 1$, we get

$$\sup_{s \in [x,y]} |f''_\mu(s)| \leq D_{x,y}^{\mu,p}\, y^{p-3}\, e^{\max(-\mu x, -\mu y)}\,,$$

which, combined to (5.27), yields the announced result. □

5.3.3 Criterion for the selection of the approximation by trapezoidal rule or differences

In order to choose between the two approximation methods (trapezoidal or difference), we propose to select the one yielding the smallest absolute error. To this aim, we consider the ratio between $|\widehat{\Delta} I_{\text{diff}}|$ and $|\widehat{\Delta} I_{\text{trapezoid}}|$, i.e.,

$$\forall x \neq y, \quad R_{x,y}^{\mu,p} = \frac{|\widehat{\Delta} I_{\text{diff}}|}{|\widehat{\Delta} I_{\text{trapezoid}}|} ,$$

which is an approximation of the ratio between the effective relative errors ΔI_{diff} and $\Delta I_{\text{trapezoid}}$. Then, we will approximate $I_{x,y}^{\mu,p}$ by $I_{\text{trapezoid}}$ when $R_{x,y}^{\mu,p} > 1$, or by I_{diff} otherwise. Notice that for the practical evaluation of $R_{x,y}^{\mu,p}$, we will use again a mantissa-exponent representation $R_{x,y}^{\mu,p} = \rho_r \cdot e^{\sigma_r}$, where

$$\rho_r = \frac{12 \cdot (m_A + m_B \cdot (1 + |n_B| + |n_A|)\, e^{n_B - n_A} + \rho_{\text{diff}}\, |\sigma_{\text{diff}}|)\, \varepsilon_{\text{machine}}}{D_{x,y}^{\mu,p}},$$

$$\text{and} \quad \sigma_r = \sigma_{\text{diff}} - \sigma_t ,$$

noting (m_A, n_A) and (m_B, n_B) the mantissa-exponent representations of the quantities A and B, returned by Algorithms 12 and 13, noting $\rho_{\text{diff}} = m_A - m_B \cdot e^{nB-nA}$, $\sigma_{\text{diff}} = n_A$, and $\sigma_t = 3 \log(y - x) + \max(0, p - 3) \cdot \log y + \max(-\mu x, -\mu y)$. We validate the ability of this criterion to automatically select the most accurate approximation in Figure 5.5.

5.4 Discussion on the evaluation of the complete gamma function

The computation of the complete gamma function $\Gamma(p)$ for $p \in \mathbb{N}$, $\mathbb{R}$ or $\mathbb{C}$ is itself a wide subject of research. The object of this section is to compare several methods from the litterature and end up with a practical efficient algorithm for computing the quantity $\Gamma(p)$, which is needed to compute $I_{x,y}^{\mu,p}$ as right-hand difference (5.9), i.e.

$$I_{x,y}^{\mu,p} = \frac{\Gamma(p)}{\mu^p} - \gamma_\mu(p, x) - \Gamma_\mu(p, y) .$$

Since $\Gamma(p)$ gets huge as p increases, in practice we approximate its logarithm $\log \Gamma(p)$. Note that when p is a positive integer, as it is the case in this chapter,

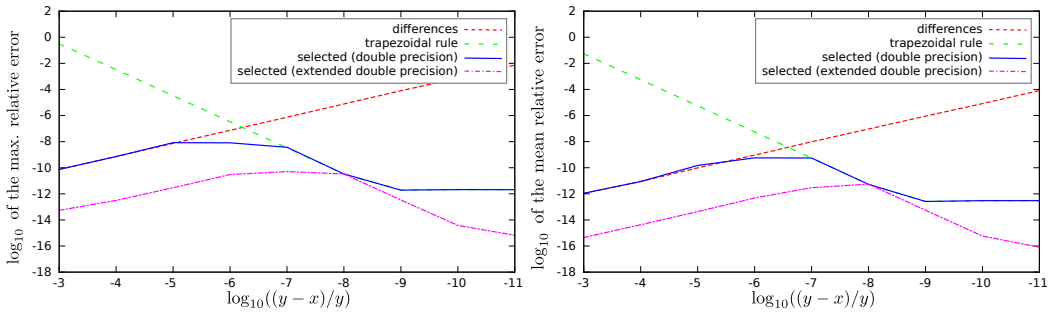


Figure 5.5: Control of maximum and mean relative errors associated to the computation of $I_{x,y}^{\mu,p}$ using differences or a first order trapezoidal rule. In Section 5.3.2, we proposed to approximate the integral $I_{x,y}^{\mu,p}$ by a difference of generalized incomplete gamma functions, $I_{x,y}^{\mu,p} \approx I_{\text{diff}} = A - B$ (see Table 5.1 to derive the values of A and B according to x, y, μ, p), or using a trapezoidal rule, $I_{x,y}^{\mu,p} \approx I_{\text{trapezoid}}$. We proposed in Section 5.3.3 an explicit criterion, based on the computation of a ratio of (some estimates of) the absolute errors $|\Delta I_{\text{diff}}|$ and $|\Delta I_{\text{trapezoid}}|$ associated to those two approximations, which can be used to automatically select which approximation should be used. For several values of $\delta_r = (y - x)/y$, we computed $I_{x,y}^{\mu,p}$ for a large range of parameters ($\mu = \pm 1$, p integer in $[1, 1000]$, y integer in $[1, 1000]$, and x being the floating-point number closest to $y\,(1 - \delta_r)$). We display here the evolution, as a function of $\log_{10}(\delta_r)$, of the maximal (left-side) and mean (right-side) relative error observed when using the approximation by differences I_{diff} (dashed red curve, standard double precision implementation), or when using the approximation by trapezoidal rule $I_{\text{trapezoid}}$ (dashed green curve, standard double precision implementation), or when automatically selecting the computation method (plain blue curve for the standard double precision implementation, dotted purple curve for the extended double precision implementation), thanks to the criterion proposed in Section 5.3.3. We see that the plain blue curve lies almost everywhere below the dashed curves, showing that the criterion efficiently selects the best accurate approximation. We can see also that the error can be improved by three orders of magnitude using extended double precision, except when $\delta_r \approx 10^{-8}$ (in that case, the precision is limited by the fact that we only use one term in the trapezoidal rule).

we have $\Gamma(p) = (p-1)!$ so that $\log \Gamma(p)$ can be easily computed using

$$\log (p-1)! = \sum_{k=1}^{p-1} \log k \,.$$

However, the numerical computation of this sum becomes rapidely inaccurate when p is large, because of the cumulation of small numerical errors made at each step of the summation. Besides, in order to facilitate the adaptation of this chapter to noninteger values of p, we prefer to focus on more general methods.

The first evaluation method that we will consider was proposed in [Lanczos

1964], and uses a Stirling formula-like approximation:

$$\forall p > 1, \quad \Gamma(p) = \sqrt{2\pi}\left(p+\gamma-\tfrac{1}{2}\right)^{p-\frac{1}{2}} e^{-(p+\gamma-\frac{1}{2})}\left(A_\gamma(p-1)+\varepsilon_\gamma\right), \tag{5.28}$$

where $\gamma > 0$ is a numerical parameter (different from the Euler-Mascheroni constant), $A_\gamma(p-1)$ is a truncated rational fraction of type $A_\gamma(p-1) = c_0(\gamma) + \sum_{k=1}^{N_\gamma} \frac{c_k(\gamma)}{p-1+k}$, and N_γ and the coefficients $\{c_k(\gamma)\}_{0\le k\le N_\gamma}$ depend on the value of γ. In the case $\gamma = 5$, Lanczos claims that the relative error $|\varepsilon_5|$ associated to (5.28) satisfies $|\varepsilon_5| < 2\cdot 10^{-10}$, and this claim was confirmed by our numerical experiments. In the case $\gamma = 5$, we have $N_\gamma = 6$ and the numerical values of the coefficients $\{c_k(\gamma)\}_{0\le k\le N_\gamma}$ are available in [Lanczos 1964]. These values are refined to double floating-point precision in [Press et al. 1992], so we used them in our implementation of (5.28).

A more recent computation method (see [Char 1980, Olver et al. 2010, Cuyt et al. 2008] and references therein), also based on a Stirling approximation, consists in computing

$$\forall p > 1, \quad \Gamma(p) = \sqrt{2\pi}\, e^{-p}\, p^{p-\frac{1}{2}}\, e^{J(p)}, \quad \text{where} \quad J(p) = \frac{a_0}{p+}\,\frac{a_1}{p+}\,\frac{a_2}{p+}\,\cdots, \tag{5.29}$$

where some numerical approximations, with 40 decimal digits of precision, of the coefficients $\{a_k\}_{0\le k\le 40}$ of the continued fraction $J(p)$ can be found in [Char 1980].

The last approximation that we present, and that we will select in practice as the most simple an accurate method, is a refinement of the Lanczos formula, proposed in [Pugh 2004]. In his work, Pugh adapted (5.28) into

$$\forall p > 1, \quad \Gamma(p) \approx 2\sqrt{\frac{e}{\pi}}\left(\frac{p+r-\frac{1}{2}}{e}\right)^{p-\frac{1}{2}}\left[d_0 + \sum_{k=1}^{N_r}\frac{d_k}{p-1+k}\right], \tag{5.30}$$

where r is again a numerical parameter (which replaces the parameter γ of (5.28), to avoid confusion with the Euler-Mascheroni constant). Pugh studied the accuracy of the approximation (5.30) for differents settings r. In the case $r = 10.900511$, he sets $N_r = 11$, and gives the numerical values of the coefficients $\{d_k\}_{0\le k\le 10}$ with 20 significant decimal digits (see Table 5.2). According to Pugh, this setting yields a relative error less than 10^{-19}, which is effectively what we observed when computing (5.30) with Maple for $1 \le p \le 10^4$ in multiprecision (30 digits).

In order to select which method will be used in our algorithms, we used the three approximations (5.28), (5.29), and (5.30) to compute $\log\Gamma(p)$ for $1 \le p \le 10^4$.

k	d_{2k}	d_{2k+1}
0	2.48574089138753565546E-5	1.05142378581721974210E+0
1	-3.45687097222016235469E+0	4.51227709466894823700E+0
2	-2.98285225323576655721E+0	1.05639711577126713077E+0
3	-1.95428773191645869583E-1	1.70970543404441224307E-2
4	-5.71926117404305781283E-4	4.63399473359905636708E-6
5	-2.71994908488607703910E-9	

Table 5.2: coefficients $\{d_k\}_{0\leq k\leq 10}$ of Equation (5.30) with 20 significant decimal digits [Pugh 2004].

Algorithm 14: Accurate computation of $\log\Gamma(p)$ using Pugh's method.

Input: A real number $p \geq 1$.

Output: An accurate estimation of $\log\Gamma(p)$.

Require: Coefficients $\{d_k\}_{0\leq k\leq 10}$ defined in Table 5.2.

return
$\log\left(2\sqrt{\frac{e}{\pi}}\left[d_0 + \sum_{k=0}^{10} \frac{d_k}{p-1+k}\right]\right) - \left(p-\frac{1}{2}\right) + \left(p-\frac{1}{2}\right)\log\left(p+10.900511-\frac{1}{2}\right)$

The values of $\Gamma(p) = e^{\log\Gamma(p)}$ were converted into scientific notation using (5.22), in order to avoid the overflow issues that fatally occur when computing $\Gamma(p)$ directly. The accuracy was evaluated using Maple, the results (restricted to $1 \leq p \leq 5000$) are displayed in Figure 5.6. It follows from our experiments that the approximations (5.29) and (5.30) are the most accurate, except for small values of p where the continued fraction is inacurrate (certainly because more than 40 approximants are needed for those values of p). Besides, both methods suffer from the loss of accuracy involved by the conversion of $\Gamma(p)$ into scientific notation from the value of its logarithm $\log\Gamma(p)$, as discussed in Section 5.2.3. Again, this loss of precision can be compensated by improving the floating-point accuracy, as illustrated in Figure 5.6. The Lanczos method yields a relative error close to $2\cdot 10^{-10}$, and in this case, the loss of accuracy is due to the approximation itself. Finally we decided to use Pugh's method, for its simplicity and the nice theoretical study provided in [Pugh 2004]. Our implementation is described in Algorithm 14.

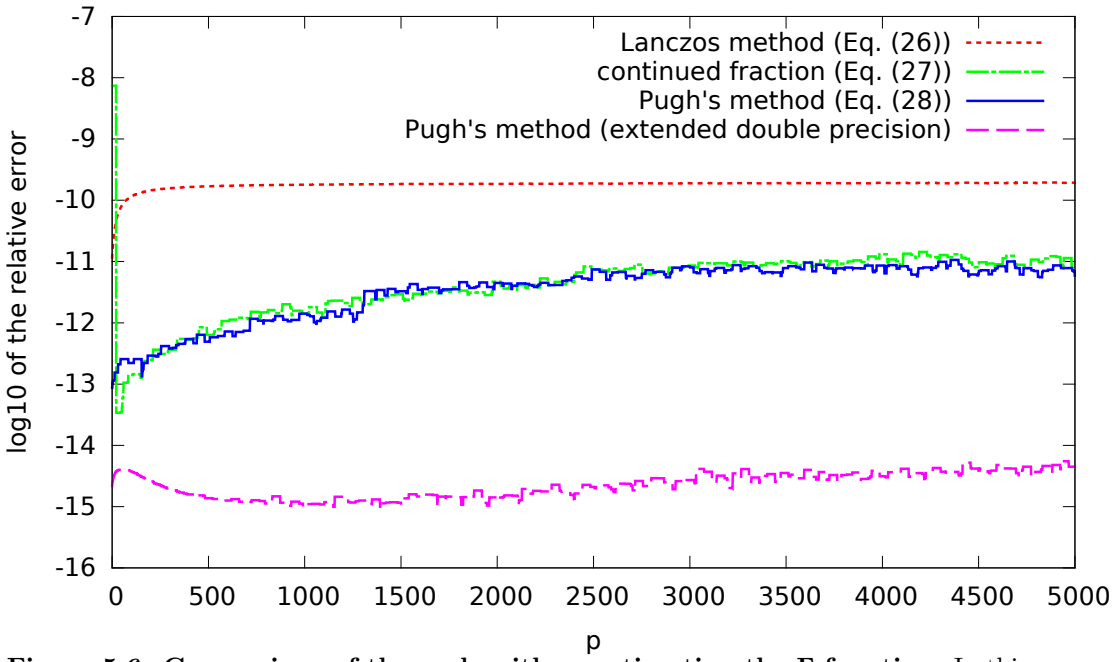


Figure 5.6: Comparison of three algorithms estimating the Γ function. In this experiment, we used (5.28), (5.29) and (5.30) to compute $\log\Gamma(p)$ for $1 \leq p \leq 5000$, and estimated the relative error using Maple. We here display the graphs of the relative errors related to each approximation, as a function of the parameter p (the curves were smoothed by replacing the relative error associated to $\Gamma(p)$ by its maximum value over the range $[p-20, p+20]$, in order to improve the readability). The three top curves were obtained using a C-implementation in standard double precision of the three approximation methods. We observe that the continued fraction approximation (5.29) is inacurrate for small values of p. However, we see that the precision reached by the Pugh's approximation (5.30) (and also the continued fraction, excepting for small values of p) is only limited by the loss of precision occuring when formatting $\Gamma(p)$ in scientific notation from its mantissa-exponent representation $\Gamma(p) = \rho \cdot e^{\sigma}$ (where $\rho = 1$ and $\sigma = \log\Gamma(p)$), since the bound predicted in (5.23) is attained (for instance, we check that for $p = 800$ (respectively $p = 5000$), we have $\sigma = \log\Gamma(p) \approx 4545$ (respectively $\sigma = 37582$), so that the optimal relative error predicted in (5.23) is close to 10^{-12} (respectively 10^{-11}), which is more or less the value attained by the two algorithms). This is not the case for the Lanczos approximation (5.28), whose precision is limited by the approximation itself. The last curve (bottom) corresponds to the implementation of Pugh's method in C langage using extended double precision (long double datatype); the loss of precision resulting from the formatting of $\Gamma(p)$ in scientific notation is significantly reduced.

5.5 Comparison with Fullerton's Algorithm

In this section, we compare our algorithm with Algorithm 435, proposed in [Fullerton 1972], for the evaluation of the generalized incomplete gamma function

$I^{\mu,p}_{x,y}$. Remind that, in his work, Fullerton focused on a slightly different integral than us, since he proposed an algorithm for the evaluation of the integral $J^p_{x,y}$ defined in (5.6), however the computation of $I^{\mu,p}_{x,y}$ using $J^p_{x,y}$, or conversely of $J^p_{x,y}$ using $I^{\mu,p}_{x,y}$, is immediate, as we already explained in Section 5.1. Similarily, the function

$$\gamma'(p,x) = \int_0^x |s|^{p-1}\, e^{-s}\, ds, \quad -\infty < x \leq +\infty\,,$$

he introduced is also closely related to our lower generalized incomplete gamma function γ_μ, since for any $x \geq 0$, and any $p > 0$, we have $\gamma_\mu(p,x) = \mu^{-p}\,\gamma'(p,x)$ when $\mu > 0$, and $\gamma_\mu(p,x) = -|\mu|^{-p}\,\gamma'(p,x)$ when $\mu < 0$. Fullerton poposed an algorithm for the numerical evaluation of γ', then suggested to evaluate the integral $J^p_{x,y}$ using the difference

$$J^p_{x,y} = e^x \int_x^y |s|^{p-1}\, e^{-s}\, ds = e^x\,(\gamma'(a,y) - \gamma'(a,x))$$

when $1 \leq p \leq 2$, and using a forward (when $p > 2$) or backward (when $p < 1$) recurrence relation, leading back to the computation of a quantity $J^q_{x,y}$, with $1 \leq q \leq 2$. In the case $1 \leq p \leq 2$, the evaluation of $\gamma'(p,x)$ relies on different approximation methods (such as continued fractions, approximation using Chebyshev polynomials, or asymptotic expansions), according to the value of x.

As already pointed in [Schoene 1978], Algorithm 435 suffers from several numerical instabilities, arising when $p > 2$. We indeed observed in our own numerical experiments, presented in Tables 5.3, 5.4 and 5.5, some computed values with very low accuracy, or incorrect sign, typically when $p \geq 10$, or when $x \leq p \leq y$. We also also observed some overflow issues, for instance when working with $p \geq 100$.

In the experiments of Tables 5.3-5.5, we evaluated the integral $I^{\mu,p}_{x,y}$, for several sets of parameters x, y, μ, p, using both Fullerton's Algorithm 435 and Algorithm 15. The accuracy of the returned result was controlled with the software Maple™, using the instruction

```
evalf(Int(s^(p-1)*exp(-mu*s),s=x..y,digits=30));
```

to approximate the integral with 30 digits of precision, and the software Mathematica™ in [Wolfram Research Inc 1998] for the online evaluation of $I^{\mu,p}_{x,y}$.

5.6 Conclusion and perspectives

In this chapter, we proposed an algorithm for the accurate evaluation of the generalized incomplete gamma function $I^{\mu,p}_{x,y}$. According to our experiments, the

implementation of this algorithm with a standard double floating-point precision yields a relative error less than 10^{-10} (in the worst case scenario), and in general less than 10^{-13} for a large range of parameters, which is a drastic gain of accuracy in comparison to that obtained using the algorithm proposed in [Fullerton 1972]. Besides, our algorithm delivers the estimated value of the integral $I_{x,y}^{\mu,p}$ under a mantissa-exponent representation $I_{x,y}^{\mu,p} = \rho \cdot e^{\sigma}$, which greatly extends the range over which it can be computed (which proved useful in [Abergel et al. 2015], where the computation of sums and ratios of generalized incomplete gamma functions $I_{x,y}^{\mu,p}$ was required).

Note also that the general accuracy of the algorithm we propose could certainly be improved further using a floating-point arithmetic with more digits and a higher order generalization of the trapezoidal rule that we use for nearly identical integral bounds. Another interesting perspective would be to extend our approach to the computation of complex values for the integral $I_{x,y}^{\mu,p}$ (for instance, when p is noninteger and $\mu < 0$, or when x or y takes a nonreal value). The continued fractions we used remain valid for complex values of x and noninteger values of p (except when $I_{x,y}^{\mu,p}$ is indefinite), but as the recursive integration by part (5.14) cannot be used any more when p is noninteger and $\mu < 0$, another strategy would be needed to cover the corresponding parameters region.

Acknowledgments

The authors would like to thank the GDS Mathrice 2754 as well as MathStic and LAGA (Laboratoire Analyse, Géométrie et Applications) at Université Paris 13, for having kindly provided us an access to the computing server GAIA.

Algorithm 15: Accurate computation of $I_{x,y}^{\mu,p} = \int_x^y s^{p-1}\, e^{-\mu s}\, ds$.

Input: Three numbers $\mu \in \mathbb{R} \setminus \{0\}$, $x \in \mathbb{R}_+$, $y \in \mathbb{R}_+ \cup \{+\infty\}$ such as $y \geq x$, and a positive integer $p \geq 1$. Notice that the value $y = +\infty$ is allowed only when $\mu > 0$.

Output: Two numbers $\rho \in \mathbb{R}$ and $\sigma \in \mathbb{R} \cup \{-\infty\}$ such as $I_{x,y}^{\mu,p} \approx \rho \times e^{\sigma}$.

Requirements: Functions γ_μ^{ibp} (Algo. 12: select recursive integration by parts), $\gamma_\mu^{\text{cfrac}}$ (Algo. 12: select continued fraction), $\Gamma_\mu^{\text{cfrac}}$ (Algo. 13), $\log\Gamma$ (Algo. 14), and function p_{lim} (Eq. (5.24)).

if $x = y$ *at machine precision* **then** $(\rho, \sigma) \leftarrow (0, -\infty)$ **else**
- // Evaluate (m_A, n_A), (m_B, n_B) and $(\rho_{\texttt{diff}}, \sigma_{\texttt{diff}})$, some mantissa-exponent
- // representations of A, B and $I_{\texttt{diff}}$, such as $A = m_A\, e^{n_A}$, $B = m_B\, e^{n_B}$,
- // $I_{\texttt{diff}} = \rho_{\texttt{diff}}\, e^{\sigma_{\texttt{diff}}} = A - B$, and $I_{x,y}^{\mu,p} \approx I_{\texttt{diff}}$.
- **if** $\mu < 0$ **then**
  - **if** $p < p_{lim}(\mu y)$ **then** $(m_A, n_A) \leftarrow \gamma_\mu^{\text{ibp}}(p, y)$
  - **else** $(m_A, n_A) \leftarrow \gamma_\mu^{\text{cfrac}}(p, y)$
  - **if** $p < p_{lim}(\mu x)$ **then** $(m_B, n_B) \leftarrow \gamma_\mu^{\text{ibp}}(p, x)$
  - **else** $(m_B, n_B) \leftarrow \gamma_\mu^{\text{cfrac}}(p, x)$
- **else if** $\mu > 0$ **then**
  - **if** $p < p_{lim}(\mu x)$ **then**
    - $(m_A, n_A) \leftarrow \Gamma_\mu^{\text{cfrac}}(p, x)$
    - $(m_B, n_B) \leftarrow \Gamma_\mu^{\text{cfrac}}(p, y)$
  - **else if** $p_{lim}(\mu x) \leq p < p_{lim}(\mu y)$ **then**
    - $(m_A, n_A) \leftarrow (1, \log\Gamma(p) - p\,\log\mu)$
    - $(m_x, n_x) \leftarrow \gamma_\mu^{\text{cfrac}}(p, x)$
    - $(m_y, n_y) \leftarrow \Gamma_\mu^{\text{cfrac}}(p, y)$
    - $n_B \leftarrow \max(n_x, n_y)$
    - **if** $n_B = -\infty$ **then** $n_B \leftarrow 0$ // may happen when $x = 0$ and $y = +\infty$
    - $m_B \leftarrow m_x\, e^{n_x - n_B} + m_y\, e^{n_y - n_B}$
  - **else**
    - $(m_A, n_A) \leftarrow \gamma_\mu^{\text{cfrac}}(p, y)$
    - $(m_B, n_B) \leftarrow \gamma_\mu^{\text{cfrac}}(p, x)$
- $(\rho_{\text{diff}}, \sigma_{\text{diff}}) \leftarrow (m_A - m_B \cdot e^{n_B - n_A}, n_A)$
- // Compute the ratio $R_{x,y}^{\mu,p} = \widehat{\Delta} I_{\texttt{diff}} / \widehat{\Delta} I_{\texttt{trapezoid}}$ with a mantissa-exponent
- // representation (ρ_r, σ_r), as described in Section 5.3.3.
- $D \leftarrow D_{x,y}^{\mu,p}$ // explicit expression in Proposition 34
- $\rho_r \leftarrow 12 \cdot \frac{m_A + m_B \cdot (1 + |n_B| + |n_A|) e^{n_B - n_A + \rho_{\text{diff}}|\sigma_{\text{diff}}|}}{D} \cdot \varepsilon_{\text{machine}}$
- $\sigma_r \leftarrow \sigma_{\text{diff}} - 3\,\log(y - x) - \max(0, p - 3)\log y - \max(-\mu x, -\mu y)$
- **if** $m_R\, e^{n_R} > 1$ **then**
  - $n_x \leftarrow -\mu x + p\log x$
  - $n_y \leftarrow -\mu y + p\log y$
  - $\sigma \leftarrow \max(n_x, n_y)$
  - $\rho \leftarrow \frac{y-x}{2x}\, e^{n_x - \sigma} + \frac{y-x}{2y}\, e^{n_y - \sigma}$
- **else** $(\rho, \sigma) \leftarrow (\rho_{\text{diff}}, \sigma_{\text{diff}})$

return (ρ, σ)

Parameters setting ($\mu=-1$)	Algorithm 435 in [Fullerton 1972]	Relative error	Algorithm 15	Relative error
$\mu=1, x=9, y=11, p=1$	$1.067081029759719\cdot 10^{-4}$	$3\cdot 10^{-9}$	$1.0670810329643395\cdot 10^{-4}$	$6\cdot 10^{-16}$
$\mu=1, x=9, y=11, p=5$	$9.567113518714904\cdot 10^{-1}$	$1\cdot 10^{-4}$	$9.5661698023023700\cdot 10^{-1}$	$1\cdot 10^{-15}$
$\mu=1, x=9, y=11, p=10$	$1.085447578125000\cdot 10^{5}$	$2\cdot 10^{-1}$	$8.9594201765236983\cdot 10^{4}$	$1\cdot 10^{-14}$
$\mu=1, x=9, y=11, p=12$	$1.632943040000000\cdot 10^{8}$	17	$8.9310494815538749\cdot 10^{6}$	$3\cdot 10^{-15}$
$\mu=1, x=9, y=11, p=14$	$-2.977905664000000\cdot 10^{10}$	34	$9.0203414117080807\cdot 10^{8}$	$2\cdot 10^{-15}$
$\mu=1, x=9, y=11, p=100$	$-\mathrm{NaN}$	N/A	$2.5825265278752760\cdot 10^{97}$	$2\cdot 10^{-14}$
$\mu=1, x=9, y=11, p=300$	$-\mathrm{NaN}$	N/A	$1.5122076179085018\cdot 10^{305}$	$3\cdot 10^{-14}$
$\mu=1, x=9, y=11, p=1000$	$-\mathrm{NaN}$	N/A	$4.1710431880333560\cdot 10^{1033}$	$3\cdot 10^{-13}$
$\mu=1, x=100, y=120, p=1$	$3.783505853677006\cdot 10^{-44}$	$2\cdot 10^{-2}$	$3.7200759683531697\cdot 10^{-44}$	$5\cdot 10^{-15}$
$\mu=1, x=100, y=120, p=5$	$3.873433252162870\cdot 10^{-36}$	$3\cdot 10^{-9}$	$3.8734332644314730\cdot 10^{-36}$	$4\cdot 10^{-15}$
$\mu=1, x=100, y=120, p=10$	$4.083660502797843\cdot 10^{-26}$	$2\cdot 10^{-8}$	$4.0836605881700520\cdot 10^{-26}$	$8\cdot 10^{-15}$
$\mu=1, x=100, y=120, p=20$	$4.579807864502072\cdot 10^{-6}$	$9\cdot 10^{-8}$	$4.5798082802928473\cdot 10^{-6}$	$2\cdot 10^{-14}$
$\mu=1, x=100, y=120, p=21$	$+\infty$	N/A	$4.6360373381202165\cdot 10^{-4}$	$1\cdot 10^{-14}$
$\mu=1, x=100, y=120, p=100$	$-\mathrm{NaN}$	N/A	$4.2821563816534019\cdot 10^{155}$	$1\cdot 10^{-13}$
$\mu=1, x=100, y=120, p=170$	$-\mathrm{NaN}$	N/A	$4.2461593130874860\cdot 10^{299}$	$3\cdot 10^{-14}$
$\mu=1, x=100, y=120, p=1000$	$-\mathrm{NaN}$	N/A	$1.3223863318125477\cdot 10^{2024}$	$1\cdot 10^{-12}$

Table 5.3: Comparison between Algorithm 435 proposed in [Fullerton 1972] and Algorithm 15, for the computation of $I_{x,y}^{\mu,p}$ with $\mu=1$. In this series of experiments, we focus on the case $\mu=1$. We tested, for $(x,y)=(9,11)$, and $(x,y)=(100,120)$, different integer values of p between 1 and 1000. In the second column, we display the values of $I_{x,y}^{\mu,p}$ returned by Fullerton's Algorithm (that we slightly adapted to compute $I_{x,y}^{\mu,p}$ instead of $J_{x,y}^{\mu,p}$). The corresponding relative errors, evaluated using Mathematica or Maple softwares (both softwares yield the same relative error), are displayed on the third and fourth columns. We see that some numerical instabilities arise when $x\leq p$, and we observe some overflow issues, as p gets high. Some inacurrate results are also observed for low values of p, when $x=100, y=120, p=1$ (but also for many other values of (x,y,p), not represented here). Note also the settings $x=100, y=120, p\in\{20,21\}$, for which the value returned by the algorithm shifts from 10^{-6} to $+\infty$. In the fourth column, we display the values returned by Algorithm 15 (using a C implementation with standard double precision), followed by the corresponding relative errors. The relative errors reached by Algorithm 15 are nearly optimal, since they are mostly due to the loss of precision involved by the mantissa-exponent representation (see the optimality bounds predicted in (5.21) and (5.23)), showing that both mantissa and exponents are in practice computed with a relative precision close to the machine precision.

Parameters setting ($\mu = -1$)	Algorithm 435 in [Fullerton 1972]	Relative error	Algorithm 15	Relative error
$\mu = -1, x = 5, y = 10, p = 1$	$2.187916015625000 \cdot 10^{4}$	$5 \cdot 10^{-5}$	$2.1878052635704163 \cdot 10^{4}$	$1 \cdot 10^{-15}$
$\mu = -1, x = 5, y = 10, p = 3$	$1.803647750000000 \cdot 10^{6}$	$3 \cdot 10^{-7}$	$1.8036471714694066 \cdot 10^{6}$	$1 \cdot 10^{-16}$
$\mu = -1, x = 5, y = 10, p = 10$	$1.129511596851200 \cdot 10^{13}$	$4 \cdot 10^{-8}$	$1.1295115549498505 \cdot 10^{13}$	$4 \cdot 10^{-15}$
$\mu = -1, x = 5, y = 10, p = 60$	$+\infty$	N/A	$3.1530071119035434 \cdot 10^{62}$	$2 \cdot 10^{-14}$
$\mu = -1, x = 5, y = 10, p = 100$	$-\mathrm{NaN}$	N/A	$2.0040499509396790 \cdot 10^{102}$	$1 \cdot 10^{-14}$
$\mu = -1, x = 5, y = 10, p = 300$	$-\mathrm{NaN}$	N/A	$7.1060487642415961 \cdot 10^{301}$	$9 \cdot 10^{-14}$
$\mu = -1, x = 5, y = 10, p = 1000$	$-\mathrm{NaN}$	N/A	$2.1808595556561760 \cdot 10^{1001}$	$3 \cdot 10^{-13}$
$\mu = -1, x = 20, y = 25, p = 1$	$7.151973171200000 \cdot 10^{10}$	$3 \cdot 10^{-8}$	$7.1519734141975967 \cdot 10^{10}$	$2 \cdot 10^{-15}$
$\mu = -1, x = 20, y = 25, p = 10$	$2.068890077987267 \cdot 10^{23}$	$3 \cdot 10^{-2}$	$2.0016822370845540 \cdot 10^{23}$	$9 \cdot 10^{-16}$
$\mu = -1, x = 20, y = 25, p = 20$	$1.821993954177914 \cdot 10^{37}$	$2 \cdot 10^{-1}$	$1.4733948083664500 \cdot 10^{37}$	$1 \cdot 10^{-15}$
$\mu = -1, x = 20, y = 25, p = 30$	$+\infty$	N/A	$1.1449672725827719 \cdot 10^{51}$	$3 \cdot 10^{-15}$
$\mu = -1, x = 20, y = 25, p = 210$	$-\mathrm{NaN}$	N/A	$1.1321187815658028 \cdot 10^{302}$	$2 \cdot 10^{-14}$
$\mu = -1, x = 20, y = 25, p = 1000$	$-\mathrm{NaN}$	N/A	$6.1186720860190186 \cdot 10^{1405}$	$2 \cdot 10^{-13}$

Table 5.4: Same as Table 5.3, but for $\mu = -1$. We performed here a similar experiment as in Table 5.3, but in the case $\mu < 0$. We tested, for $(x, y) = (5, 10)$ and $(x, y) = (20, 25)$, different values of p between 1 and 1000. As we can see, Fullerton's algorithm is unbable to provide accurate estimates as p increases. In contrast, the relative errors reached by Algorithm 15 remain nearly optimal, mostly limited by the mantissa-exponent representation according to the optimality bounds derived in (5.21) and (5.23).

Parameters setting	Algorithm 435 in [Fullerton 1972]	Relative error	Algorithm 15	Selected approx.	Relative error
$\mu = 1, x = d(5 - 10^{0}), y = 5, p = 10$	$8.598737304687500 \cdot 10^{3}$	$2 \cdot 10^{-8}$	$8.5987371691242424 \cdot 10^{3}$	difference	$7 \cdot 10^{-17}$
$\mu = 1, x = d(5 - 10^{-1}), y = 5, p = 10$	$1.263989379882812 \cdot 10^{3}$	$8 \cdot 10^{-7}$	$1.2639903706449711 \cdot 10^{3}$	difference	$1 \cdot 10^{-15}$
$\mu = 1, x = d(5 - 10^{-3}), y = 5, p = 10$	$1.315382766723632 \cdot 10^{1}$	$7 \cdot 10^{-5}$	$1.3154789325748350 \cdot 10^{1}$	difference	$1 \cdot 10^{-13}$
$\mu = 1, x = d(5 - 10^{-4}), y = 5, p = 10$	$1.317400574684143 \cdot 10^{0}$	$1 \cdot 10^{-3}$	$1.3159526336877760 \cdot 10^{0}$	difference	$2 \cdot 10^{-11}$
$\mu = 1, x = d(5 - 10^{-5}), y = 5, p = 10$	$1.322335302829742 \cdot 10^{-1}$	$5 \cdot 10^{-3}$	$1.3160000091971793 \cdot 10^{-1}$	trap. rule	$2 \cdot 10^{-12}$
$\mu = 1, x = d(5 - 10^{-6}), y = 5, p = 10$	$1.179141830652952 \cdot 10^{-2}$	$1 \cdot 10^{-1}$	$1.3160047470408107 \cdot 10^{-2}$	trap. rule	$2 \cdot 10^{-14}$
$\mu = 1, x = d(5 - 10^{-7}), y = 5, p = 10$	0	1	$1.3160052243091611 \cdot 10^{-3}$	trap. rule	$4 \cdot 10^{-16}$
$\mu = 1, x = d(17 - 10^{0}), y = 17, p = 17$	$3.725839564800000 \cdot 10^{12}$	$8 \cdot 10^{-1}$	$2.0551230250736027 \cdot 10^{12}$	difference	$3 \cdot 10^{-14}$
$\mu = 1, x = d(17 - 10^{-1}), y = 17, p = 17$	$2.998156984320000 \cdot 10^{11}$	$5 \cdot 10^{-1}$	$2.0202925544709274 \cdot 10^{11}$	difference	$2 \cdot 10^{-13}$
$\mu = 1, x = d(17 - 10^{-3}), y = 17, p = 17$	$2.941651456000000 \cdot 10^{9}$	$5 \cdot 10^{-1}$	$2.0146022707357914 \cdot 10^{9}$	difference	$1 \cdot 10^{-11}$
$\mu = 1, x = d(17 - 10^{-4}), y = 17, p = 17$	$2.928078720000000 \cdot 10^{8}$	$5 \cdot 10^{-1}$	$2.0145489617316359 \cdot 10^{8}$	trap. rule	$4 \cdot 10^{-11}$
$\mu = 1, x = d(17 - 10^{-6}), y = 17, p = 17$	$6.554762500000000 \cdot 10^{6}$	$2 \cdot 10^{0}$	$2.0145430981932636 \cdot 10^{6}$	trap. rule	$2 \cdot 10^{-15}$
$\mu = 1, x = d(17 - 10^{-9}), y = 17, p = 17$	0	1	$2.0145432036144500 \cdot 10^{3}$	trap. rule	$2 \cdot 10^{-15}$
$\mu = -1, x = d(21 - 10^{0}), y = 21, p = 10$	$5.859836137154984 \cdot 10^{20}$	$5 \cdot 10^{-2}$	$5.5623377927217197 \cdot 10^{20}$	difference	$4 \cdot 10^{-15}$
$\mu = -1, x = d(21 - 10^{-1}), y = 21, p = 10$	$1.025911814748488 \cdot 10^{20}$	$5 \cdot 10^{-2}$	$9.7609411144076116 \cdot 10^{19}$	difference	$7 \cdot 10^{-15}$
$\mu = -1, x = d(21 - 10^{-3}), y = 21, p = 10$	$1.099215584270221 \cdot 10^{18}$	$5 \cdot 10^{-2}$	$1.0467611548908131 \cdot 10^{18}$	difference	$6 \cdot 10^{-12}$
$\mu = -1, x = d(21 - 10^{-5}), y = 21, p = 10$	$1.045801880623513 \cdot 10^{16}$	$2 \cdot 10^{-3}$	$1.0475015408230294 \cdot 10^{16}$	trap. rule	$2 \cdot 10^{-11}$
$\mu = -1, x = d(21 - 10^{-7}), y = 21, p = 10$	0	1	$1.0475089604363028 \cdot 10^{14}$	trap. rule	$2 \cdot 10^{-15}$
$\mu = -1, x = d(21 - 10^{-9}), y = 21, p = 10$	0	1	$1.0475091089401447 \cdot 10^{12}$	trap. rule	$3 \cdot 10^{-15}$

Table 5.5: Comparison between Fullerton's algorithm and Algorithm 15, for the computation of $I_{x,y}^{\mu,p}$ when $x \approx y$. In this last experiment, we compute $I_{x,y}^{\mu,p}$ in the case $x \approx y$ (the notation $d(s)$ used in the left column denotes the double-precision floating-point number that is closest to s). We see that the relative error reached by Algorithm 435 deteriorates as x and y get close to each other, and as already remarked before, Algorithm 435 is very inacurrate when $\mu x < p < \mu y$. In contrast, the relative errors observed with Algorithm 15 never exceed 10^{-10}, thanks to the first order estimate (keyword "trap. rule" in column 5) that takes over to avoid cancellation errors when x and y are very close to each other.

Chapter 6

A-contrario Algorithms for Computing Motion Correspondence in a Noisy Point Set Sequence

Contents

The content of this chapter has been partially presented in the conference paper [Abergel and Moisan 2014], at the occasion of the 22nd European Signal Processing Conference (EUSIPCO 2014). Moreover, a software is distributed at the address http://www.math-info.univ-paris5.fr/~rabergel/cutastre.html, it corresponds to our C language implementations of the algorithms presented in this chapter.

Abstract

The detection of smooth trajectories in a (noisy) point set sequence can be realized optimally with the ASTRE (A-contrario Smooth TRajectory Extraction) algorithm, but the quadratic time and memory complexity of this algorithm with respect to the number of frames is prohibitive for many practical applications. We here propose a variant that cuts the input sequence into overlapping temporal chunks that are processed in a sequential (but non-independent) way, which results in a linear complexity with respect to the number of frames. Surprisingly, the performances are not affected by this acceleration strategy, and are in general even slightly above those of the original ASTRE algorithm.

6.1 Introduction

Many image processing tasks that concern video or image sequences are related to various forms of motion analysis like optical flow, object tracking, trajectory detection, motion compensation, etc. In this chapter, we consider the fundamental problem of finding reliable trajectories in a point set sequence that has been previously extracted from an image sequence, without any attribute attached to each point. This task, which has been considered several times in the literature [Veenman et al. 2001, Shafique and Shah 2003, Khan et al. 2005, Berclaz et al. 2011, Collins 2012], is the core of various applications including, e.g., particle velocimetry in fluid mechanics, dynamic analysis of fluorescent probes in biology, study of ant or termite behavior, pedestrian and car tracking, etc. For the case where smooth trajectories (more precisely, trajectories having a small maximum acceleration) are to be detected among a potentially high number of incoherent noise points, an algorithm with optimality guarantees, called ASTRE [Primet and Moisan 2012], has been recently built, but it turns out that it is nearly impossible to use it on long image sequences (say $K \geq 1000$ frames) because its time and memory complexity is quadratic with respect to K. We here propose to break this complexity limitation and describe a new algorithm for which the complexity is linear with respect to K, which substantially increases the possibilities of real-world applications.

In Section 6.2, we describe the original ASTRE algorithm, and show that the introduction of a maximum speed threshold may bring an important speed-up, but does not break the $\mathscr{O}(K^2)$ complexity. This is why in Section 6.3 we present a new algorithm named CUTASTRE, which cuts the original image sequence into overlapping small temporal chunks and processes these chunks sequentially with

the ASTRE algorithm, using an incremental strategy to detect long trajectories. This $\mathscr{O}(K)$ algorithm, though theoretically sub-optimal in terms of detection performances, still offers (like ASTRE) a rigorous control of false detections in pure noise data. The principle of CUTASTRE is presented in Section 6.3.1, and a pseudocode description of the algorithm is proposed in 6.3.2. The new parameters (chunk size and overlapping ratio) are analyzed in Section 6.3.3, and appear to be rather easy to set. Moreover, numerical experiments on both synthetic and natural point set sequences reveal that the detection performances of CUTASTRE are very similar to those of ASTRE, which, considering the dramatic speed-up offered by CUTASTRE, opens very interesting perspectives.

6.2 The ASTRE Algorithm

6.2.1 Principle

Based on the a-contrario methodology [Desolneux et al. 2008], the ASTRE algorithm is designed to perform trajectory detection over a sequence of K frames $f_1, f_2, \ldots, f_K$ with domain Ω, such as each frame f_k contains N_k points. We will first recall the principles of ASTRE in the continuous setting, when Ω is the square $[0,1] \times [0,1]$, and when the number of points per frame is constant (for any k, $N_k = N$). Then we will explain how the model can be modified to handle data-quantization (that is, discrete Ω) and variable number of points over the frame sequence.

The continuous framework with constant number of points

General a-contrario algorithms are based on two main ingredients: a naive model (called $\mathcal{H}_0$) describing what could be pure noise data, and a measurement function that characterizes the kind of structures looked for (that is here, smooth trajectories). The basic idea of the a-contrario methodology (which is motivated by the *Helmoltz principle*) is to detect the structures according to a principle of rejection of $\mathcal{H}_0$. More precisely we are interested in the detection (and the extraction) of the structures which are too rare to happen by chance in $\mathcal{H}_0$. In the case of ASTRE, the naive model $\mathcal{H}_0$ is a uniform draw of N points in each of the K frames, and the measurement function associated to a random trajectory $T = (X_{i_1}^{k_0}, X_{i_2}^{k_0+1}, \ldots, X_{i_\ell}^{k_0+\ell-1})$ with length $\ell \geq 3$ (we do not consider trajectories

with less than 3 points) is its acceleration

$$a(T) = \max_{p=3,\dots,\ell} \left\| X_{i_p}^{k_0+p-1} - 2X_{i_{p-1}}^{k_0+p-2} + X_{i_{p-2}}^{k_0+p-3} \right\|,$$

where X_i^k is the i-th (random) point of frame f_k. We shall denote the random trajectory T by

$$T = X_{i_1}^{k_0} \to X_{i_2}^{k_0+1} \to \cdots \to X_{i_\ell}^{k_0+\ell-1},$$

and a link between two successive points X_i^k and X_j^{k+1} by $X_i^k \to X_j^{k+1}$. Similarly, the local discrete acceleration will be written

$$a(u \to v \to w) = \|w - 2v + u\|.$$

The amount of surprise when observing an actual trajectory t with length ℓ and acceleration $a(t)$ can be estimated by using a simple (but precise) upper bound of the probability of observing a trajectory with acceleration smaller than $a(t)$ in $\mathcal{H}_0$ (see Proposition 2 in [Primet and Moisan 2012]),

$$\mathbb{P}_{\mathcal{H}_0}(a(T) \leq a(t)) \leq \left(\pi a(t)^2\right)^{\ell-2}. \tag{6.1}$$

It turns out that the smaller is the quantity $(\pi a(t)^2)^{\ell-2}$, the smoother is trajectory t and the less it is likely to have been generated by the naive model $\mathcal{H}_0$. Therefore, we might be tempted to use $s(t) := (\pi a(t)^2)^{\ell-2}$ as an inverted score for the trajectory t (inverted because we want this score as small as possible), and to use a threshold on this score in order to decide if trajectory t should be extracted or not. However, this score does not take into account all the parameters of the model (in particular the number of points per frame N, and the total number of frames K of the sequence). For instance, as K and N get large, the probability of observing by chance a trajectory with small acceleration in a random sequence following $\mathcal{H}_0$ increases, thus, the amounts of surprise related to the observation of a trajectory t with acceleration $a(t)$ should decrease as K and N increase. Consequently, the threshold used for the score s should depend on the model parameters K and N. Finally, the weakness of this approach is that the choice of the threshold would require a double expertise from the user, who must be not only familiar with its data, but also with the mathematical content of the algorithm.

Using the a-contrario methodology (see Proposition 2 in [Grosjean and Moisan 2009], we will also give more details below), we go a step further and design below a *Number of False Alarms* (NFA) for the measurement $a(t)$. Let $\mathbb{T}_\ell$ denote the

set of all trajectories of length ℓ (for $\ell \in \mathbb{N}$) in the considered point set sequence. We set

$$\forall \ell \geq 3,\ \forall t \in \mathbb{T}_\ell, \quad \mathrm{NFA}_\ell(a(t)) = K(K-\ell+1)N^\ell \left(\pi a(t)^2\right)^{\ell-2} . \tag{6.2}$$

First, we see that this formula merges the model parameters (K, N, ℓ), with the score $s(t) = (\pi a(t)^2)^{\ell-2}$, and more precisely, it weights the score $s(t)$ with a quantity depending on the model parameters. Second, and before giving more details about how was built the NFA formula (6.2), let us announce the so-called NFA-property which is one of the most fundamental result inherited from the a-contrario methodology. Let $\mathbb{T} = \bigcup_{\ell=3}^{K} \mathbb{T}_\ell$ denote the set of all trajectories with length $\ell \in [3, K]$ that can be found in the point set sequence, we have

$$\forall \varepsilon > 0, \quad \mathbb{E}_{\mathcal{H}_0}\left[\#\left\{T \in \mathbb{T} \;\middle|\; \mathrm{NFA}_{\ell(T)}(a(T)) \leq \varepsilon\right\}\right] \leq \varepsilon\,, \tag{6.3}$$

where, as usual, the notation $\#S$ stands for the cardinality of the set S, and $\ell(T)$ denotes the length of the trajectory T. Formulated differently, the NFA-property (6.3) states that in pure noise sequences (i.e. the random sequences following $\mathcal{H}_0$), the average number of trajectories with NFA less than ε is less than ε. In practice, we will use the NFA (6.2) as a score for each trajectory, so that we will manage to extract all trajectories having a NFA less than a given threshold $\varepsilon > 0$. In the following, we will say that a trajectory is ε-meaningful (or detected at level ε) when it has a NFA smaller than ε. Of course, we immediately remark that

$$\forall \ell \geq 3,\ \forall t \in \mathbb{T}_\ell, \quad \mathrm{NFA}_\ell(a(t)) \leq \varepsilon \Leftrightarrow (\pi a(t)^2)^{\ell-2} \leq \frac{\varepsilon}{K(K-\ell+1)N^\ell}\,,$$

so that thresholding $\mathrm{NFA}_\ell(a(t))$ using the threshold ε amounts to threshold the previously considered score $s(t) = (\pi a(t)^2)^{\ell-2}$ using an adaptive threshold (which explicitly depends on K, N, and ℓ). Besides, the NFA-property (6.3) gives a practical and very intuitive interpretation for the threshold parameter ε, which is the unique parameter involved in the ASTRE model. Indeed, ε is simply an upper bound of the number of detections (in fact, false detections) allowed in a pure noise sequence. Usually one sets $\varepsilon = 1$, so that the NFA-property ensures that in average, less than one detection is done in $\mathcal{H}_0$.

Finally, the NFA-score proposed in (6.2) smartly combines all the parameters of the model (the number of frames K of the sequence, the number of points per frame N, the length ℓ of each tested trajectory) with the measurements that we designed (here, the acceleration of each tested trajectory) to produce a formula

easy to threshold, in the sense that the threshold parameter ε comes with a concrete, simple and intuitive interpretation for the user, but also with a strong guarantee provided by (6.3), about the number of false detections expected in pure noise data sequences. The practical algorithm proposed in [Primet and Moisan 2012] for the extraction of the ε-meaningful trajectories is greedy, it looks for a trajectory t with minimal NFA-score m and, when $m \leq \varepsilon$, the trajectory t is detected at level ε, its points are removed from the sequence, and the process is repeated until no more trajectory with NFA less than ε can be found in the sequence. The algorithm will be detailed in Section 6.2.2.

Details about the NFA construction

The remarkable NFA-property (6.3) gives a tangible meaning to the threshold parameter ε. Let us recall how the NFA formula (6.2) was built by directly applying the generic method proposed in [Grosjean and Moisan 2009] which explains how to design NFA formulas that automatically satisfy the NFA-property. This generic method consists in grouping the structure of interest (here the trajectories) using a particular real-valued weighting family $\{w_t\}_{t\in\mathbb{T}}$ which must satisfy $\sum_{t\in\mathbb{T}} \frac{1}{w_t} \leq 1$. In the case of ASTRE, the structures are grouped according to their length, we set

$$\forall t \in \mathbb{T}, \quad w_t = w_{\ell(t)} := K \cdot \#\mathbb{T}_{\ell(t)} = K(K - \ell(t) + 1)N^{\ell(t)}\,, \tag{6.4}$$

which indeed satisfies

$$\sum_{t\in\mathbb{T}} \frac{1}{w_t} = \sum_{\ell=3}^{K} \sum_{t\in\mathbb{T}_\ell} \frac{1}{K \cdot \#\mathbb{T}_\ell} = \sum_{\ell=3}^{K} \frac{1}{K} \leq 1\,.$$

Then, we will show that one defines a NFA for the measurement $\delta = a(T)$ by setting

$$\forall T \in \mathbb{T},\ \forall \delta \in \mathbb{R}, \quad \mathrm{NFA}_{\ell(T)}(\delta) = w_{\ell(T)} \cdot F(\delta)\,, \tag{6.5}$$

where $F = \delta \mapsto \mathbb{P}_{\mathcal{H}_0}(a(T) \leq \delta)$ is the cumulative function of the random measurement $a(T)$. Before showing that (6.5) does satisfy the NFA-property, remark that

$$\mathrm{NFA}_{\ell(T)}(\delta) \leq \varepsilon \Leftrightarrow F(\delta) \leq \tfrac{\varepsilon}{w_{\ell(T)}}\,,$$

therefore, we see that the weighting family $\{w_T\}_{T\in\mathbb{T}}$ gives in practice a way to adjust the detection thresholds for each structure T (the particular choice done here of grouping the trajectories according to their length, by giving the same

weight to trajectories having the same length, is discussed in [Primet and Moisan 2012]).

Now, let $\mathbb{1}_{\mathrm{NFA}_{\ell(T)}(a(T))\leq\varepsilon}$ be the random variable taking the value 1 in the case $\mathrm{NFA}_{\ell(T)}(a(T)) \leq \varepsilon$, and the value 0 otherwise. We have

$$\begin{aligned}\mathbb{E}_{\mathcal{H}_0}\left[\#\left\{T \in \mathbb{T} \;\middle|\; \mathrm{NFA}_{\ell(T)}(a(T)) \leq \varepsilon\right\}\right] &= \sum_{T\in\mathbb{T}} \mathbb{E}_{\mathcal{H}_0}\left[\mathbb{1}_{\mathrm{NFA}_{\ell(T)}(a(T))\leq\varepsilon}\right] \\ &= \sum_{T\in\mathbb{T}} \mathbb{P}_{\mathcal{H}_0}\left(F(a(T)) \leq \frac{\varepsilon}{w_{\ell(T)}}\right).\end{aligned}$$

Besides, using a *p-value* property (see Lemma 1 in [Grosjean and Moisan 2009]), we get

$$\forall T \in \mathbb{T},\ \forall s \in \mathbb{R}_+, \quad \mathbb{P}_{\mathcal{H}_0}(F(a(T)) \leq s) \leq s\,,$$

so that we have

$$\mathbb{E}_{\mathcal{H}_0}\left[\#\left\{T \in \mathbb{T} \;\middle|\; \mathrm{NFA}_{\ell(T)}(a(T)) \leq \varepsilon\right\}\right] \leq \sum_{T\in\mathbb{T}} \frac{\varepsilon}{w_{\ell(T)}} \leq \varepsilon\,,$$

which means that (6.5) satisfies the NFA-property (6.3). Last, remark that when the cumulative function F involved in (6.5) is not easy to compute, it can be replaced by any function G satisfying $F \leq G$, since in that case, the choice $\mathrm{NFA}_{\ell(T)}(\delta) = w_{\ell(T)} \cdot G(\delta)$ yields

$$\begin{aligned}\mathbb{E}_{\mathcal{H}_0}\left[\#\left\{T \in \mathbb{T} \;\middle|\; \mathrm{NFA}_{\ell(T)}(a(T)) \leq \varepsilon\right\}\right] &= \sum_{T\in\mathbb{T}} \mathbb{P}_{\mathcal{H}_0}\left(G(a(T)) \leq \frac{\varepsilon}{w_{\ell(T)}}\right) \\ &\leq \sum_{T\in\mathbb{T}} \mathbb{P}_{\mathcal{H}_0}\left(F(a(T)) \leq \frac{\varepsilon}{w_{\ell(T)}}\right) \leq \varepsilon\,.\end{aligned}$$

In the case of ASTRE, the exact computation of the cumulative function F is difficult (some local accelerations of the 3-uple are less represented near the edges of Ω), this is the reason why the (tight) bound (6.1) was needed to build the NFA (6.2).

Variable number of points

For the practical application of ASTRE to real datasets, it is important to address the case where the number of points of the sequence is non-constant, that is, when each frame f_i contains N_i points (with potentially $N_i \neq N_j$ for two given frames f_i, f_j of the sequence). As remarked in [Primet and Moisan 2012],

since the NFA is an upper bound on the average number of false detections made in $\mathcal{H}_0$, we can simply take $N = \max_{1\leq k\leq K} N_k$ in (6.2). However, this choice is imprecise in the sense that the actual expected number of detections at level ε in $\mathcal{H}_0$ may be much less than the ε, especially when the sequence $\{N_k\}_{1\leq k\leq K}$ exhibits many values N_k being far from the maximum N. In order to obtain more accurate results, we can refine the NFA definition by grouping the trajectories according to their length ℓ and their starting frame k_0 in (6.4). Given (ℓ, k_0) such as $3 \leq \ell \leq K$ and $1 \leq k_0 \leq K - \ell + 1$, and noting $\mathbb{T}_{\ell,k_0}$ the set of all trajectories with length ℓ starting at frame k_0, we set

$$\forall t \in \mathbb{T}_{\ell,k_0}, \quad w_t = w_{k_0,\ell} := K(K-\ell+1) \cdot \#\mathbb{T}_{\ell,k_0} = K(K-\ell+1) \prod_{k=k_0}^{k_0+\ell-1} N_k \,.$$

By construction, we have

$$\sum_{t\in\mathbb{T}} \frac{1}{w_t} = \sum_{\ell=3}^{K} \sum_{k_0=1}^{K-\ell+1} \sum_{t\in\mathbb{T}_{\ell,k_0}} \frac{1}{K(K-\ell+1)\#\mathbb{T}_{\ell,k_0}} \leq 1 \,,$$

thus, applying again the generic methodology presented above, we obtain a NFA for the measurement $a(t)$ by setting

$$\forall t \in \mathbb{T}_{\ell,k_0}, \quad \mathrm{NFA}_{\ell,k_0}(a(t)) = K(K-\ell+1) \left(\prod_{k=k_0}^{k_0+\ell-1} N_k \right) \cdot \left(\pi a(t)^2\right)^{\ell-2} , \quad (6.6)$$

for any (ℓ, k_0) such as $3 \leq \ell \leq K$ and $1 \leq k_0 \leq K - \ell + 1$.

The Discrete framework

Another important point to take into account is the data quantization, since in most applications, the point detection is realized on a discrete grid of integer pixel coordinates. Consequently, it may happen that three consecutive points (or more) in the sequence have null acceleration, which is a strong contradiction with the (continuous) $\mathcal{H}_0$ model (in which the points are uniformly sampled over the continuous domain $\Omega = [0,1] \times [0,1]$), since such an event arises with null probability in $\mathcal{H}_0$. Besides when a trajectory t with null acceleration is observed, (6.2) (or (6.6)) yields a null NFA-score, which is the smallest NFA-score observable in the sequence. Therefore, any trajectory t' containing the points of t, but such as $a(t') \neq 0$, will present a NFA higher than t, so that the greedy ASTRE algorithm

will first cut the longer trajectory t' into chunks to isolate the (optimal) null-NFA subtrajectory t.

This undesirable behaviour can be avoided by using a quantized $\mathcal{H}_0$ model that we note $\mathcal{H}_0^{\mathrm{d}}$, in which the points of the sequence are uniformly sampled on a bounded discrete domain Ω of $\mathbb{Z}^2$. A probability bound similar to (6.1) can be computed in $\mathcal{H}_0^{\mathrm{d}}$, one can show that

$$\forall \ell \geq 3,\ \forall t \in \mathbb{T}_\ell, \quad \mathbb{P}_{\mathcal{H}_0^{\mathrm{d}}}(a(T) \leq a(t)) \leq \left(a^d(t)\right)^{\ell-2}, \tag{6.7}$$

where

$$a^d(t) = \frac{\#\left(\mathbb{Z}^2 \cap \mathscr{B}(0, a(t))\right)}{\#\Omega}, \tag{6.8}$$

noting $\mathscr{B}(0,\delta)$ the continuous closed ball with center 0 and radius δ. The quantity $a^d(t)$ defined in (6.8) is called the *discrete acceleration of trajectory* t, it measures the ratio between the number of pixels enclosed in the discrete disc $\mathbb{Z}^2 \cap \mathscr{B}(0, a(t))$ with that enclosed in Ω, and it can be viewed as a discrete equivalent of the continuous acceleration area $\pi a(t)^2$ involved in (6.1). As we did with the continuous acceleration in the continuous framework, the discrete acceleration will be used as a measurement of the amounts of surprise of observing a given trajectory in $\mathcal{H}_0^{\mathrm{d}}$. In particular, the observation of a trajectory $t \in \mathbb{T}_\ell$ with null (continuous) acceleration in $\mathcal{H}_0^{\mathrm{d}}$ yields the probability bound $(a^d(t))^{\ell-2} = (1/\#\Omega)^{\ell-2}$, which is nonzero anymore. Thanks to (6.7) and applying one more time the above presented generic methodology, we obtain a NFA for the measurement $a^d(t)$ by setting

$$\forall \ell \geq 3, \quad \forall t \in \mathbb{T}_\ell, \quad \mathrm{NFA}_{\ell,k_0}(a^d(t)) = K(K-\ell+1)N^\ell \left(a^d(t)\right)^{\ell-2}, \tag{6.9}$$

in the case of a constant number N of points per frame, or, by setting

$$\forall t \in \mathbb{T}_{\ell,k_0}, \quad \mathrm{NFA}_{\ell,k_0}(a^d(t)) = K(K-\ell+1)\left(\prod_{k=k_0}^{k_0+\ell-1} N_k\right)\left(a^d(t)\right)^{\ell-2}, \tag{6.10}$$

for any $\ell \in [3, K]$ and $k_0 \in [1, K-\ell+1]$, in the case of a variable number of points.

6.2.2 The proposed greedy algorithm

To compute the smallest NFA among all possible trajectories, a dynamic programming strategy (see [Bellman 1954]) is used. The idea is that, if two

trajectories t_1, t_2 have same length ℓ but different discrete accelerations (say $a^d(t_1) < a^d(t_2)$), then we automatically have $\mathrm{NFA}_\ell(a^d(t_1)) < \mathrm{NFA}_\ell(a^d(t_2))$, thus t_1 realizes a better NFA-score than t_2. Consequently, we can avoid the prohibitive systematic computation of the NFA of all the trajectories of the sequence in the following way: given a frame index $k \in [3, K]$, a pair of points $(x^k, y^{k-1}) \in f_k \times f_{k-1}$, and a trajectory length $\ell \in [3, k]$, let us call $\mathscr{G}(x^k, y^{k-1}, \ell)$ the smallest discrete acceleration of a trajectory $t \in \mathbb{T}_\ell$ ending with link $y^{k-1} \to x^k$ (we note $t = \cdots \to y^{k-1} \to x^k$), that is

$$\mathscr{G}(x^k, y^{k-1}, \ell) = \min_{t \in \mathbb{T}_\ell} a^d(t) \quad \text{subject to} \quad t = \cdots \to y^{k-1} \to x^k \,.$$

The quantity $\mathscr{G}$ can be recursively computed in $\mathscr{O}(N^3K^2)$ operations ($N = \max_{1 \le k \le K} N_k$ denotes the maximal number of points observed in a frame of the sequence) using the dynamic programming Algorithm 16, thanks to the Bellman Equation

$$\mathscr{G}(x^k, y^{k-1}, \ell) = \begin{cases} \frac{1}{\#\Omega} & \text{if } \ell = 2\,, \\ \min\limits_{z^{k-2} \in f_{k-2}} \overline{\mathscr{G}}(x^k, y^{k-1}, z^{k-2}, \ell) & \text{otherwise,} \end{cases} \tag{6.11}$$

where

$$\overline{\mathscr{G}}(x^k, y^{k-1}, z^{k-2}, \ell) = \max\left(a^d(z^{k-2} \to y^{k-1} \to x^k), \mathscr{G}(y^{k-1}, z^k, \ell - 1)\right).$$

Then, the smallest NFA over all trajectories of the input sequence is equal to $\mathrm{NFA}^d_{\ell_m}(\mathscr{G}(x^{k_m}, y^{k_m - 1}, \ell_m))$, where

$$(k_m, x_m, y_m, \ell_m) \in \operatorname*{argmin}_{\substack{k \in [3,K],\ \ell \in [3,k], \\ (y^{k-1}, x^k) \in f_{k-1} \times f_k}} \mathrm{NFA}^d_\ell(\mathscr{G}(x^k, y^{k-1}, \ell)), \tag{6.12}$$

and this minimal NFA is realized by a (non necessarily unique) trajectory with length ℓ_m and ending at frame k_m with the link $y_m \to x_m$ (that is, of the type $t_m = \cdots \to y_m \to x_m \in \mathbb{T}_{\ell_m}$). The minimal NFA over the sequence, as well as a 4-uple (k_m, x_m, y_m, ℓ_m) realizing the argmin (6.12) can be computed in $\mathscr{O}(N^2K^2)$ operations using Algorithm 17. Last, one can extract a trajectory t_m having minimal NFA using an iterative backtracking strategy on Equation (6.11). Indeed, a predecessor z_m (non necessarily unique) of y_m can be obtained by computing

$$z_m \in \operatorname*{argmin}_{z \in f_{k_m - 2}} \max\left(a^d(z \to y_m \to x_m), \mathscr{G}(y_m, z, \ell_m - 1)\right),$$

and we can reiterate this backtracking process by changing (x_m, y_m, ℓ_m) into $(y_m, z_m, \ell_m - 1)$ to find a predecessor of z_m, etc. Finally, a trajectory having minimal NFA can be extracted in $\mathscr{O}(NK)$ operations, using Algorithm 18.

Algorithm 16: `compute_`$\mathscr{G}$, dynamic programming computation of $\mathscr{G}$.

Input: The set of frames, $\mathscr{F} = \{f_1, \dots, f_K\}$, containing the points of the sequence.

Output: $\mathscr{G}$, such as for any $k \in [1, K]$, for any pair of points $(x, y) \in f_k \times f_{k-1}$, and for any $\ell \in [3, k]$, the quantity $\mathscr{G}(x, y, \ell)$ equals the smallest acceleration of a trajectory with length ℓ ending with link $y \to x$.

Requirements: The discrete acceleration $t \mapsto a^d(t)$ defined in (6.8).

for $2 \leq k \leq K$ **do**
  for $x \in f_k$ **do**
    for $y \in f_{k-1}$ **do**
      $\mathscr{G}(x, y, 2) \leftarrow \frac{1}{\#\Omega}$ `// the smallest discrete acceleration`
      for $3 \leq \ell \leq k$ **do**
        $\mathscr{G}(x, y, \ell) \leftarrow +\infty$
        for $z \in f_{k-2}$ **do**
          $a \leftarrow \max\left(a^d(z \to y \to x), \mathscr{G}(y, z, \ell - 1)\right)$
          $\mathscr{G}(x, y, \ell) \leftarrow \min\left(a, \mathscr{G}(x, y, \ell)\right)$

return $\mathscr{G}$

Algorithm 17: `minimal_NFA`, compute the minimal NFA in the sequence.

Input: The point set sequence $\mathscr{F} = \{f_1, \dots, f_K\}$.

Output: $\text{NFA}_{\min}$, the minimal NFA among all trajectories in $\mathscr{F}$, and (k_m, x_m, y_m, ℓ_m) such as $(y_m, x_m) \in f_{k_m-1} \times f_{k_m}$, and a trajectory with length ℓ_m ending with link $y_m \to x_m$ having NFA equal to $\text{NFA}_{\min}$ can be found in $\mathscr{F}$.

Requirements: The NFA_{ℓ,k_0} formula defined in (6.10).

$\mathscr{G} \leftarrow$ `compute_`$\mathscr{G}(\mathscr{F})$
$\text{NFA}_{\min} \leftarrow +\infty$
for $2 \le k \le K$ **do**
  for $x \in f_k$ **do**
    for $y \in f_{k-1}$ **do**
      for $3 \le \ell \le k$ **do**
        $k_0 \leftarrow k - \ell + 1$
        if $\text{NFA}_{\ell,k_0}(\mathscr{G}(x, y, \ell) < \text{NFA}_{\min}$ **then**
          $\text{NFA}_{\min} \leftarrow \text{NFA}_{\ell,k_0}(\mathscr{G}(x, y, \ell))$
          $(k_m, x_m, y_m, \ell_m) \leftarrow (k, x, y, \ell)$

return $(\text{NFA}_{\min}, k_m, x_m, y_m, \ell_m)$

Algorithm 18: `backtrack_trajectory`, perform trajectory backtracking.

Input: The point set sequence $\mathscr{F} = \{f_1, \dots, f_K\}$, a frame index $k_m \in [3, K]$, a pair of points $(y_m, x_m) \in f_{k_m-1} \times f_{k_m}$, and a trajectory length $\ell_m \in [3, k_m]$.

Output: A trajectory t_m having the smallest NFA among those of the type $t = \dots \to y_m \to x_m \in \mathbb{T}_{\ell_m}$.

set $t_m = y_m \to x_m$ `// initalization of` t_m `with` $y_m \to x_m \in \mathbb{T}_2$
while $\ell_m > 2$ **do**
  find $z_m \in \underset{z \in f_{k_m-2}}{\text{argmin}} \max\left(a^d(z \to y_m \to x_m), \mathscr{G}(y_m, z, \ell_m - 1)\right)$
  set $t_m = z_m \to t_m$ `// link` z_m `to the starting point of` t_m
  $(k_m, x_m, y_m, \ell_m) \leftarrow (k_m - 1, y_m, z_m, \ell_m - 1)$

return t_m

The final extraction scheme in [Primet and Moisan 2012] is greedy: if m, the minimal NFA among all possible trajectories is less than ε, a trajectory having NFA equal to m is extracted, its points are removed from the sequence and the process is repeated until no trajectory with NFA less than ε can be found any more. Besides, it is often possible to save computation time by extracting several trajectories at once without recomputing the function $\mathscr{G}$ each time, because removing points from the current data set cannot decrease any value of $\mathscr{G}$. Therefore, if two trajectories realize the two smallest NFA of the sequence without sharing any point, those two trajectories can be extracted at the same time, without recomputation of $\mathscr{G}$. This yields Algorithm 19.

Algorithm 19: `ASTRE`, greedy trajectories extraction according to their NFA.

Input: The set of frames, $\mathscr{F} = \{f_1, \ldots, f_K\}$, containing the points of the sequence, and ε, the maximal NFA of a trajectory to be extracted.

Output: $\mathscr{T} = \{t_1, \ldots, t_M\}$, the set of the extracted trajectories.

$\mathscr{T} \leftarrow \emptyset$
$\mathscr{F}' \leftarrow \mathscr{F}$
repeat
  $\mathscr{G} \leftarrow$ `compute_`$\mathscr{G}(\mathscr{F}')$
  $(\text{NFA}_{\min}, k_m, x_m, y_m, \ell_m) \leftarrow$ `minimal_NFA`$(\mathscr{F}')$
  stop $\leftarrow$ `false`
  while $m \leq \varepsilon$ *and stop* = `false` **do**
    $t_m \leftarrow$ `backtrack_trajectory`$(\mathscr{F}, k_m, x_m, y_m, \ell_m)$
    if t_m *shares some points with a trajectory* $t \in \mathscr{T}$ **then**
      `/* a recomputation of` $\mathscr{G}$ `is needed */`
      $\mathscr{F} \leftarrow \mathscr{F}'$
      stop $\leftarrow$ `true`
    else `/* the detection of` t_m `is accepted */`
      $\mathscr{T} \leftarrow \mathscr{T} \cup \{t_m\}$
      remove all the points of t_m from $\mathscr{F}'$
      $(\text{NFA}_{\min}, k_m, x_m, y_m, \ell_m) \leftarrow$ `minimal_NFA`$(\mathscr{F}')$
until $m > \varepsilon$ *or there are no more points in the sequence* $\mathscr{F}$
return $\mathscr{T}$

6.2.3 Improvement of the execution time

The main weakness of ASTRE is its quadratic time and memory complexity with respect to K, due to the extensive computation of $\mathscr{G}$. A very simple way to reduce the execution time of this algorithm is to introduce a threshold $\mathcal{S}_{\text{thre}}$ on the speed of the trajectories: as soon as the distance between two points x and y of two consecutive frames is higher than $\mathcal{S}_{\text{thre}}$, we consider that link $y \to x$ cannot exist. Hence we can avoid the computation of $\mathscr{G}(x, y, \ell)$ for those pairs of points.

A speed threshold is a physical parameter that can be easily adjusted in many applications. The use of this additional knowledge restricts the number of linking possibilities among the sequence, and reduces very significantly the execution time in general. However the complexity remains $\mathscr{O}(K^2)$ and ASTRE is still inapplicable to long data sequences, as can be seen in the ASTRE columns of Table 6.1.

6.3 An accelerated variant with linear complexity

6.3.1 Principle of the CUTASTRE Algorithm

Our approach consists in grouping consecutive frames of the full sequence $\mathscr{F} = \{f_1, \dots, f_K\}$ into overlapping chunks $\mathscr{B}_1, \dots, \mathscr{B}_n$ (an example of grouping is proposed in Figure 6.1, and the practical cutting procedure is detailed below). Trajectories will be detected within each chunk (starting from chunk $\mathscr{B}_n$) with an algorithm similar to ASTRE, which will be adapted to extend trajectories extracted from a chunk $\mathscr{B}_k$ when processing its predecessor $\mathscr{B}_{k-1}$. The function $\mathscr{G}$ will be computed only within a single chunk, allowing a drastic reduction of the time and memory complexity.

The cutting procedure.

Let c and o be two positive integers such as $c \leq K$ and $2 \leq o < c$. We will now regroup the frames $f_1, \dots, f_K$ into overlapping chunks $\mathscr{B}_1, \dots, \mathscr{B}_n$ of the type $\mathscr{B}_i = \{f_{k^i_{\text{start}}}, \dots, f_{k^i_{\text{end}}}\}$, such as, as far as possible, each chunk $\mathscr{B}_i$ contains c frames, and each overlapping area $\mathscr{R}_{i \leftrightarrow i+1} := \mathscr{B}_i \cap \mathscr{B}_{i+1}$ contains o frames. This condition will be in practice satisfied exactly in all the chunks $\mathscr{B}_1, \cdots, \mathscr{B}_{n-1}$, but might be unsatisfied for the last chunk $\mathscr{B}_n$, because of an inappropriate total number of frames K. To properly deal with this eventual edge effect, we impose to the last chunk $\mathscr{B}_n$ to contain at least c frames. It follows that n, the total

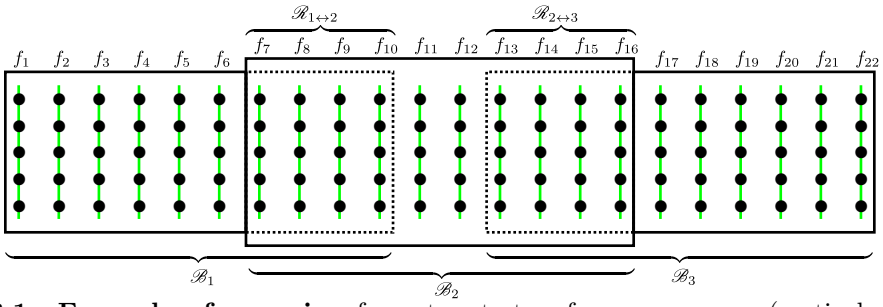


Figure 6.1: Example of grouping for a twenty-two frames sequence (vertical segments) containing five points each (black disks). Here frames are grouped into three chunks of ten frames each, with four frames overlap.

number of chunks is the highest integer i satisfying $k^i_{\text{start}} + c - 1 \leq K$, and since we can easily check that $k^i_{\text{start}} = 1 + (i-1)\cdot(c-o)$, we finally get

$$n = 1 + \left\lfloor \frac{K-c}{c-o} \right\rfloor . \tag{6.13}$$

Finally, the cutting procedure that we propose consists in setting

$$\forall i \in [1,n], \quad k^i_{\text{start}} = 1 + i\cdot(c-o), \quad k^i_{\text{end}} = \begin{cases} k^i_{\text{start}} + c - 1 & \text{if } i < n\,, \\ K & \text{if } i = n\,. \end{cases} \tag{6.14}$$

In the following we will denote by K_i the number of frames contained in the chunk $\mathscr{B}_i$, that is $K_i = k^i_{\text{end}} - k^i_{\text{start}} + 1$.

The sequential processing of the chunks.

The first chunk to be processed is $\mathscr{B}_n$. For this very first chunk we simply replace K by K_n in the NFA formula (6.2), and we apply the ASTRE algorithm to extract all the (ε/n)-meaningful trajectories from the corresponding subsequence of frames (we will explain later this choice of thresholding). When ASTRE terminates, we put back in the sequence all points that have been removed from the overlapping block of frames $\mathscr{R}_{n-1\leftrightarrow n}$. We remove the corresponding links from the detected trajectories, except those linking points of the two last frames of $\mathscr{R}_{n-1\leftrightarrow n}$. Finally, when a trajectory is entirely included into $\mathscr{R}_{n-1\leftrightarrow n}$, all its links are removed (this is the case of trajectory (4) in Figure 6.2-(a)). This ends the process for the chunk $\mathscr{B}_n$.

Let us now describe the process for the other chunks $(\mathscr{B}_i)_{1\leq i<n}$. We say that a trajectory t is extendable to $\mathscr{B}_i$ if t has been extracted while processing chunk

$\mathscr{B}_{i+1}$ (not $\mathscr{B}_i$) and reaches the two last frames (that we denote $\mathscr{F}_0^i$ and $\mathscr{F}_1^i$) of $\mathscr{R}_{i\leftrightarrow i+1}$ (see Figure 6.2-(a), trajectories (1) and (2) are extendable to $\mathscr{B}_2$, as both reach frames $\mathscr{F}_0^2 = f_{15}$ and $\mathscr{F}_1^2 = f_{16}$). To process these chunks, we need to adapt the computation of $\mathscr{G}(x, y, \ell)$ and $\mathrm{NFA}_\ell(\mathscr{G}(x, y, \ell))$ when x or y belongs to an extendable trajectory. Let us say we focus on chunk $\mathscr{B}_i$, three cases must be distinguished:

(i) neither x nor y belongs to a trajectory extendable to $\mathscr{B}_i$,

(ii) $x \in \mathscr{F}_1^i$ and x belongs to a trajectory t extendable to $\mathscr{B}_i$ that also contains y (necessarily $y \in \mathscr{F}_0^i$),

(iii) any other case.

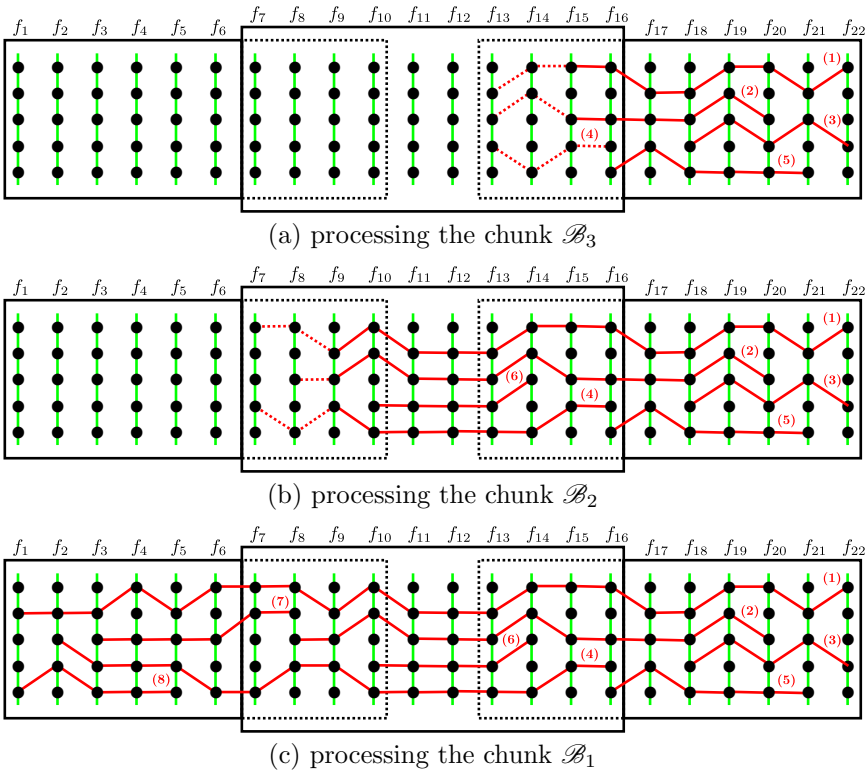
In case (i), K is replaced by K_i in (6.2), and $\mathrm{NFA}_\ell(\mathscr{G}(x, y, \ell))$ is computed exactly as in [Primet and Moisan 2012]. In case (ii), by construction, t starts with the link $y \to x$ (that is, $t = y \to x \to \cdots$), because all links before $y \to x$ were suppressed the trajectory t was extracted from the chunk $\mathscr{B}_{i+1}$. Therefore, any trajectory t' of $\mathscr{B}_i$ having length ℓ and ending with link $y \to x$ (that is, $t' = \cdots \to y \to x$, and $\ell(t') = \ell$) is an extension of t. Let t_0 be the sub-trajectory of t that is obtained by removing from t all points that do not belong to chunk $\mathscr{B}_{i+1}$ (or said differently, t_0 is the restriction of t to the chunk $\mathscr{B}_{i+1}$). We propose to take into account the trajectory extension by replacing, the quantities $\mathscr{G}(x, y, \ell)$, ℓ and K as follows when computing $\mathrm{NFA}_\ell(\mathscr{G}(x, y, \ell)$:

(a) replace $\mathscr{G}(x, y, \ell)$ by $\max(\mathscr{G}(x, y, \ell), a^d(t_0))$, which represents the smallest acceleration of an extension of t_0 by a trajectory $t' \in \mathbb{T}_\ell$ of $\mathscr{B}_i$ of the type $t' = \cdots \to y \to x$;

(b) replace ℓ by $\ell + \ell(t_0) - 2$, which represents the length of the extension of t_0 by t';

(c) replace K by the number of frames contained in the union of the two consecutive chunks $\mathscr{B}_i$ and $\mathscr{B}_{i+1}$, noted $\mathscr{B}_i \cup \mathscr{B}_{i+1}$.

Last, in case (iii) we simply set $\mathscr{G}(x, y, \ell) = +\infty$ and $\mathrm{NFA}_\ell(\mathscr{G}(x, y, \ell)) = +\infty$ in order to avoid the detection of a trajectory ending with link $y \to x$ (or equivalently, we just do not compute $\mathscr{G}(x, y, \ell)$ for such pairs of points). The strategy concerning trajectories extraction is exactly the same as in [Primet and Moisan 2012]; when all (ε/n)-meaningful trajectories are extracted, we set back again to the sequence all points belonging to $\mathscr{R}_{i-1\leftrightarrow i}$ and unless $\mathscr{B}_i$ is chunk $\mathscr{B}_1$, we repeat the link suppression process (see Figure 6.2, (b) and (c)).



(a) processing the chunk $\mathscr{B}_3$

(b) processing the chunk $\mathscr{B}_2$

(c) processing the chunk $\mathscr{B}_1$

Figure 6.2: Illustration of the trajectory extraction process operated by CUTASTRE. **(a)** Trajectories are being detected at level ε/n using ASTRE, taking $K_3 = 10$ instead of $K = 22$ for the NFA computation. Each time a trajectory is extracted, all its points are removed from the sequence. Once all trajectories are extracted, we set back to the sequence all points of overlapping frames (frames f_{13} to f_{16}). The corresponding links are removed (dotted links) excepting those linking a point of frame f_{15} with a point of frame f_{16}. Trajectory (4) being entirely included into $\mathscr{R}_{2\leftrightarrow 3}$, all its links are removed. **(b)** extraction of the (ε/n)-meaningful trajectories within chunk $\mathscr{B}_2$ with several links suppression within $\mathscr{R}_{1\leftrightarrow 2}$. Trajectories (1) and (2) have been extended, trajectory (4) is detected again but in a slightly different way (the link between frames f_{13} and f_{14} has changed), and is now longer. **(c)** extraction of the (ε/n)-meaningful trajectories within chunk $\mathscr{B}_1$, trajectories (1), (2) and (4) have been extended, no link suppression must be done in $\mathscr{B}_1$, since the trajectories will not be extended anymore.

The choice of locality.

The cutting strategy of the whole frame sequence into chunks $\mathscr{B}_1, \ldots, \mathscr{B}_n$ adopted by CUTASTRE breaks the quadratic complexity of ASTRE, yielding a $\mathscr{O}(K)$ complexity (the time and memory necessary to process the whole sequence is roughly the time and memory necessary to process a chunk, which is $\mathscr{O}(1)$, multiplied by the number of chunks n which is $\mathscr{O}(K)$). This drastic reduction of the

complexity will be illustrated with some practical experiments in Section 6.3.3.

We would like to emphasize that, beyond this improvement of the algorithmic complexity, the cutting strategy of CUTASTRE is also a way to introduce some locality (in time) to the underlying motion correspondence problem. Indeed, with ASTRE the trajectory extraction is based on a ranking of the trajectories according their global NFA, which takes into account the points of the whole sequence. With CUTASTRE we adopt a sliding window strategy, as the trajectories are first detected (in part) in a single chunk (according to their local NFA, computed within a single chunk) and can progressively extend from one chunk to another. When being extended, a trajectory is always seen as a structure of a two-consecutive-chunks sized sequence (thanks to the restriction of the considered extensions to the union $\mathscr{B}_i \cup \mathscr{B}_{i+1}$ arising in the case (ii) described above), and never more.

The role of the overlap areas.

The locality provided by CUTASTRE also introduced some difficulties. Since in practice a trajectory may start at any frame of the sequence, the limited temporal scope involved by the cutting of the whole sequence into smaller chunks can be problematic for a trajectory starting near the edge of a chunk, where some wrong linking can be decided due to a lack of information. These edge effects could have been in practice limited using overlapping chunks. Indeed, thanks to the link suppression process operated in the overlap areas, the CUTASTRE algorithm does not necessarily redraw removed links when a trajectory extension is done (look carefully at trajectory (4) in Figure 6.2). Therefore, these overlap areas can be seen as decision areas and removed links as hypotheses that can be validated or not when trajectories are being extended. We will show in Section 6.3.3 that, provided a good setting of the chunk and overlap size parameters, the CUTASTRE algorithm achieve better detection performances, both on synthetic and real datasets.

Preservation of the NFA property.

A natural question arises: do we still control the number of false detections by (6.3) like the ASTRE algorithm? The answer is yes, and it simply comes from the fact that, in $\mathcal{H}_0^{\mathrm{d}}$, the average number of new trajectories extracted in a given chunk is less than ε/n, so that the average total number of extracted trajectories in the whole sequence is less than ε. Although the preservation of this NFA-property remains interesting for applications where the number of false detections must be precisely controlled, this result is in fact rather poor. Indeed, once we impose to trajectories to be detected at level ε/n in a single chunk, whatever the trajectory

extension strategy adopted, the NFA-property will be satisfied. For instance, we can decide to extend all trajectories detected at level ε/n in the chunk $\mathscr{B}_n$ by randomly selecting their links in the chunks $\{\mathscr{B}_j\}_{1\leq j<n}$, which would produce very unrealistic trajectories although the NFA-property would remain true.

6.3.2 A pseudocode description of CUTASTRE

The implementation of CUTASTRE is quite similar to that of ASTRE, in particular, our implementation of CUTASTRE uses the algorithms 16 and 18 (`compute_`$\mathscr{G}$, `backtrack_trajectory`), which are kept unchanged, and are simply applied to the chunks $\{\mathscr{B}_i\}_{1\leq i\leq n}$ instead of the full sequence $\mathscr{F}$. The principal modification involved by CUTASTRE appears when we look for the minimal NFA in given a chunk, since we must take into account some eventual trajectory extensions. This can be done using Algorithm 20.

A simplified implementation of CUTASTRE is proposed in Algorithm 21, which is rather easy to read but suboptimal in term of computation time (it computes $\mathscr{G}$ before each trajectory detection). An accelerated implementation of CUTASTRE is finally proposed in Algorithm 22, which avoids this systematic computation of $\mathscr{G}$, using the same strategy as in Algorithm 19. Notice that the outputs of these implementations of CUTASTRE are a bit different from that of the ASTRE Algorithm 19, since instead of returning a set of trajectories, CUTASTRE adds some links between the points of the sequence $\mathscr{F}$, and returns a sequence of linked points $\mathscr{F}_{\text{linked}}$.

Algorithm 20: `minimal_NFA_chunk`, compute the min. NFA within a chunk.

Input: Two consecutive chunks $\mathscr{B}_i$ and $\mathscr{B}_{i+1}$ (with $\mathscr{B}_{i+1} = \emptyset$ when $i = n$).

Output: NFA_{min}, the minimal NFA computed within chunk $\mathscr{B}_i$ (taking into account some eventual trajectory extensions), and (k_m, x_m, y_m, ℓ_m), such as there exists a trajectory t of the type

$$t = t_m \to t' := \underbrace{\cdots \to y_m \to x_m}_{\substack{t_m \in \mathbb{T}_{\ell_m} \text{ and} \\ t_m \text{ is included in } \mathscr{B}_i}} \to \underbrace{\cdots\cdots}_{t' \notin \mathscr{B}_i}$$

(if t' is nonempty, then t_m extends t' to the chunk $\mathscr{B}_i$, and $y_m \in \mathscr{F}_0^i$, $x_m \in \mathscr{F}_1^i$) having NFA equal to NFA_{min}.

Requirements: The function $f(\ell, k_0) = \prod_{k=k_0}^{k_0+\ell-1} N_k$, where N_k denotes the number of points contained in the frame f_k.

$\mathscr{G} \leftarrow$ `compute_`$\mathscr{G}(\mathscr{B}_i)$
$\text{NFA}_{\text{min}} \leftarrow +\infty$
$K_i \leftarrow k_{\text{end}}^i - k_{\text{start}}^i + 1$ `// number of frames in` $\mathscr{B}_i$
for $k_{start}^i \le k \le k_{end}^i$ **do**
  for $x \in f_k$ **do**
    if $x \in \mathscr{F}_1^i$ *and* x *belongs to a trajectory* t *extendable to* $\mathscr{B}_i$ **then**
      $y \leftarrow$ predecessor of x in t `//` $y \in f_{k-1} (= \mathscr{F}_0^i)$
      $t_0 \leftarrow$ restriction of t to the chunk $\mathscr{B}_{i+1}$ `//` $t_0 = y \to x \to \cdots$
      for $3 \le \ell \le K_i$ **do** `// a potential extension of` t `is being tested. Compute the actual acceleration, length, and number of frames, that will be took into account to compute the NFA.`
        $a^\star \leftarrow \max\left(a^d(t_0), \mathscr{G}(x, y, \ell)\right)$
        $\ell^\star \leftarrow \ell + \ell(t_0) - 2$
        $K^\star \leftarrow$ number of frames in $\mathscr{B}_i \cup \mathscr{B}_{i+1}$
        $m \leftarrow K^\star(K^\star - \ell^\star + 1) f(\ell^\star, k - \ell + 1)(a^\star)^{\ell^\star - 2}$ `// computed NFA`
        if $m < \text{NFA}_{\text{min}}$ **then**
          $(\text{NFA}_{\text{min}}, k_m, x_m, y_m, \ell_m) \leftarrow (m, k, x, y, \ell)$
    else
      `/* the tested trajectory does not extend any other one */`
      for $y \in f_{k-1}$ *such as* y *does not belong to any previously detected trajectory* **do**
        for $3 \le \ell \le K_i$ **do**
          $m \leftarrow K^\star(K_i - \ell + 1) f(\ell, k - \ell + 1)(\mathscr{G}(x, y, \ell))^{\ell - 2}$
          if $m < \text{NFA}_{\text{min}}$ **then**
            $(\text{NFA}_{\text{min}}, k_m, x_m, y_m, \ell_m) \leftarrow (m, k, x, y, \ell)$

return $(\text{NFA}_{\text{min}}, k_m, x_m, y_m, \ell_m)$

Algorithm 21: `CUTASTRE`, simplified but suboptimal version ($\mathscr{G}$ is systematically recomputed after each trajectory extraction).

Input: The set of frames, $\mathscr{F} = \{f_1, \ldots, f_K\}$, containing the points of the sequence, ε, the maximal NFA of a trajectory to be extracted, c and o, the chunk and overlap sizes used for cutting the sequence.

Output: $\mathscr{F}_{\text{linked}}$, the same sequence as $\mathscr{F}$ where points may be linked together.

Initialization: Cut the full sequence $\mathscr{F}$ into n overlapping chunks $\mathscr{B}_1, \ldots, \mathscr{B}_n$, with overlapping areas $\mathscr{R}_{i \leftrightarrow i+1} = \mathscr{B}_i \cap \mathscr{B}_{i+1}$ using (6.13) and (6.14). By convention, we set $\mathscr{B}_{n+1} = \emptyset$ and $\mathscr{R}_{0 \leftrightarrow 1} = \emptyset$.

```
/* Process the chunks sequentially, from B_n to B_1.                    */
```
$i \leftarrow n$
while $i \geq 1$ **do**
  repeat
    $\mathscr{G} \leftarrow$ `compute_`$\mathscr{G}(\mathscr{B}_i)$
    $(\text{NFA}_{\min}, k_m, x_m, y_m, \ell_m) \leftarrow$ `minimal_NFA_chunk`$(\mathscr{B}_i, \mathscr{B}_{i+1})$
    `/* start the link supression process in` $\mathscr{R}_{i-1 \leftrightarrow i}$`.  */`
    if t_m *is fully included in* $\mathscr{R}_{i-1 \leftrightarrow i}$ **then**
      `// the points of` t_m `will have again a chance to be linked together when processing the chunk` $\mathscr{B}_{i-1}$`.`
      $t_m \leftarrow \emptyset$
    else if $i > 1$ *and* t_m *has a point in frame* $f_{k_{end}^{i-1}-1}$ **then**
      `//` t_m `is extendable from` $\mathscr{B}_i$ `to` $\mathscr{B}_{i-1}$`. Remove from` t_m `all the points in the overlapping area` $\mathscr{R}_{i-1 \leftrightarrow i}$ `excepts those in frames` $\mathscr{F}_0^{i-1}$ `(=` $f_{k_{\text{end}}^{i-1}-1}$`) and` $\mathscr{F}_1^{i-1}$ `(=` $f_{k_{\text{end}}^{i-1}}$`).`
      $t_m \leftarrow$ restriction of t_m to $\{f_k\}_{k \geq k_{\text{end}}^{i-1}-1}$
    reproduce each link of t_m in $\mathscr{F}_{\text{linked}}$ and remove all the points of t_m from $\mathscr{B}_i$
  until $m > \varepsilon/n$ *or there are no more points in the chunk* $\mathscr{B}_i$
  put back all removed points in $\mathscr{B}_i$
  $i \leftarrow i - 1$
return $\mathscr{F}_{linked}$

Algorithm 22: `CUTASTRE`, accelerated version (avoids the systematic recomputation of $\mathscr{G}$).

Input: The set of frames, $\mathscr{F} = \{f_1, \dots, f_K\}$, containing the points of the sequence, ε, the maximal NFA of a trajectory to be extracted, c and o, the chunk and overlap sizes used for cutting the sequence.

Output: $\mathscr{F}_{\text{linked}}$, the same sequence as $\mathscr{F}$ where points may be linked together.

Initialization: Cut the full sequence $\mathscr{F}$ into n overlapping chunks $\mathscr{B}_1, \dots, \mathscr{B}_n$, with overlapping areas $\mathscr{R}_{i \leftrightarrow i+1} = \mathscr{B}_i \cap \mathscr{B}_{i+1}$ using (6.13) and (6.14). By convention, we set $\mathscr{B}_{n+1} = \emptyset$ and $\mathscr{R}_{0 \leftrightarrow 1} = \emptyset$.

```
/* Process the chunks sequentially, from B_n to B_1.                    */
i ← n
while i ≥ 1 do
    B'_i ← B_i
    repeat
        G ← compute_G(B_i)
        (NFA_min, k_m, x_m, y_m, ℓ_m) ← minimal_NFA_chunk(B_i, B_{i+1})
        stop ← false
        while m ≤ ε/n and stop = false do
            t_m ← backtrack_trajectory(B_i, k_m, x_m, y_m, ℓ_m)
            t_{m_1} ← restriction of t_m to B_i \ {f_{k^i_end − 1}, f_{k^i_end}}
            if a link of t_{m_1} already exists in F_linked then
                /* a recomputation of G is needed                       */
                B_i ← B'_i
                stop ← true
            else
                /* start the link supression process in R_{i−1↔i}.      */
                if t_m is fully included in R_{i−1↔i} then
                    t_m ← ∅
                else if i > 1 and t_m has a point in frame f_{k^{i−1}_end − 1} then
                    t_m ← restriction of t_m to {f_k}_{k ≥ k^{i−1}_end − 1}
                reproduce each link of t_m in F_linked
                remove the points of t_m from B'_i
                (NFA_min, k_m, x_m, y_m, ℓ_m) ← minimal_NFA_chunk(B'_i, B_{i+1})
    until m > ε/n or there are no more points in the chunk B_i
    put back all removed points in B_i
    i ← i − 1
return F_linked
```

6.3.3 Experiments

We first compare the ASTRE and CUTASTRE algorithms on synthetic sequences produced by the Point-Set Motion Generation (PSMG) algorithm described in [Verestóy and Chetverikov 2000] (see also [Primet and Moisan 2012], Section 4.2):

— The initial position of a trajectory is chosen uniformly on the (continuous) image domain Ω;

— The initial velocity magnitude is $\nu_0 = \alpha|Z|$, where $Z \sim \mathcal{N}(\mu = 5, \sigma = 0.5)$ is a Gaussian random variable with mean μ and variance σ^2, and α is a scale factor whose setting will be detailed later. The initial velocity angle β_o is uniformly chosen on $[0, 2\pi]$.

— The velocity magnitude and angle are updated in each frame using

$$\begin{cases} \nu_{k+1} = |Z|, \quad \text{where } Z \sim \mathcal{N}(\nu_k, \alpha \cdot \sigma_\nu) \\ \beta_{k+1} \sim \mathcal{N}(\beta_k, \sigma_\beta). \end{cases}$$

— The generation ends when the trajectory reaches the last time index or when it goes outside Ω. Once the trajectory is generated, its points are quantized on a discrete grid.

In all our experiments we set $\sigma_\beta = 0.2$, $\sigma_\nu = 0.2$ or 0.5, and the frame domain Ω is quantized in 1000×1000 pixels. The setting of the other parameters (length K of the sequence, length ℓ of the trajectories) will be signaled in each experiment. The scale factor α is equal to $(\#\Omega/(100 \times 100))^{1/2}$ (that is, $\alpha = 10$ in our experiments), it can be use to change the domain quantization while maintaining the acceleration and speed characteristics of the trajectories. Last, when a trajectory does not cover the whole sequence (that is $\ell < K$), its starting frame index is chosen uniformly among $\{1, \ldots, K - \ell + 1\}$.

The detection performances are evaluated using the F_1-score criterion defined by

$$F_1\text{-score} = 2 \cdot \frac{\text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}},$$

where $1 - \text{precision}$ measures the proportion of false positive links among found links, and $1 - \text{recall}$ measures the proportion of false negative links among actual links, that is,

$$\text{precision} = \frac{\#\text{ correct links found}}{\#\text{ links found}}, \quad \text{recall} = \frac{\#\text{ correct links found}}{\#\text{ actual links}}.$$

Unless explicitly signaled, we systematically take $\varepsilon = 1$ for both algorithms.

Setting the chunk-size and overlap-size parameters

The ASTRE algorithm has the NFA threshold ε as unique parameter, which is remarkably easy to set (ε is a simple bound on the average number of detections that would be done in pure noise data, usually one chooses $\varepsilon = 1$). The CUTASTRE algorithm introduces two new parameters, which are the chunk and overlap sizes. Fortunately, the setting of these parameters appears to be quite simple, according to the experiments performed on synthetic and real-life data (see Fig. 6.3 and 6.5-left). Indeed, a standard setup like $c = 30$ (or more) and $o = c/2$ seems to lead to near-optimal performances in most situations.

Performances ASTRE versus CUTASTRE

We evaluated both algorithms on synthetic data sequences (with different characteristics, see Figure 6.4) but also on a real one (see Figure 6.5). It turns out from our experiments that ASTRE and CUTASTRE lead to similar performances when dealing with highly accelerated trajectories ($\sigma_\nu = 0.5$) and a high level of noise. Conversely, CUTASTRE achieved better detection on the smooth synthetic data set ($\sigma_\nu = 0.2$) and the snow sequence (when $\varepsilon = 1$).

Time and space complexity

The introduction of a speed threshold discussed in Section 6.2.3 can be applied to both algorithms; it decreases the execution time, but does not change the time and memory complexities, which are respectively $\mathscr{O}(N^3K^2)$ and $\mathscr{O}(N^2K^2)$ for ASTRE, and respectively $\mathscr{O}(N^3K)$ and $\mathscr{O}(N^2K)$ for CUTASTRE. Examples of practical execution time are given in Table 6.1.

Tuning the threshold parameter ε

In general, the *numbers of false alarms* of a-contrario algorithms are built using a probability upper bound (like (6.1)) that is not necessarily sharp. Furthermore, in the case of ASTRE, trajectories are greedily extracted, thus the number of trajectories extracted at level ε in any data sequence is always less than the number of ε-meaningful trajectories. As a consequence, ε is in practice a pessimistic estimation of the number of detections that really occur in pure noise data, and the user can usually obtain better detection results by increasing the NFA-threshold parameter ε, as illustrated in Figure 6.5 and 6.6.

K	N_{noise}	no speed threshold		use $\mathcal{S}_{\text{thre}} = 150$	
		ASTRE	CUTASTRE	ASTRE	CUTASTRE
200	10	30	1.4	1.2	0.09
500	10	270	3.6	11	0.26
1000	10	2160	7.6	80	0.51
3000	10	N/A	24.8	1230	1.64
5000	10	N/A	29.7	N/A	2.76
200	50	718	27	18	0.91
500	50	10^4	88	226	2.88
1000	50	N/A	158	1686	5.13
3000	50	N/A	444	N/A	15.20
5000	50	N/A	743	N/A	24.87

Table 6.1: Comparison of typical execution times on synthetic sequences ($\sigma_\nu = 0.2$) with various values of the number of frames K. Each sequence contains $K/10$ trajectories with length $\ell \in [100, 200]$ and $\mathrm{N}_{\text{noise}} \in \{10, 50\}$ spurious points per frame. We compare the execution time (in seconds) of ASTRE and CUTASTRE algorithms, with and without speed threshold (we took $\mathcal{S}_{\text{thre}} = 150$, which was three times the typical maximal speed that we could observe in the data). This experiment shows that the use of a speed threshold (even pessimistic) reduces significantly the execution time (for both algorithms), but does not break the $\mathcal{O}(K^2)$ complexity of ASTRE, which is prohibitive for long data sequences. With CUTASTRE, the execution time increases linearly with the number of frames.

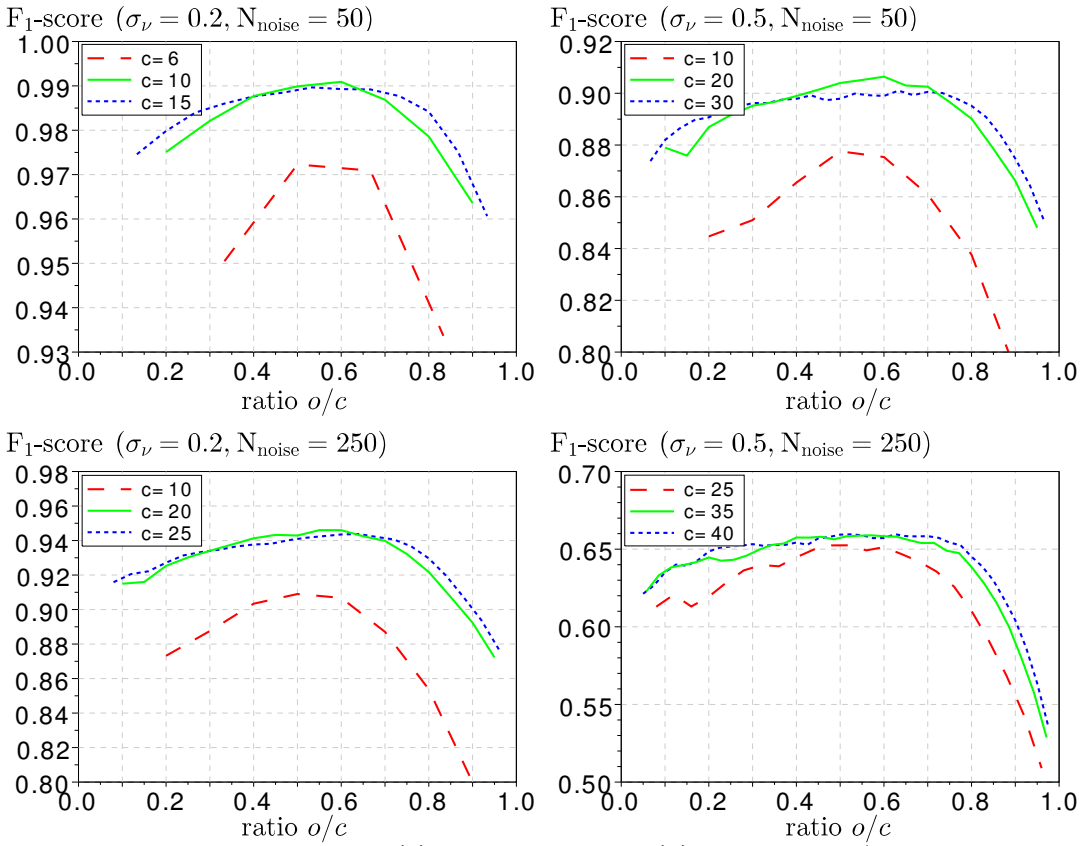


Figure 6.3: Setting chunk size (c) and overlap size (o). We compute (over 50 realizations) the average F_1-score obtained with CUTASTRE on synthetic sequences ($\sigma_\nu = 0.2$ or 0.5) of $K = 90$ frames, each containing 20 trajectories with length $\ell \geq 45$ and $N_{noise} \in \{50, 250\}$ spurious points (uniformly drawn) per frame. Several chunk sizes (c) are tested for all possible overlap sizes ($2 \leq o \leq c - 1$). We display the F_1-score as a function of the ratio o/c. As could be expected, the optimal chunk size c_{opt} increases with σ_ν and N_{noise} (the algorithm needs bigger chunks to catch the temporal coherence of the motion), but surprisingly enough it remains quite small compared to K. The performance is stable according to the choice of c as soon as c is not chosen too small. Conversely, once c is set, taking $o = \frac{c}{2}$ seems to be the optimal (or at least a reasonable) choice for the overlap size.

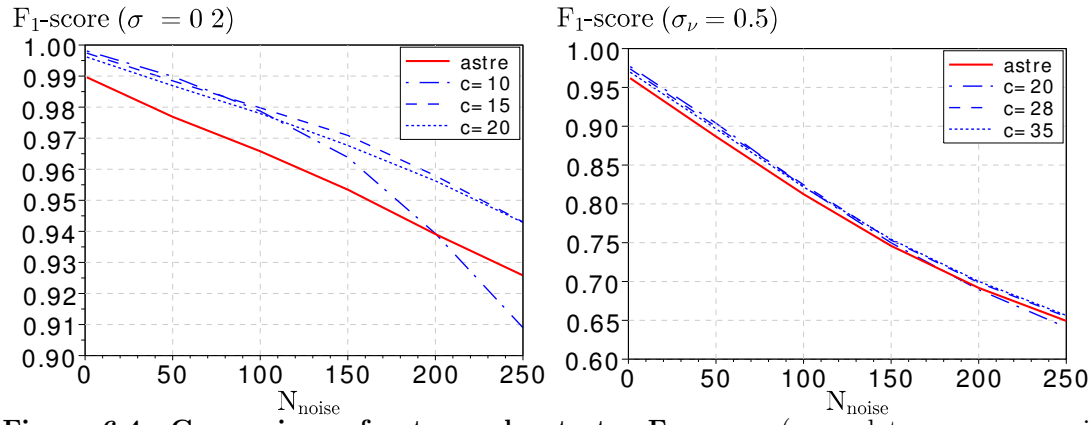


Figure 6.4: Comparison of astre and cutastre F_1-scores (same data sequences as in Fig. 6.3). CUTASTRE is tested for several (near optimal) chunk sizes and the overlap size is set to $c/2$ (integer part). We observe that better performances can be reached with CUTASTRE especially for data sequences with $\sigma_\nu = 0.2$ (smooth trajectories). When working with $\sigma_\nu = 0.5$ (trajectories with high accelerations), CUTASTRE remains slightly better but performances are very close.

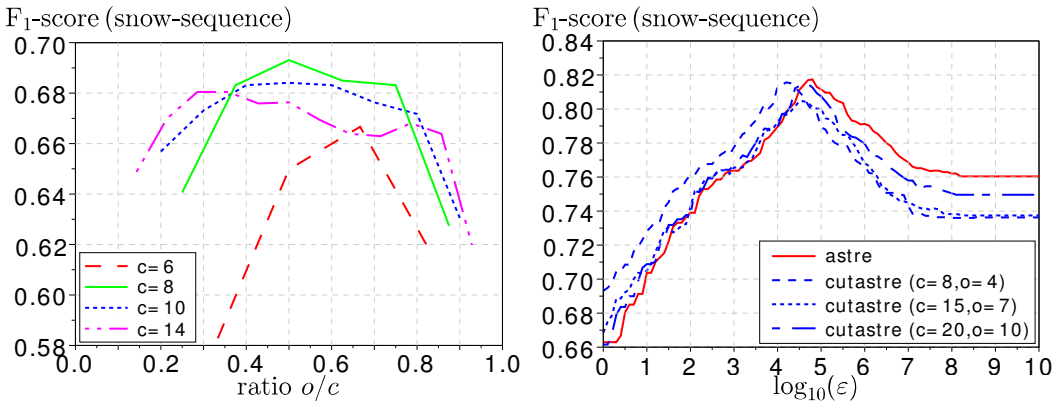


Figure 6.5: Performances evaluation on a real sequence. We evaluated the algorithms on the *snow sequence* described in [Primet and Moisan 2012] (available online at http://www.mi.parisdescartes.fr/~moisan/astre/). On the left, we reproduce the parameter exploration of Figure 6.3. We find $c_{\text{opt}} = 8$ and the performance is stable for $c \geq c_{\text{opt}}$. Also, the setting $o = c/2$ remains a good choice (often the best) once the parameter c is set. On the right, we display the evolution of the F_1-score with respect to $\log_{10}(\varepsilon)$. We can see, as in Fig. 6.4, that the two algorithms achieve similar performances (actually slightly better for CUTASTRE when the parameters c and o are optimally set).

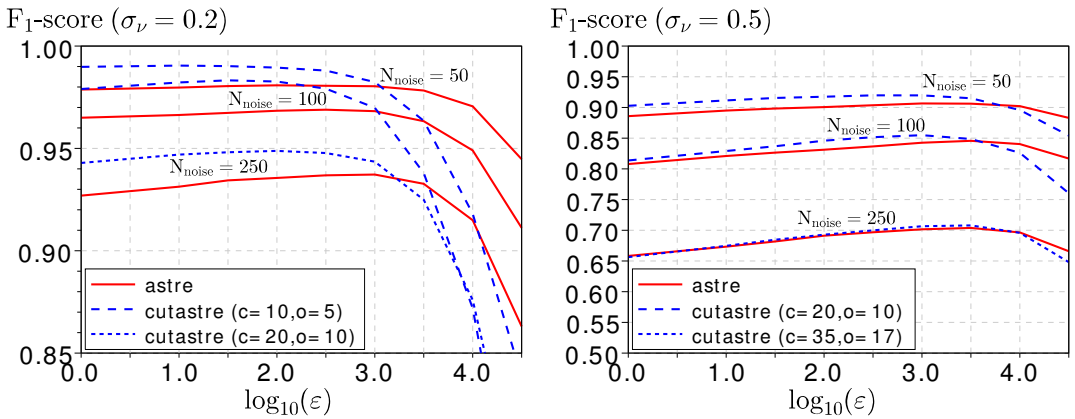


Figure 6.6: Influence of the threshold parameter ε (same data sequences as in Figure 6.3-6.4). We display the (average) F_1-score as a function of ε. Algorithm CUTASTRE is used with a near-optimal setting of parameters c and o (which revealed to be robust to ε changes). For both algorithms, the F_1-score increases with ε up to a global maximum, then it falls down. We observe as in Figure 6.4 that the performances of CUTASTRE are similar to those of ASTRE (and even slightly better for low accelerations and low noise levels).

Chapter 7

Conclusion

In this thesis, we managed to design (or make use of already known) efficient algorithms in order to provide, in a reasonable computation time, a correct approximation of the output data defined by the considered mathematical models (in most cases the output was an image, excepting in Chapters 5 and 6). In particular, we explained how the STV based optimization problems could be efficiently handled using the celebrated Chambolle-Pock algorithm [Chambolle and Pock 2011] combined with the Fast Fourier Transform algorithms [Cooley and Tukey 1965, Frigo and Johnson 2005]. We adapted to the case of Poisson noise the TV-ICE model proposed in [Louchet and Moisan 2014] that can be viewed as a fast variant of the TV-LSE model proposed in [Louchet and Moisan 2008, 2013], yielding a fixed-point numerical scheme which exhibits a linear convergence rate. We developed a fast algorithm dedicated to the accurate evaluation of a generalized incomplete gamma function. Last, we proposed CUTASTRE as a fast variant of the ASTRE algorithm of Primet [2011], which allowed a drastic reducing of the time and memory complexity, and which is already being used in [Dimiccoli et al. 2016] in a more complex fluorescent particle tracking algorithm. In this last chapter, we give a summary of the main contributions of this thesis and propose some perspectives for future works.

7.1 The Shannon total variation

In Chapter 3, we remarked that the use of TV^{d} (that is, the discretization of the TV functional using a finite differences scheme) as a regularizer for image processing problems generally leads to images that are aliased, and thus difficult to interpolate. We focused then on the Shannon total variation (STV) variant which

is inspired from the Shannon sampling theory and consists in approximating the continuous total variation of the Shannon interpolation of the discrete image using a Riemann sum. We derived some preliminary theoretical properties regarding the choice of the sampling step of this Riemann sum, however, it would be interesting as a future work to get more precise results about this choice, and more generally, to derive more mathematical properties about STV, for instance by addressing the following questions: Does STV_2 controls STV_∞? What are the differences between STV_n and $n^{-1}\,\mathrm{TV}^{\mathrm{d}} \circ Z_n$? In particular, can we provide a mathematical study of the convergence speed with respect to n of those two estimates of STV_∞, at least on a simple class of signals?

If the mathematical study of STV is still largely open, we provided many practical and useful results about its use as a regularizer. We showed how the STV based energies could be efficiently handled with modern dual algorithms, and that replacing TV^{d} by STV in the classical optimization problems does not raise any theoretical or numerical difficulties. We illustrated through many examples (denoising, deblurring, spectrum extrapolation) the improved quality in terms of sub-pixel accuracy of the images produced using this STV model. In particular, we showed that those images can be nicely interpolated (that is, without artifact) using the discrete Shannon interpolation, and are therefore well sampled according to the Shannon sampling theory.

The same approach was used to define the Huber Shannon total variation (HSTV), which can be viewed as the Huber variant of STV (in analogy with the Huber variant $\mathrm{HTV}^{\mathrm{d}}$ of TV^{d} presented in Chapter 2). We showed that the HSTV regularizer could be handled using dual algorithms, and we experimentally checked that, even if both $\mathrm{HTV}^{\mathrm{d}}$ and HSTV models produce images without staircasing artifact, only those produced using HSTV could be nicely interpolated. The ease with which we could adapt the classical duality tools in order to handle the minimization of STV or HSTV based energies leads us to believe that this approach could be generalized to many other variants of TV^{d}, such as for instance the Total Generalized Variation (TGV) introduced by Bredies, Kunisch, and Pock [2010], which involves high order derivatives (that is, derivatives with order higher than one) of the image. The STV approach could be particularly relevant to improve the TGV model since the partial derivatives of the Shannon interpolation of a discrete image can be easily and exactly computed.

Then, we moved a bit from the classical restoration models by proposing a new one which involves a data-fidelity term formulated in the frequency domain. This model makes use of a frequency weight mapping γ which can be used to control the relative importance in the minimization process of the data-fidelity term and the

regularity term with respect to the frequency position. Different choices of γ yield different applications, in particular we showed how γ could be easily set to remove the aliasing from an image, or given an image which is difficult to interpolate, could produce a visually similar image being easily interpolable. An interesting perspective would be to focus more carefully on the choice of γ according to the targeted application. For instance in the case of aliases removal, the design of the frequency mapping could be driven using an aliasing detector such as that proposed in [Coulange and Moisan 2010], in order to focus the processing on the aliased frequencies. We could also combine the aliasing removal with a spectrum extrapolation step in order to perform dealiasing, that is, in order to but back the aliases to their correct position in the spectrum of the image.

Besides, a nice improvement we could try to achieve for this weighted frequencies based model would be to figure out a way to reduce the loss of contrast in the produced image. In classical applications, the data-fidelity term is formulated in the spatial domain and mathematical studies (see for instance [Meyer 2001, Strong and Chan 2003]) point out that an important loss of contrast occurs when the data-fidelity term is taken equal to the ℓ^2 square distance to the initial image. A better contrast preservation can be achieved by replacing the ℓ^2 distance by a ℓ^1 one (see [Chan and Esedoglu 2005] and references therein). In the case of the weighted frequencies based model, the data-fidelity term, formulated as a (weighted) ℓ^2 square distance in the Fourier domain, is also responsible for loss of contrast in the produced image. Can a better contrast preservation be achieved by replacing again the ℓ^2 distance in the frequency domain by a ℓ^1 one as well? If numerical experiments could be easily done to empirically observe how much the contrast can be preserved using a ℓ^1 distance, providing some theoretical justifications could be challenging.

Last, we presented some preliminary results which indicate an excellent level of isotropy provided by the STV model. This should obviously be explored further and carefully compared to the recent advances on this subject, such as those proposed in [Chambolle et al. 2011, Condat 2016].

7.2 The Poisson TV-ICE model

In Chapter 4, we proposed a variant of the recent TV-ICE denoising model of Louchet and Moisan [2014] that we adapted for the processing of images corrupted with a Poisson noise. We provided a theoretical study of this new denoising Poisson TV-ICE model. In particular, we proved the absence of staircasing artifact for the images generated by this model, and we proved that the iterative

scheme associated to the Poisson TV-ICE model exhibits a linear convergence rate. The practical computation of the Poisson TV-ICE recursion involved some important numerical issues such as underflow or overflow errors, integral approximation, cancellation errors, etc. We explained how those numerical issues could be handled and we proposed a practical algorithm for computing the Poisson TV-ICE iterations. Then, we confirmed the two main theoretical results (the absence of staircasing for the processed images and the linear convergence rate of the numerical scheme) with some numerical experiments. Besides, the absence of staircasing and the better-quality restored images attested by experiments make Poisson TV-ICE a good alternative to Poisson TV-MAP.

Remark that the comparison between the Poisson TV-ICE and Poisson TV-LSE (the variant of the TV-LSE approach [Louchet and Moisan 2008, 2013] designed in the Gaussian case), both from a theoretical or practical viewpoint is still open: can we give statistical interpretations of the image produced by the Poisson TV-ICE model? Does it lie in the vicinity of that produced by the Poisson variant of the TV-LSE model? We have for the moment no answer to give to these questions. Another direction for future works would be to adapt this Poisson TV-ICE approach in order to handle more complex inverse problems (such as those considered in [Figueiredo and Bioucas-Dias 2010]), which would considerably enlarge the interest of this model for real applications.

From the practical viewpoint, a challenging but relevant perspective for this work would be to improve further the computation speed of the algorithm, since at the moment, one iteration of the Poisson TV-ICE scheme is about a 100 times slower than one iteration of TV-MAP implemented with the Chambolle-Pock algorithm. This is mainly due to the fact that at each iteration of the process, the update of each gray levels of the image involves the numerical computation of several integrals followed by several evaluations of logarithms and exponentials (since the evaluation of each integral is done with a mantissa-exponent representation). Instead of focusing on the speed of the algorithm dedicated to the accurate computation of those integrals (which is developed in details in Chapter 5 and that we believe to be quite fast considered the obtained level of accuracy), we can imagine some strategies consisting in implementing some less accurate but faster and still robust estimations method to perform the Poisson TV-ICE iterations (for instance which approximates directly the ratio of sums of integrals involved by this scheme). In that case, the accurate implementation that we proposed in Chapter 5 would be useful to control quality of the approximation.

7.3 Fast and accurate evaluation of the generalized incomplete gamma function

In Chapter 5, we focused on the evaluation of the generalized incomplete gamma function $I_{x,y}^{\mu,p}$ (which was needed in Chapter 4 to perform the Poisson TV-ICE iterations), and proposed a numerical procedure for its fast and accurate evaluation. We performed a careful numerical validation of this procedure for a large range of parameters, and showed that the double floating-point implementation of this procedure achieves a relative error less than 10^{-10} (in the worst case), and in general less than 10^{-13}, which is, compared to the accuracy obtained using the method of Fullerton [1972], a significant improvement. Besides, by computing the integral $I_{x,y}^{\mu,p}$ with a mantissa-exponent representation $I_{x,y}^{\mu,p} = \rho \cdot e^{\sigma}$ (with ρ and σ evaluated in double floating-point precision), this procedure greatly extends the range over which the integral can be represented, and this mantissa-exponent representation of the estimated integral was particularly useful to avoid underflow and overflow errors when computing some ratios of sums of integrals $I_{x,y}^{\mu,p}$ in Chapter 4. This work boils down to a C-language software which is available at www.math-info.univ-paris5.fr/~rabergel/softwares/deltagammainc.zip.

As future research directions, we already evoked the possibility to improve further the accuracy of this algorithm, and its extension to handle the computation of complex values of the integral. Besides, while performing our numerical experiments, we remarked that the method proposed by Pugh [2004] (that we presented in Section 5.4, and used in the algorithm we proposed) to evaluate the complete gamma function, involves a numerical parameter ($r = 10.900511$) which is not representable using standard double floating-point precision numbers (because this number has an infinite development in base 2). Consequently, although the theoretical study provided by Pugh predicts a relative accuracy less than 10^{-19}, which was confirmed by numerical experiments done in Maple (which allows computation with arbitrary precision), it happens that changing $r = 10.900511$ into its closest double floating-point number ($r \approx 10.9005109999999998$) involves a significant deterioration of the relative accuracy. Since the accurate evaluation of the gamma function is involved in many applications, it would be relevant to reconsider this choice to improve the accuracy achieved when the implementation is done in double floating-point precision, which is the commonly used precision in most programs.

7.4 The CUTASTRE Algorithm

In Chapter 6, we focused on the generic problem of the detection of smooth trajectories from a noisy point set sequence. This detection can be done optimally (according to an a contrario criterion) using the ASTRE algorithm proposed by Primet [2011]. However the ASTRE algorithm exhibits a quadratic time and space complexity with respect to the number of frame of the given input sequence (noted K), which is prohibitive for most applications. We proposed a new variant of ASTRE, called CUTASTRE, which consists in cutting the sequence into overlapping chunks of frames, and processing those chunks sequentially with an algorithm similar to ASTRE and a strategy of trajectory extension between two consecutive chunks. This variant exhibits a linear time and memory complexity (that is, $\mathscr{O}(K)$), and thus breaks the $\mathscr{O}(K^2)$ time and memory complexity of ASTRE, while showing at the same time a similar (or even slightly higher) detection performance, according to the numerical experiments that we performed on real or synthetic data. Some standalone implementations of ASTRE and CUTASTRE in C-language are available at http://www.math-info.univ-paris5.fr/~rabergel/cutastre.html.

A practical improvement that could be easily done would be to reconsider the choice made in [Primet 2011] which consists in processing the frame sequence in reverse order (that is from its last frame to its first frame) in the proposed dynamic programming algorithm. Because of this choice (that we did not changed here in order to focus more on the cutting strategy of CUTASTRE), the full sequence must be entirely loaded before ASTRE or CUTASTRE can be run. In fact, rewriting both algorithms to process the sequence in the right order (that is, starting from the first frame of the sequence) is straightforward and opens the possibility of processing the chunks of frames on the fly with CUTASTRE.

In terms of future works, this variant could be extended to handle missing points (trajectories "with holes"), as this functionality is already available with ASTRE but is still too computationally expensive for many applications, because of the $\mathscr{O}(K^5)$ complexity it exhibits. Besides, according to the nature of the data, it could be interesting to introduce some new features, for instance adding an orientation to the points of the input sequence would be very convenient for applications involving microtubules or small insects. A more challenging perspective would be to enclose the feature (position, orientation, or more generic features ...) detection step to the whole process, leading to a generic tracking algorithm that could be used to detect and track objects directly from an input sequence of images.

Bibliography

R. Abergel and L. Moisan. Accelerated a-contrario detection of smooth trajectories. In *Proceedings of the 22nd European Signal Processing Conference (EUSIPCO)*, pages 2200–2204. IEEE, 2014.

R. Abergel and L. Moisan. Fast and accurate evaluation of a generalized incomplete gamma function. Preprint MAP5, 2016.

R. Abergel, C. Louchet, L. Moisan, and T. Zeng. Total variation restoration of images corrupted by poisson noise with iterated conditional expectations. In *Proceedings of the 5th International Conference on Scale Space and Variational Methods in Computer Vision (SSVM)*, pages 178–190. Springer, 2015.

M. Abramowitz and I. A. Stegun. *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*. Number 55. Courier Corporation, 1964.

C. Aguerrebere, J. Delon, Y. Gousseau, and P. Musé. Study of the digital camera acquisition process and statistical modeling of the sensor raw data. *HAL*, 2012.

C. Akinlar and C. Topal. Edcircles: A real-time circle detector with a false detection control. *Pattern Recognition*, 46(3):725–740, 2013.

A. Aldroubi, M. Unser, and M. Eden. Cardinal spline filters: Stability and convergence to the ideal sinc interpolator. *Signal Processing*, 28(2):127–138, 1992.

A. Almansa, V. Caselles, G. Haro, and B. Rougé. Restoration and zoom of irregularly sampled, blurred, and noisy images by accurate total variation minimization with local constraints. *Multiscale Modeling & Simulation*, 5(1):235–272, 2006.

F. Alter, S. Durand, and J. Froment. Adapted total variation for artifact free decompression of JPEG images. *Journal of Mathematical Imaging and Vision*, 23(2):199–211, 2005.

K. J. Arrow, L. Hurwicz, and H. Uzawa. *Studies in linear and non-linear programming*. Stanford University Press, 1958.

J.-F. Aujol and A. Chambolle. Dual norms and image decomposition models. *International Journal of Computer Vision*, 63(1):85–104, 2005.

J.-F. Aujol and C. Dossal. Stability of over-relaxations for the forward-backward algorithm, application to FISTA. *SIAM Journal on Optimization*, 25(4):2408–2433, 2015.

J. F. Aujol, G. Aubert, L. Blanc-Féraud, and A. Chambolle. Image decomposition into a bounded variation component and an oscillating component. *Journal of Mathematical Imaging and Vision*, 22(1):71–88, 2005.

S. D. Babacan, R. Molina, and A. K. Katsaggelos. Total variation super resolution using a variational approach. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 641–644, 2008.

Y. Bar-Shalom. On hierarchical tracking for the real world. *IEEE Transactions on Aerospace and Electronic Systems*, 42(3):846–850, 2006.

Y. Bar-Shalom, T. Fortmann, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, 8(3):173–184, 1983.

A. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Transactions on Image Processing*, 18(11):2419–2434, 2009a.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009b.

R. Bellman. The theory of dynamic programming. *Bulletin of the American Mathematical Society*, 60(6):503–515, 1954.

J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(9):1806–1819, 2011.

G. P. Bhattacharjee. Algorithm AS 32: The incomplete gamma integral. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 19(3):285–287, 1970.

G. Blanchet and L. Moisan. An explicit sharpness index related to global phase coherence. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1065–1068. IEEE, 2012.

M. Bleicher and P. Nicolini. Large extra dimensions and small black holes at the lhc. In *Journal of Physics: Conference Series*, volume 237, page 012008. IOP Publishing, 2010.

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

K. Bredies, K. Kunisch, and T. Pock. Total generalized variation. *SIAM Journal on Imaging Sciences*, 3(3):492–526, 2010.

T. Briand and J. Vacher. Linear filtering : From the continuous spectral definition to the numerical computations. IPOL preprint, 2015.

A. Buades, B. Coll, and J.-M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2):490–530, 2005.

C. J. Cannon and I. M. Vardavas. The effect of redistribution on the emission peaks from chromospheric-type stellar atmospheres. *Astronomy and Astrophysics*, 32:85, 1974.

V. Caselles, A. Chambolle, and M. Novaga. Total variation in imaging. *Handbook of Mathematical Methods in Imaging*, pages 1455–1499, 2015.

A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1-2):89–97, 2004.

A. Chambolle. Total variation minimization and a class of binary mrf models. In *Energy minimization methods in computer vision and pattern recognition*, pages 136–152. Springer, 2005.

A. Chambolle and C. Dossal. On the convergence of the iterates of the "fast iterative shrinkage/thresholding algorithm". *Journal of Optimization Theory and Applications*, 166(3):968–982, 2015.

A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

A. Chambolle, V. Caselles, D. Cremers, M. Novaga, and T. Pock. An introduction to total variation for image analysis. *Theoretical foundations and numerical methods for sparse recovery*, 9:263–340, 2010.

A. Chambolle, S. E. Levine, and B. J. Lucier. An upwind finite-difference method for total variation-based image smoothing. *SIAM Journal on Imaging Sciences*, 4(1):277–299, 2011.

A. Chambolle, V. Duval, G. Peyré, and C. Poon. Geometric properties of solutions to the total variation denoising problem. Preprint arXiv, 2016.

T. F. Chan and S. Esedoglu. Aspects of total variation regularized L^1 function approximation. *SIAM Journal on Applied Mathematics*, 65(5):1817–1837, 2005.

T. F. Chan and C.-K. Wong. Total variation blind deconvolution. *IEEE transactions on Image Processing*, 7(3):370–375, 1998.

T. F. Chan, A. Marquina, and P. Mulet. High-order total variation-based image restoration. *SIAM Journal on Scientific Computing*, 22(2):503–516, 2000.

T. F. Chan, A. M. Yip, and F. E. Park. Simultaneous total variation image inpainting and blind deconvolution. *International Journal of Imaging Systems and Technology*, 15(1):92–102, 2005.

B. W. Char. On Stieltjes's continued fraction for the gamma function. *Mathematics of Computation*, 34(150):547–551, 1980.

M. A. Chaudhry and S. M. Zubair. *On a class of incomplete gamma functions with applications*. CRC Press, 2001.

D. Chetverikov and J. Verestoy. Feature point tracking for incomplete trajectories. *Computing*, 62:321–338, 1999.

R. T. Collins. Multitarget data association with higher-order motion models. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1744–1751, 2012.

P. L. Combettes. Iterative construction of the resolvent of a sum of maximal monotone operators. *Journal of Convex Analysis*, 16(4):727–748, 2009.

P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011.

P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.

L. Condat. Discrete total variation: New definition and minimization. Preprint GIPSA-lab, 2016.

J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

B. Coulange and L. Moisan. An aliasing detection algorithm based on suspicious colocalizations of fourier coefficients. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 2013–2016, 2010.

I. Csiszar. Why least squares and maximum entropy? an axiomatic approach to inference for linear inverse problems. *The annals of statistics*, pages 2032–2066, 1991.

A. Cuyt, F. Backeljauw, and C. Bonan-Hamada. *Handbook of continued fractions for special functions*. Springer Science & Business Media, 2008.

K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007.

J. Darbon and M. 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.

I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on pure and applied mathematics*, 57(11):1413–1457, 2004.

C.-A. Deledalle, F. Tupin, and L. Denis. Poisson NL means: Unsupervised non local means for poisson noise. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 801–804, 2010.

A. Desolneux. When the a contrario approach becomes generative. *International Journal of Computer Vision*, 116(1):46–65, 2016.

A. Desolneux and F. Doré. An anisotropic a contrario framework for the detection of convergences in images. *Journal of Mathematical Imaging and Vision*, 56(1): 32–56, 2016.

A. Desolneux, L. Moisan, and J.-M. Morel. Meaningful alignments. *International Journal of Computer Vision*, 40(1):7–23, 2000.

A. Desolneux, L. Moisan, and J.-M. Morel. Edge detection by Helmholtz principle. *Journal of Mathematical Imaging and Vision*, 14(3):271–284, 2001.

A. Desolneux, L. Moisan, and J.-M. Morel. Maximal meaningful events and applications to image analysis. *Annals of Statistics*, pages 1822–1851, 2003.

A. Desolneux, L. Moisan, and J.-M. Morel. *From Gestalt Theory to Image Analysis. A Probabilistic Approach.* Springer-Verlag, collection "Interdisciplinary Applied Mathematics", 2008.

M. Dimiccoli, J.-P. Jacob, and L. Moisan. Particle detection and tracking in fluorescence time-lapse imaging: a contrario approach. *Machine Vision and Applications*, 27(4):511–527, 2016.

DLMF. NIST Digital Library of Mathematical Functions. URL http://dlmf.nist.gov/, Release 1.0.10, 2015. Online companion to [Olver et al. 2010].

F. Doré. *Convergences de structures linéaires dans les images: modélisation stochastique et applications en imagerie médicale.* PhD thesis, Université René Descartes-Paris V, 2014.

J. Douglas and H. H. Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American mathematical Society*, 82(2):421–439, 1956.

Y. Drori, S. Sabach, and M. Teboulle. A simple algorithm for a class of nonsmooth convex-concave saddle-point problems. *Operations Research Letters*, 43(2):209–214, 2015.

J. Eckstein and D. P Bertsekas. On the douglas-rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1-3):293–318, 1992.

I. Ekeland and R. Témam. *Convex Analysis and Variational Problems*, volume 28. Society for Industrial and Applied Mathematics (SIAM), 1999.

G. Facciolo, A. Almansa, J.-F. Aujol, and V. Caselles. Irregular to regular sampling, denoising and deconvolution. *Multiscale Modeling & Simulation*, 7(4): 1574–1608, 2009.

J. M. Fadili and G. Peyré. Total variation projection with first order schemes. *Transactions on Image Processing*, 20(3):657–669, 2011.

M. A. T. Figueiredo and J. M. Bioucas-Dias. Restoration of poissonian images using alternating direction optimization. *Transactions on Image Processing*, 19 (12):3133–3145, 2010.

F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multi-Camera People Tracking with a Probabilistic Occupancy Map. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(2):267–282, February 2008.

M. Frigo and S. G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. Special issue on "Program Generation, Optimization, and Platform Adaptation".

W. Fullerton. Algorithm 435: Modified incomplete gamma function [S14]. *Communications of the ACM*, 15(11):993–995, November 1972. ISSN 0001-0782. doi: 10.1145/355606.361891.

D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976.

W. Gautschi. A computational procedure for incomplete gamma functions. *ACM Transactions On Mathematical Software*, 5(4):466–481, December 1979. ISSN 0098-3500.

W. Gautschi. The incomplete gamma functions since tricomi. In *Tricomi's Ideas and Contemporary Applied Mathematics, Atti dei Convegni Lincei, Accademia Nazionale dei Lincei*, volume 147, pages 203–237, 1998.

P. Getreuer. Linear methods for image interpolation. *Image Processing On Line*, 1, 2011.

G. Gilboa. A spectral approach to total variation. In *Proceedings of the 4th International Conference on Scale Space and Variational Methods in Computer Vision (SSVM)*, volume 7893, pages 36–47, 2013.

R. Glowinski and P. Le Tallec. *Augmented Lagrangian and operator-splitting methods in nonlinear mechanics*, volume 9. SIAM, 1989.

Y. Gousseau and J.-M. Morel. Are natural images of bounded variation? *SIAM Journal on Mathematical Analysis*, 33(3):634–648, 2001.

B. Grosjean and L. Moisan. A-contrario detectability of spots in textured backgrounds. *Journal of Mathematical Imaging and Vision*, 33:3:313–337, 2009.

F. Guichard and F. Malgouyres. Total variation based interpolation. In *Proceedings of the 9th European Signal Processing Conference (EUSIPCO)*, volume 3, pages 1741–1744, 1998.

I. I. Guseinov and B. A. Mamedov. Evaluation of Incomplete Gamma Functions Using Downward Recursion and Analytical Relations. *Journal of Mathematical Chemistry*, 36(4):341–346, August 2004.

J. G. Hills. Effect of binary stars on the dynamical evolution of stellar clusters. II-analytic evolutionary models. *The Astronomical Journal*, 80:1075–1080, 1975.

P. J. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

P. J. Huber. Robust regression: asymptotics, conjectures and Monte Carlo. *The Annals of Statistics*, pages 799–821, 1973.

W. B. Jones and W. J. Thron. *Continued fractions: analytic theory and applications*. Number 11 in Encyclopedia of mathematics and its applications. Addison-Wesley Pub. Co, 1980. ISBN 978-0-201-13510-7.

R. Kannan and C. K. Krueger. *Advanced analysis: on the real line*. Springer Science & Business Media, 2012.

Z. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(11):1805–1819, 2005.

L. Kissel, R. H. Pratt, and S. C. Roy. Rayleigh scattering by neutral atoms, 100 eV to 10 MeV. *Physical Review A*, 22:1970–2004, Nov 1980. doi: 10.1103/PhysRevA.22.1970.

M.-J. Lai, B. Lucier, and J. Wang. The convergence of a central-difference discretization of Rudin-Osher-Fatemi model for image denoising. In *Proceedings of the 2nd International Conference on Scale Space and Variational Methods in Computer Vision (SSVM)*, pages 514–526. Springer, 2009.

C. Lanczos. A precision approximation of the gamma function. *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, 1(1):86–96, 1964.

A. Leclaire and L. Moisan. No-reference image quality assessment and blind deblurring with sharpness metrics exploiting fourier phase information. *Journal of Mathematical Imaging and Vision*, 52(1):145–172, 2015.

W. J. Lentz. Generating bessel functions in mie scattering calculations using continued fractions. *Applied Optics*, 15(3):668–671, 1976.

V. Linetsky. Pricing equity derivatives subject to bankruptcy. *Mathematical finance*, 16(2):255–282, 2006.

C. Louchet and L. Moisan. Total variation denoising using posterior expectation. In *Proceedings of the 16th European Signal Processing Conference (EUSIPCO)*, pages 1–5. IEEE, 2008.

C. Louchet and L. Moisan. Posterior expectation of the total variation model: properties and experiments. *SIAM Journal on Imaging Sciences*, 6(4):2640–2684, 2013.

C. Louchet and L. Moisan. Total variation denoising using iterated conditional expectation. In *Proceedings of the 22nd European Signal Processing Conference (EUSIPCO)*, pages 1592–1596. IEEE, 2014.

D. G. Lowe. *Perceptual Organization and Visual Recognition*, volume 5 of *The Kluwer International Series in Engineering and Computer Science*. Springer US, 1985.

D. G Luenberger and Y. Ye. *Linear and nonlinear programming*, volume 2. Springer, 1984.

F. Malgouyres and F. Guichard. Edge direction preserving image zooming: a mathematical and numerical analysis. *SIAM Journal on Numerical Analysis*, 39(1):1–37, 2001.

S. Masnou and J.-M. Morel. Level lines based disocclusion. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 259–263, 1998.

W. Metzger. *Gesetze des sehens*. Kramer, 1975.

Y. Meyer. *Oscillating patterns in image processing and nonlinear evolution equations: the fifteenth Dean Jacqueline B. Lewis memorial lectures*, volume 22. American Mathematical Society, 2001.

W. Miled, J. Pesquet, and M. Parent. A convex optimization approach for depth estimation under illumination variation. *IEEE Transactions on Image Processing*, 18(4):813–830, 2009.

L. Moisan. How to discretize the total variation of an image? In *the 6th International Congress on Industrial Applied Mathematics, Proceedings in Applied Mathematics and Mechanics*, volume 7(1), pages 1041907–1041908, 2007.

L. Moisan. Periodic plus smooth image decomposition. *Journal of Mathematical Imaging and Vision*, 39(2):161–179, 2011.

J.-J. Moreau. Inf-convolution des fonctions numériques sur un espace vectoriel. *Comptes Rendus de l'Academie des Sciences de Paris*, 256:125–129, 1963.

J.-J. Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société mathématique de France*, 93:273–299, 1965.

Y. Moreno, R. Pastor-Satorras, and A. Vespignani. Epidemic outbreaks in complex heterogeneous networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 26(4):521–529, 2002.

Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983.

Y. Nesterov. *Introductory Lectures on Convex Optimization*, volume 87 of *Applied Optimization*. Springer US, 2004.

Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez. Video inpainting of complex scenes. *SIAM Journal on Imaging Sciences*, 7(4):1993–2019, 2014.

M. Nikolova. Local strong homogeneity of a regularized estimator. *SIAM Journal on Applied Mathematics*, 61(2):633–658, 2000.

M. Nikolova. Model distortions in bayesian map reconstruction. *Inverse Problems and Imaging*, 1(2):399, 2007.

P. Ochs, Y. Chen, T. Brox, and T. Pock. iPiano: inertial proximal algorithm for nonconvex optimization. *SIAM Journal on Imaging Sciences*, 7(2):1388–1419, 2014.

F. W. J. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark, editors. *NIST Handbook of Mathematical Functions*. Cambridge University Press, New York, NY, 2010. Print companion to [DLMF].

N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):123–231, 2013.

T. Pock and A. Chambolle. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1762–1769, 2011.

J. Preciozzi, P. Musé, A. Almansa, S. Durand, A. Khazaal, and B. Rougé. SMOS images restoration from L1A data: A sparsity-based variational approach. In *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 2487–2490, 2014.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes in C: the art of scientific computing*. Cambridge University Press, 2nd edition, 1992.

M. Primet. *Probabilistic methods for point tracking and biological image analysis*. PhD thesis, MAP5, 2011.

M. Primet and L. Moisan. Point tracking: an a-contrario approach. 2012.

G. R. Pugh. *An analysis of the Lanczos gamma approximation*. PhD thesis, University of British Columbia, 2004.

J. Rabin, J. Delon, and Y. Gousseau. A statistical approach to the matching of local features. *SIAM Journal on Imaging Sciences*, 2(3):931–958, 2009.

H. Raguet, J. M. Fadili, and G. Peyré. A generalized forward-backward splitting. *SIAM Journal on Imaging Sciences*, 6(3):1199–1226, 2013.

K. Rangarajan and M. Shah. Establishing motion correspondence. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 103–108, 1991.

D. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979.

W. Ring. Structural properties of solutions to total variation regularization problems. *ESAIM: Modélisation Mathématique et Analyse Numérique*, 34(4):799–810, 2000.

A. Robin, G. Mercier, G. Moser, and S. Serpico. An a-contrario approach for unsupervised change detection in radar images. In *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, volume 4, pages 240–243, 2009.

A. Robin, L. Moisan, and S. Le Hégarat-Mascle. An a-contrario approach for sub-pixel change detection in satellite imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(11):1977–1993, 2010.

R. T. Rockafellar. Convex analysis (Princeton mathematical series). *Princeton University Press*, 46:49, 1970.

R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, 14(5):877–898, 1976.

R. T. Rockafellar and R. Wets. *Variational Analysis*, volume 317. Springer: Grundlehren der Math. Wissenschaften., 1998.

B. Rougé and A. Seghier. Nonlinear spectral extrapolation: new results and their application to spatial and medical imaging. In *Proceedings of the SPIE's International Symposium on Optical Science, Engineering, and Instrumentation*, pages 279–289. International Society for Optics and Photonics, 1995.

D. L. Ruderman. The statistics of natural images. *Network: computation in neural systems*, 5(4):517–548, 1994.

L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.

K. D. Schmidt. *On the covariance of monotone functions of a random variable*. Professoren des Inst. für Math. Stochastik, 2003.

A. Y. Schoene. Remark on "algorithm 435: Modified incomplete gamma function [S14]". *ACM Trans. Math. Softw.*, 4(3):296–304, September 1978. ISSN 0098-3500. doi: 10.1145/355791.355803.

S. Setzer, G. Steidl, and T. Teuber. Deblurring poissonian images by split Bregman techniques. *Journal of Visual Communication and Image Representation*, 21(3):193–199, 2010.

K. Shafique and M. Shah. A non-iterative greedy algorithm for multi-frame point correspondence. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 110–115, 2003.

L. Simon and J.-M. Morel. Influence of unknown exterior samples on interpolated values for band-limited images. *SIAM Journal on Imaging Sciences*, 9(1):152–184, 2016.

D. Strong and T. F. Chan. Edge-preserving and scale-dependent properties of total variation regularization. *Inverse problems*, 19(6):S165–S187, 2003.

P. Thévenaz, T. Blu, and M. Unser. Interpolation revisited [medical images application]. *IEEE Transactions on Medical Imaging*, 19(7):739–758, 2000.

I. Thompson. Algorithm 926: Incomplete gamma functions with negative arguments. *ACM Transactions On Mathematical Software*, 39(2):14:1–14:9, February 2013. ISSN 0098-3500. doi: 10.1145/2427023.2427031.

I. J. Thompson and A. R. Barnett. Coulomb and bessel functions of complex arguments and order. *Journal of Computational Physics*, 64(2):490–509, 1986.

F. G. Tricomi. Sulla funzione gamma incompleta. *Annali di Matematica Pura ed Applicata*, 31(1):263–279, 1950.

M. Unser. Ten good reasons for using spline wavelets. In *Optical Science, Engineering and Instrumentation'97*, pages 422–431. International Society for Optics and Photonics, 1997.

M. Unser. Sampling-50 years after shannon. *Proceedings of the IEEE*, 88(4): 569–587, 2000.

M. Unser, A. Aldroubi, and M. Eden. Fast b-spline transforms for continuous image representation and interpolation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, (3):277–285, 1991.

C. J. Veenman, M. J. T. Reinders, and E. Backer. Resolving motion correspondence for densely moving points. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(1):54–72, 2001.

C. J. Veenman, M. J. T. Reinders, and E. Backer. Motion tracking as a constrained optimization problem. *Pattern Recognition*, 36(9):2049–2067, 2003.

J. Verestóy and D. Chetverikov. Experimental comparative evaluation of feature point tracking algorithms. In *Performance Characterization in Computer Vision*, pages 167–178. Springer, 2000.

L. A Vese and S. J. Osher. Modeling textures with total variation minimization and oscillating patterns in image processing. *Journal of scientific computing*, 19(1-3):553–572, 2003.

L. A. Vese and S. J. Osher. Image denoising and decomposition with total variation minimization and oscillatory functions. *Journal of Mathematical Imaging and Vision*, 20(1):7–18, 2004.

C. R. Vogel and M. E. Oman. Fast, robust total variation-based reconstruction of noisy, blurred images. *IEEE Transactions on Image Processing*, 7(6):813–824, 1998.

R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. Lsd: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, (4):722–732, 2008a.

R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. On straight line segment detection. *Journal of Mathematical Imaging and Vision*, 32(3):313–347, 2008b.

J. Wang and B. J. Lucier. Error bounds for finite-difference methods for Rudin-Osher-Fatemi image smoothing. *SIAM Journal on Numerical Analysis*, 49(2): 845–868, 2011.

P. Weiss. *Algorithmes rapides d'optimisation convexe. Applications à la reconstruction d'images et à la détection de changements.* PhD thesis, Université Nice Sophia Antipolis, 2008.

P. Weiss and L. Blanc-Féraud. A proximal method for inverse problems in image processing. In *Proceedings of the 17th European Signal Processing Conference (EUSIPCO)*, pages 1374–1378. IEEE, 2009.

P. Weiss, L. Blanc-Féraud, and G. Aubert. Efficient schemes for total variation minimization under constraints in image processing. *SIAM journal on Scientific Computing*, 31(3):2047–2080, 2009.

M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic Huber-L1 Optical Flow. In *Proceedings of the British Machine Vision Conference*, volume 1, page 3, 2009.

M. Wertheimer. Untersuchungen zur lehre von der gestalt. II. *Psychological Research*, 4(1):301–350, 1923.

S. Winitzki. Computing the incomplete gamma function to arbitrary precision. In *Proceedings of the International Conference on Computational Science and Its Applications: Part I*, ICCSA'03, pages 790–798. Springer-Verlag, 2003. ISBN 3-540-40155-5.

Wolfram Research Inc. Generalized incomplete gamma function, 1988. URL http://reference.wolfram.com/language/ref/Gamma.html. (documentation page).

Wolfram Research Inc. Generalized incomplete gamma function, 1998. URL http://functions.wolfram.com/webMathematica/FunctionEvaluation.jsp?name=Gamma3. (online evaluation page).

G.-S. Xia, J. Delon, and Y. Gousseau. Accurate junction detection and characterization in natural images. *International journal of computer vision*, 106(1): 31–56, 2014.

L. P. Yaroslavsky. Signal sinc-interpolation: a fast computer algorithm. *Bioimaging*, 4(4):225–231, 1996.

K. Yosida. *Functional Analysis*. Springer Berlin Heidelberg, 1980. Originally published as volume 123 in the series: Grundlehren der mathematischen Wissenschaften, 1968.

M. Zhu and T. Chan. An efficient primal-dual hybrid gradient algorithm for total variation image restoration. UCLA CAM Report, 2008.

W. P. Ziemer. *Weakly differentiable functions: Sobolev spaces and functions of bounded variation*, volume 120. Springer Science & Business Media, 2012.