Variational methods for PDE-based image and video compression

Thomas Jacumin

Institut de Recherche en Informatique, Mathématiques, Automatique et Signal
Université de Haute–Alsace
18 rue des frères Lumière
68093 Mulhouse

Thèse de Doctorat de l'Université de Haute-Alsace

École Doctorale de Mathématiques, Sciences de l'Information et de l'Ingénieur

Spécialité

Mathématiques Appliquées

Présentée par

Thomas Jacumin

Pour obtenir le grade de

Docteur de l'Université de Haute–Alsace

Techniques Variationnelles pour l'Analyse d'Images et la Compression Vidéo.

Soutenue le 15 décembre 2022 *devant le jury composé de :*

Rapporteur :	Faker Ben Belgacem	*Professeur,*	*Université de Technologie de Compiègne*
Rapporteur :	Frédéric Hecht	*Professeur,*	*Université de la Sorbonne*
Examinateur :	Nicolas Juillet	*Professeur,*	*Université de Haute–Alsace*
Examinateur :	Andreas Langer	*Professeur,*	*Université de Lund (Suède)*
Examinateur :	Yannick Privat	*Professeur,*	*Université de Strasbourg*
Directeur de thèse :	Zakaria Belhachmi	*Professeur,*	*Université de Haute–Alsace*

Résumé

Dans cette thèse, nous allons proposer des modèles et des méthodes mathématiques basées sur les équations aux dérivées partielles permettant de compresser des images ainsi que des vidéos. Ces nouveaux algorithmes de compression seront également capables de prendre en compte d'éventuelles détériorations de l'image dues aux capteurs d'acquisitions, au stockage ou à la transmission de celles-ci. Ainsi, le but serait de pouvoir enlever une partie de ces détériorations de l'image (bruit) via l'action de la compression plutôt que d'effectuer une étape de pré-filtrage sur l'image à compresser. Plus précisément, dans la première partie, nous allons proposer un critère mathématique permettant de localiser les meilleurs pixels à garder au sein d'une image dans le cas où la méthode d'inpainting, permettant la reconstruction de l'image, est l'équation de la chaleur (avec les conditions aux bords adéquates). Pour cela, nous écrirons le problème de compression comme un problème d'optimisation de forme, dont le coût sera une "erreur" entre l'image d'origine et l'image décompressée/reconstruite. L'existence d'une solution à ce problème sera démontrée dans le cadre de la Γ-convergence. Dans un premier temps, nous nous restreindrons au coût L^2, important pour le débruitage, ainsi qu'à la première itération d'un schéma explicite en temps de l'équation de la chaleur. Nous pourrons expliciter un critère de sélection au moyen du gradient topologique et un critère relâché en imposant qu'un ensemble admissible soit une réunion de "fat pixels" (boules dont on fait tendre le rayon vers 0). Nous étendrons ce critère au cas d'une procédure itérative de sélection des pixels en considérant cette fois-ci une itération quelconque de l'inpainting, améliorant ainsi les performances de la méthode. Dans un second temps, nous étudierons le cas d'un coût plus général, dont un critère sera donné par la méthode de l'adjoint. Des résultats numériques seront alors présentés dans le cas du coût L^2 et du coût L^1, également important pour la suppression du bruit dans une image. Dans la seconde partie, nous nous intéresserons aux cas de la compression vidéo en étudiant le mouvement existant entre deux images consécutives. Cette estimation du mouvement sera faite grâce au flot optique pour de petits déplacements. Nous implémenterons quelques méthodes de calcul du flot optique dans le but de pouvoir les utiliser pour la compression de vidéos dans la partie suivante. Ensuite, nous proposerons un nouveau modèle de flot optique utilisant le transport optimal et la formule de Benamou-Brenier. Enfin, dans la troisième partie, nous implémenterons un codec vidéo complet et nous le comparerons avec les codecs existants dans la vie courante. Plus particulièrement, nous comparerons la taille effective de la vidéo compressée ainsi que la qualité de la vidéo décompressée. Pour finir, nous proposerons du calcul sur GPU afin d'accélérer la résolution numérique des équations.

Mots clés Compression vidéo – Optimisation de formes – Inpainting d'image – Flot optique – Γ-convergence – Transport optimal – Calcul numérique – GPU

Abstract

In this thesis, we will propose mathematical models and methods based on partial differential equations to compress images and videos. These new compression algorithms will also be able to take into account possible deteriorations of the image due to the acquisition sensors, storage or transmission. Thus, the aim would be to be able to remove part of these image deteriorations (noise), via the action of compression rather than carrying out a pre-filtering stage on the image to be compressed. More precisely, in the first part, we will propose a mathematical criterion allowing us to locate the best pixels to keep within an image in the case where the inpainting is the heat equation (with the appropriate boundary conditions). To do this, we will write the compression problem as a shape optimization problem, where the cost function will be an "error" between the original image and the decompressed image. The existence of a solution to this problem will be proved in the context of Γ-convergence. Initially, we will restrict ourselves to the cost L^2, which is important for denoising, and to the first iteration of a time-explicit scheme of the heat equation. We will be able to make explicit a selection criterion by means of the topological gradient and a relaxed criterion by imposing that an admissible set is a union of "fat pixels" (balls whose radius is made to tend to 0). We will extend this criterion to the case of an iterative pixel selection procedure, this time considering any iteration of the inpainting, and thus improving the performance of the method. In a second step, we will study the case of a more general cost, a criterion of which will be given by the adjoint method. Numerical results will then be presented in the case of the L^2 cost and the L^1 cost, which is also important for noise removal in an image. In the second part, we will focus on the case of video compression by studying the motion existing between two consecutive images. This motion estimation will be done using the optical flow for small displacements. We will implement some methods for computing the optical flow in order to be able to use them for video compression in the following part. Then, we will propose a new optical flow model using the optimal transport and the Benamou-Brenier formula. Then, in the third part, we will implement a complete video codec and compare it with existing codecs in real life. In particular, we will compare the effective size of the compressed video, as well as the quality of the decompressed video. Finally, we will propose GPU implementation of our algorithms in order to speed up the computation time.

Keywords Video compression – Shape optimization – Image Inpainting – Optical flow – Γ-convergence – Optimal transport – Numerical computation – GPU

Remerciements

En premier lieu, je remercie le Professeur Zakaria Belhachmi, mon directeur de thèse au laboratoire IRIMAS de l'Université de Haute-Alsace, pour m'avoir prodigué des conseils et remarques constructives tout au long de mes trois années de thèse malgré une période difficile due aux conditions sanitaires liées au COVID-19.

Je tiens également à remercier les membres du jury : le Professeur Faker Ben Belgacem de l'université de Technologie de Compiègne, le Professeur Nicolas Juillet de l'université de Haute–Alsace, le Professeur Andreas Langer de l'université de Lund et plus particulièrement le Professeur Frédéric Hecht du laboratoire Jacques-Louis Lions de l'Université de la Sorbonne ainsi que le Professeur Yannick Privat du laboratoire IRMA de l'Université de Strasbourg pour avoir également accepté de faire partie de mon comité de suivi de thèse.

J'associe à ces remerciements toutes les personnes présentes au département de mathématiques de l'IRIMAS ainsi que son directeur, le Professeur Abdenacer Makhlouf, pour leur accueil. Je remercie le Professeur Nicolas Juillet pour son dynamisme au sein du laboratoire ainsi que pour sa mise en place du groupe de travail sur le transport optimal au cours duquel j'ai pu découvrir cette branche des mathématiques et m'entraîner à préparer des exposés. Je remercie le Professeur Augustin Fruchard pour ses problèmes mathématiques intéressants mais rarement résolus. Je tiens tout particulièrement à saluer la secrétaire du département Mme Viviane Kuhn pour sa bonne humeur, sa disponibilité et son soutien logistique. Un grand merci au Professeur Abdenacer Makhlouf et à Michel Masiano pour nous avoir aidé, mon groupe de musique et moi, à organiser la fête de la musique de 2022 à l'Université.

Merci également à l'ensemble des doctorants, Anissa, Leila, Rabeb, Rahma, Imène, Otávio, Salih, Quentin, Ícaro, Hamilton, Armand, Andrea, Atef et Mohamed, pour tous les bons moments passés ensemble. Plus particulièrement, je remercie Salih pour nos sorties nocturnes à Paris ainsi que pour ses conseils concernant les post docs, Otávio pour nos duo de guitares, Hamilton et Ícaro pour m'avoir encouragé (et accompagné) à aller à la salle de sport. J'ai une pensée toute particulière pour mes camarades Quentin et Armand, qui m'ont fait découvrir et apprécier l'Alsace, avec notamment des randonnées dans les massifs Vosgiens et des repas à la ferme auberge du Baerenbach. J'ai une pensée toute particulièrement reconnaissante pour leur relecture enrichissante et leurs conseils pour l'oral des différents exposés que j'ai été amené à faire.

Ces remerciements ne peuvent s'achever sans une pensée pour ma famille et plus particulièrement pour ma première fan (et correctrice des fautes d'orthographes de cette thèse !) : ma mère. Leurs encouragements et leur soutien sans faille sont pour moi les piliers fondateurs de ce que je suis et de ce que je fais.

Pour terminer, je remercie toutes les personnes que je n'ai pas nommées, mais qui ont participé à cette aventure.

Contents

List of Figures

List of Tables

List of Algorithms

Introduction générale et résumé

Contents

Avec le déploiement massif de l'internet haut-débit partout dans le monde au cours de ces dernières années, les services de streaming de musiques, de vidéos et plus récemment de jeux vidéos ont connu un véritable essor. Ainsi, de nouvelles problématiques ont émergé concernant l'efficacité des méthodes de compression des flux de données, comme par exemple le taux de compression pour optimiser la consommation de données, la qualité visuelle et auditive, le temps de calcul pour diminuer le temps de latence ou encore la consommation énergétique. Le but de la compression est de minimiser la taille en octets d'une image/vidéo sans réduire la qualité de l'image à un niveau inacceptable. Comme les données d'image et de vidéo occupent une grande partie de la bande passante pendant la transmission, elles doivent être compressées sans perte ou avec une faible perte de qualité. La compression d'une vidéo est sensiblement différente de la compression de données binaires brutes puisque les données contenues dans une image ou dans une vidéo sont fortement corrélées. Ces redondances sont de trois types : les redondances à l'intérieur d'une image de la vidéo, appelées redondances spatiales, les redondances entre les images, appelées redondances temporelles et les redondances psycho-visuelles dues aux informations qui sont ignorées par le système visuel humain.

État-de-l'art de la compression vidéo

Une image est un signal 2D traité par le système visuel humain. Les signaux représentant les images sont généralement sous forme analogique, mais pour le traitement, le stockage et la transmission par des applications informatiques, ils sont convertis de la forme analogique à la forme numérique. Une image numérique est un tableau bidimensionnel de pixels.

La compression de données ou codage source est l'opération informatique consistant à transformer une séquence de bits A en une séquence de bits B plus courte permettant de restituer la même information, ou une information similaire, à l'aide d'un algorithme de décompression [29, 93, 114, 143]. Un algorithme de *compression sans perte* restitue après décompression une séquence de bits strictement identique à l'original. Les algorithmes de compression sans perte sont utilisés pour les archives, les fichiers exécutables ou les textes. Avec un algorithme de *compression avec perte*, la séquence de bits obtenue après décompression

est plus ou moins proche de l'original selon la qualité souhaitée. Ils sont utiles pour les images, le son et la vidéo. La compression d'images ou de vidéos est une application de la compression de données aux images numériques. Le but de cette compression est de réduire la redondance des données (*compression sensing*) dans une image afin qu'elle puisse être stockée sans prendre beaucoup de place ou transmise rapidement.

Dans cette section, nous proposons une revue non exhaustive des différentes méthodes de compression. Inspirés par [117], nous proposons une classification des principales méthodes de compression d'image/vidéo dans la Figure 1.

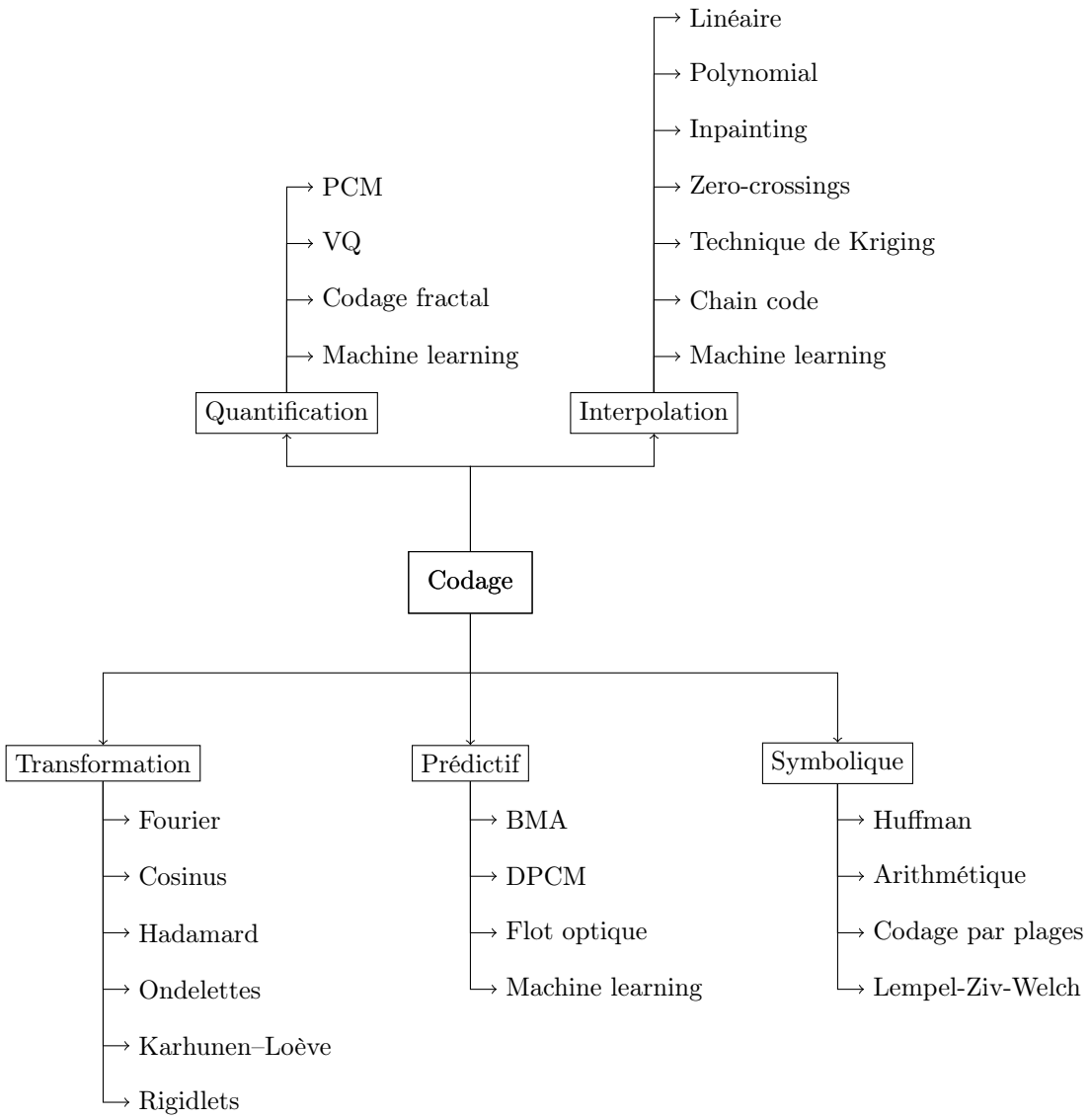


Figure 1: Classification non-exhaustive des techniques de compression vidéo.

Le codage par transformation

Dans le *codage par transformation*, on souhaite exploiter la corrélation spatial des pixels d'une image afin de pouvoir retirer de l'information sans trop détériorer la qualité de l'image. Pour cela, on divise une image en sous-images, puis celles-ci sont transformées en un ensemble de coefficients dé-corrélés grâce à une fonction de transformation. Les coefficients sont ensuite échantillonnés et codés pour la transmission à la place des pixels originaux de l'image. Une grande partie de la compression résulte de l'élimination des coefficients suffisamment petits et de l'échantillonnage des autres, en fonction de la qualité de l'image voulue. Au niveau du récepteur, les bits reçus sont décodés en coefficients de transformation. Une transformation inverse est appliquée pour reconstruire l'image.

Voici quelques exemples de transformées utilisées dans les codecs : la transformée en ondelettes discrète (DWT) [166], la transformée de Karhunen–Loève [159], la transformée de Hadamard [137], la transformée en cosinus discrète [4], la transformée en rigidlets [54], ou encore la transformée de Fourier discrète [71]. Nous allons voir dans la suite que la méthode de codage par transformation est au coeur du fonctionnement des codecs existants de la vie de tous les jours, ainsi que détailler la transformée en cosinus.

Enfin, en 1983, les auteurs ont proposé dans [31] une méthode de compression d'image à mi-chemin entre le codage par transformation et le codage prédictif (voir paragraphe suivant). Cette méthode consiste en la création d'une pyramide de Laplace, c'est-à-dire que l'on va appliquer une série de filtres de plus en plus lissants et réduire de plus en plus la taille de l'image. On obtient ainsi de nouvelles images, $(g_i)_i$, et on sauvegarde les différences entre deux images consécutives i.e. $g_i - g_{i+1}$. Ces différences sont en effet fortement dé-corrélées et très régulières : on peut les sous-échantillonner sans perdre trop d'informations.

Le codage prédictif

Dans *le codage prédictif*, la redondance des données vidéo est déterminée à partir de la corrélation des pixels voisins, qui existe dans les images ou entre les images. Ainsi, la base du codage prédictif est de prédire la valeur d'un pixel à partir des valeurs de pixels qui ont déjà été transmises.

Le codage prédictif le plus courant est celui utilisant des méthodes de compensation de mouvements. La compensation de mouvements en informatique est une technique algorithmique utilisée pour prédire une image dans une vidéo, compte tenu des images précédentes et/ou futures, en tenant compte du mouvement de la caméra et/ou des objets dans la vidéo.

L'exemple le plus simple d'algorithme de compensation de mouvements est la différenciation des images. Cette méthode n'utilise aucun modèle de mouvements : l'image de référence est, sans aucune modification, directement utilisée comme prédiction pour l'image actuelle. La méthode connue sous le nom de modulation d'impulsion codée différentielle (DPCM), quantifie l'erreur de prédiction (c'est-à-dire la différence entre le pixel réel et la valeur du pixel prédit) puis la code. Étant donné que les pixels entre les images sont fortement corrélés, l'erreur de prédiction tend à être faible.

Un autre exemple d'algorithmes sont les algorithmes basés sur une estimation du mouvement. Il s'agit d'un processus de détermination des vecteurs de mouvements qui décrivent la transformation d'une image à une autre, généralement à partir d'images adjacentes dans une séquence vidéo. Il s'agit d'un problème mal posé, car le mouvement est en trois dimensions, mais les images sont une projection sur un plan 2D. Les vecteurs de mouvements peuvent se rapporter à l'image entière comme dans le flot optique [72] ou à des parties spécifiques, comme des blocs rectangulaires, comme les algorithmes de block-matching (BMA) [118, 88, 121, 70, 92, 91, 46]. Une fois le mouvement estimé, un algorithme est invoqué pour utiliser les informations de mouvements provenant de l'estimation du mouvement pour modifier le contenu de l'image de référence, selon le modèle de mouvements, afin de produire une prédiction de l'image actuelle. La prédiction est appelée prédiction à compensation de mouvements ou trame déplacée (DF) [173].

Les prédicteurs linéaires (par exemple [111]) ont été étudiés en utilisant la théorie générale de la prédiction linéaire [80]. Dans cet article, la prédiction pour le prochain échantillon de signal est simplement la somme des échantillons des signaux précédents, chacun étant multiplié par un facteur de pondération approprié.

Dans [153], les auteurs présentent des algorithmes pour prédire les changements d'intensité dans des images de télévision successives.

Enfin, il existe des méthodes utilisant du machine learning pour obtenir l'estimation d'un mouvement comme dans [151].

Le codage par quantification

Le *codage par quantification* est le processus de mise en correspondance de valeurs d'entrée d'un grand ensemble, parfois un ensemble continu, avec des valeurs de sortie dans un ensemble plus petit avec un nombre fini d'éléments. L'arrondi et la troncature sont des exemples typiques de processus de quantification.

La modulation par impulsions et codage (PCM) n'est rien d'autre qu'une représentation du signal discrète en temps et en amplitude [136]. Elle a été appliquée pour la première fois aux signaux de télévision analogique par Goodall en 1951 [74].

Dans la quantification vectorielle (VQ), l'idée de base est de développer un dictionnaire de vecteurs de taille fixe [75]. Pour ce faire, une image donnée est divisée en blocs appelés vecteurs d'image. Ensuite,

chaque vecteur d'image est codé par son index dans le dictionnaire. Ainsi, chaque image est représentée par une séquence d'indices.

La compression fractale repose sur la détection de la récurrence des motifs [11, 87, 167]. Cette méthode convient mieux aux textures et aux images naturelles, car elle repose sur le fait que certaines parties d'une image ressemblent souvent à d'autres parties de la même image.

Une méthode utilisant du machine learning peut être associé au codage par quantification [103].

Le codage par interpolation

Dans le *codage par interpolation*, un sous-ensemble d'éléments de l'image est transmis et les autres sont interpolés [79]. Cette technique de codage a été largement étudiée dans le passé pour le codage numérique des images [98, 82]. La qualité de l'image qui en résulte dépend du nombre et du type d'échantillons éliminés et de la méthode d'interpolation.

La plupart des méthodes d'interpolation ont utilisé des moyennes pondérées, soit en utilisant des lignes droites, soit des polynômes de degré supérieur [51, 53, 101, 112]. Il semble que l'interpolation à l'aide de lignes droites soit assez efficace et que l'interpolation à l'aide de polynômes de degré supérieur n'apporte pas grand-chose.

Une autre méthode de codage par interpolation se basant sur les zero-crossings a été proposée dans [47, 141]. La détection des arêtes d'une image est réduite à la résolution du problème des zero-crossings. Cette méthode de compression utilise une extension du théorème de Logan [107] et l'interpolation spatio-temporelle proposée dans [135].

Il existe aussi des méthodes basées sur l'inpainting d'images [20]. L'objectif est de trouver et de sauvegarder les meilleurs pixels dans une image pour pouvoir reconstruire les pixels manquants grâce à un problème d'inpainting. On détaillera ces méthodes dans la partie I de cette thèse.

Dans [90, 163] les auteurs proposent d'utiliser la technique de Kriging comme opérateur d'interpolation. La technique de Kriging est une méthode d'interpolation qui permet de produire des prédictions de valeurs non observées à partir d'observations de sa valeur à des endroits proches. Elle confère des poids à chaque point en fonction de sa distance par rapport à la valeur inconnue. En fait, ces prédictions sont traitées comme des combinaisons linéaires pondérées des valeurs connues. La méthode de Kriging est plus précise que l'interpolation polynomiale lorsque la valeur non observée est suffisamment proche des valeurs observées.

Une autre méthode, proposée dans [36], consiste à reconstruire une image à partir de la connaissance partielle de ses coefficients de Fourier. L'interpolation est donc réalisée dans le domaine fréquentiel au lieu du domaine spatial.

Il existe également des méthodes utilisant des collections de points d'intérêt de l'espace d'échelle (scale space) [171] comme entrée pour l'algorithme de reconstruction comme [94]. Un point d'intérêt (top point) est un point critique (le gradient de l'image est nul) auquel le déterminant de la hessienne de l'image est également nul. L'interpolation utilisée est l'algorithme de reconstruction proposé par Janssen [89]. Celui-ci tente de minimiser une fonction d'énergie régularisée.

Un code en chaîne (chain code) [64, 175] est une méthode de compression sans perte pour les images binaires basée sur la segmentation d'image. Le principe de base des codes en chaîne est de coder séparément chaque composante connectée dans l'image. Pour chacune de ces régions, un point sur la frontière est sélectionné et ses coordonnées sont transmises. Le codeur se déplace ensuite sur le bord de la région et, à chaque étape, transmet un symbole représentant la direction de ce mouvement jusqu'à ce que le codeur revienne à la position de départ.

Enfin, il existe des méthodes utilisant du machine learning comme par exemple [5].

Le codage symbolique

Le *codage symbolique* est une méthode de compression des données, généralement sans perte, qui code les symboles en utilisant une quantité de bits inversement proportionnelle à la probabilité des symboles. Cette étape élimine la redondance du codage en utilisant des techniques comme le codage de Huffman, le codage arithmétique ou le codage Lempel-Ziv-Welch (LZW) [143]. Les idées de base du codage arithmétique et du codage de Huffman trouvent leur origine dans les travaux originaux de Shannon [150].

Le codage arithmétique [139, 109] et le codage de Huffman [86] attribuent un numéro unique à une séquence entière qui agit comme une étiquette pour cette séquence. En d'autres termes, ce nombre est

un code pour cette séquence : une représentation binaire de ce nombre est un code binaire pour cette séquence. Comme cette étiquette est unique, en théorie, le décodeur peut reconstruire toute la séquence à partir de cette étiquette. Afin de mettre en œuvre cette idée, nous avons besoin d'une bijection entre la séquence et l'étiquette.

Le codage Lempel-Ziv-Welch [172] est un codage par dictionnaire (ou codage par facteur). Comme son nom l'indique, ce codage est basé sur un dictionnaire ou une liste de mots. Ici, l'objectif va être de remplacer une séquence de symboles (un mot) par sa position dans le dictionnaire.

Enfin, le codage par plages (RLE) [140] est une forme de compression de données sans perte dans laquelle les séquences où le même symbole apparaît plusieurs fois de manière consécutive, sont stockées comme un symbole et son nombre de répétition.

Historique des codecs existants

Le premier codec vidéo fut le H.120 et a été proposé en 1984. Celui-ci était basé sur la modulation par codage impulsionnel différentielle (DPCM). La modulation par codage impulsionnel est une méthode utilisée pour représenter numériquement des signaux analogiques échantillonnés. C'est la forme standard de l'audio numérique dans les ordinateurs, les CD audio ou la téléphonie numérique.

En 1988, le standard H.261 fut le premier à utiliser la transformée en cosinus discrète (DCT) et la notion de compensation de mouvements (voir la section suivante). Suite à ce standard, le MPEG-1 à été proposé en 1991 comme une évolution du H.261.

À partir de 1994, la norme MPEG-2/H.262 fut utilisée pour stocker des vidéos sur les DVD.

En 1999, le codec MPEG-4/H.263 introduisit un nouveau codage entropique basé sur des codes à longueur variable.

À partir de 2000, la société *On2 Technologies* développa en parallèle aux standards MPEG-x/H.26x la famille de codec VPx, dont le premier fut le VP3, qui donnera naissance à Theora. Ce dernier utilisera en plus le sous-échantillonage chromatique (*chroma subsampling*). Cette technique se base sur le fait que la perception visuelle de l'homme est plus sensible à la variation de la luminosité plutôt qu'aux variations de couleurs. Ainsi, en décomposant une image de façon adéquat (YCbCr plutôt que RGB), on peut sous-échantillonner les canaux Cb et Cr.

En 2003, le standard H.264/MPEG-4 AVC permit d'améliorer le temps de calcul lors du codage en utilisant notamment une méthode de transformée en cosinus discrète plus efficace. La même année, Motion JPEG2000 proposait un codage sans compensation de mouvements, c'est-à-dire que chaque frame d'une vidéo est encodée indépendamment des autres. Cela évite la propagation d'erreurs d'une frame à l'autre et permet un accès direct à une frame en particulier, au prix d'un taux de compression amoindri. Ce codec était principalement utilisé pour le cinéma.

En 2008, VP8 utilisa la transformée de Hadamard (WHT).

En 2013, la norme HEVC/H.265/MPEG-H Part 2, utilisa une segmentation en blocs de taille variable tandis que le VP9 introduisit la transformée en cosinus discrète asymétrique (ADST).

En 2018, le codec AV1 proposa une utilisa une segmentation en blocs de taille variable plus poussée grâce à un partitionnement récursif de l'image.

Enfin, en 2020, le H.266/VVC apporta principalement de nombreuses fonctionnalités comme un large choix de résolutions, le support des vidéos 360° ou encore le *high dynamic range* (HDR).

Limites

Comme on peut le remarquer dans l'historique de la section précédentes la familles de méthodes de compression (codec) pour la vidéo la plus populaire reste de nos jours la famille MPEG [41]. Ces codecs reposent sur une combinaison de prédiction et de codage par transformation des images : les images dites *intra frame* composant une vidéo sont prédites sans aucune référence aux autres images, tandis que les images dites *inter frame* sont estimées à l'aide des images précédentes ou suivantes ainsi que d'un champ de mouvements entre elles.

On peut interpréter le codage des *intra frames* par de la compression d'image. La plupart des méthodes de compression d'image avec perte consistent à décomposer une image (ou un morceau de celle-ci) comme une somme d'images "de base" et de négliger les images "de base" ayant un effet insignifiant sur la qualité

de l'image. Par exemple, le standard JPEG [165] fonctionne de la manière suivante : une image est découpée en blocs de taille 8 par 8 pixels, que l'on décrit par une fonction,

$$F : \{0, \dots, 7\}^2 \to \mathbb{R}.$$

Ainsi, $F(i, j)$ nous donne l'intensité (le niveau de gris) du pixel situé à la position i, j dans le bloc. On peut alors définir les images de base comme la famille $(F_{u,v})$, pour $0 \leq u, v \leq 7$ de la manière suivante :

$$F_{u,v} : (i, j) \mapsto C \ \cos\left(\frac{(2i+1)u\pi}{16}\right) \ \cos\left(\frac{(2j+1)v\pi}{16}\right).$$

Ici, le nom "images de base" n'est pas choisi au hasard, car la famille $(F_{u,v})$ forme une base pour l'espace des fonctions allant de $\{0, \dots, 7\}^2$ dans $\mathbb{R}$, et est de plus orthonormale (pour des C bien choisis) pour le produit scalaire,

$$\langle G, H \rangle := \sum_{0 \leq i,j \leq 7} G(i, j)\, H(i, j).$$

On peut alors décomposer F comme,

$$F = \sum_{0 \leq u,v \leq 7} c_{u,v}\, F_{u,v},$$

où $c_{u,v} := \langle F, F_{u,v} \rangle$. Pour la compression, ce sont ces $c_{u,v}$ qui sont sauvegardés. Avec cette décomposition, les coefficients $c_{u,v}$ sont proches de 0 et certains peuvent être négligés, en fonction du taux de compression ou de la qualité souhaité.

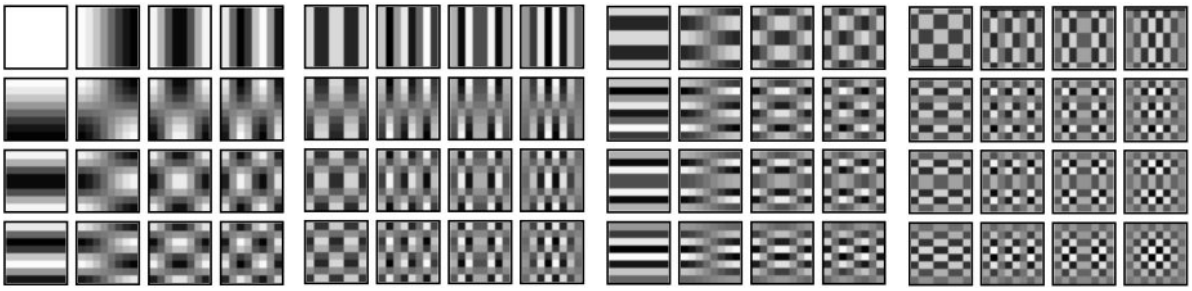


Figure 2: Les 64 images de base pour le standard JPEG.

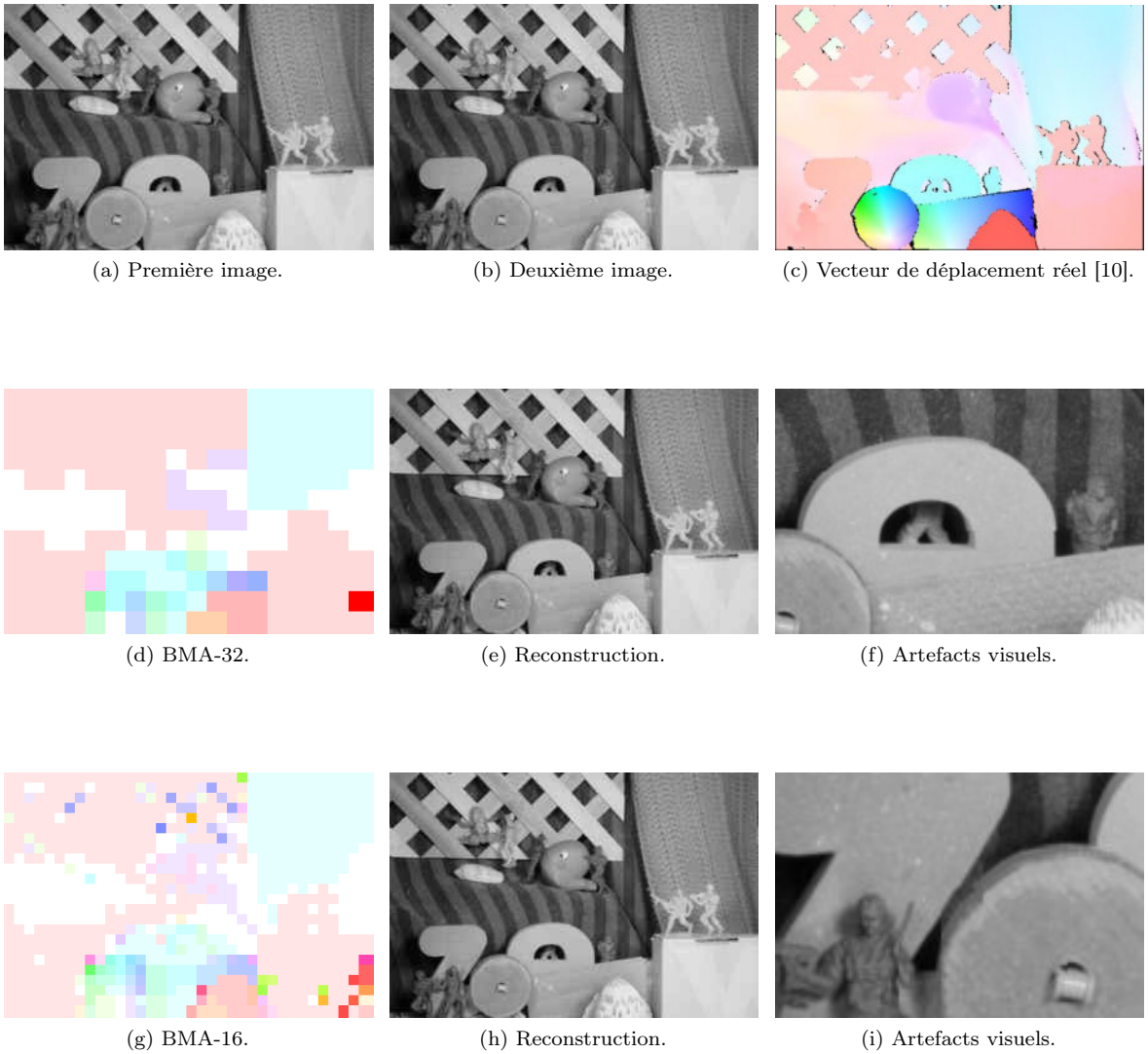
Plus généralement, les différentes méthodes de compression d'image actuelles diffèrent principalement par la taille des blocs ainsi que par le choix des images de bases [155, 21]. Le problème majeur de ces codecs vient de la décomposition de l'image en blocs qui introduit l'apparition d'artefacts visuels lors de la décompression, comme illustré ci-dessous :

(a) Image originale. (b) JPEG. (c) JPEG2000. (d) WEBP.

Figure 3: Apparition d'artefacts visuels lors de forts taux de compression.

La prédiction des *inter frames* est effectuée à l'aide de techniques de compensation de mouvement (motion compensation). La compensation de mouvement est une technique algorithmique utilisée pour prédire une image dans une vidéo, compte tenu des images précédentes et/ou futures, en tenant compte du mouvement de la caméra et/ou des objets dans la vidéo. La plupart des standards de codage vidéo, tels que les formats H.26x et MPEG [41, 154], utilisent la transformée discrète en cosinus compensée en mouvement (motion-compensated DCT). Ici encore, l'image à traiter est divisée en blocs de taille fixe (par exemple 16 par 16) et pour passer d'une image à la suivante, les blocs sont translatés grâce à un vecteur de mouvement. Ces types d'algorithmes font partie de la famille des *Block-Matching Algorithms* (BMA).

Plusieurs stratégies de calcul pour trouver le vecteur de mouvement sont possibles, mais l'approche la plus simple, bien que coûteuse en terme de temps de calcul, est la recherche exhaustive [88]. Cet algorithme consiste à déplacer chaque bloc dans toutes les directions possibles dans un voisinage du bloc afin de minimiser une erreur locale entre le bloc déplacé et l'image suivante (ou précédente). Il existe d'autres algorithmes de Block-Matching, avec comme principale différence leur précision et le temps de calcul [1].



(a) Première image. (b) Deuxième image. (c) Vecteur de déplacement réel [10].

(d) BMA-32. (e) Reconstruction. (f) Artefacts visuels.

(g) BMA-16. (h) Reconstruction. (i) Artefacts visuels.

Figure 4: Apparition d'artefacts visuels dans le vecteur de mouvement avec les BMA.

Tout comme pour les *intra frames*, chaque bloc est considéré indépendamment des autres. Ainsi, le principal inconvénient de la compensation de mouvement par blocs est qu'elle introduit des discontinuités aux limites des blocs.

Contributions

Le but de cette thèse est de proposer de nouvelles méthodes de compression d'image et d'estimation du mouvement, où les artefacts sont absents. En 2009, Schmaltz *et al* ont réussi à surpasser la compression JPEG2000 en utilisant des méthodes de compression basées sur l'inpainting d'image par équations aux dérivées partielles (EDP) [147]. Le principe de ces méthodes est de reconstruire une image à partir d'un faible nombre de pixels connus, la difficulté majeure étant de choisir les "bons" pixels à garder afin d'obtenir une "bonne" reconstruction. L'avantage de ces méthodes est de considérer la totalité de l'image et ainsi d'éviter les artefacts visuels. De plus, l'utilisation des EDP permet d'obtenir une image plus régulière et donc, plus agréable visuellement.

(a) JPEG. (b) JPEG2000. (c) [66] (EDP). (d) [147] (EDP).

Figure 5: Comparaison des méthodes de compression par décomposition avec celles par inpainting [147].

Concernant l'estimation du mouvement, S. Andris *et al* ont proposé en 2016 dans [8] de calculer le vecteur de mouvement à l'aide du *flot optique*. Une fois de plus, ces méthodes utilisent des EDP et permettent d'éviter l'apparition des artefacts visuels.

(a) Vecteur de mouvement réel [10]. (b) Vecteur de mouvement [72]. (c) Reconstruction.

Figure 6: Vecteur de mouvements par flot optique.

Le principal objectif dans cette thèse est de proposer de nouveaux modèles de compression prenant en compte le fait que les vraies images contiennent du bruit. Le but est d'obtenir un codec permettant, en une seule étape, de compresser et de débruiter une image ou une vidéo.

Organisation de la thèse

La partie I, est consacrée à la compression des *inter frames*, donc la compression d'image par inpainting. Dans le chapitre 1, nous allons proposer un modèle de compression prenant en compte le bruit *gaussien* dans les images et donner un critère analytique pour localiser les meilleurs pixels à sauvegarder. Dans le chapitre 2, nous allons étendre les résultats précédents à une approche itérative afin d'améliorer le débruitage lors de la compression. Notamment, l'approche itérative nous permet de modifier au cours des itéraions l'ensemble des pixels retenus pour la compression. Cet ensemble est ainsi construit de manière dynamique par un processus d'enrichisement-délestage à travers l'échange de pixels (les moins "influents"

avec les plus significatifs). Le débruitage se trouve ainsi amélioré par la prise en compte des nouvelles valeurs de l'intensité (valeur à l'étape n) et du masque remis à jour.

Enfin, dans le chapitre 3, nous allons présenter une nouvelle méthode pour déterminer l'ensemble de compression basée sur la notion d'état adjoint. Cette méthode conduit à un critère de sélection plus lisse (soft-tresholding) qui permet une meilleure reconstruction et qui s'applique avec la même éfficacité au bruit *gaussien* et au bruit *salt and pepper*. Nous présentons dans un petit chapitre a part, les résultats préliminaires concernant la mise en ouvre des algorithmes dans cette thèse par carte graphique pour accéler les calculs et permettre à des méthodes variationnelles de réaliser des performances en simulations numériques qui approchent le traitement en temps réel.

Dans la partie II, nous allons étudier l'estimation du mouvement au sein d'une vidéo, dans le but de la compresser. Nous allons proposer dans le chapitre 4 une nouvelle formulation du flot optique basée sur le transport optimal [164]. On présentera des formulations classiques du problème de la détermination du flot optique, notamment avec prise en compte de la variation de la luminosité, puis la nouvelle formulation basée sur le transport de densités à la place des particules (pixels).

La partie III, nous allons la consacrer à appliquer les méthodes des deux parties précédentes pour effectuer la compression video. Nous présentons le principe du fonctionnement général du codec vidéo. Dans le chapitre 5, différentes étapes de cette compression vidéo sont mises en oeuvre sur des exemples et des comparaisons de la qualité visuelle des reconstructions sont données.

En annexes, nous donnons quelques détails et démonstrations techniques. L'annexe A est liée aux chapitres 1 et 2 et la démonstration de résultats utilisés. L'annexe B, concerne les démonstrations de certains résultats utlisés au chapitre 3. Enfin l'annexe C est dédié à des rappels d'outils généraux évoqués le long de la thèse.

Description détaillée du contenu

Compression d'image par "inpainting"

L'utilisation d'équations aux dérivées partielles (EDPs) pour la compression d'image est un domaine de recherche relativement récent et qui gagne en popularité depuis quelques années. Cependant, la majorité de ces techniques sont en réalité couplées avec des méthodes de compression plus traditionnelles comme le JPEG [138], le JPEG 2000 [155] ou la transformation en ondelettes [21]. En effet, les EDPs sont plutôt utilisées pour effectuer un filtrage de l'image avant ou après l'avoir compressée avec ces codecs. On peut citer par exemple le lissage d'image ou le débruitage d'image [40, 113, 160, 141, 2]. Le but de la compression d'image par inpainting est de réduire la quantité de données nécessaires pour stocker l'image, en utilisant uniquement des méthodes basées sur les équations aux dérivées partielles. Le principal objectif de ce champ de recherche est de trouver tout d'abord quelles méthodes d'inpainting utiliser en fonction des caractéristiques de l'image à compresser (images texturées, bruitées, cartoon, ...), mais également de savoir quels sont les "bons" pixels à garder pour avoir une reconstruction satisfaisante [52, 146, 15, 106]. Récemment, ces techniques ont été appliquées à des signaux autres que des images, comme pour la compression audio [130]. Dans ce chapitre, nous ferons une liste non exhaustive des méthodes de compression par inpainting d'image. En particulier, il existe des méthodes utilisant le machine learning qui ne seront pas détaillée ici [148, 133].

Inpainting d'images

La technique d'inpainting trouve son origine dans la restauration de tableaux anciens où certaines parties de peintures sont manquantes. Ces dernières sont repeintes en s'aidant des couleurs et des motifs aux alentours. En analyse d'image, cette méthode est également utilisée pour restaurer une image lorsque celle-ci est détériorée, mais également pour enlever certains éléments de l'image [20, 146]. L'inpainting est un problème mal posé au sens d'Hadamard. Soit K un sous-ensemble de D dans lequel l'image à traiter n'est pas détériorée. On appelle K le *masque* d'inpainting et $D \setminus K$ le domaine d'inpainting. Le modèle général de l'inpainting d'image est le suivant :

Problem 0.0.1. *Trouver u dans V tel que,*

$$\begin{cases} A(u) = 0, & \text{dans } D \setminus K, \\ u = f, & \text{dans } K, \\ \frac{\partial u}{\partial \mathbf{n}} = 0, & \text{sur } \partial D, \end{cases}$$

où $A : V \to \mathbb{R}$ est un opérateur donné. Dans la suite, nous allons voir quelques exemples d'opérateurs pour l'inpainting. D'autres types d'opérateurs existent, comme par exemple le cas où A est un réseau de neurones [174, 125], mais nous allons nous focaliser dans la suite sur les méthodes de reconstruction (interpolation) par EDPs.

Example 1 (Diffusion homogène). On commence par donner l'exemple de la diffusion homogène [171], énoncé de la manière suivante :

Problem 0.0.2. *Trouver u dans $H^1(D)$ tel que,*

$$\begin{cases} -\Delta u = 0, & \text{dans } D \setminus K, \\ u = f, & \text{dans } K, \\ \frac{\partial u}{\partial \mathbf{n}} = 0, & \text{sur } \partial D. \end{cases}$$

Cet interpolation est efficace pour reconstruire les zones homogènes de l'image, mais rend les arêtes floues. Une illustration en est donnée dans la figure 7 (b). On peut généraliser à d'autres opétareurs de diffusion par exemple le p-laplacien.

Example 2 (Diffusion de Charbonnier). Introduit dans le cas des méthodes dites scale-space methods [129], cet opérateur corrige l'effet régularisant (floutage).

Problem 0.0.3. *Trouver u dans $H^1(D)$ tel que,*

$$\begin{cases} -\operatorname{div}\left(g(|\nabla u|^2)\,\nabla u\right) = 0, & \text{dans } D \setminus K, \\ u = f, & \text{dans } K, \\ \frac{\partial u}{\partial \mathbf{n}} = 0, & \text{sur } \partial D, \end{cases}$$

où $g : \mathbb{R} \to \mathbb{R}$ est une fonction, dite de diffusion, positive, décroissante et telle que $g(0) = 1$. Le but de g est de réduire la diffusion au niveau des arêtes. En effet, lorsque $|\nabla u|$ est grand (proche des arêtes), g est faible, donc on diffuse peu. Au contraire, dans les zone homogènes, $|\nabla u|$ est faible et donc g est proche de 1. Pour la diffusion de Charbonnier [42], on utilise la fonction,

$$g(s^2) = \frac{1}{\sqrt{1 + (s/\lambda)^2}},$$

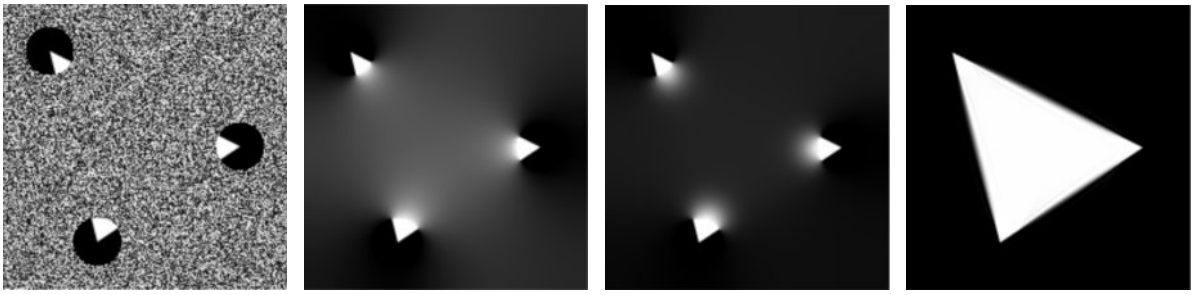
avec $\lambda > 0$ un paramètre dit de *contraste*. Une variante de la diffusion de Charbonnier a été proposée dans [37]. Ici, $g(|\nabla u|^2)$ est remplacée par $g(|\nabla u_\sigma|^2)$ avec u_σ une version plus lisse de u dans le but d'être moins sensible au bruit. Une illustration de cette reconstruction est donnée dans la figure 7 (c).

Example 3 (Edge-enhancing diffusion). Dans l'exemple précédent, l'interpolation anisotropique tentait de pallier au principal défaut de la diffusion homogène. Pour celà, les auteurs ont ajouté un paramètre G pour contrôler la diffusion en fonction de module du gradient de u. L'edge-enhancing diffusion [169] est une amélioration permettant de prendre en compte également la direction de ∇u. Afin de ne pas flouter les arêtes, on veut que la diffusion dans la direction perpendiculaire à celles-ci soit faible.

Problem 0.0.4. *Trouver u dans $H^1(D)$ tel que,*

$$\begin{cases} -\operatorname{div}\left(G(\nabla u_\sigma \nabla u_\sigma^T)\,\nabla u\right) = 0, & \text{dans } D \setminus K, \\ u = f, & \text{dans } K, \\ \frac{\partial u}{\partial \mathbf{n}} = 0, & \text{sur } \partial D. \end{cases}$$

Cette fois, G va de $\mathbb{R}^{d\times d}$ dans $\mathbb{R}^{d\times d}$. Si $M \in \mathbb{R}^{d\times d}$ est diagonalisable, on pose $G(M) := \mathrm{diag}\big(g(\lambda_i(M)\big)$ avec g de Charbonnier comme dans l'exemple 2 et λ_i les valeurs propres de M comptées avec multiplicité. Puisque $\nabla u_\sigma \nabla u_\sigma^T$ est symétrique, elle est diagonalisable d'après le théorème spectral. Une illustration de cet inpainting est donnée dans la figure 7 (d).



(a) image d'entrée. (b) Diffusion homogène. (c) Charbonnier régularisé. (d) Edge-enhancing diffusion.

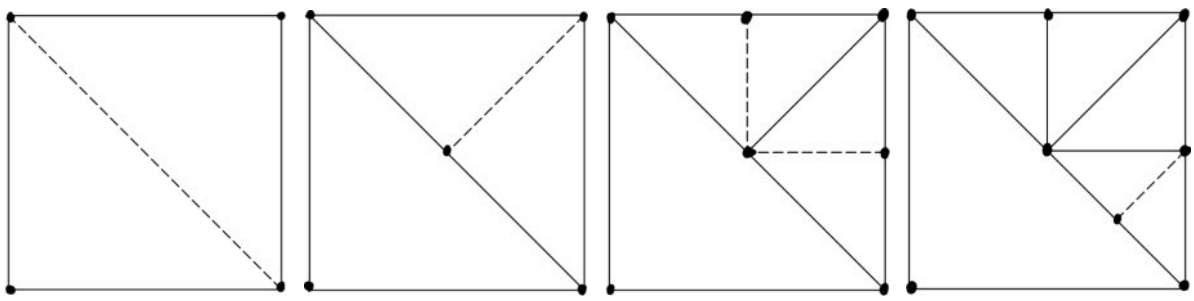
Figure 7: Illustration de quelques méthodes d'inpainting tirée de [146]

Compression par l'algorithme B-Tree

En 1997, Distasi *et al* [53] ont proposé un algorithme permettant la décomposition d'une image en triangles de plus en plus petits. Une fois cette décomposition faite, ils ont sauvegardé la valeur des pixels aux sommets de chaque triangle. Ces informations, couplées avec une interpolation linéaire, ont permis de compresser et de décoder une image de manière plus rapide que les principales méthodes utilisées jusque là. De plus, le découpage pouvant être représenté sous la forme d'un arbre binaire, cette méthode est également extrêmement simple à sauvegarder. Plus précisément, il s'agit d'un algorithme récursif comme suit :

Data: Image f, seuil ϵ de l'erreur acceptée, Masque d'inpainting K.
Result: Nouveau masque d'inpainting K.
Ajouter le schéma de points dans K (par exemple, un point pour les 3 sommets du triangle);
if *On peut encore diviser* f **then**
  Calculer la reconstruction par inpainting u à partir du masque K et de f;
  if *Erreur*$(u, f) > \varepsilon$ **then**
    Diviser f en deux sous-images f_1 et f_2, en créant un triangle supplémentaire;
    Appliquer l'algorithme sur la première sous-image i.e. $K_1 \leftarrow$ B-Tree(f_1, ϵ, K);
    Appliquer l'algorithme sur la deuxième sous-image i.e. $K_2 \leftarrow$ B-Tree(f_2, ϵ, K);
    Fusionner les deux sous-masques i.e. $K \leftarrow K_1 \cup K_2$;
  end
end

Algorithm 1: B-Tree.



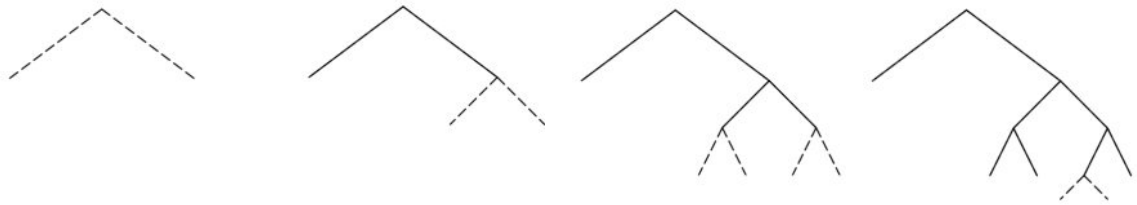


Figure 8: Illustration du B-Tree.

En 2005, Galić *et al* [66] proposèrent de remplacer la simple interpolation linéaire par une interpolation effectuée par inpainting, par exemple par de l'edge-enhancing diffusion [169].

Enfin, en 2014, Schmaltz *et al* [146] proposèrent de remplacer la division en triangles par une division en rectangles et ont déterminé que les meilleurs points d'interpolation pour chaque rectangle à sauvegarder sont les quatre coins plus le centre. De plus, ils ont comparé différentes méthodes d'interpolation (i.e. inpainting) et, parmi celles testées, l'edge-enhancing diffusion (exemple 3) semblait la plus performante.

(a) Masque B-Tree (10%). (b) Reconstruction. (c) Masque B-Tree (5%). (d) Reconstruction.

Figure 9: Compression B-Tree avec l'inpainting de diffusion homogène.

Compression par optimisation de forme

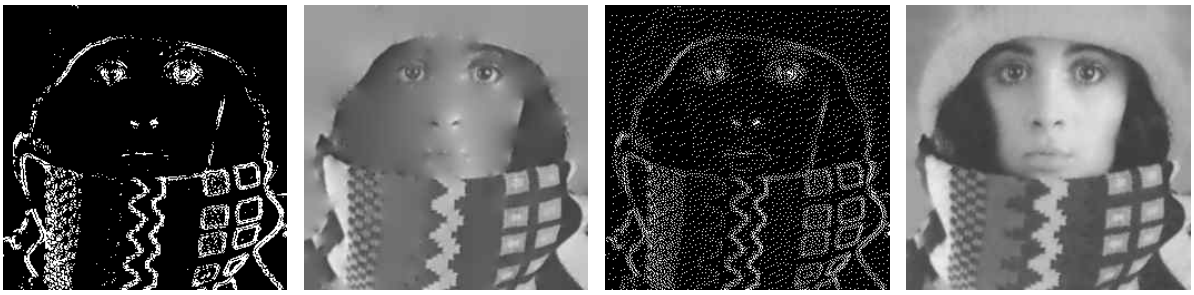
Soit D un sous-ensemble de $\mathbb{R}^2$ le support d'une image bi-dimensionnelle en niveau de gris f. Pour une méthode de reconstruction donnée, le but est de déterminer, s'il et possible, un ensemble $K \subset D$, qu'on va appeler masque,, constitué de pixels à retenir dans la phase de compression et qui soit optimal au regard d'un certain critère à optimiser. Celà revient à minimiser une fonction coût $\mathcal{F} : V \to \mathbb{R}$ ayant comme argument K, sur un ensemble de contraintes non linéaires, dont la solution u_K du problème 0.0.1 et la taille de K, soit $m(K) \leq c$ où m est une mesure de K bien choisie et $c > 0$. Le problème de compression peut s'écrire alors comme le problème de minimisation suivant,

$$\min_{K \subset D,\ m(K) \leq c} \{\mathcal{F}(u_K) \mid u_K \text{ solution de problème 0.0.1}\}.$$

Dans l'article [15], les auteurs ont étudié le cas où le problème d'inpainting est donné par la diffusion homogène (exemple 1) et où,

$$\mathcal{F}(u) := \frac{1}{2} \int_D |\nabla u - \nabla f|^2,\ \forall u \in H^1(D).$$

L'analyse de ce problème d'optimisation de forme permet de trouver la localisation optimale des points à garder, à savoir, là où l'intensité varie le plus, c'est-à-dire les arêtes de l'image. Deux méthodes ont été proposées, la première où seuls les points qui maximisent $|\Delta f|$ sont gardés (figure 10 (a)-(b)), et la seconde, où la densité des pixels gardés augmente avec $|\Delta f|$ (figure 10 (c)-(d)).



(a) Masque gradient topologique. (b) Reconstruction avec (a). (c) Masque "fat pixels". (d) Reconstruction avec (c).

Figure 10: Masque et reconstruction avec les méthodes proposées dans [15] (10% des pixels dans le masque).

Compression par algorithmes probabilistes

En 2012, Mainberger *et al* proposèrent un algorithme pour la sélection des pixels qu'il nommèrent algorithme de *sparsification* [110]. Ils commence par un masque contenant l'ensemble des pixels de l'image, puis à chaque itération, supprime de manière aléatoire une fraction des pixels du masque et reconstruisent l'image. Si l'erreur, calculée uniquement à chaque pixel supprimé est grande, il est remit, de sorte que soit ajouté au masque un sous-ensemble des pixels supprimés présentant l'erreur la plus importante. Ainsi, les pixels les moins significatifs sont retirés jusqu'à atteindre le nombre de pixels souhaitée pour la compression.

Data: Image f, fraction p de pixels dans l'ensemble candidat, fraction q de pixels supprimés à chaque itération, densité d souhaitée dans le masque.
Result: Masque d'inpainting K.
$K \leftarrow D$;
while $|K| > d\,|D|$ **do**
  Choisir aléatoirement $p\,|K|$ indices de pixels de K dans un ensemble T;
  Mettre à jour $K \leftarrow \{x_i \in K \mid i \notin T\}$;
  Calculer la reconstruction par inpainting u à partir du masque K et de f;
  Calculer l'erreur $e := |u - f|$;
  Prendre les $(1-q)|D|$ pixels x_i qui maximisent la valeur $e(x_i)$ et les re-mettre dans K;
end

Algorithm 2: Algorithme de *sparsification*.

L'inconvénient de la méthode présentée précédemment est qu'une fois qu'un pixel est retiré du masque, il n'y retourne plus même s'il pouvait être significatif pour la reconstruction. Mainberger *et al* ont proposé dans le même article l'algorithme de *nonlocal pixel exchange* [110, 131]. Celui-ci représente une étape de post-traitement de l'algorithme de *sparsification*. À chaque itération, on choisit au hasard une quantité fixe de pixels hors du masque. Un sous-ensemble de ceux qui présentent la plus grande erreur de reconstruction est échangé avec des pixels de masque choisis eux aussi au hasard. Si l'erreur de reconstruction pour le nouveau masque ne diminue pas, on réinitialise le masque à sa configuration précédente. Sinon, on garde le nouveau masque.

Notons que l'analyse de tels algorithmes n'est pas faite et que l'essentiel de l'approche est purement heuristique (convergence, lois de probabilité utilisées -autre que la loi uniforme-, ...).

Data: Image f, masque d'inpainting K, taille m de l'ensemble des candidats, nombre m de pixels echangés par itération.
Result: Masque d'inpainting K.

Calculer la reconstruction par interpolation de u à partir du masque K et de f;
for do
  Choisir aléatoirement $m \leq |K|$ pixels de $D \setminus K$ dans un ensemble T;
  Pour x dans T, calculer $e(x) := |u(x) - f(x)|$;
  Enlever aléatoirement $n \leq |T|$ pixels de K et stocker ce nouveau masque dans K^{new};
  Ajouter dans K^{new} les n pixels x de T ayant les plus grandes erreurs $e(x)$;
  Calculer la reconstruction par inpainting u^{new} à partir du masque K^{new} et de f;
  if *Erreur*$(u, f) >$ *Erreur*(u^{new}, f) **then**
    $u \leftarrow u^{\text{new}}$;
    $K \leftarrow K^{\text{new}}$;
  end
end

Algorithm 3: Algorithme de *nonlocal pixel exchange*.

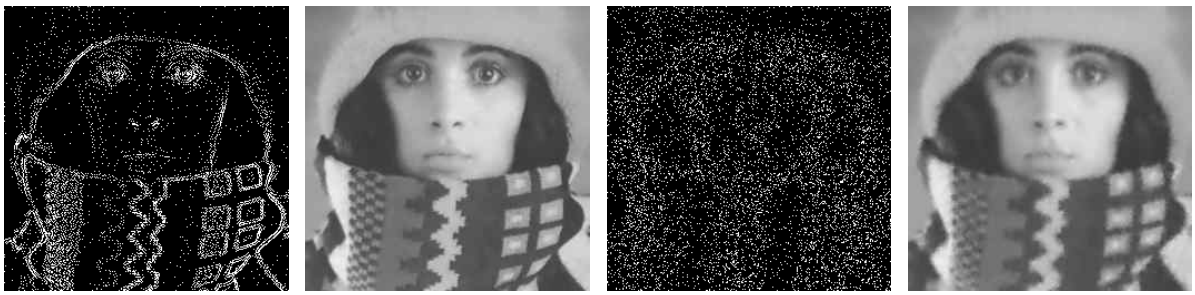
En 2016, Peter *et al* proposèrent l'algorithme de *densification* [134]. On commence par un masque vide et, pour chaque itération, on choisit aléatoirement α pixels qui n'appartiennent pas au masque. On ajoute ensuite l'unique pixel qui améliore le plus l'erreur de reconstruction globale par rapport à l'image.

Data: Image f, nombre α de candidats, densité d souhaitée dans le masque.
Result: Masque d'inpainting K.

$K \leftarrow \emptyset$;
while *La densité de pixel dans le masque est inférieure à celle souhaitée i.e.* $|K| \leq d\,|D|$ **do**
  Choisir aléatoirement un ensemble $A \subset \{1, \cdot, M\,N\}$, de taille α, d'indices de pixels absent de K;
  for $i \in A$ **do**
    Créer un masque temporaire $K^i = K \cup \{x_i\}$ identique à K mais avec le pixel $x_i \in D \setminus K$ en plus;
    Calculer la reconstruction par inpainting u^i à partir du masque K^i et de f;
  end
  Mettre à jour $K := \text{argmin}_{K^i}\, \text{Erreur}(u^i, f)$;
end

Algorithm 4: Algorithme de *densification*.

Dans la figure 11, on propose une illustration de ces méthodes dites probabilistes. Pour l'algorithme de *densification*, on utilise une version modifiée où on ajoute plusieurs pixels à chaque itération plutôt qu'un seul (voir chapitre 2).



(a) Masque par *sparsification* (sans *nonlocal pixel exchange*). (b) Reconstruction avec (a). (c) Masque par *densification*. (d) Reconstruction avec (c).

Figure 11: Compression par algorithmes probabilistes avec l'inpainting de diffusion homogène. (10% des pixels dans le masque).

Améliorations

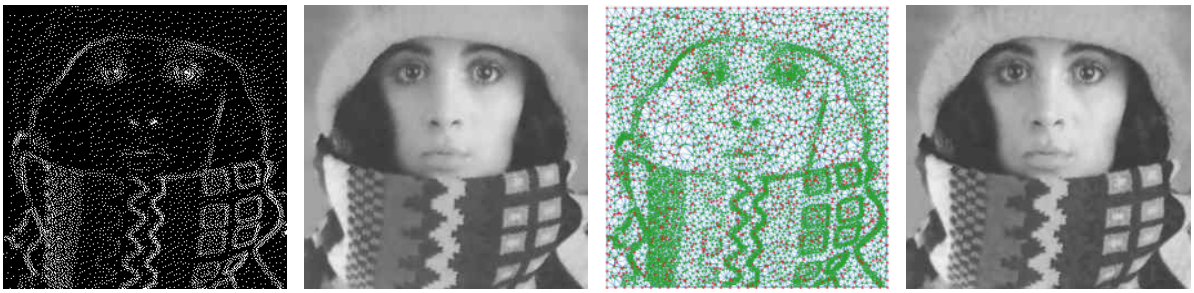
En 2010, Bae *et al* ont proposé dans [9] l'*interpolation swapping*. Il s'agit d'une étape d'inpainting supplémentaire où le rôle des pixels connus et inconnus est échangé. C'est-à-dire qu'une fois que la reconstruction u_K^f est trouvée, on calcule $u_{D\setminus K}^{u_K^f}$. Cela permet d'obtenir une reconstruction finale plus lisse.

En 2012, Mainberger *et al* [110] proposèrent d'améliorer la reconstruction de l'image en changeant la valeur des pixels gardés dans un masque d'inpainting fixé. C'est ce qu'ils ont appelé de l'*optimisation tonale*. Plus précisément, si on note la reconstruction u_K^f par inpainting, on pose,

$$\delta f := \text{argmin}_{\delta f} \|f - u_K^{f+\delta f}\|^2.$$

Alors, on sauvegarde $(f + \delta f)|_K$ comme donnée de la compression.

En 2021, Chizhov *et al* [43] utilisèrent les éléments finis plutôt que les différences finies pour résoudre le problème d'inpainting. Cela permet d'une part de réduire le nombre d'inconnues (points rouges sur la Figure 12 (c)) lors de la discrétisation, ce qui induit un temps de calcul réduit, mais également d'avoir une reconstruction plus régulière et donc plus agréable visuellement.



(a) Masque "fat pixels" [15]. (b) Différences finies. (c) Maillage. (d) Éléments finis $\mathbb{P}_1$.

Figure 12: Masque, maillage et reconstruction avec les méthodes proposées dans [15] et [43] (10% des pixels dans le masque).

Organisation de la partie I

Dans le chapitre 1, nous introduirons des modèles basés sur l'optimisation de formes pour trouver les meilleures données d'interpolation dans la compression d'images avec du bruit. Nous considérerons un modèle d'EDP stationnaire continu, obtenu en se concentrant sur la première itération de l'EDP discrétisée dépendante du temps que nous analyserons dans le cadre de la Γ-convergence. Nous obtiendrons des informations ponctuelles sur la pertinence de chaque pixel par une analyse asymptotique. Ensuite, nous introduirons un cadre en dimension finie du modèle continu basé sur des boules avec un rayon positif, et nous étudierons par Γ-convergence l'asymptotique lorsque le rayon tend vers zéro. Dans le chapitre 2, nous étendrons les résultats obtenus à un problème parabolique où l'ensemble des pixels stockés dépend du temps. Nous considérerons le système dynamique semi-discret associé au modèle qui donne lieu à une méthode itérative où les données stockées sont modifiées pendant les itérations pour obtenir les meilleurs résultats. Nous effectuerons l'analyse et dériverons plusieurs algorithmes itératifs que nous implémenterons et comparerons pour obtenir les stratégies les plus efficaces de compression et d'inpainting pour les images bruitées. Nous présenterons quelques calculs numériques pour confirmer les résultats théoriques. Enfin, nous proposerons un modèle modifié qui permet aux données d'inpainting de changer avec l'itération. Pour finir, dans le chapitre 3, nous proposerons d'étudier le problème de compression d'image dans le cas où l'erreur est plus générale que celle considérée dans les deux chapitres précédents. De cette façon, notre nouvelle méthode, dérivée de la méthode de l'adjoint, sera plus robuste aux différents types de bruit que les méthodes précédentes.

Estimation avec prise en compte de la variation de la luminosité

Le flot optique est un champ de vecteurs donnant le déplacement de l'intensité des pixels d'une image animée au cours du temps. Dans le cas d'une vidéo, lorsqu'on a pas de changement de scène, deux images

successives sont visuellement très semblables et donc, le flot optique sera très homogène. C'est pour cela que, plutôt que de sauvegarder les deux images indépendamment, les auteurs de [2] proposent de calculer le flot optique et de le sous-échantillonner. Cette dernière étape ne devrait pas avoir une grande influence sur la qualité de la reconstruction grâce à la régularité du flot optique.

Dans cette partie, nous allons voir quelques méthodes classiques pour le calcul du flot optique.

Contrainte fondamentale du flot optique en petits déplacements

Considérons une séquence d'images successives dont l'intensité d'un pixel à la position $\mathbf{x}$ de D au moment $t \geq 0$ est donnée par,

$$f : D \times [0,T] \to [0,1],\ (\mathbf{x},t) \mapsto f(\mathbf{x},t).$$

La contrainte fondamentale du flot optique est l'hypothèse principale pour l'estimation du flot optique. Nous allons voir uniquement le cas où l'on suppose que la luminosité reste constante au cours du temps. La luminosité constante se traduit par,

$$f(\mathbf{x}+\delta\mathbf{x}, t+\delta t) = f(\mathbf{x},t).$$

De plus, nous ne supposons que le cas des petits déplacements, c'est-à-dire que lorsque $\delta\mathbf{x}$ est petit. On passe aux coordonnées lagrangiennes. Puisqu'on est dans le cas de la luminosité constante, on suppose que l'intensité d'un point reste constante le long de sa trajectoire, c'est-à-dire que pour $(\mathbf{x_0}, t_0)$, il existe une trajectoire $t \mapsto (\mathbf{x}(t), t)$, telle que $(\mathbf{x}(t_0), t_0) = (\mathbf{x_0}, t_0)$ et pour tout $t > 0$, $f(\mathbf{x}(t), t) = f(\mathbf{x_0}, t_0)$. En petits déplacements, il est raisonnable de supposer que f est dérivable, d'où en dérivant par rapport à t et en prenant $t = t_0$, on a,

$$\partial_t f(\mathbf{x_0}, t_0) + \partial_t \mathbf{x}(t_0) \cdot \nabla f(\mathbf{x_0}, t_0) = 0.$$

En posant $\mathbf{u}(\mathbf{x_0}) = \partial_t \mathbf{x}(t_0)$, et puisque l'équation ci-dessus est vérifiée pour tout $(\mathbf{x_0}, t_0)$, on obtient la *contrainte fondamentale du flot optique*,

$$\nabla f(\mathbf{x},t) \cdot \mathbf{u}(\mathbf{x},t) + \partial_t f(\mathbf{x},t) = 0.$$

Formulation variationnelle

Utiliser uniquement la contrainte fondamentale du flot optique ne suffit pas à déterminer les composantes du flot optique $\mathbf{u}$. Ce problème est appelé problème de l'ouverture (*aperture problem*) et on doit alors ajouter une deuxième contrainte.

En 1981, Lucas et Kanade [108] proposent la *méthode locale*, qui cherche à minimiser $\mathbf{u}$ pour tout $t \in [0,T]$ et $\mathbf{x} \in D$,

$$\mathcal{D}_{\mathrm{l}}(\mathbf{u}) = K_\rho \star \Big(\nabla f(\mathbf{x},t) \cdot \mathbf{u}(\mathbf{x},t) + \partial_t f(\mathbf{x},t) \Big).$$

Ici, $\star$ désigne le produit de convolution et K_ρ désigne le *noyau de la chaleur*, solution fondamentale de l'équation de la chaleur, qui, une fois convolué avec une fonction g a pour effet de lisser g. Si g est une image, cette opération est un *flou gaussien* de déviation standard $\rho > 0$. L'équation d'Euler-Lagrange nous donne en dimension 2,

$$\begin{cases} K_\rho \star (f_x)^2\ u_1 + K_\rho \star (f_x f_y)\ u_2 = -K_\rho \star (f_x f_t) & , \quad \text{dans } [0,T] \times D, \\ K_\rho \star (f_y f_x)\ u_1 + K_\rho \star (f_y)^2\ u_2 = -K_\rho \star (f_y f_t) & , \quad \text{dans } [0,T] \times D, \end{cases}$$

Le système linéaire précédent peut être résolu ponctuellement, ce qui présente un avantage en terme de temps de calcul.

La même année, Horn et Schunck [85] proposèrent la *méthode globale*, qui consiste à minimiser sur $\mathbf{u}$ une énergie associée à la *contrainte fondamentale du flot optique*,

$$\mathcal{D}_{\mathrm{g}}(\mathbf{u}) = \int_D \nabla f(\mathbf{x},t) \cdot \mathbf{u}(\mathbf{x},t) + \partial_t f(\mathbf{x},t)\ dx + \alpha\, \mathcal{R}(\mathbf{u}).$$

Si $\mathcal{R}(\mathbf{u}) = \sum_{i=1}^{d} \int_D |\nabla u_i|^2\ dx$, l'équation d'Euler-Lagrange nous donne en dimension 2,

$$\begin{cases} -\alpha\Delta u_1 + (f_x)^2\ u_1 + (f_x f_y)\ u_2 = -(f_x f_t) & , \quad \text{dans } [0,T] \times D, \\ -\alpha\Delta u_2 + (f_y f_x)\ u_1 + (f_y)^2\ u_2 = -(f_y f_t) & , \quad \text{dans } [0,T] \times D, \\ \frac{\partial u_1}{\partial \mathbf{n}} = \frac{\partial u_2}{\partial \mathbf{n}} = 0 & , \quad \text{sur } [0,T] \times \partial D. \end{cases}$$

En 2005, Bruhn *et al* [27], proposèrent la *méthode globale-locale*, qui est la combinaison des deux méthodes précédentes, et ainsi de minimiser $\mathbf{u}$ pour tout $t \in [0, T - \delta t[$,

$$\mathcal{D}_{\text{gl}}(\mathbf{u}) = \int_D K_\rho(\mathbf{x}) \star \Big(\nabla f(\mathbf{x}, t) \cdot \mathbf{u}(\mathbf{x}, t) + \partial_t f(\mathbf{x}, t) \Big)\, dx + \alpha \mathcal{R}(\mathbf{u}).$$

Si $\mathcal{R}(\mathbf{u}) = \sum_{i=1}^{d} \int_D |\nabla u_i|^2 \, dx$, l'équation d'Euler-Lagrange nous donne en dimension 2,

$$\begin{cases} -\alpha \Delta u_1 + K_\rho \star (f_x)^2 \; u_1 + K_\rho \star (f_x f_y) \; u_2 = -K_\rho \star (f_x f_t) & , \quad \text{dans } [0,T] \times D, \\ -\alpha \Delta u_2 + K_\rho \star (f_y f_x) \; u_1 + K_\rho \star (f_y)^2 \; u_2 = -K_\rho \star (f_y f_t) & , \quad \text{dans } [0,T] \times D, \\ \frac{\partial u_1}{\partial \mathbf{n}} = \frac{\partial u_2}{\partial \mathbf{n}} = 0 & , \quad \text{sur } [0,T] \times \partial D. \end{cases}$$

Dans cette dernière méthode, on a deux paramètres à choisir : ρ et α. Le premier sert à rendre plus robuste la méthode face au bruit dans f, tandis que le second, associé à la régularisation, permet de favoriser un flot optique régulier. Ainsi, une valeur élevée de α va avoir pour effet de flouter les informations proches des arêtes et une valeur faible de α ajoute trop d'informations inutiles dans les parties homogènes. L'idéal serait d'avoir α faible lorsqu'on est proche des arêtes et élevé dans les zones homogènes.

C'est pour cela qu'ont été introduites les méthodes adaptatives [16, 17, 72] du choix de α. Dans celles-ci, on suppose que le paramètre de régularisation α n'est plus constant mais constant par morceaux. On utilise la méthode des éléments finis, on a α dans $\mathbb{P}_0$ et si on note η_K l'erreur a-posteriori sur l'élément K, on pose pour α^n donné,

$$\alpha^{n+1}|_K = \max \left(\frac{\alpha^n|_K}{1 + \kappa \max \left(\frac{\eta_K}{\|\eta_K\|_\infty} - \frac{\xi}{100}, 0 \right)}, \alpha_{\text{s}} \right).$$

Le paramètre κ est entre 5 et 10 et α_{s} est un paramètre de seuil pour pas avoir une régularisation trop petite. Avec ce choix de α^{n+1}, si l'erreur relative est plus petite que $\xi\%$ on réduit $\alpha^{n+1}|_K$, sinon, on garde le même $\alpha^n|_K$. On a alors la méthode itérative suivante :

Data: f_1, f_2, N, κ, α_{s}.
Result: u.
Calcul de u^0 sur le maillage cartésien T^0 entre $f1$ et $f2$;
for n *in* $\{0, \ldots, N\}$ **do**
  Maillage adaptatif T^{n+1};
  Calcul de α^{n+1};
  Calcul de u^{n+1} avec la méthode globale-locale;
end
$u \leftarrow u^N$;

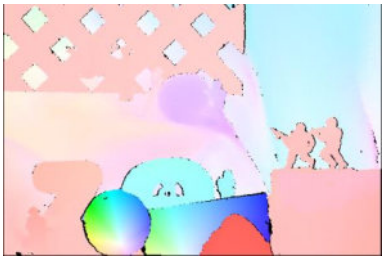
Algorithm 5: Méthodes adaptatives pour le flot optique.

Dans la pratique, on choisira α^0 constant et grand puisque le paramètre α ne peut que diminuer au cours des itérations.

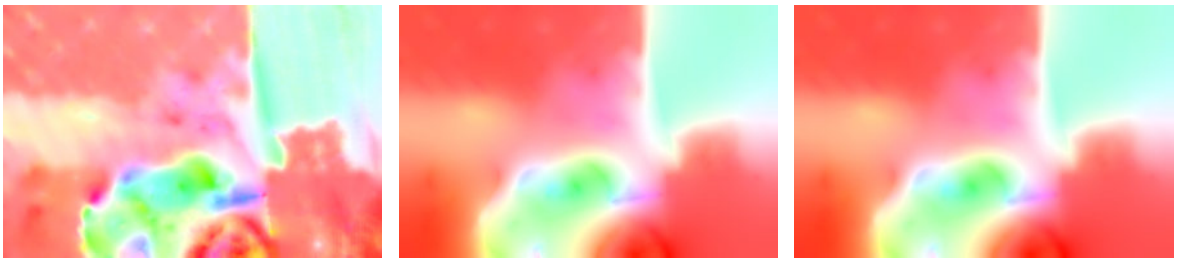
On donne dans la figure 13 quelques résultats numériques pour le flot optique. On utilise la carte de coloration pour la représentation des champs de vecteurs donnée par le site de Middleburry [10].

(a) Première image $f1$.

(b) Deuxième image $f2$.



(c) Flot optique réel [10].



(d) Méthode locale $\rho = 10$. (e) Méthode globale $\alpha = 1$. (f) Méthode globale-locale $\alpha = 1$ et $\rho = 2$.

(g) Rapprochement itératif. (h) Coarse-to-fine. (i) Contrôle adaptatif de α.

Figure 13: Quelques résultats numériques pour le flot optique avec luminosité constante.

Formulation du flot optique avec variation de la luminosité

Une approche proposée par Gennert et Negahdaripour [69] pour la déterminer le flot optique $\mathbf{u}$ est basée sur l'hypothèse que l'intensité d'un pixel varie linéairement entre deux images successives. Cette hypothèse introduit une nouvelle inconnue, à savoir la variation de la luminosité. Plus précisément, la loi de conservation est la suivante

$$f(\mathbf{x} + \delta\mathbf{x}, t + \delta t) = M(\mathbf{x}, t) f(\mathbf{x}, t) + C(\mathbf{x}, t). \tag{1}$$

Pour les petits déplacements et dans de nombreuses situations physiques, cette hypothèse est raisonnable et cohérente avec les lois de l'optique. De plus, on peut supposer que $C(\mathbf{x}, t) = 0$ et que $M(\mathbf{x}, t)$ est proche de l'identité. On écrit alors $M = I_d + \delta m$, avec I_d la matrice identité de taille d et δm, une "petite" variation de la luminosité. Sous ces nouvelles hypothèses, l'équation (1) s'écrit,

$$f(\mathbf{x} + \delta\mathbf{x}, t + \delta t) = \big(I_d + \delta m(\mathbf{x}, t)\big) f(\mathbf{x}, t).$$

En utilisant le développement de Taylor sur le terme de gauche, nous obtenons

$$\nabla f \cdot \delta\mathbf{x} + \partial_t f \delta t - f \delta m + o(t) = 0.$$

En divisant par δt et en faisant tendre δt vers zéro, nous obtenons la contrainte fondamentale du flux optique avec les variations de luminosité pour les petits déplacements, c'est-à-dire

$$\nabla f \cdot \mathbf{u} + \partial_t f - f m_t = 0,$$

with $u_i := \lim_{\delta t \to 0} \frac{\delta x_i}{\delta t}$, $m_t := \lim_{\delta t \to 0} \frac{\delta m}{\delta t}$ et $c_t := \lim_{\delta t \to 0} \frac{\delta c}{\delta t}$. Comme pour dans le cas de la luminosité constante, nous devons ajouter des contraintes à l'équation fondamentale du flot optique en régularisant les inconnues u_1, u_2, m_t et c_t. De la même façon que dans [72], on peut supposer que le coefficient de translation c_t s'annule et utiliser la méthode *globale-locale* comme seconde contrainte [27]. On cherche alors à minimiser $(\mathbf{u}, m_t)$ pour $t \in [0, 1 - \delta t[$,

$$\int_D K_\rho(\mathbf{x}) \star \Big(\nabla f \cdot \mathbf{u} + \partial_t f - f m_t\Big)\, dx + \alpha \sum_{i=1}^{d} \int_D |\nabla u_i|^2 \, dx + \lambda \int_D |\nabla m_t|^2 \, dx.$$

En dimension 2, les équations d'Euler-Lagrange nous donnent,

$$\begin{cases} -\alpha\Delta u_1 + K_\rho \star (f_x)^2\ u_1 + K_\rho \star (f_x f_y)\ u_2 - K_\rho \star (f_x f)\ m_t = -K_\rho \star (f_x f_t), & \text{dans } [0,1]\times D, \\ -\alpha\Delta u_2 + K_\rho \star (f_y f_x)\ u_1 + K_\rho \star (f_y)^2\ u_2 - K_\rho \star (f_y f)\ m_t = -K_\rho \star (f_y f_t), & \text{dans } [0,1]\times D, \\ -\lambda\Delta m_t + K_\rho \star (f f_x)\ u_1 + K_\rho \star (f f_y)\ u_2 - K_\rho \star (f^2)\ m_t = K_\rho \star (f f_t), & \text{dans } [0,1]\times D, \\ \frac{\partial u_1}{\partial \mathbf{n}} = \frac{\partial u_2}{\partial \mathbf{n}} = \frac{\partial m_t}{\partial \mathbf{n}} = 0, & \text{sur } [0,1]\times \partial D. \end{cases}$$

Cette méthode populaire pour déterminer le flot optique utilise une régularisation via le laplacien sur les composantes de $\mathbf{u}$ et sur m_t ainsi que le lissage des données sur f. Cela conduit à des solutions homogènes et denses qui peuvent contenir trop d'informations inutiles.

Une autre possibilité pour modéliser la variation de la luminosité est de considérer la constance du gradient de la luminosité [26],

$$\nabla f(\mathbf{x} + \delta\mathbf{x}, t + \delta t) = \nabla f(\mathbf{x}, t).$$

Il existe d'autres contraintes fondamentales et d'autres régularisations, présentées dans [124].

(a) Optical flow $\mathbf{u}$. (b) Variations m_t. (c) Reconstruction x.

Figure 14: Flot optique avec variation de la luminosité ($\rho = 0.4$, $\alpha = 0.1$ et $\lambda = 0.5$).

On peut également considérer le cas des grands déplacements [72]. Puisqu'on ne peut plus utiliser un développement de Taylor comme dans le cas des petits déplacements, on a un problème de minimisation non-linéaire,

$$\min_{\mathbf{u}} \underbrace{\int_D K_\rho \star \big(f(\mathbf{x}+\mathbf{u}, t+\delta t) - f(\mathbf{x}, t)\big)^2}_{\text{non-linéaire}} + \alpha\,\mathcal{R}(\mathbf{u}).$$

Organisation de la partie II

Dans le chapitre 4, nous proposons un modèle de flot optique basé sur la mécanique des fluides en utilisant l'équation de continuité et nous faisons un parallèle entre ce nouveau modèle et le problème de transport optimal grâce au théorème de Benamou-Brenier. Notre formulation inclut l'hypothèse de la variation de la luminosité sans qu'il soit nécessaire de supposer d'autres hypothèses afin de régulariser le problème du flux optique puisque l'inconnue supplémentaire, à savoir la variation de la luminosité. Contrairement à l'approche classique, la variation de la luminosité est donnée par la divergence du vecteur flot optique et traduit la "variation de volume" lors du transport de densités (intensité de l'image). Celà donne une interprétation "naturelle" de cette variation proche de ce que suggère les lois de l'optique.

La comparaison des approches classiques et de celle proposée ici nous permet de tirer les premières conclusions et perspectives pour la modélisation et l'analyse de mouvement en vision par ordinateurs.

Part I

PDE-Inpainting based Image Compression

Image Compression by Inpainting: State-of-the-Art

Contents

The use of partial differential equations (PDEs) for image compression is a relatively recent area of research that has been gaining popularity in recent years. However, most of these techniques are actually coupled with more traditional compression methods such as JPEG [138], JPEG 2000 [155] or wavelet transform [21]. Indeed, the EDPs are rather used to carry out a filtering of the image before or after having compressed it with these codecs. One can quote for example the image smoothing or the image denoising [40, 113, 160, 141, 2]. The goal of image compression by inpainting is to reduce the amount of data needed to store the image, using only methods based on partial differential equations. The main objective of this field of research is to find first of all which inpainting methods to use according to the characteristics of the image to be compressed (textured images, noisy images, cartoons, ...), but also to know which are the "good" pixels to keep in order to have a satisfactory reconstruction [52, 146, 15, 106]. Recently, these techniques have been applied to signals other than images, as for audio compression [130]. In this chapter, we will make a non-exhaustive list of image inpainting compression methods. In particular, there are methods using machine learning which will not be detailed here [148, 133].

Image Inpainting

The technique of inpainting originates from the restoration of old paintings where some parts of the paintings are missing. These are repainted with the help of the surrounding colors and patterns. In image analysis, this method is also used to restore an image when it is deteriorated, but also to remove certain elements of the image [20, 146]. Inpainting is an ill-posed problem in Hadamard's sense. Let K be a subset of D in which the image to be processed is not deteriorated. We call K the inpainting *mask* and $D \setminus K$ the inpainting domain. The general model of image inpainting is the following:

Problem. *Find u in V such that,*

$$\begin{cases} A(u) = 0, & \text{in } D \setminus K, \\ u = f, & \text{in } K, \\ \frac{\partial u}{\partial \mathbf{n}} = 0, & \text{on } \partial D, \end{cases} \tag{2}$$

where $A : V \to \mathbb{R}$ is a given operator. In the following, we will see an example of an operator for inpainting. Other types of operators exist, such as the case where A is a neural network [174, 125], but we will focus in the following on inpainting methods by PDEs.

Example 4 (Homogeneous diffusion). We give the example of homogeneous diffusion inpainting [171], stated as follows:

> *Problem* 0.0.5. *Find u in $H^1(D)$ such that,*
>
> $$\begin{cases} -\Delta u = 0, & \text{in } D \setminus K, \\ u = f, & \text{in } K, \\ \frac{\partial u}{\partial \mathbf{n}} = 0, & \text{on } \partial D. \end{cases}$$

This inpainting is effective in reconstructing the homogeneous areas of the image, but it blurs the edges of the image.

Compression by the B-Tree Algorithm

In 1997, Distasi *et al* [53] proposed an algorithm allowing the decomposition of an image in triangles of increasingly small size. Once this decomposition done, they saved the value of the pixels at the vertices of each triangle. This information, coupled with a linear interpolation, made it possible to compress and decode an image in a faster way than the principal methods used until then. Moreover, since the slicing can be represented as a binary tree, this method is also extremely simple to save.

In 2005, Galić *et al* [66] proposed to replace the simple linear interpolation by an interpolation performed by inpainting, for example by edge-enhancing diffusion inpainting [169].

Finally, in 2014, Schmaltz *et al* [146] proposed to replace the triangle division with a rectangle division and determined that the best interpolation points for each rectangle to be saved are the four corners plus the center. In addition, they compared different interpolation methods (i.e. inpainting) and, among those tested, the edge-enhancing diffusion seemed to be the most efficient.

Compression by Shape Optimization

Let D be a subset of $\mathbb{R}^2$ the support of a two-dimensional grayscale image f. For a given inpainting method, the goal of image compression by inpainting is to find an inpainting mask $K \subset \mathbb{R}^2$ so as to minimize a cost function $\mathcal{F} : V \to \mathbb{R}$ having as argument the solution u_K of (2). Moreover, since we wish to do compression, it is necessary to add a constraint on K in order to limit its size under the form $m(K) \leq c$ where m is a well chosen measure of K and $c > 0$. The inpainting image compression problem can then be written as the following minimization problem,

$$\min_{K \subset D,\ m(K) \leq c} \{\mathcal{F}(u_K) \mid u_K \text{ solution de (2)}\}.$$

This is an ill-posed inverse problem. This is why it is common to write the cost function as the sum of a data fidelity term $\mathcal{Q}$ and a regularization term $\mathcal{R}$, with regularization parameter $\alpha > 0$,

$$\mathcal{F} := \mathcal{Q} + \alpha\, \mathcal{R}.$$

We can generalize this model for R^N, $N \in \mathbb{N}^*$ signals. In the paper [15], the authors studied the case where the inpainting problem is the homogeneous diffusion and where,

$$\mathcal{F}(u) := \frac{1}{2} \int_D |\nabla u - \nabla f|^2,\ \forall u \in H^1(D).$$

The analysis of this shape optimization problem allows to find the optimal location of the points to keep, namely, where the color varies the most, i.e. the edges of the image. Two methods have been proposed, the first one where only the points that maximize $|\Delta f|$ are kept, and the second, where the density of the kept pixels increases with $|\Delta f|$.

Compression by Probabilistic Algorithms

In 2012, Mainberger *et al* proposed a probabilistic algorithm for pixel selection that he named the *sparsification* algorithm [110]. We start with a mask containing all the pixels of the image, then at each iteration, we randomly remove a fraction of the pixels of the mask, we inpaint, we calculate the error only at each pixel removed and we put in the mask a subset of the pixels removed with the largest error. Thus, the least significant pixels are removed little by little until the desired pixel density remains.

The disadvantage of the method presented above is that once a pixel is removed from the mask, it will never be returned even if it could be significant for the reconstruction. Mainberger *et al* proposed in the same article the algorithm of *nonlocal pixel exchange* [110, 131]. This one represents a post-optimization step within the iterations of the *sparsification* algorithm. In each iteration, a fixed amount of out-of-mask pixels are randomly selected from a set of candidates. A subset of those with the largest inpainting error is swapped with mask pixels also chosen at random. If the inpainting error for the new mask does not decrease, we reset the mask to its previous configuration. Otherwise, we keep the new mask.

In 2016, Peter *et al* proposed the *densification* algorithm [134]. We start with an empty mask and, for each iteration, we randomly select α pixels that do not belong to the mask. We then add the single pixel that improves the global reconstruction error the most compared to the image.

Improvements

In 2010, Bae *et al* proposed in [9] the *interpolation swapping*. This is an additional inpainting step where the role of known and unknown pixels is swapped. That is, once the reconstruction u_K^f is found, we compute $u_{D\setminus K}^{u_K^f}$. This results in a smoother final reconstruction.

In 2012, Mainberger *et al* [110] proposed to improve image reconstruction by changing the value of pixels kept in a fixed inpainting mask. This is what they called *tonal optimization*. More precisely, if we note the reconstruction u_K^f by inpainting, we pose,

$$\delta f := \operatorname{argmin}_{\delta f} \|f - u_K^{f+\delta f}\|^2.$$

Then, we save $(f + \delta f)|_K$ as a data of the compression.

In 2021, Chizhov *et al* [43] used finite elements rather than finite differences to solve the inpainting problem. This allows on the one hand to reduce the number of unknowns during the discretization, which induces a reduced computation time, but also to have a more regular and thus more visually pleasing reconstruction.

Organization of this Part

In the Chapter 1, we will introduce models based on shape optimization to find the best interpolation data in image compression with noise. We will consider a continuous stationary PDE model, obtained by focusing on the first iteration of the discretized time-dependent PDE which we will analyze in the Γ-convergence framework. We will obtain pointwise information about the relevance of each pixel by an asymptotic analysis. Then, we will introduce a finite dimensional framework of the continuous model based on balls with positive radius, and we will study by Γ-convergence the asymptotics when the radius tends to zero. In Chapter 2, we will extend the results obtained to a parabolic problem where the set of stored pixels depends on time. We will consider the semi-discrete dynamical system associated with the model which gives rise to an iterative method where the stored data is modified during the iterations to obtain the best results. We will perform the analysis and derive several iterative algorithms that we will implement and compare to obtain the most efficient compression and inpainting strategies for noisy images. We will present some numerical calculations to confirm the theoretical results. Finally, we will propose a modified model that allows the inpainting data to change with iteration. Finally, in Chapter 3, we propose to study the image compression problem in the case where the error is more general than

the one considered in the two previous chapters. In this way, our new method, derived from the adjoint method, will be more robust to different types of noise than the previous methods.

Chapter 1

Optimal Interpolation Data for PDE-based Compression of Images with Noise

Published : Communications in Nonlinear Science and Numerical Simulation, 109 (2022), p. 106278.

Contents

Abstract

We introduce and discuss shape-based models for finding the best interpolation data in the compression of images with noise. The aim is to reconstruct missing regions by means of minimizing a data fitting term in the L^2-norm between the images and their reconstructed counterparts using time-dependent PDE inpainting. We analyze the proposed models in the framework of the Γ-convergence from two different points of view. First, we consider a continuous stationary PDE model, obtained by focusing on the first iteration of the discretized time-dependent PDE, and get pointwise information on the "relevance" of each pixel by a topological asymptotic method. Second, we introduce a finite dimensional setting of the continuous model based on "fat pixels" (balls with positive radius), and we study by Γ-convergence the asymptotics when the radius vanishes. Numerical computations are presented that confirm the usefulness of our theoretical findings for non-stationary PDE-based image compression.

Keywords image compression – shape optimization – Γ-convergence – image interpolation – inpainting – PDEs – gaussian noise – image denoising

Introduction

The aim of PDE-based compression is to reconstruct a given image, by inpainting from a set of few "relevant pixels", denoted by K, with a suitable partial differential operator. The compression is a two steps process which consists of coding part, that is the choice of the set K, then the decoding phase where the image is entirely recovered. Therefore, it appears intuitively that a balance between the choice quality of the set K, with respect to constraints such as its "size", as small as possible, and the location of its pixels on one hand, and the achievable accuracy of the reconstructed image, is a major key to success of the PDE-based compression. We quote a picture from [146] which expresses nicely this idea : "PDE-based data compression suffers from poverty, but enjoys liberty [9, 15, 66, 147] : Unlike in pure inpainting research [113, 20], one has an extremely tight pixel budget for reconstructing some given image. However, one is free to choose where and how one spends this budget". Besides that, any image compression approach should take into account the nature of the considered images (e.g., noisy, textured, cartoons) and measure its impact on the selection of K.

The goal of the present chapter is to optimize the choice of such sets K and to obtain, as far as possible, an analytic criteria to build it, in the spirit of [15], but when the images are noisy. Optimizing over sets is a well-known field in shape optimization analysis, and many advanced theories and analytic works have been developed for various kinds of constraints on shapes and on differential operators. Our approach fits under this general framework and show the deep links between this field and the mathematical image analysis.

We emphasize that a comprehensive and satisfactory treatment of PDE-based compression must include both the choice of the pixels, the grey (or color) values stored and that of the inpainting operator. Actually, we know from several previous works that (e.g., [65, 160, 23, 66, 15, 147, 9]):

- Optimal sets, seen as optimal shapes, are not exhaustive with respect to all constraints that might be suited for image compression (e.g., easy storage, sparsity).
- The stability of an "optimal" set with respect to some perturbations, when it holds, is rather weak, as it requires topologies of convergence of sets. In particular, this stability is under investigated for the case of noisy data or when some stored values are changed.
- An optimal set, in the sens of optimal shape, is highly dependent on the inpainting operator, whereas a "good" operator may compensate a sub-optimal choice of pixels.

Nevertheless, finding an analytic optimal set remains in our opinion a very reasonable objective to enforce PDE-based compression methods.

Related Works

Several works, notably in the field of PDE image compression, were undertaken to optimize the choice of pixels to store in the coding phase to ensure high reconstruction quality with as few as possible selected points, we refer the reader to [9] and the references therein. In particular, in [15] the authors studied the choices of "the best set" of pixels as finding an optimal shape minimizing the semi-norm H^1 between the reconstructed solution and the initial noiseless image. They obtain such optimal shape in the framework of Γ-convergence approach and they give an analytic expression to build it from topological asymptotics. In [83], the authors introduced a mix of probabilistic and PDE based approach to deal with both finding optimal pixels and tonal data for discrete homogeneous harmonic inpainting. Loosely summarized, they start with a data sparsification step which consists of selecting randomly a set of pixels, then they correct this choice within an iterative procedure which consists of a nonlocal exchange of pixels. Lastly, they optimize the grey values at these inpainting points by a least squares minimization method. This procedure is more complete than the first one as it consider both the choice of pixels and the grey values. Notice that for a fixed set K, the harmonic inpainting is an elliptic problem and small perturbation of the data (grey values) leads to a small perturbation on the reconstructed solution, thus, optimizing the selection of the set K appears more critical for the final outcome. Whereas, a small perturbation of the sets is only "weakly stable" (in the sense of γ-convergence of sequences of sets, see [15]). Therefore, it seems reasonable to seek a more general problem of finding an optimal set with some stability properties.

In this paper, we consider a shape-based analysis taking into account noisy data. We study and analyze the problem of finding a fixed set K for the time harmonic linear diffusion, extending this way

the approach of [15]. We obtain some selection criteria which are suited to the noise level. We compare different methods proposed and existing in the related literature in the presence of noise.

Let us now give a mathematical formulation of the problem considered. Let $D \subset \mathbb{R}^2$ the support of an image (say a rectangle) and $f : D \longrightarrow \mathbb{R}$, an image which is assumed to be known only on some region $K \subset D$. There are several PDE models to interpolate f and give an approximation of the missing data. One of the basic ways is to approach $f|_{D\setminus K}$ by the solution of the heat equation, having the Dirichlet boundary data $f|_K$ on K and homogeneous Neumann boundary conditions on ∂D, i.e. to solve

Problem 1.0.1. *For $t > 0$, find $u(t,\cdot)$ in $H^1(D)$ such that*

$$\begin{cases} \partial_t u(t,\cdot) - \Delta u(t,\cdot) = 0, & \text{in } D \setminus K, \\ u(t,\cdot) = f, & \text{in } K, \\ \frac{\partial u(t,\cdot)}{\partial \mathbf{n}} = 0, & \text{on } \partial D, \end{cases} \tag{1.1}$$

$$u(0,\cdot) = u_0, \text{ in } D.$$

We assume given $f \in H^1(D)$ and $\Delta f \in L^2(D)$ with $\frac{\partial f}{\partial \mathbf{n}} = 0$, for simplicity though in practice f is a function of bounded variations with a non trivial jump set. In fact, the whole analysis in the paper extends to the case of $f \in L^2$.

To ensure the compatibility conditions with the non-homogeneous "boundary" conditions, we take $u(0,.) = f$ in D. Thus we may rewrite the problem with $v(t,x) = u(t,x) - f(x)$

Problem 1.0.2. *For $t > 0$, find $v(t,\cdot)$ in $H^1(D)$ such that*

$$\begin{cases} \partial_t v(t,\cdot) - \Delta v(t,\cdot) = \Delta f, & \text{in } D \setminus K, \\ v(t,\cdot) = 0, & \text{in } K, \\ \frac{\partial v(t,\cdot)}{\partial \mathbf{n}} = 0, & \text{on } \partial D, \end{cases} \tag{1.2}$$

$$v(0,\cdot) = 0, \text{ in } D.$$

Denoting by $v_K = u_K - f$ the solution of Problem 1.0.2, the question is to identify the region K which gives the "best" approximation u_K, in a suitable sense, for example which minimizes some L^p or Sobolev norms, e.g. in [15]

$$\int_D |\nabla u_K - \nabla f|^2 \, dx,$$

(associated to a harmonic interpolation of f in $D \setminus K$). As we want to take into account noisy images, and at the same time to perform the inpainting with denoising, a better choice a priori is to minimize the L^p-norms of $u_K - f$ and its gradient, particularly for $p = 1$ and $p = 2$, known to be good filters for a large class of noises. In this chapter, we restrict ourselves to linear time harmonic reconstruction, thus we only consider the L^p-norm, $p = 1$ or $p = 2$ for the data term. The choice of the set K, that is to say the coding part, being performed at the first step, We associate a semi-implicit discrete system to solve Problem 1.0.2. Omitting the indices $n \in \mathbb{N}$ and looking for the set K at the first iteration, with the initial condition $v_0 = u_0 - f = 0$ in D, we are led to consider the elliptic equation :

Problem 1.0.3. *Find u in $H^1(D)$ such that*

$$\begin{cases} u - \alpha \Delta u = f, & \text{in } D \setminus K, \\ u = f, & \text{in } K, \\ \frac{\partial u}{\partial \mathbf{n}} = 0, & \text{on } \partial D. \end{cases} \tag{1.3}$$

for $\alpha = \delta t$, the time step, or equivalently

Problem 1.0.4. *Find v in $H^1(D)$ such that*

$$\begin{cases} v - \alpha\Delta v = \alpha\Delta f, & \text{in } D \setminus K, \\ v = 0, & \text{in } K, \\ \frac{\partial v}{\partial \mathbf{n}} = 0, & \text{on } \partial D. \end{cases} \tag{1.4}$$

Thus, finding the set of pixels which gives the best approximation u_K in the L^2 sense is a shape analysis problem for the state equation of Problem 1.0.2 (i.e. Problem 1.0.3). We recall that if u_K is the solution of Problem 1.0.3, then u_K is the minimizer of

$$\min_{u\in H^1(D),\ u=f \text{ in } K} \frac{1}{2}\int_D (u-f)^2\ dx + \frac{\alpha}{2}\int_D |\nabla(u-f)|^2\ dx - \alpha\int_D \Delta f\,(u-f)\ dx,$$

which is equivalent to

$$\min_{u\in H^1(D),\ u=f \text{ in } K} \frac{1}{2}\int_D (u-f)^2\ dx + \frac{\alpha}{2}\int_D |\nabla u|^2\ dx.$$

Following [15], we develop two directions of finding that optimal set. The first is to set a continuous PDE model and search pointwise information by a topological asymptotic method. The second direction is to simulate in the continuous frame a finite dimensional shape optimization problem by imposing K to be the union of a finite number of "fat pixels". Performing the asymptotic analysis by Γ-convergence when the number of pixels is increasing (in the same time that the fatness vanishes), we obtain useful information about the optimal distribution of the best interpolation pixels.

Organization of the Chapter

In Section 1.1, we introduce a mathematical model of the compression problem and its relaxed formulation. In Section 1.2, we compute the topological gradient of our minimization problem in order to find a mathematical criterion to construct our set of interpolation points. In Section 1.3, we change our point of view, by considering "fat pixels" instead of a general set of interpolation points. Finally, in Section 1.4, we expose some numerical results.

1.1 The Continuous Model

1.1.1 Min-max Formulation

Let D be a bounded open subset of $\mathbb{R}^2$. We consider the shape optimization problem

$$\min_{K\subseteq D, m(K)\leq c} \{\mathcal{E}_p(u_K) \mid u_K \text{ solution of Problem 1.0.3}\}, \tag{1.5}$$

where $\mathcal{E}_p$, is defined by

$$\mathcal{E}_p(u) = \frac{1}{p}\int_D |u-f|^p\ dx + \frac{\alpha}{2}\int_D |\nabla(u-f)|^2\ dx,\ \forall u \in H^1(D)\cap L^p(D), \tag{1.6}$$

m is a measure, to be chosen, and $c > 0$. We notice that (1.6) as a cost functional corresponds to the L^p data fitting term with Tikhonov regularization [157]. Hence $\mathcal{E}_p$ is the simplest, and widely used, denoising PDE models at least for the values $p = 1$ or $p = 2$. The image compression problem aims to find an optimal set of pixels from which an accurate reconstruction of (noisy) image will be performed. Actually, the data term does not affect the Γ-convergence analysis, so we only consider the case $p = 2$ and we drop the index 2 by denoting $\mathcal{E}$ the energy. Thanks to Proposition C.1.1 in Appendix C, the analysis of the continuous model is similar to the H^1-semi norm case in [15], we give the main analysis result and the main steps of the proof in the next section.

Let u_K be the solution of Problem 1.0.3, it is straightforward to obtain

Proposition 1.1.1. *The optimization problem* (1.5) *is equivalent to*

$$\max_{K\subseteq D, m(K)\leq c} \min_{u\in H^1(D), u=f \text{ in } K} \frac{\alpha}{2}\int_D |\nabla u|^2 \, dx + \frac{1}{2}\int_D (u-f)^2 \, dx.$$

Proof. The weak formulation of Problem 1.0.3 with $u_K - f$ as test function gives,

$$\alpha\int_D |\nabla u_K|^2 \, dx + \int_D (u_K - f)^2 \, dx = \alpha\int_D \nabla u_K \cdot \nabla f \, dx.$$

Thus,

$$\begin{aligned}
(1.5) &\Leftrightarrow \min_{K\subseteq D, m(K)\leq c} \frac{1}{2}\int_D (u_K-f)^2 \, dx + \frac{\alpha}{2}\int_D |\nabla(u_K-f)|^2 \, dx \\
&\Leftrightarrow \min_{K\subseteq D, m(K)\leq c} \frac{1}{2}\int_D (u_K-f)^2 \, dx + \frac{\alpha}{2}\int_D |\nabla u_K|^2 \, dx - \alpha\int_D \nabla u_K \cdot \nabla f \, dx \\
&\Leftrightarrow \min_{K\subseteq D, m(K)\leq c} \frac{1}{2}\int_D (u_K-f)^2 \, dx + \frac{\alpha}{2}\int_D |\nabla u_K|^2 \, dx - \alpha\int_D |\nabla u_K|^2 \, dx - \int_D (u_K-f)^2 \, dx \\
&\Leftrightarrow \min_{K\subseteq D, m(K)\leq c} -\frac{1}{2}\int_D (u_K-f)^2 \, dx - \frac{\alpha}{2}\int_D |\nabla u_K|^2 \, dx \\
&\Leftrightarrow \max_{K\subseteq D, m(K)\leq c} \frac{1}{2}\int_D (u_K-f)^2 \, dx + \frac{\alpha}{2}\int_D |\nabla u_K|^2 \, dx.
\end{aligned}$$

Since a is symmetric, by using Lax-Milgram theorem [57], we have the result. □

Finally, problem (1.5) can be rewritten under the unconstrained form by penalizing the constraint $m(K) \leq c$ as follows :

$$\max_{K\subseteq D} \min_{u\in H^1(D), u=f \text{ in } K} \frac{\alpha}{2}\int_D |\nabla u|^2 \, dx + \frac{1}{2}\int_D (u-f)^2 \, dx - \beta m(K), \tag{1.7}$$

for $\beta > 0$. The well-posedness of (1.5) depends of the choice of the measure m. In [15], it has been proven that, in the laplacian case, choosing the ν-capacity as measure m leads to the existence of a relaxed formulation and the well-posedness of this optimization problem. Consequently, we will study (1.5) when m is the ν-capacity. The next section is devoted to the analysis within the γ-convergence (see Appendix C) approach follows the same lines as in [15] with slight changes. Without loss of generality, we consider the capacity when $\nu = 1$.

1.1.2 Analysis of the Model

The optimization problem (1.5) can be rewritten by penalizing the Dirichlet boundary condition $u = f$ in K

$$\max_{K\subseteq D} \min_{u\in H^1(D)} \frac{\alpha}{2}\int_D |\nabla u|^2 \, dx + \frac{1}{2}\int_D (u-f)^2 \, dx + \frac{1}{2}\int_D (u-f)^2 \, d\infty_K - \beta \operatorname{cap}_\nu(K),$$

where the measure ∞_K is defined in Appendix C. It is well known that such shape optimization problems do not always have a solution (e.g. [15]), we seek a relaxed formulation, which under the capacity constraint yields a relaxed solution, that is to say, a capacity measure (see Appendix C). Thus, we consider the problem

$$\max_{\mu\in\mathcal{M}_0(D)} \min_{u\in H^1(D)} \frac{\alpha}{2}\int_D |\nabla u|^2 \, dx + \frac{1}{2}\int_D (u-f)^2 \, dx + \frac{1}{2}\int_D (u-f)^2 \, d\mu - \beta \operatorname{cap}_\nu(\mu),$$

where μ is in $\mathcal{M}_0(D)$. As the L^2- norm is continuous, referring to Proposition C.1.1 in Appendix C, we may drop from the following Γ-convergence analysis, the term

$$\frac{1}{2}\int_D (u-f)^2 \, dx.$$

For every μ in $\mathcal{M}_0(D)$ and u in $H^1(D)$, we define F_μ, from $H^1(D)$ into $\mathbb{R} \cup \{+\infty\}$, by

$$F_\mu(u) := \begin{cases} \alpha \int_D |\nabla u|^2 \, dx + \int_D (u-f)^2 \, d\mu, & \text{if } |u| \leq |f|_\infty, \\ +\infty, & \text{otherwise.} \end{cases}$$

We have that F_μ is equi-coercive with respect to μ, for any μ in $\mathcal{M}_0(D)$. Indeed, let u be in $H^1(D)$ such that $|u|_\infty \leq |f|_\infty$. It is easy to see that

$$F_\mu(u) \geq \alpha \int_D |\nabla u|^2 \, dx - 2|f|_\infty^2 \mu(D).$$

For every μ in $\mathcal{M}_0(D)$, we define E, from $\mathcal{M}_0(D)$ into $\mathbb{R}$, by

$$E(\mu) := \min_{u \in H^1(D)} F_\mu(u).$$

For a given μ in $\mathcal{M}_0(D)$, $E(\mu)$ corresponds to the energy of

Problem 1.1.1. *Find u in $H^1(D)$ such that*

$$\begin{cases} -\alpha \Delta u + \mu(u-f) = 0, & \text{in } D, \\ \frac{\partial u}{\partial \mathbf{n}} = 0, & \text{on } \partial D. \end{cases} \tag{1.8}$$

Thus, if u is a solution of Problem 1.1.1 for a given μ in $\mathcal{M}_0(D)$, then $F_\mu(u) = E(\mu)$ and the function u satisfies the maximum principle $|u| \leq |f|_\infty$. The existence of the relaxed optimal shape follows the same lines as in [15]. We sketch the proof of Theorem 1.1.4 for the existence of the relaxed solution, for details see in Appendix A.1. Since we want to include balls centered at points x_0 in D, that we do not want to be too close to the boundary of D, we introduce the following notations for $\delta > 0$,

$$D^{-\delta} := \{x \in D \mid d(x, \partial D) \geq \delta\} \subseteq D,$$

$$\mathcal{K}^\delta(D) := \{K \subseteq D \mid K \text{ closed}, \ K \subseteq D^{-\delta}\},$$

and

$$\mathcal{M}_0^\delta(D) := \{\mu \in \mathcal{M}_0(D) \mid \mu_{|D \setminus D^{-\delta}} = 0\} \subseteq \mathcal{M}_0(D).$$

Let us consider the problem

$$\max_{\mu \in \mathcal{M}_0^\delta(D)} \min_{u \in H^1(D)} \frac{\alpha}{2} \int_D |\nabla u|^2 \, dx + \frac{1}{2} \int_D (u-f)^2 \, d\mu - \beta \, \mathrm{cap}(\mu).$$

Using the compactness of $\mathcal{M}_0(D)$ for the γ-convergence (Proposition C.5.4 in Appendix C) and the locality of the $\gamma(F)$-convergence (Proposition C.5.5 in Appendix C), we have the following result

Proposition 1.1.2 (γ-compactness of $\mathcal{M}_0^\delta(D)$)**.** *The set $\mathcal{M}_0^\delta(D)$ defined above is compact with respect to the γ-convergence.*

We have also the density theorem (the proof is given in Appendix A.1)

Theorem 1.1.1. *We have*

$$\mathrm{cl}_\gamma \, \mathcal{K}_\delta(D) = \mathcal{M}_0^\delta(D),$$

i.e., $\mathcal{K}_\delta(D)$ is dense into $\mathcal{M}_0^\delta(D)$ with respect to the $\gamma(F)$-convergence.

Similarly to Lemma 3.4 in [15], we have

Theorem 1.1.2. *Let $\mu_n \in \mathcal{K}_\delta(D)$. If μ_n γ-converge to μ, then $\mathrm{cap}_\nu(\mu_n) \to \mathrm{cap}_\nu(\mu)$.*

Theorem 1.1.3. *If $(\mu_n)_n$ in $\mathcal{M}_0^\delta(D)$ γ-converges to μ, then μ is in $\mathcal{M}_0^\delta(D)$ and F_{μ_n} Γ-converges to F_μ in $L^2(D)$.*

The proof of the last theorem is also given in Appendix A.1. Finally, we can state the main result of this section (proof in Appendix A.1).

Theorem 1.1.4. *We have*

$$\sup_{K\in\mathcal{K}_\delta(D)} \big(E(\infty_K) - \beta\,\mathrm{cap}_\nu(\infty_K)\big) = \max_{\mu\in\mathcal{M}_0^\delta(D)} \big(E(\mu) - \beta\,\mathrm{cap}_\nu(\mu)\big).$$

Replacing F_n with $F_n + G$, $G := \frac{1}{2}\int_D (u-f)^2\,dx$ and with Proposition C.1.1 in Appendix C, we get the existence of an optimal solution to the relaxed formulation.

Note. In order to solve the relaxed problem

$$\min_{u\in H^1(D)} \alpha\int_D |\nabla u|^2\,dx + \int_D (u-f)^2\,dx + \int_D (u-f)^2\,d\infty_K - \beta\,\mathrm{cap}_\nu(K),$$

we may use a shape derivative with respect to the measures μ. However, such a method yields diffuse measures, thus too thick sets whereas we seek discrete sets of pixels.

In the next two sections, we aim to find an explicit characterization of the set K using topological asymptotic.

1.2 Topological Gradient

Here, we aim to compute the solution of our optimization problem (1.5) by using a topological gradient-based algorithm as in [100, 68]. The computation of the topological gradient is different than the case of a pure Laplacian problem as it uses fundamental solution of $-\Delta u + u = f$, namely the Bessel functions of second kind (see Appendix A.2). This kind of algorithm consists in starting with $K = \bar{D}$ and determining how making small holes in K affect the cost functional to find the balls which have the most decreasing effect. To this end, let us define K_ε the compact set $K \setminus B(x_0,\varepsilon)$ where $B(x_0,\varepsilon)$ is the ball centered in $x_0 \in D$ with radius $\varepsilon > 0$ such that $B(x_0,\varepsilon) \subset K$. From now, we consider the functional :

$$j^* : A \subset D \mapsto \min_{u\in H^1(D), u=f \text{ in } A} \frac{1}{2}\int_D (u-f)^2\,dx + \frac{\alpha}{2}\int_D |\nabla u|^2\,dx,$$

or equivalently,

$$j : A \subset D \mapsto \min_{v\in H^1(D), v=0 \text{ in } A} \frac{1}{2}\int_D v^2\,dx + \frac{\alpha}{2}\int_D |\nabla v|^2\,dx - \int_D gv\,dx,$$

where $g := \alpha\Delta f$. Finally, we denote by v_ε the minimizer of $j(K_\varepsilon)$. Then, we have

Proposition 1.2.1. *With notations from above, we have when ε tends to 0,*

$$j(K_\varepsilon) - j(K) = \frac{\pi}{2}\big(g(x_0)\big)^2\varepsilon^2\ln(\varepsilon) + O(\varepsilon^2).$$

Proof.

$$j(K_\varepsilon) - j(K) = \frac{\alpha}{2}\int_{B(x_0,\varepsilon)} |\nabla v_\varepsilon|^2 \, dx + \frac{1}{2}\int_{B(x_0,\varepsilon)} v_\varepsilon^2 \, dx - \int_{B(x_0,\varepsilon)} g v_\varepsilon \, dx.$$

The weak formulation of Problem 1.0.3 leads to

$$\begin{aligned} j(K_\varepsilon) - j(K) &= \frac{1}{2}\int_{B(x_0,\varepsilon)} g v_\varepsilon \, dx - \int_{B(x_0,\varepsilon)} g v_\varepsilon \, dx \\ &= -\frac{1}{2}\int_{B(x_0,\varepsilon)} g v_\varepsilon \, dx. \end{aligned}$$

We have $g(x) = g(x_0) + \|x - x_0\| O(1)$, and hence

$$j(K_\varepsilon) - j(K) = -\frac{1}{2} g(x_0) \int_{B(x_0,\varepsilon)} v_\varepsilon \, dx + \varepsilon \, O(1) \int_{B(x_0,\varepsilon)} v_\varepsilon \, dx.$$

It is enough to compute the fundamental term in the asymptotic development of the expression $\int_{B(x_0,\varepsilon)} v_\varepsilon \, dx$. This is done by using Proposition A.2.1 in Appendix A. □

Since for $\varepsilon < 1$, $\ln \varepsilon < 0$, the result above suggests to keep the points x_0 where $|\Delta f(x_0)|^2$ is maximal, when ε small enough. From a practical point of view, this is the main result of our local shape analysis. In practice, we compute the quantity $|\Delta f|$ and we apply a hard-thresholding to select the maximal points x_0 in one single step. In the next section, we will see that such a strict threshold rule might be relaxed.

1.3 Optimal Distribution of Pixels : The "Fat Pixels" Approach

In this section, we change our point of view by considering "fat pixels" instead of a general set of interpolation points. In the sequel, we will follow [15, 33]. We restrict our class of admissible sets as an union of balls which represent pixels. For $m > 0$ and $n \in \mathbb{N}$, we define

$$\mathcal{A}_{m,n} := \Big\{ \overline{D} \cap \bigcup_{i=1}^{n} \overline{B(x_i, r)} \;\Big|\; x_i \in D_r, \; r = mn^{-1/2} \Big\},$$

where D_r is the r-neighborhood of D. The following analysis remains unchanged in $\mathbb{R}^d$, but for the sake of simplicity we restrict ourselves to the case $d = 2$. We consider problem (1.5) for every $K \in \mathcal{A}_{m,n}$ i.e.

$$\min_{K \in \mathcal{A}_{m,n}} \Big\{ \frac{1}{2}\int_D (u_K - f)^2 \, dx + \frac{\alpha}{2}\int_D |\nabla u_K - \nabla f|^2 \, dx \;\Big|\; u_K \text{ solution of Problem 1.0.3} \Big\}.$$

Like in the previous section, we set $v_K := u_K - f$. This last optimization problem can be reformulated as a compliance optimization problem :

$$\min_{K \in \mathcal{A}_{m,n}} \Big\{ \frac{1}{2}\int_D g \, v_{K_n} \, dx \;\Big|\; u_K := v_K + f \text{ solution of Problem 1.0.3} \Big\}, \tag{1.9}$$

where $g := \alpha \Delta f$, like in the previous sections. Here, we do not need to specify a size constraint on our admissible domains. Indeed, imposing $K \in \mathcal{A}_{m,n}$ implies a volume constraint and a geometrical constraint on K since K is formed by a finite number of balls with radius $mn^{-1/2}$. We deal with Neumann boundary conditions on D. However, it is possible to cover the boundary with $\frac{2C_D}{m} n^{1/2}$ balls so that we have formally homogeneous Dirichlet boundary conditions on D. The well-posedness of such a problem has been studied in the Laplacian case in [33]. Without significant change we have

Theorem 1.3.1. *If D is an open bounded subset of $\mathbb{R}^2$ and if $g \geq 0$ is in $L^2(D)$, then the problem (1.9) admits a unique solution.*

If we denote by K_n^{opt} the solution, then we have that $\infty_{K_n^{\text{opt}}}$ γ-converge to ∞_D as n tends to $+\infty$. However, the number of pixels x_0 in D to keep goes also to infinity. Thus, it gives no relevant information on the distribution of the points to retain. As pointed out in [28], the local density of K_n^{opt} can be obtained by using a different topology for the Γ-convergence of the rescaled energies. In this new frame, the minimizers are unchanged but their behavior is seen from a different point of view. We define the probability measure μ_K for a given set K in $\mathcal{A}_{m,n}$ by

$$\mu_K := \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i}.$$

We define the functional F_n from $\mathcal{P}(\bar{D})$ into $[0,+\infty]$ by

$$F_n(\mu) := \begin{cases} n\int_D g v_K \, dx, & \text{if } \exists K \in \mathcal{A}_{m,n}, \text{ s.t. } \mu = \mu_K, \\ +\infty, & \text{otherwise.} \end{cases}$$

The following Γ-convergence of F_n theorem is similar to the one given in Theorem 2.2. in [33].

Theorem 1.3.2. *If $g \geq 0$, then the sequence of functionals F_n, defined above, Γ-converge with respect to the weak $\star$ topology in $\mathcal{P}(\bar{D})$ to*

$$F(\mu) := \int_D \frac{g^2}{\mu_a}\theta(m\mu_a^{1/2}) \, dx,$$

where $\mu = \mu_a dx + \nu$ is the Radon–Nikodym–Lebesgue decomposition of μ ([62], Theorem 3.8) with respect to the Lebesgue measure and

$$\theta(m) := \inf_{K_n \in \mathcal{A}_{m,n}} \liminf_{n\to+\infty} n \int_D g v_{K_n} \, dx,$$

$v_{K_n} := u_{K_n} - f$, u_{K_n} solution of Problem 1.0.3.

As a consequence of the Γ-convergence stated in the theorem above, the empirical measure $\mu_{K_n^{\text{opt}}} \to \mu^{\text{opt}}$ weak $\star$ in $\mathcal{P}(\mathbb{R}^d)$ where μ^{opt} is a minimizer of F. Unfortunately, the function θ is not known explicitly. We establish here after that θ is positive, non-increasing and vanish after some point which will be enough for practical exploration. The next theorem gives an estimate of the function θ defined above. The proof is given in Section A.3.

Theorem 1.3.3. *We have, for m in $(0,t_1)$,*

$$C_1(\alpha)|\ln(m)| - C_2(\alpha) \leq \theta(m) \leq C_3(\alpha)|\ln(m)|,$$

where C_1, C_2 and C_3 are constants depending on α.

Note. We can extend the results above to any g since we may formally split the discussion on the sets $\{g \geq 0\}$ and $\{g < 0\}$.

These estimates on θ suggest that to minimize F, when $|g|$ is large, μ_a should be large in order for θ to be close to its vanishing point, while when $|g|$ is small μ_a could be small. Formal Euler-Lagrange equation and the estimates on θ give the following information : to minimize

$$F(\mu) := \int_D \frac{g^2}{\mu_a}\theta(m\mu_a^{1/2}) \, dx,$$

one have to take

$$\frac{\mu_a^2}{|1-\log\mu_a|} \approx c_{m,f}\, g^2.$$

This introduces a soft-thresholding with respect to the first approach. To sum up, we can choose the interpolation data such that the pixel density is increasing with $|g| = |\Delta f|$. This soft-thresholding rule can be enforced with a standard digital halftoning. According to [15, 161, 3], digital halftoning is a method of rendering that convert a continuous image to a binary image, for example black and white image, while giving the illusion of color continuity. This color continuity is simulated for the human eye by a spacial distribution of black and white pixels. Two different kinds of halftoning algorithms exist : dithering and error diffusion halftoning. The first one is based on a so-called dithering mask function, while the other one is an algorithm which propagate the error between the new value (0 or 1) and the old one (in the interval $[0, 1]$). An ideal digital halftoning method would conserves the average value of gray while giving the illusion of color continuity.

1.4 Numerical Results

In this section, we present some numerical simulations to validate the previous theoretical analysis and we compare to other commonly used methods of image compression. We discretize the PDEs with a standard implicit finite difference scheme on a quasi-uniform mesh in order to make the comparisons easy. We have considered the method presented in this chapter that we will denote by L^2-methods, more precisely we call *L2-T* the algorithm based on hard thresholding with the criteria obtained in Section 1.2, *L2-H* the algorithm based on the fat pixels variant (soft thresholding) and each algorithm is used with, respectively without, the halftoning based on Floyd-Steinberg dithering algorithm [61]. The methods that we use for comparison purposes are the B-Tree algorithm [53] and a random mask selection. Next we discuss and present some extensions of the method in several ways : first, we allow a data modification on the compression set K to test how under the same framework and analysis the selected masks may be eventually improved. Secondly, we consider images corrupted with Salt and Pepper noise, though the L^2-norms based reconstruction are less efficient. Finally, we consider the case of color images. We will denote f the initial image, f_δ its noisy version and u the reconstructed one.

1.4.1 Numerical Simulations and Comparisons

For the L^2-methods, respectively, H^1 based methods, we implement the hard threshold criteria, namely we select the pixels where $|\Delta f|$ is maximum and the soft threshold algorithm of the fat pixels approach, where the selected pixels are chosen according to the distribution of $|\Delta f|$. The last algorithm uses a dithering procedure [61].

In Table 1.1, Table 1.2 and Table 1.3, we give the L^2-errors between the images f and u, as a function of the noise level for each method. We notice that the L^2-errors are better than with B-Tree and random choices when the noise magnitude of the data is not too high whereas it deteriorates increasingly with the noise. In fact, the locations where $|\Delta f|$ is high includes more noisy pixels which is reflected in the mask selection. This effect of taking more noisy pixels is amplified with compression ratio. We emphasize that our comparisons are only concerned with the influence of noise on the coding phase in compression and are by no means exhaustive. In particular, when the noise level is too high the criterion based on the locations where the Laplacian is maximum appears less efficient with respect to B-Tree (which include by construction an amount of denoising) or even the random choice of pixels, we will see how to improve the criterion in these cases. We can observe that the hard-thresholding method gives higher error without noise than with noisy image. This might by due to the laplacian in the mask criterium that allows a better distribution of the pixels inside the mask in the presence of noise.

In Figure 1.2, Figure 1.3 and Figure 1.4, we present various masks obtained and the corresponding reconstructed images. We notice that with no noise or low level ones, the masks consist of pixels located on, and close to, the edges which is intuitively expected. The soft threshold method includes few pixels from the homogeneous areas leading to a better reconstruction results. As the noise magnitude grows, increasingly noisy pixels are selected in the mask leading to poor reconstructions.

Noise	L2-T	L2-H	B-tree		Rand
	$\|f-u\|_2$	$\|f-u\|_2$	α	$\|f-u\|_2$	$\|f-u\|_2$
0	39.17	9.56	10000	9.88	15.02
0.03	13.47	12.14	30	10.61	15.49
0.05	17.10	15.12	70	11.80	16.13
0.1	31.43	23.98	87	15.50	19.01
0.2	75.48	41.92	68	24.87	27.64

Table 1.1: L^2-error between the original image f and the reconstruction u (build from f_δ) with 5% of total pixels saved.

Noise	L2-T	L2-H	B-tree		Rand
	$\|f-u\|_2$	$\|f-u\|_2$	α	$\|f-u\|_2$	$\|f-u\|_2$
0	25.57	4.92	10000	6.44	10.94
0.03	9.36	8.59	80	7.56	11.85
0.05	13.91	12.66	62	9.57	13.14
0.1	27.39	23.19	56	15.16	16.94
0.2	61.46	43.29	51	26.73	27.89

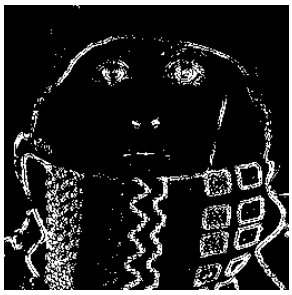
Table 1.2: L^2-error between the original image f and the reconstruction u with 10% of total pixels saved.

Noise	L2-T	L2-H	B-tree		Rand
	$\|f-u\|_2$	$\|f-u\|_2$	α	$\|f-u\|_2$	$\|f-u\|_2$
0	18.21	3.32	10000	4.55	9.03
0.03	8.21	7.67	51	6.54	9.93
0.05	13.05	12.07	15	8.91	11.35
0.1	25.91	23.11	22	15.36	16.78
0.2	54.66	43.34	12	28.20	28.77

Table 1.3: L^2-error between the original image f and the reconstruction u with 15% of total pixels saved.

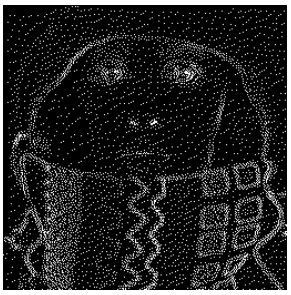
(a) Input image. (b) $\sigma = 0.03$, $\|f - f_\delta\|_2 = 7.65$.(c) $\sigma = 0.05$, $\|f - f_\delta\|_2 = 12.81$.(d) $\sigma = 0.1$, $\|f - f_\delta\|_2 = 25.45$.

Figure 1.1: Input images with and without gaussian noise of standard deviation σ.



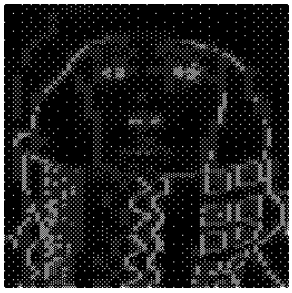
(a) Mask with L2-T method.

(b) Reconstruction with L2-T method.



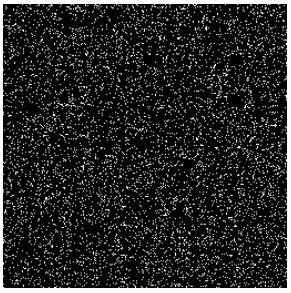
(c) Mask with L2-H method.

(d) Reconstruction with L2-H method.



(e) Mask with B-TREE method.

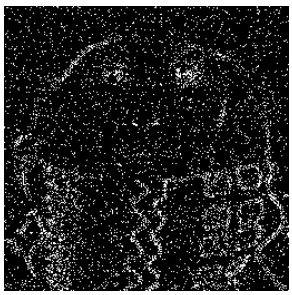
(f) Reconstruction with B-TREE method.



(g) Mask with RAND method.

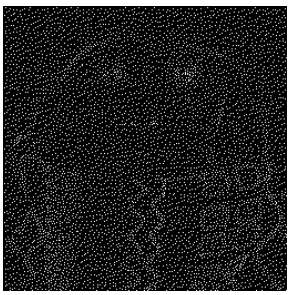
(h) Reconstruction with RAND method.

Figure 1.2: Masks and reconstructions for Table 1.2 when the input image is noiseless ($\sigma = 0$).



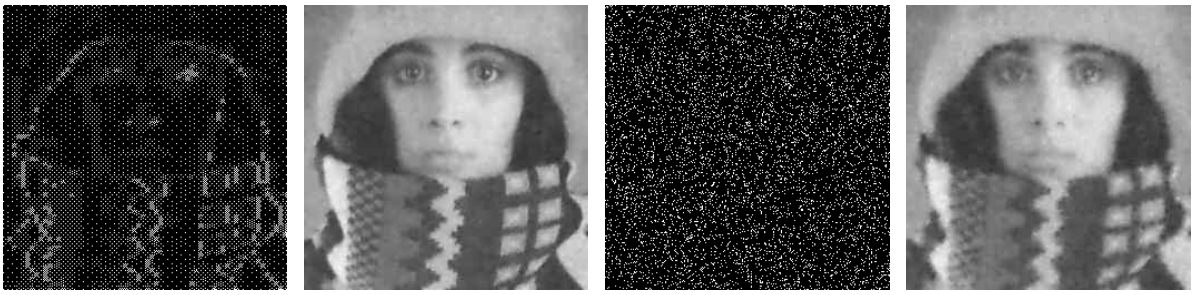
(a) Mask with L2-T method.

(b) Reconstruction with L2-T method.



(c) Mask with L2-H method.

(d) Reconstruction with L2-H method.



(e) Mask with B-TREE method. (f) Reconstruction with B-TREE method. (g) Mask with RAND method. (h) Reconstruction with RAND method.

Figure 1.3: Masks and reconstructions for Table 1.2 when the input image is affected by gaussian noise ($\sigma = 0.03$).

(a) Mask with L2-T method. (b) Reconstruction with L2-T method. (c) Mask with L2-H method. (d) Reconstruction with L2-H method.

(e) Mask with B-TREE method. (f) Reconstruction with B-TREE method. (g) Mask with RAND method. (h) Reconstruction with RAND method.

Figure 1.4: Masks and reconstructions for Table 1.2 when the input image is affected by gaussian noise ($\sigma = 0.05$).

1.4.2 Comparison with B-Tree

B-Tree algorithms seem in some examples to perform slightly better in terms of the L^2-error. Actually, this is not surprising as B-Tree approach build the masks for compression by optimizing with respect to this norm. However, this is not a disadvantage to our approach when we compare with respect to some other constraints, e.g.

- the cost of the compression is higher than with our method in term of CPU time. This becomes worst with images of high resolution,

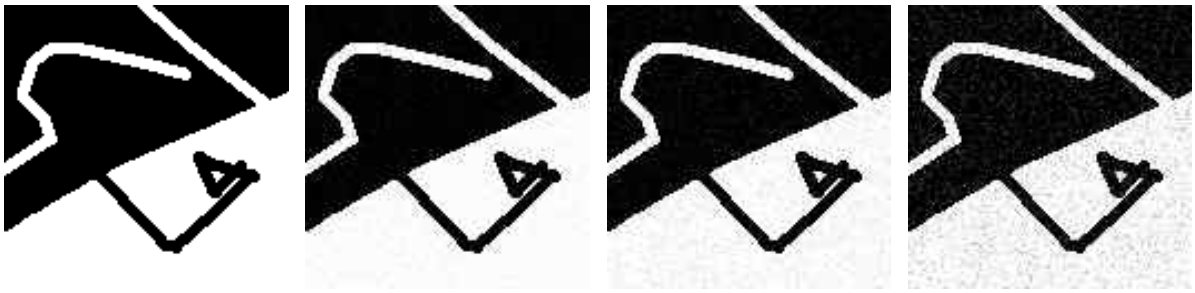
- B-Tree works only with regular grids, which is a serious limitation for images where features of importance (e.g. edges) are located outside the grid. For images with high anisotropy this shortcoming is more critical,

- in B-Tree algorithms refinement are necessary such as the choice of the parameters α which make them more image dependent.

Our approach gives an analytic criteria which allows to overcome most of these difficulties. We added more comparisons elements between the two approaches in the revised version to give a more complete image on their outcomes Table 1.4 and Figure 1.6.

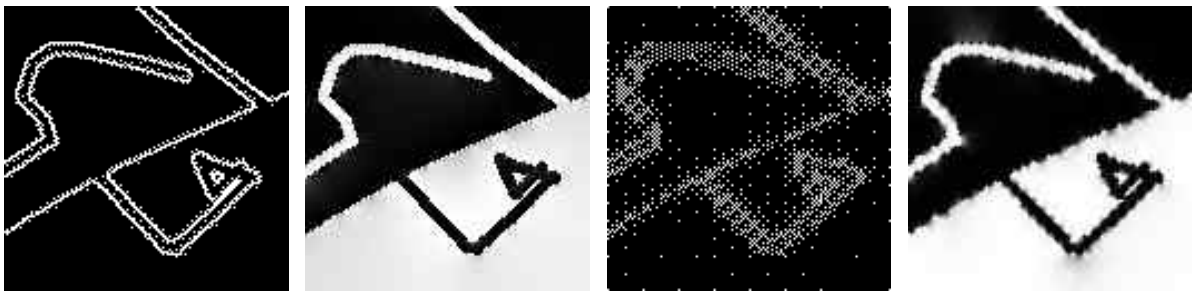
Noise	**L2-H**		**B-tree**		
	$\|f-u\|_2$	time (s)	α	$\|f-u\|_2$	time (s)
0	9.36	0.37	10000	15.65	198.23
0.03	12.19	0.38	19.59	16.16	33.92
0.05	15.91	0.38	12.20	16.42	20.58
0.1	23.26	0.39	16.57	17.80	13.71
0.2	36.30	0.37	10.00	21.85	20.70

Table 1.4: L^2-error between the original image f and the reconstruction u (build from f_δ) with 9% of total pixels saved.

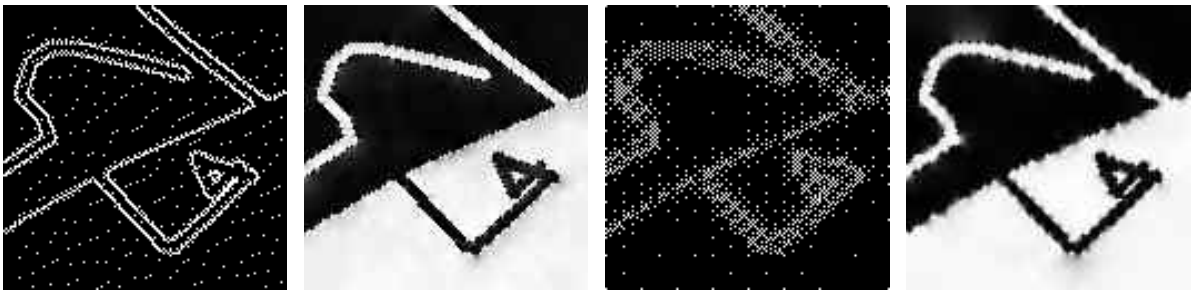


(a) Input image. (b) $\sigma = 0.03$, $\|f - f_\delta\|_2 = 2.70$.(c) $\sigma = 0.05$, $\|f - f_\delta\|_2 = 4.55$.(d) $\sigma = 0.1$, $\|f - f_\delta\|_2 = 9.17$.

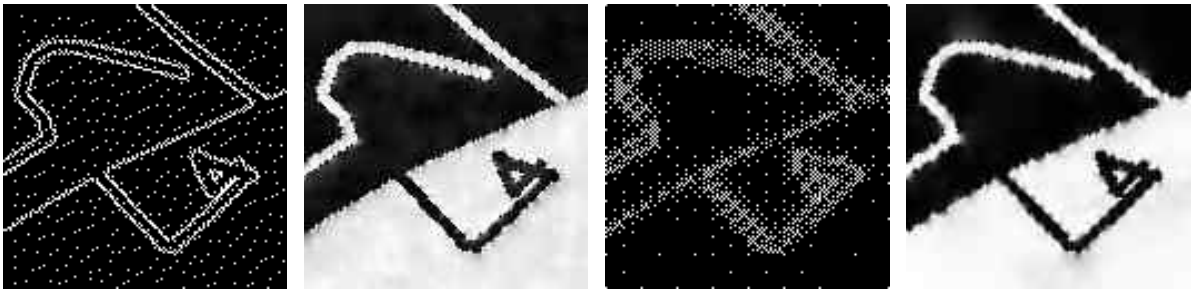
Figure 1.5: Input images with and without gaussian noise of standard deviation σ.



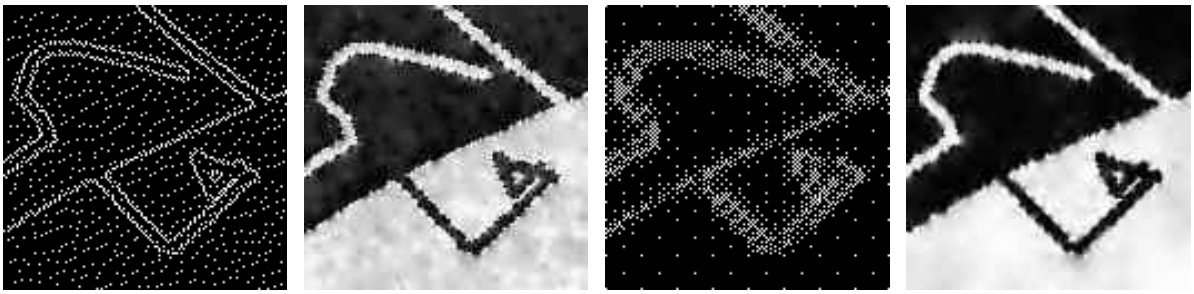
(a) Mask with L2-H method. (b) Reconstruction with L2-H method. (c) Mask with B-Tree method. (d) Reconstruction with B-Tree method.



(e) Mask with L2-H method. (f) Reconstruction with L2-H method. (g) Mask with B-Tree method. (h) Reconstruction with B-Tree method.



(i) Mask with L2-H method. (j) Reconstruction with L2-H method. (k) Mask with B-Tree method. (l) Reconstruction with B-Tree method.



(m) Mask with L2-H method. (n) Reconstruction with L2-H method. (o) Mask with B-Tree method. (p) Reconstruction with B-Tree method.

Figure 1.6: Masks and reconstructions for Table 1.4.

1.4.3 Improving the Selection Criteria

The issue raised from the above experiments and analysis is how to improve the mask selection as $|\Delta f|$ appears to be very sensitive to the noise magnitude? A way considered by [110, 83] is to resort to tonal optimization where the data is simultaneously modified on K. In this paper we investigate first a more basic idea on how to modify the criterion under the same analysis presented in the previous sections. It is clear that any change in the data on the Dirichlet condition on K will causes a modification (e.g. a correction) on the final criterion. Intuitively, taking $u = g$ on K where g is either less noisy than f or a copy of f with enhanced edges, it would lead a best pixels selection. The simplest ways to this are presented now.

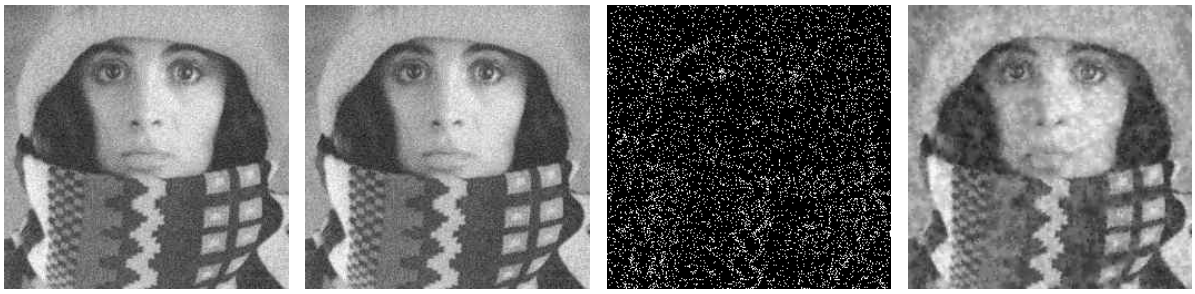
Sharpening the Edges

We replace f by $\tilde{f} := f - \beta\Delta f$, $\beta > 0$. Then the criterion become :

$$\max|\Delta\tilde{f}| \Leftrightarrow \max|\Delta f - \beta\Delta^2 f|.$$

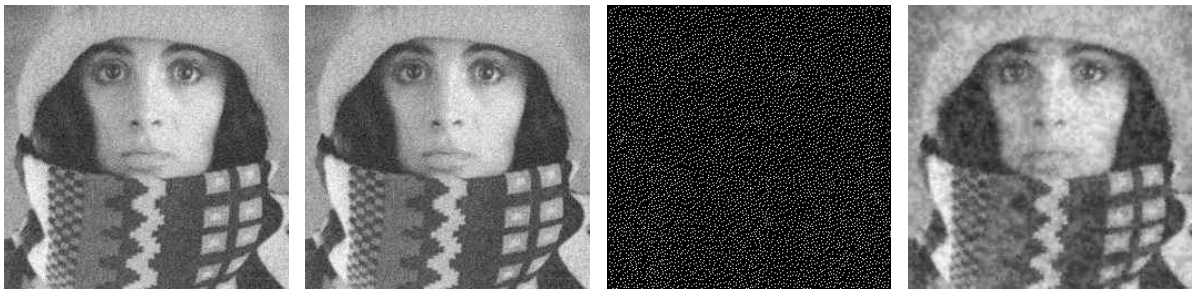
(a) $\sigma = 0$. (b) $\tilde{f}$. (c) Mask with $\tilde{f}$. (d) Reconstruction with $\tilde{f}$.

Figure 1.7: With L2-H $\beta = 0.18$, $\|f - u\|_2 = 4.78$.



(a) $\sigma = 0.05$. (b) $\tilde{f}$. (c) Mask with $\tilde{f}$. (d) Reconstruction with $\tilde{f}$.

Figure 1.8: With L2-T $\beta = 0.00005$, $\|f - u\|_2 = 13.91$.



(a) $\sigma = 0.05$. (b) $\tilde{f}$. (c) Mask with $\tilde{f}$. (d) Reconstruction with $\tilde{f}$.

Figure 1.9: With L2-H $\beta = 0.0009$, $\|f - u\|_2 = 12.57$.

We notice that sharpening the edges lead to a slight improvement of the accuracy of the reconstruction, however this effect decreases as the noise level increases. This is due to the action of the operators Δ and Δ^2 which enforce the edges but also amplify the noise.

(Pre)-Filtering the data

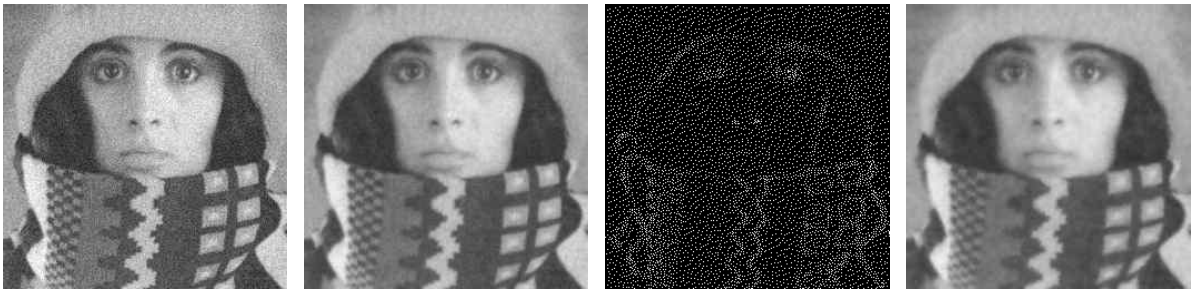
The idea here is to perform a small amount of filtering of the initial data. This may be performed on coarse mesh with the goal of reducing slightly the noise level. Thus, we replace f by $\tilde{f} \in H^1(D)$ solution of $\tilde{f} - \beta\Delta\tilde{f} = f$ in D. The criterion become :

$$\max|\Delta\tilde{f}| \Leftrightarrow \max|\tilde{f} - f|.$$

We notice a significant improvement in the accuracy and the obtained mask. It is also important to notice that there is no need of blind and strong denoising as the linear filter is applied on coarse mesh with the

aim of small reduction of the noise in the homogeneous areas, where intuitively we expect few pixels in the mask, and the similar action near the edges, with even some amount of blurring which do not affect too much the selection criterion.

We emphasize that others possible improvements within the same framework may be considered but clearly, a more systematic study in the spirit of tonal optimization is certainly a better choice in this direction.

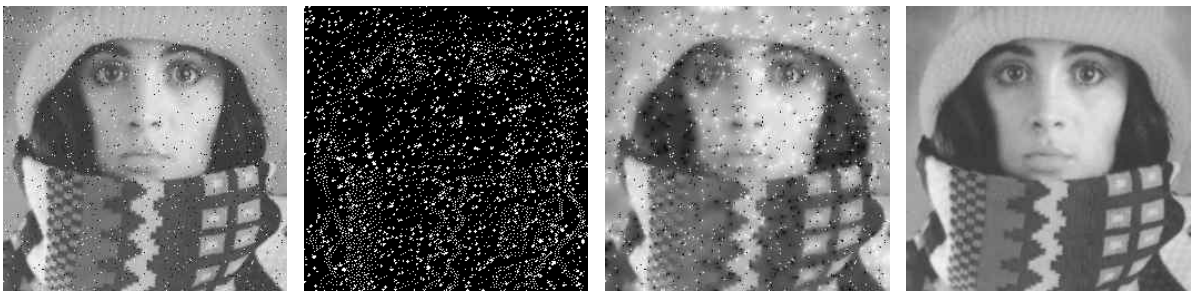


(a) $\sigma = 0.05$. (b) $\tilde{f}$. (c) Mask with $\tilde{f}$. (d) Reconstruction with $\tilde{f}$.

Figure 1.10: With L2-H $\beta = 1.2$, $\|f - u\|_2 = 7.77$.

1.4.4 Impulse Noise

We consider now images corrupted with impulse noise. In Figure 1.11, Figure 1.12 and Figure 1.13 we make some experiments with 2% of salt and pepper noise, respectively 1% of only salt noise and 1% of only pepper noise. These numerical simulations show that L^2-methods do not give a satisfying reconstruction with this sort of noise. In fact, the Laplacian takes large values at the noisy pixels. Thus such pixels are selected in the inpainting mask, whereas linear diffusion denoising as it is well known, do not perform well (e.g. large stains in Figure 1.11 (c)). This suggests to minimize the L^1-errors (instead of L^2 [119, 120]) to remove the impulse noise like salt and pepper noise as shown in the figures.



(a) Original image with 2% of salt and pepper noise. (b) L2-H mask. (c) Reconstruction. (d) L^1-error minimizer.

Figure 1.11: Image reconstruction with 10% of total pixels saved and 2% of salt and pepper noise applied to the input image.

(a) Original image with 1% of salt noise. (b) L2-H mask. (c) Reconstruction. (d) L^1-error minimizer.

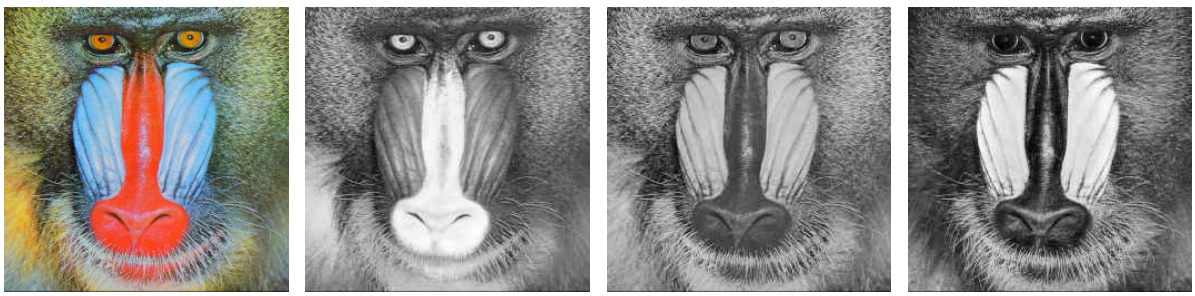
Figure 1.12: Image reconstruction with 10% of total pixels saved and 1% of salt noise applied to the input image.

(a) Original image with 1% of pepper noise. (b) L2-H mask. (c) Reconstruction. (d) L^1-error minimizer.

Figure 1.13: Image reconstruction with 10% of total pixels saved and 1% of pepper noise applied to the input image.

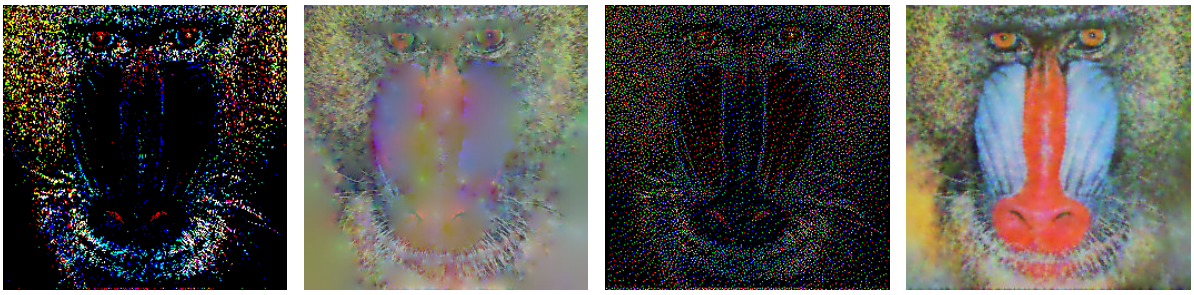
1.4.5 Colored Images

A colored image can be modeled by a function f from D to $[0,1]^3$, $x \mapsto \big(f_{\mathrm{R}}(x), f_{\mathrm{G}}(x), f_{\mathrm{B}}(x)\big)^{\mathrm{T}}$, where functions f_{R}, f_{G} and f_{B} are from D to $\mathbb{R}$, represent red channel, green channel and blue channel respectively (Figure 1.14). Our strategy is to create three masks, one for each channel. This is done in Figure 1.15 where (a) is the original image and (c) is the reconstruction by keeping 10% of total pixels for each mask. Since we compute a mask with a fixed number of pixels for each channel, the final mask, where the three masks are combined, may not have the same number of pixels. In fact, the three masks may not have common pixels or only some common pixels. More efficient strategies that use YCbCr color space instead of RGB space, have been investigated in [134, 132, 115].



(a) Original image. (b) Red channel to grayscale. (c) Green channel to grayscale. (d) Blue channel to grayscale.

Figure 1.14: Reconstructions for colored images.



(a) Mask *L2-T*. (b) Reconstruction *L2-T*. (c) Mask *L2-H*. (d) Reconstruction *L2-H*.

Figure 1.15: Reconstructions for colored images.

Summary and Conclusions

We have considered the problem of finding the best interpolation data in PDE-based compression problems for images with noise. We aim to have a unified framework for both compression and denoising since it is not clear that doing this two tasks separately leads to satisfying results. We introduced a geometric variational model to determine a set K which minimizes the L^p-distance between the initial image and its reconstruction from the datum in K. We extended the shape optimization approach introduced in [15] based on the analysis in the framework of Γ-convergence. In particular, we studied the two approaches considered there which differ in the way a single pixel in K is taken. Both theoretical findings emphasize the importance of the Laplacian of appropriate data and highlight the deep connection between the geometric set and the inpainting operator (in our case time harmonic). We have performed several numerical tests and comparisons which demonstrate the efficiency of the approach in handling images with noise. Besides, ongoing research addresses on one hand, a systematic study of tonal optimization techniques as a further step towards a drastic reduction of the "size" of K without loss of accuracy. Secondly, extending the shape analysis methods to nonlinear reconstruction operators will open some exciting perspectives in the fields of PDE based image compression.

This work may have several application like compression of "real world" images (since they always contain noise), video compression using only variational methods [8], microscopy imaging or denoising by inpainting [2].

Chapter 2

Iterative Approach to Image Compression with Noise : Optimizing Spatial and Tonal Data

Submitted

Contents

We consider some iterative methods for finding the best interpolation data in the images compression with noise. The interpolation data consists of the set of pixels and their grey/color values. The aim in the iterative approach is to allow the change of the data dynamically during the inpainting process for a reconstruction of the image that includes the enhancement and denoising effects. The governing PDE model of this approach is a fully parabolic problem where the set of stored pixels is time dependent. We consider the semi-discrete dynamical system associated to the model which gives rise to an iterative method where the stored data are modified during the iterations for best outcomes. Finding the compression sets follows from a shape-based analysis within the Γ-convergence tools developed in [15, 18], in particular well suited topological asymptotic and a "fat pixels" approach are considered to obtain an analytic characterization of the optimal sets in the sense of shape optimization theory. We perform the analysis and derive several iterative algorithms that we implement and compare to obtain the most efficient strategies of compression and inpainting for noisy images. Some numerical computations are presented to confirm the theoretical findings. Finally, we propose a modified model that allows the inpainting data to change with the iteration and compare the resulting new method to the "probabilistic" ones from the state-of-the-art.

keywords image compression – shape optimization – Γ-convergence – image interpolation – inpainting – PDEs – gaussian noise – image denoising

Introduction

Image compression and *inpainting* aim to reconstruct missing parts of an image from a chosen set of few known pixels, called *inpainting mask* [113, 20]. PDE-based models have emerged among the most efficient methods in this field as they allow high quality reconstruction even from small set of pixels (masks) [147, 65] and the references therein. Actually, diffusion-based models [66, 147, 15] are known to be efficient in image processing, particularly in image and video compression [8, 7], and similar problems such as zooming or denoising [2, 102].

It is commonly admitted in image compression problems, that both a well chosen inpainting operator and mask play a crucial role on the quality of the reconstruction and the efficiency of the compression strategy [110, 9, 66]. In fact, one aims to use a simple differential operator (e.g. laplacian) for easy and cheap reconstruction, but without sacrificing important features of the image at hand (e.g. singularities -edges-). This may appears somehow contradictory (because of the regularizing effects of the operator) or at least calls for finding a good balance between the mentioned simplicity of the differential operator and the necessary care about too strong regularization. Now, fixing the operator, the main question is the following : is a good "choice" of the mask possible? and what is a good choice? Many studies show that apart from noise or textures, good mask candidates should include pixels from contours of objects in the image. A mathematical theory addressing such questions from the shape optimization point of view may be found in [15], adapted to noisy images in [18]. In particular, it is proven there that an optimal mask under "size constraints" exists and is easily characterized by an analytical criterium. The most significant advantage then being the selection of the mask for general meshes and without any cost (other than the storage of the pixels).

Recently, many research works ([146, 43, 18]) noticed that choosing the mask at once may be less efficient than improving it somehow iteratively, however without giving a clear and sound justified strategy for the iterative process.

In this chapter, we consider the linear heat equation as the inpainting operator and study the selection of a "time-dependent" set of optimal pixels using shape optimization tools, extending this way the results in [18]. We present several strategies combining the shape optimization approach giving analytic criteria for masks selection and an enhancement process allowing to enrich/adapt such choice of pixels as well as the modification of the data on this points (*tonal optimization*). This turns to be a nice tool for image compression, particularly those with noise.

Related works

There exists important literature and several works addressing the problem of images compression (see [53, 52, 110, 84, 146] and the references therein). Most of these methods require a significant amount of computation time for the masks selection as the set of pixels selected are built upon a process based on local choices at the scale of each pixel. Moreover, mostly, the notion of optimality of the selection criteria, in the mathematical sense, is barely considered, though some of these methods give very good results and may include powerful features such as a low storage cost, handling complex images (e.g. with noise or textures). Beside finding an optimal mask, [110] introduced a data optimization strategy allowing to modify the values stored on the inpainting mask to improve the reconstructed images. Loosely speaking, they optimize the spatial distribution of the inpainting data, with a probabilistic data sparsification followed by a nonlocal pixel exchange (randomly remove and add pixels in the mask to avoid being trapped in a local minimizer) and then they optimize the grey values in these inpainting points using a least squares approach. Other data optimization approaches have been proposed in [43] by using finite element method instead of finite differences method and deep learning techniques for both choosing the inpainting mask and values [5].

In this chapter, we aim to propose some iterative methods using both the mathematical results in the spirit of [15] and an iterative process to enhance the quality of the mask within a data optimization approach for images with noise. We study and analyze the problem of finding a set K for the time harmonic linear diffusion that is "time-dependent", in the sense of allowing mask's changes (by exchanging,

removing, or adding pixels) and changes of the stored values during the process. We propose a deterministic all-in-one strategy for the mask choice and the tonal optimization. The enhancement of the mask selection is encoded in the criterium which is dependent on the previous values of the reconstructed image and the tonal optimization is performed by replacing the original values of the noisy image with those given by the iterative reconstruction. We compare different methods proposed here and those existing in the related literature, particularly in the presence of noise.

The standard PDE-based compression problem reads : let $D \subset \mathbb{R}^2$ the support of an image (say a rectangle) and $f : D \longrightarrow \mathbb{R}$, be an image which is assumed to be known only on some region $K \subset D$. There are several PDE models to interpolate f in $D \setminus K$, the simplest being the use of the linear elliptic equation, with Dirichlet boundary data $f|_K$ on K and homogeneous Neumann boundary conditions on ∂D. In our approach, we consider the iterative system of equations

Problem 2.0.1. *For $n \in \mathbb{N}$, given u^n, find u^{n+1} in $H^1(D)$ such that*

$$\begin{cases} u^{n+1} - \alpha \Delta u^{n+1} = u^n, & \text{in } D \setminus K_n, \\ u^{n+1} = f, & \text{in } K_n, \\ \frac{\partial u^{n+1}}{\partial \mathbf{n}} = 0, & \text{on } \partial D, \end{cases} \tag{2.1}$$

with the initial conditions u_0 and K_0. The data f in K_n may be replaced during the iteration by a function h_n, possibly depending on n, for tonal optimization and which in our case might be a "smoothed" version of f upon the iterations. Formally, the family of Problems 2.0.1, with $\alpha = \delta t$, the time step, is a semi-implicit discrete system to solve

Problem 2.0.2. *For $t > 0$, find $u(t, \cdot)$ in $H^1(D)$ such that*

$$\begin{cases} \partial_t u(t,\cdot) - \Delta u(t,\cdot) = 0, & \text{in } D \setminus K_t, \\ u(t,\cdot) = f, & \text{in } K_t, \\ \frac{\partial u(t,\cdot)}{\partial \mathbf{n}} = 0, & \text{on } \partial D, \end{cases} \tag{2.2}$$

$$u(0,\cdot) = u_0, \text{ in } D,$$

It might be possible and interesting to make precise the word "formally" by providing a well suited topology of sets and addressing the question of convergence of the semi-discrete system to Problem 2.0.2 but this is a quite deep question beyond the scope of this chapter.

Problem 2.0.1 with a fixed K has been considered in [18] and may appears as a particular case of the iterative process where one look for an optimal inpainting mask once and for all. The goal in solving Problem 2.0.1 is to modify K_n according to the change of the data (we notice that u^n is less noisy than f) while preserving the optimality at each step n. Thus, the iterative approach leads to adaptive improvement of the whole data with : topological changes in K_n and stored values changes that is $h_n\,|_{K_n}$ which may differ from f (*tonal optimization*).

We assume given $f \in H^1(D)$ and $\Delta f \in L^2(D)$ with $\frac{\partial f}{\partial \mathbf{n}} = 0$, for simplicity though in practice f is a function of bounded variations with a non trivial jump set. In fact, the whole analysis in the paper extends to the case of $f \in L^2(D)$.

To ensure the first compatibility condition for the full parabolic system, with the non-homogeneous "boundary" condition, we take $u(0,.) = f$ in D. Note that this choice is not in contradiction with image compression since the entire image is available during the encoding step. Denoting by u_{K_n} the solution of Problem 2.0.1, the question is to identify the region K_n which gives the "best" approximation u_{K_n}, in a suitable sense, for example which minimizes some L^p or Sobolev norms, e.g. in [15]

$$\int_D |\nabla u_{K_n} - \nabla f|^2 \, dx,$$

For noisy images, a well suited choice for the reconstruction is to minimize the L^p-norms of $u_{K_n} - f$, particularly for $p = 1$ and $p = 2$, which are known to be good filters for a large class of additive noises. In this chapter, we will take $p = 2$ for easy computations of the topological gradient but the main analysis holds for $p = 1$.

Following [15], we develop at each step $n > 0$, two methods of finding an optimal shape. The first is based on a topological asymptotic (a gradient method) which gives pointwise information on the set of pixels to select and yields a hard thresholding approach. The second method consists in a finite dimensional shape optimization problem, where K_n is taken as a union of small balls, i.e. a finite number of "fat pixels". Then, performing the asymptotic analysis by Γ-convergence when the number of pixels is increasing (in the same time that the fatness vanishes), we obtain useful information about the optimal distribution of the best interpolation pixels as a density function leading to soft thresholding approach.

In all cases, we obtain an optimal mask, in the sense mentioned above, and the values to be stored within the iterative method.

Organisation of the Chapter

In Section 2.1, we recall the mathematical model of the compression problem and we summarize the analysis steps and results referring interested readers to [15, 18] for details and recall the two methods proposed to construct our set of interpolation points at each step n. In Section 2.2, we propose three different algorithms based on the hard/soft-thresholding criterium stated in the previous section. In Section 2.3 we numerically compare the proposed algorithms for both image compression and image denoising. Finally, in Section 2.4, we propose a modified model that allows the inpainting data to change with the iteration and compare the resulting new methods with the so-called *sparsification* and *densification* algorithm [2, 110].

2.1 Review of the Continuous Model

In this section, we mainly recall and adapt results found in [18] in the case of the non-homogeneous linear diffusion inpainting.

2.1.1 Min-max Formulation

Let D be a smooth bounded open subset of $\mathbb{R}^2$. The shape optimization problem we study is, for a given $n \in \mathbb{N}$,

$$\min_{K_n \subseteq D,\ \mathrm{cap}(K_n) \leq c} \Big\{ \frac{1}{2} \int_D |u_{K_n} - f|^2 \, dx + \frac{\alpha}{2} \int_D |\nabla (u_{K_n} - f)|^2 \, dx \ \Big|\ u_{K_n} \text{ solution of Problem 2.0.1} \Big\}, \quad (2.3)$$

where $\mathrm{cap}(E)$ is the capacity of a subset E in D i.e.

$$\mathrm{cap}(E) = \inf \Big\{ \int_D |\nabla u|^2 \, dx + \int_D u^2 \, dx \ \Big|\ u \in H_0^1(D),\ u \geq 1 \text{ a.e. in } E \Big\},$$

and $c > 0$. The image compression problem aims to find an optimal set of pixels from which an accurate reconstruction of the (noisy) image will be performed. If we denote by u_{K_n} the solution of Problem 2.0.1, it is straightforward to obtain

Proposition 2.1.1. *The optimization problem* (2.3) *is equivalent to*

$$\max_{K_n \subseteq D} \min_{u \in H^1(D),\ u = f \text{ in } K_n} \frac{\alpha}{2} \int_D |\nabla u|^2 \, dx + \frac{1}{2} \int_D (u - u^n)^2 \, dx - \beta \, \mathrm{cap}(K_n), \quad (2.4)$$

for $\beta > 0$.

The next section is devoted to the analysis within the γ-convergence approach follows the same lines as in [15, 18] with slight changes.

2.1.2 Analysis of the Model

It is well known that such shape optimization problems do not always have a solution (e.g. [15]), we seek a relaxed formulation, which yields a relaxed solution, that is to say, a capacity measure. Thus, we consider the relaxed problem

$$\max_{\mu_n \in \mathcal{M}_0(D)} \min_{u \in H^1(D)} \frac{\alpha}{2} \int_D |\nabla u|^2 \, dx + \frac{1}{2} \int_D (u - u^n)^2 \, dx + \frac{1}{2} \int_D (u - f)^2 \, d\mu_n - \beta \, \mathrm{cap}(\mu_n), \tag{2.5}$$

where $\mathcal{M}_0(D)$ is the set of all non negative Borel measures μ on D, such that

- $\mu(B) = 0$, for every Borel set B subset of D with $\mathrm{cap}(B) = 0$,
- $\mu(B) = \inf \big\{ \mu(U) \;\big|\; U \text{ quasi-open}, \; B \subseteq U \big\}$, for every Borel subset B of D,

and $\mathrm{cap}(\mu)$ is the measure capacity i.e. for μ in $\mathcal{M}_0(D)$,

$$\mathrm{cap}(\mu) := \inf_{u \in H_0^1(D)} \int_D |\nabla u|^2 \, dx + \nu \int_D u^2 \, dx + \int_D (u - 1)^2 \, d\mu.$$

Next we give a natural way to identify a set to a measure of $\mathcal{M}_0(D)$. Let E be a Borel subset of D. We denote by ∞_E the measure of $\mathcal{M}_0(D)$ defined by

$$\infty_E(B) := \begin{cases} +\infty, & \text{if } \mathrm{cap}(B \cap E) > 0, \\ 0, & \text{otherwise.} \end{cases}, \text{ for all } B \text{ Borel subset of } D.$$

Formally, when $\mu_n = \infty_A$ in (2.5), we penalize the functional which is equal to $+\infty$ until $u = f$ in A. Somehow, we embed the Dirichlet condition into the min-max problem thanks to this penalization. For technical reasons, we want to include balls centered at points x_0 in D, that we do not want to be too close to the boundary of D, we introduce the following notations for $\delta > 0$,

$$D^{-\delta} := \{ x \in D \mid d(x, \partial D) \geq \delta \} \subseteq D,$$

$$\mathcal{K}^\delta(D) := \{ K \subseteq D \mid K \text{ closed}, \; K \subseteq D^{-\delta} \},$$

and

$$\mathcal{M}_0^\delta(D) := \{ \mu \in \mathcal{M}_0(D) \mid \mu_{|D \setminus D^{-\delta}} = 0 \} \subseteq \mathcal{M}_0(D).$$

We have that (2.5) is the relaxed formulation of the optimization problem (2.4) in the sense of the γ-convergence (Theorem 1.4 in [18]) :

Theorem 2.1.1. *We have,*

$$\sup_{K \in \mathcal{K}_\delta(D)} \big(E(\infty_K) - \beta \, \mathrm{cap}(\infty_K) \big) = \max_{\mu \in \mathcal{M}_0^\delta(D)} \big(E(\mu) - \beta \, \mathrm{cap}(\mu) \big),$$

where

$$E(\mu) := \min_{u \in H^1(D)} F_\mu(u),$$

and

$$F_\mu(u) := \begin{cases} \alpha \int_D |\nabla u|^2 \, dx + \int_D (u - f)^2 \, d\mu, & \textit{if } |u| \leq |f|_\infty, \\ +\infty, & \textit{otherwise.} \end{cases}$$

This result gives us the existence of a relaxed solution which γ-converges to the "solution" of our shape optimization problem. In order to solve the relaxed problem, we may use a shape derivative with respect to the measures μ_n. However, such a method yields diffuse measures, thus too thick sets whereas we seek discrete sets of pixel.

2.1.3 Topological Gradient

Here, we aim to compute the solution of our optimization problem (2.3) by using a topological gradient-based algorithm as in [100, 68]. This kind of algorithm consists in starting with $K_n = \bar{D}$ and determining how making small holes in K_n affect the cost functional to find the balls which have the most decreasing effect. To this end, let us define K_ε^n the compact set $K_n \setminus B(x_0, \varepsilon)$ where $B(x_0, \varepsilon)$ is the ball centered in $x_0 \in D$ with radius $\varepsilon > 0$ such that $B(x_0, \varepsilon) \subset K_n$. From now, we consider the variable $v_{K_n} := u_{K_n} - f$, with u_{K_n} solution of Problem 2.0.1. Let us denote by j the functional

$$j : A \subset D \mapsto \min_{v \in H^1(D),\ v=0 \text{ in } A} \frac{\alpha}{2} \int_D |\nabla v|^2 \, dx + \frac{1}{2} \int_D v^2 \, dx \ - \int_D g^n v \, dx,$$

where $g^n := u^n - f + \alpha \Delta f$. We denote by v_ε^n the minimizer of $j(K_\varepsilon^n)$ and we have

Proposition 2.1.2. *With notations from above, we have when ε tends to 0,*

$$j(K_\varepsilon^n) - j(K_n) = \frac{\pi}{2} \big(g^n(x_0)\big)^2 \varepsilon^2 \ln(\varepsilon) + O(\varepsilon^2).$$

Since for $\varepsilon < 1$, $\ln \varepsilon < 0$, the result above suggests to keep the points x_0 where $|u^n(x_0) - f(x_0) + \alpha \Delta f(x_0)|^2$ is maximal, when ε small enough. In the next section, we will see that such a strict threshold rule might be relaxed.

2.1.4 Optimal Distribution of Pixels : The "Fat Pixels" Approach

In this section, we change our point of view by considering "fat pixels" instead of a general set of interpolation points. In the sequel, we will follow [33, 15, 18]. We restrict our class of admissible sets as an union of balls which represent pixels. For $m > 0$ and $k \in \mathbb{N}$, we define

$$\mathcal{A}_{m,k} := \Big\{ \overline{D} \cap \bigcup_{i=1}^{k} \overline{B(x_i, r)} \ \Big| \ x_i \in D_r, \ r = mk^{-1/2} \Big\},$$

where D_r is the r-neighborhood of D. We consider problem (2.3) for every $K_n \in \mathcal{A}_{m,k}$ i.e.

$$\min_{K_n \in \mathcal{A}_{m,k}} \Big\{ \frac{1}{2} \int_D (u_{K_n} - f)^2 \, dx + \frac{\alpha}{2} \int_D |\nabla u_{K_n} - \nabla f|^2 \, dx \ \Big| \ u_{K_n} \text{ solution of Problem 2.0.1} \Big\}. \tag{2.6}$$

By setting $v_{K_n} := u_{K_n} - f$, this last optimization problem can be reformulated as a compliance optimization problem. We set $g^n := u^n - f + \alpha \Delta f$ like in the previous section. Here, we do not need to specify a size constraint on our admissible domains. Indeed, imposing $K_n \in \mathcal{A}_{m,k}$ implies a volume constraint and a geometrical constraint on K_n since K_n is formed by a finite number of balls with radius $m\, k^{-1/2}$. The well-posedness of such a problem has been studied in the laplacian case in [33]. As pointed out in [28], the local density of $K_k^{n,\text{opt}}$, the optimal solution, can be obtained by using a different topology for the Γ-convergence of the rescaled energies. In this new frame, the minimizers are unchanged but their behavior is seen from a different point of view. We define the probability measure μ_K for a given set K in $\mathcal{A}_{m,k}$ by

$$\mu_K := \frac{1}{k} \sum_{i=1}^{k} \delta_{x_i}.$$

We define the functional F_k^n from $\mathcal{P}(\bar{D})$ into $[0, +\infty]$ by

$$F_k^n(\mu) := \begin{cases} k \int_D g^n v_{K_n} \, dx, & \text{if } \exists K_n \in \mathcal{A}_{m,k}, \text{ s.t. } \mu = \mu_{K_n}, \\ +\infty, & \text{otherwise.} \end{cases}$$

The following Γ-convergence of F_k^n theorem is similar to the one given in Theorem 2.2. in [33].

Theorem 2.1.2. *Then the sequence of functionals F_k^n, defined above, Γ-converge when k tends to $+\infty$ with respect to the weak $\star$ topology in $\mathcal{P}(\bar{D})$ to*

$$F^n(\mu^n) := \int_D \frac{(g^n)^2}{\mu_a^n} \theta\big(m(\mu_a^n)^{1/2}\big)\ dx, \tag{2.7}$$

where $\mu^n = \mu_a^n dx + \nu^n$ is the Radon-Nikodym-Lebesgue decomposition of μ^n ([62], Theorem 3.8) with respect to the Lebesgue measure and

$$\theta(m) := \inf_{K_k \in \mathcal{A}_{m,k}} \liminf_{k\to+\infty} k \int_D g^n v_{K_k}\ dx,$$

$v_{K_n} := u_{K_n} - f$, u_{K_n} solution of Problem 2.0.1.

As a consequence of the Γ-convergence stated in the theorem above, the empirical measure $\mu_{K_k^{n,\mathrm{opt}}} \to \mu^{n,\mathrm{opt}}$ weak $\star$ in $\mathcal{P}(\mathbb{R}^d)$ where $\mu^{n,\mathrm{opt}}$ is a minimizer of F^n. Unfortunately, the function θ is not known explicitly. Formal Euler-Lagrange equation and the estimates on θ [18] give the following information : to minimize (2.7) one have to take

$$\frac{(\mu_a^n)^2}{|1 - \log \mu_a^n|} \approx c_{m,f}\,(u^n - f + \alpha\Delta f)^2.$$

This introduces a soft-thresholding with respect to the first approach. To sum up, we can choose the interpolation data such that the pixel density is increasing with $|u^n - f + \alpha\Delta f|$.

2.2 The Iterative Methods

With the model proposed in previous sections it is clear that we do need the complete sequence of optimal set $(K_n)_n$ in order to reconstruct the solutions $(u^n)_n$. While it does not represent a problem for denoising only purpose, it is not conceivable to store the complete sequence of inpainting mask for compression purpose. To overcome this issue, we propose in this section three algorithms, with on one side the hard-threshold criteria, namely we select the pixels where $|u^n - f + \alpha\Delta f|$ is maximum and on the other side, the soft-threshold criteria of the fat pixels approach, where the selected pixels are chosen according to the distribution of $|u^n - f + \alpha\Delta f|$.

2.2.1 L2-INSTA

Encoding. During the encoding step, the original image is known on the whole domain D. We can thus set $u_0 = f$ in D. We provide the input image f and the desired pixel density $0 < c < 1$, and the algorithm produces the "optimal" mask K. We notice that K is only optimal in the sense that it is a limit of optimal sets K_n. At each iteration, we compute and use a new mask without preserving the previous set. The stopping criterium is a given integer N corresponding to the level of error on the reconstruction we want to achieve. We note that with this algorithm, K_0 is the same set as in the stationary case [18]. The complexity of the encoding is $O(N)$. We give the algorithm of *L2-INSTA* in Algorithm 6.

Data: Original (noisy) image f, parameter (time-step) $\alpha > 0$, desired pixel density $c \in (0,1)$, number of iterations $N \in \mathbb{N}^*$.
Result: Inpainting mask $K \subset D$, last encoding reconstruction u^N.

```
u^0 ← f;
for n in {0, ..., N − 1} do
    Save in K the c|D| pixel by using the hard/soft-thresholding criterium i.e. |u^n − f + αΔf|;
    Compute u^{n+1}, solution of Problem 2.0.1;
end
```

Algorithm 6: *L2-INSTA*.

Decoding. During the decoding step, the data are only available on K. Therefore, we must not use the image f in $D \setminus K$ for the inpainting problem. We propose to put u_0 to zero in $D \setminus K$. However,

doing this choice leads to a reconstruction u which is near zero in the unknown set when computing the homogeneous heat equation for small time t. Since we want to compute the reconstruction at the same time as in the encoding step, we have to choose $t = N\alpha$. Thus, to have large t, we can either have a large number of iteration N, either a large time-step α. But, for large α, the criterium will be close to $|\Delta f|$ and the resulting mask will not depends on u^n anymore. Thus, a good choice would be to have a small time-step α and a large iteration number N in the encoding step. Note that for the decoding step, we can directly compute the reconstruction at time $t = N\alpha$ by performing only one iteration.

2.2.2 L2-DEC

Encoding. In their article, Mainberger *et al* proposed a probabilistic algorithm to compute an inpainting mask for a given inpainting method [110]. This *sparsification* algorithm is described as follow : we start with a mask containing all the pixels of the image, then at each iteration, we randomly delete a fraction of the pixels of the mask, we inpaint, we calculate the local error at each deleted pixel and we put back in the mask a subset of the deleted pixels presenting the most important error. We propose a modified algorithm based on the *sparsification* one that we call *L2-DEC*. The difference here is that, instead of selectionning a small random candidate set of pixels for each iteration, we directly choose an optimal set with respect to the criterion stated in this chapter for the current iteration. Using this strategy eliminate the probabilistic nature of the algorithm. The complexity of the encoding is $O(\frac{c}{q})$, with q the fraction of finally deleted pixel at the end of the iteration. Despite we give the algorithm of *L2-DEC* in Algorithm 7 for completness, we believe that *L2-DEC* is not pertinent. Indeed, for every point x_0 in K_n, $u^n(x_0) = f(x_0)$ and the criterion becomes

$$|u^n(x_0) - f(x_0) + \alpha\Delta f(x_0)| = |f(x_0) - f(x_0) + \alpha\Delta f(x_0)| = \alpha|\Delta f(x_0)|.$$

The algorithm will remove pixel with the lowest values $|\Delta f|$ and produce the same mask as the optimal one for the homogeneous diffusion inpainting [15]. We propose this theoretical algorithm for completeness purpose, but we will not realize numerical computations.

Data: Original (noisy) image f, parameter (time-step) $\alpha > 0$, fraction q of pixel removed, desired pixel density $c \in (0, 1)$.
Result: Inpainting mask $K \subset D$, last encoding reconstruction u^N.

$u^0 \leftarrow f$;
$K \leftarrow D$;
$n \leftarrow 0$;
while $|K| > c\,|D|$ **do**
  Keep in K the $|K| - q\,|D|$ pixel in K by using the hard/soft-thresholding criterium i.e. $|u^n - f + \alpha\Delta f|$;
  Compute u^{n+1}, solution of Problem 2.0.1;
  $n \leftarrow n + 1$;
end

Algorithm 7: *L2-DEC.*

Decoding. As described in the original paper of the *sparsification* algorithm [110], we should use the same inpainting (with the same parameters) as the one used in the encoding iterations, namely the discretized heat equation. However, as explained in the previous section, in order to have a pertinent reconstruction we should take the time-step α large enough. Such a choice will lead to a constant inpainting mask with respect to the iterations, which is not the purpose of this paper. We propose instead to use the homogeneous diffusion inpainting [15, 110] to reconstruct the image from data in K.

2.2.3 L2-INC

Encoding. Peter *et al* proposed the *densification* algorithm [132, 2]. We start with an empty mask and, for each iteration, we randomly choose α pixels that do not belong to the mask. We then add only the single pixel that improves the global reconstruction error with respect to the image. In order to make the comparison possible with the following proposed algorithm, we propose to not only add one pixel but to add the fraction q of pixels that improves the most the global reconstruction error with respect to the

image. With our proposed algorithm, we directly choose an optimal candidate set with respect to the criterion stated in this chapter for the current iteration instead of choosing a small random candidate set of pixel for each iteration. The complexity of the encoding is $O(\frac{c}{q})$. We give the algorithm of *L2-INC* in Algorithm 8.

Data: Original (noisy) image f, parameter (time-step) $\alpha > 0$, fraction q of pixel added, desired pixel density $c \in (0, 1)$.
Result: Inpainting mask $K \subset D$, last encoding reconstruction u^N.
$u^0 \leftarrow f$;
$K \leftarrow \emptyset$;
$n \leftarrow 0$;
while $|K| < c\,|D|$ **do**
  Add the $q\,|D|$ pixels in $D \setminus K$ by using the hard/soft-thresholding criterium i.e. $|u^n - f + \alpha\Delta f|$, and add them to K;
  Compute u^{n+1}, solution of Problem 2.0.1;
  $n \leftarrow n + 1$;
end

Algorithm 8: *L2-INC*.

Decoding. With the same reasoning as the decoding step of the *L2-DEC* algorithm, we propose to use the homogeneous diffusion inpainting to reconstruct the image from data in K.

2.3 Numerical Comparison of the Iterative Methods

In this section, we will present numerical results and comparisons between the presented methods from Section 2.2. The soft-thresholding rule from the "fat pixel" point of view can be enforced with a standard digital halftoning. Digital halftoning is a method of rendering that convert a continuous image to a binary image while giving the illusion of color continuity [161, 3]. This color continuity is simulated for the human eye by a spacial distribution of black and white pixels. An ideal digital halftoning method would preserves the average value of gray while giving the illusion of color continuity. For the experiment using the soft-thresholding rule, we use the Floyd-Steinberg dithering [61] except for *L2-INC-* for reasons that will be discussed in the related section. We use the notation "name-of-the-algorithm-T" and "name-of-the-algorithm-H" to distinguish the hard- and soft-thresholding criteria used.

2.3.1 Image Compression

As discussed in the previous section, for *L2-STA-* algorithms, we take for $\alpha = 0.05$ (a small value). For *L2-INC-H*, we cannot use the Floyd-Steinberg algorithm since it is biased by the error propagation when the number of pixel demanded is to small (here, we choose to add 50 pixels per iteration). This leads to select pixel localised in the bottom-right part of the image. As an alternative, we propose to use a halftoning method based on the Lloyd's method instead of error propagation [149]. Algorithms *H1-T* and *H1-H* correspond to the methods described in [15], namely, the hard/soft-thresholding of the absolute value of the laplacian of the input image f_δ i.e. $|\Delta f_\delta|$. These two methods have the advantage to not require any reconstructions during the encoding phase. However, since our input image may contain noise, the laplacian will be highly perturbed and the information given by the edges will be lost. It will lead to a poor reconstructions for images that contain noise.

In most of the case, *L2-INC-T* gives greater or similar reconstruction quality than the other methods. It can be explained by the fact that, for each iteration, we add to the mask a small amount of best pixels for the current iteration. Thus, we keep important pixels for previous iterations and somehow store the full sequence of optimal sets $(K_n)_n$ containing a few number of pixels. Contrasting with them, the *L2-INSTA-* methods "forget" at each iteration the optimal pixel set from the previous iterations. This leads to a poorer visual quality in the reconstruction. It is also interesting to notice that our model and then our analytical criteria are more robust to noise than the one using the homogeneous diffusion as inpainting operator, namely the *H1-* methods. It confirm the pertinence of our model to handle image with gaussian noise.

In Table 2.1 and Table 2.2, we give the L^2-errors between the image f and the reconstruction u (we write L^2 in the tables), as a function of the gaussian noise standard deviation for each method for different size of mask. In Figure 2.2, Figure 2.3 and Figure 2.4 (Section 2.5), we present various masks obtained and the corresponding reconstructed images.

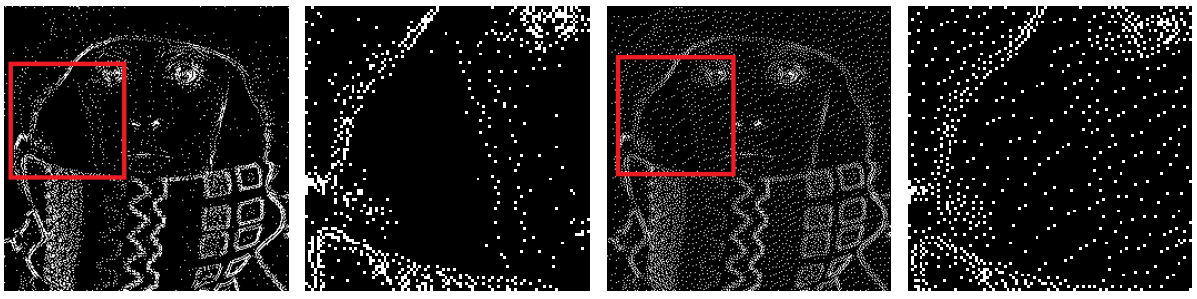
Noise	**H1-T**	**H1-H**	**L2-INSTA-T**		**L2-INSTA-H**		**L2-INC-T**		**L2-INC-H**	
	L^2	L^2	N	L^2	N	L^2	α	L^2	α	L^2
0	34.46	**11.43**	3131	29.39	6517	11.58	0.26	18.68	2.25	20.18
0.03	15.84	14.21	7905	15.76	8765	13.37	0.26	**8.46**	0.42	21.24
0.05	19.24	16.44	4341	18.82	6472	15.50	0.16	**11.74**	1.44	24.25
0.1	30.91	23.78	4507	24.81	6192	**21.52**	0.06	23.27	1.43	27.22
0.2	64.51	40.68	4717	40.24	6376	35.26	0.11	48.31	0.62	**33.06**

Table 2.1: L^2-error between the original image f and the reconstruction u (build from f_δ) with 5% of total pixels saved.

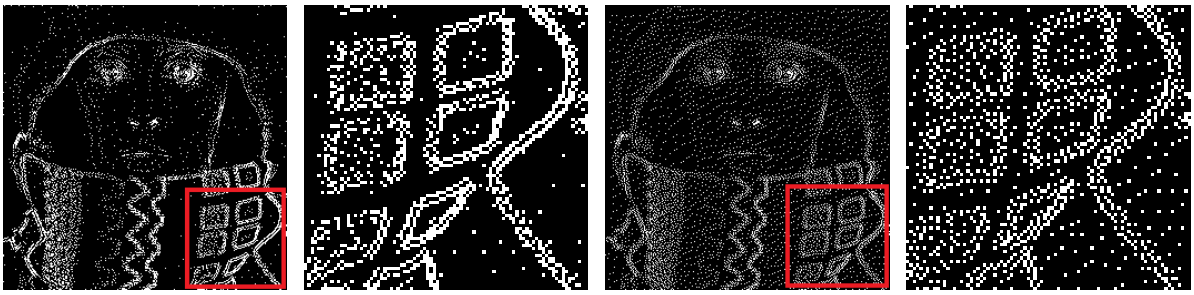
Noise	**H1-T**	**H1-H**	**L2-INSTA-T**		**L2-INSTA-H**		**L2-INC-T**		**L2-INC-H**	
	L^2	L^2	N	L^2	N	L^2	α	L^2	α	L^2
0	22.99	**6.02**	3655	13.61	4815	6.78	0.31	6.36	1.03	14.64
0.03	11.52	10.43	2315	13.69	2407	10.49	0.11	**6.96**	0.42	17.50
0.05	15.42	13.77	3131	14.50	3909	12.64	0.11	**10.94**	0.83	20.89
0.1	27.51	23.09	2143	22.17	5060	**20.34**	0.11	20.72	0.83	23.56
0.2	55.09	42.44	3007	35.60	4987	35.99	0.11	39.14	0.83	**31.87**

Table 2.2: L^2-error between the original image f and the reconstruction u (build from f_δ) with 10% of total pixels saved.

In Figure 2.1 we see in particular that, in order to reconstruct an homogeneous part of the image, *L2-INC-T* select a few amount of pixels (here 4 pixels) whereas an halftoning algorithm way more redundant pixels. It leads to more pixels available to spend near the edges in *L2-INC-T* and then, a better reconstruction.



(a) Homogeneous part with *L2-INC-T* method. (b) Zoom. (c) Homogeneous part with *L2-STA-H* method. (d) Zoom.



(e) Edges with *L2-INC-T* method. (f) Zoom. (g) Edges with *L2-STA-H* method. (h) Zoom.

Figure 2.1: Zoom on homogeneous parts and edges.

2.3.2 Image Denoising

We start by making the hereafter observation : for well chosen parameters, the error between the original image f and the solution u^n decreases during the encoding step and becomes lower than the error between the original image f and the noised one f_δ. In fact, this error on u^n is lower than the one on f_δ from the first iteration. We propose then to use the last encoding reconstruction u^N of algorithms *L2-INSTA* and *L2-INC*, Algorithm 6 and Algorithm 8 respectively, as denoised version of f_δ. For every noise level σ, we take take 1% of the total pixel in the mask for *L2-INSTA-* and 2% for *L2-INC-* methods except when $\sigma = 0.2$, we take 4% for the *L2-INC-* methods. In any case, we set $\alpha = 0.01$.

It appears that *L2-INC-* are almost as efficient as the linear diffusion filter, which gives the lowest error, but are more edges preserving than the linear diffusion filter. As expected, every of the proposed methods perform noise removing and we will exploit this feature in the next section.

We give in Table 2.3 the L^2-error between f and u^N with respect to noise level in f_δ for the methods proposed in this paper and for the linear diffusion filter of parameter η, which is known to remove gaussian noise, and the resulting images in Figure 2.5 (Section 2.6).

Noise	$\|f - f_\delta\|_{L^2(D)}$	**Lin. Filter**		**L2-INSTA-T**		**L2-INSTA-H**		**L2-INC-T**	**L2-INC-H**
		η	L^2	N	L^2	N	L^2	L^2	L^2
0.03	7.65	0.9	4.61	44	5.28	38	4.92	4.82	**4.44**
0.05	12.81	1.2	**6.16**	58	8.54	55	7.58	7.60	6.60
0.1	25.45	2.0	**8.98**	59	16.60	102	14.06	14.67	12.50
0.2	48.43	2.5	**12.72**	88	30.65	182	25.76	19.92	16.44

Table 2.3: Using u^N as a denoised version of the input image.

2.4 Numerical Comparison with the "Probabilistic" Methods

In this section, we present numerical results and comparisons with some stat-of-the-art compression by inpainting methods for image with noise. In the following we denote by *SPAR* the *sparsification* algorithm without the *nonlocal pixel exchange* post-optimization (step within every *sparsification*'s iteration to avoid source locality due to local error computation) from [110] and by *DENS* a modified version of the original *densification* algorithm from [2], that we will describe in the sequel.

The knowledge of the previous section motives us to replace f by u^n as Dirichlet boundary condition in the inpainting mask K, and thus, to replace f by u^n in the shape optimization problem (2.3) : for a given $n \in \mathbb{N}$,

$$\min_{K_n \subseteq D,\ \mathrm{cap}(K_n) \leq c} \left\{ \frac{1}{2} \int_D |u_{K_n} - u^n|^2 \, dx + \frac{\alpha}{2} \int_D |\nabla (u_{K_n} - u^n)|^2 \, dx \right\}, \tag{2.8}$$

with u_{K_n} solution of

Problem 2.4.1. *For $n \in \mathbb{N}$, given u^n, find u^{n+1} in $H^1(D)$ such that*

$$\begin{cases} u^{n+1} - \alpha \Delta u^{n+1} = u^n, & \text{in } D \setminus K_n, \\ u^{n+1} = u^n, & \text{in } K_n, \\ \frac{\partial u^{n+1}}{\partial \mathbf{n}} = 0, & \text{on } \partial D, \end{cases} \tag{2.9}$$

Since the analysis remains the same as in Section 2.1, these changes yield to a new criterion, namely $|\Delta u^n|$ for both hard- and soft-thresholding. This *tonal optimization* avoid brutal smoothing of the image during the first iterations. *Tonal optimization* consists in changing the value of pixel in a given inpainting mask in order to improve the overall image's reconstruction quality [110]. Like for the previous algorithms, we take $u_0 = f$ in D for the encoding step since the entire noisy image is available, but we set u_0 to zero on $D \setminus K$ during the decoding step.

As suggested by the authors in the original paper of the *SPAR* algorithm, the candidate set have 2% of the pixel in D, and we choose the finally removed numbers of pixel to be 50 pixels from the candidate set. To make the comparison possible, we propose to remove/add a fixed number of pixel (we choose 50 pixels) at each iteration for the appropriate algorithms. Then, we have to modify the *DENS* algorithm, since it originally add one single pixel per iteration. This modification induces lower visual quality for the reconstruction with a high gain for the computation time. Also, we choose the number of randomly selected candidates per iteration to be 100 which still require a lot of computation time. Therefore, we have 100 reconstructions to compute at each iteration. We name the method from this new model combined with Algorithm 8 and hard-thresholding *L2-INC-T-E*.

L2-INC-T-E seems to efficiently denoise the input image while compressing and outperform the *SPAR* method and the *DENS* method for high level of noise. However, our new model is designed to handle input image with gaussian noise and then, perform worst than the previous model when the image does not contain noise.

We give in Table 2.3 the L^2-error between f and u^N with respect to noise level in f_δ for the *L2-INC-T-E* method proposed in this section and the "probabilistic" methods from the state-of-the-art and we give in Figure 2.6 and Figure 2.7 (Section 2.7) some illustration.

Noise	**SPAR**	**DENS**	**L2-INC-T**	
	L^2	L^2	α	L^2
0	**3.68**	8.70	0.01	15.54
0.03	**6.94**	9.09	0.01	7.53
0.05	11.11	9.82	0.01	**7.98**
0.1	21.37	12.58	0.01	**11.79**

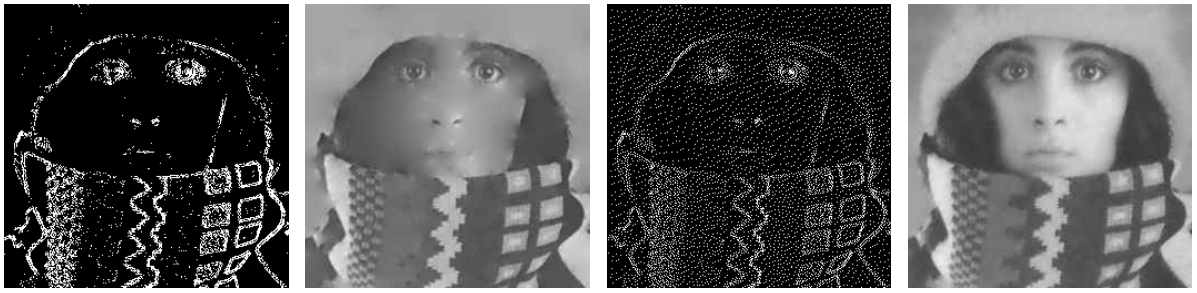
Table 2.4: L^2-error between the original image f and the reconstruction u (build from f_δ and with the new model) with 10% of total pixels saved.

Summary and Conclusions

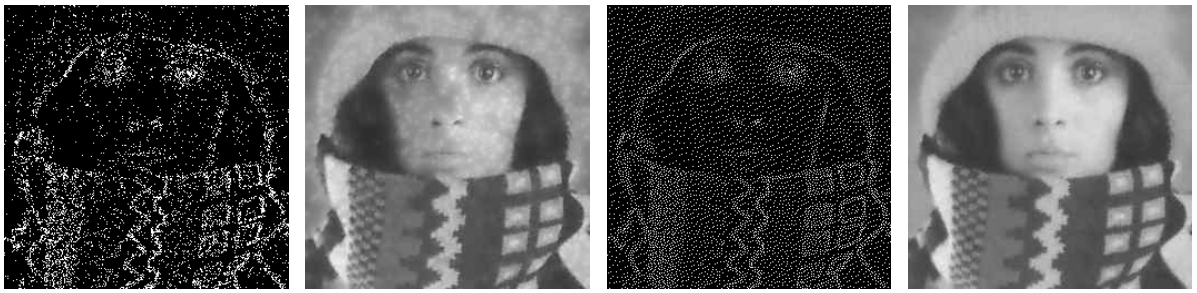
In this chapter, we have considered a new fully parabolic model based on the heat equation for PDE images compression and formulate an associated shape optimization problem for the best choice of data interpolation choice. By using the results found in [18], we proved that it has a relaxed solution in the framework of the γ-convergence and we proposed two points of view for choosing an optimal inpainting mask : a first one that use an asymptotic development similar to the topological gradient, and a second one by considering the sets formed by a finite number of balls that we called "fat pixels". Unlike the stationary case, the iterative approach include the results of the previous steps in the analytic criterion giving the best choice which is now depending on the iterations. Next, we proposed and implemented three algorithms to construct the sequences of inpainting mask $(K_n)_n$: *L2-INSTA*, *L2-DEC* and *L2-INC*, and we performed several simulations with different compression rates and different amounts of noise for grayscale images. It appears that in most cases, masks issued form *L2-INC-T* outperforms the other ones.

Finally, we performed a tonal optimization within the same method by changing the Dirichlet boundary conditions on the mask, which appears to give better results for denoising while compressing an image with noise than the *sparsification* and the *densification* algorithms [2, 110] for high level of noise.

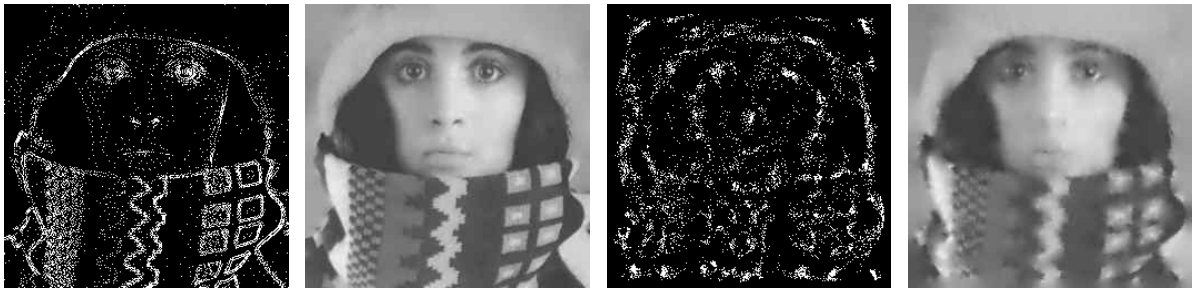
2.5 Inpainting Masks and Reconstructions for Compression from Section 2.3



(a) Mask with *H1-T* method. (b) Reconstruction with *H1-T* method. (c) Mask with *H1-H* method. (d) Reconstruction with *H1-H* method.



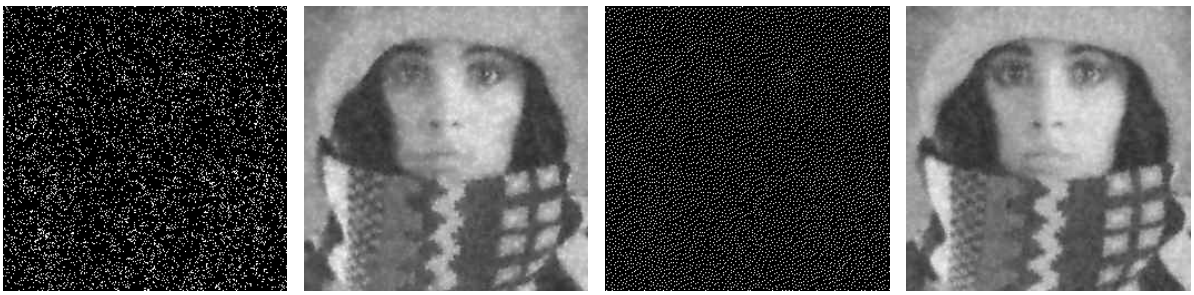
(e) Mask with *L2-INSTA-T* method. (f) Reconstruction with *L2-INSTA-T* method. (g) Mask with *L2-INSTA-H* method. (h) Reconstruction with *L2-INSTA-H* method.



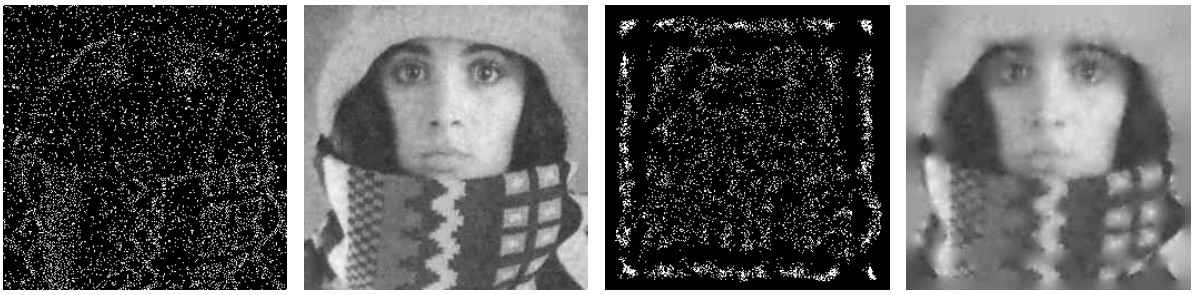
(i) Mask with *L2-INC-T* method. (j) Reconstruction with *L2-INC-T* method. (k) Mask with *L2-INC-H* method. (l) Reconstruction with *L2-INC-H* method.

Figure 2.2: Masks and reconstructions for Table 2.2 when the input image is noiseless ($\sigma = 0$).

(a) Mask with *H1-T* method. (b) Reconstruction with *H1-T* method. (c) Mask with *H1-H* method. (d) Reconstruction with *H1-H* method.

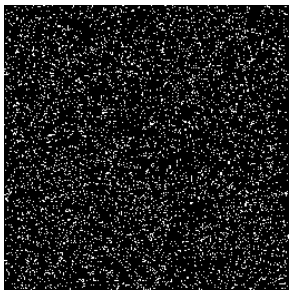


(e) Mask with *L2-INSTA-T* method. (f) Reconstruction with *L2-INSTA-T* method. (g) Mask with *L2-INSTA-H* method. (h) Reconstruction with *L2-INSTA-H* method.



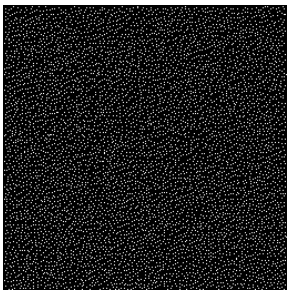
(i) Mask with *L2-INC-T* method. (j) Reconstruction with *L2-INC-T* method. (k) Mask with *L2-INC-H* method. (l) Reconstruction with *L2-INC-H* method.

Figure 2.3: Masks and reconstructions for Table 2.2 when the input image is affected by gaussian noise ($\sigma = 0.03$).



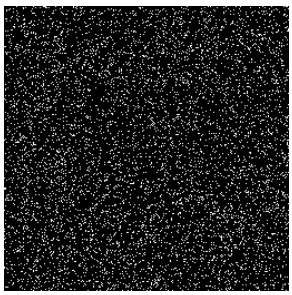
(a) Mask with *H1-T* method.

(b) Reconstruction with *H1-T* method.



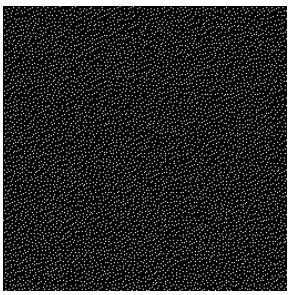
(c) Mask with *H1-H* method.

(d) Reconstruction with *H1-H* method.



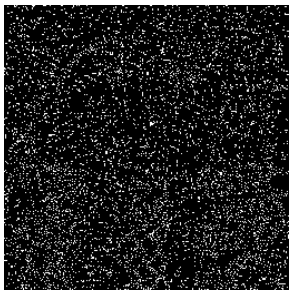
(e) Mask with *L2-INSTA-T* method.

(f) Reconstruction with *L2-INSTA-T* method.



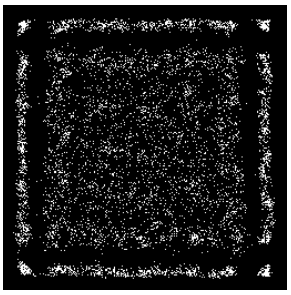
(g) Mask with *L2-INSTA-H* method.

(h) Reconstruction with *L2-INSTA-H* method.



(i) Mask with *L2-INC-T* method.

(j) Reconstruction with *L2-INC-T* method.



(k) Mask with *L2-INC-H* method.

(l) Reconstruction with *L2-INC-H* method.

Figure 2.4: Masks and reconstructions for Table 2.2 when the input image is affected by gaussian noise ($\sigma = 0.05$).

2.6 Reconstructions for Image Denoising from Section 2.3

(a) *L2-INSTA-T* ($\sigma = 0.03$). (b) *L2-INSTA-T* ($\sigma = 0.05$). (c) *L2-INSTA-T* ($\sigma = 0.1$). (d) *L2-INSTA-T* ($\sigma = 0.2$).

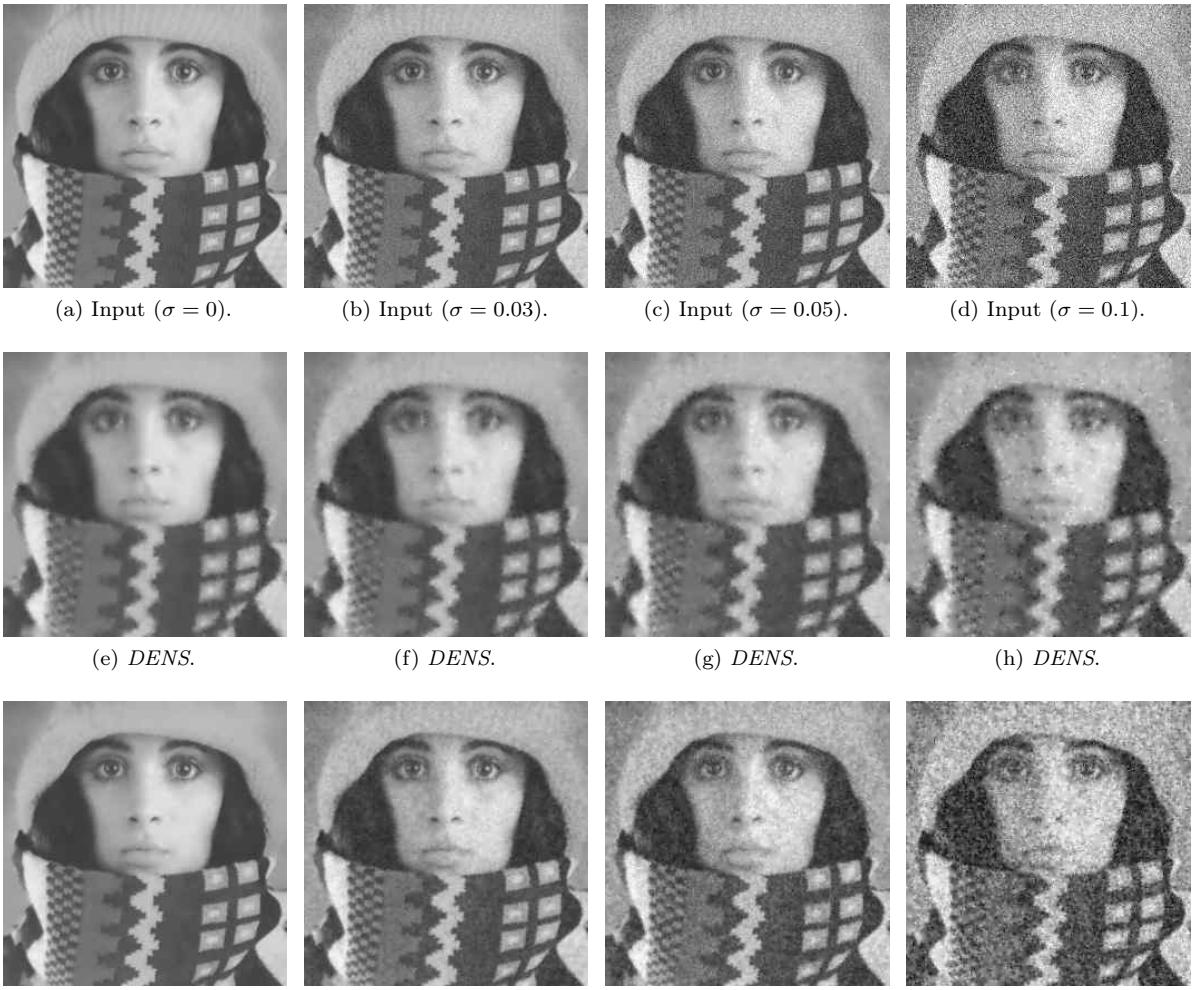
(e) *L2-INSTA-H* ($\sigma = 0.03$). (f) *L2-INSTA-H* ($\sigma = 0.05$). (g) *L2-INSTA-H* ($\sigma = 0.1$). (h) *L2-INSTA-H* ($\sigma = 0.2$).

(i) *L2-INC-T* ($\sigma = 0.03$). (j) *L2-INC-T* ($\sigma = 0.05$). (k) *L2-INC-T* ($\sigma = 0.1$). (l) *L2-INC-T* ($\sigma = 0.2$).

(m) *L2-INC-H* ($\sigma = 0.03$). (n) *L2-INC-H* ($\sigma = 0.05$). (o) *L2-INC-H* ($\sigma = 0.1$). (p) *L2-INC-H* ($\sigma = 0.2$).

Figure 2.5: u^N as a denoised version of the input image for multiple levels of noise.

2.7 Inpainting Masks and Reconstructions for the new model from Section 2.4



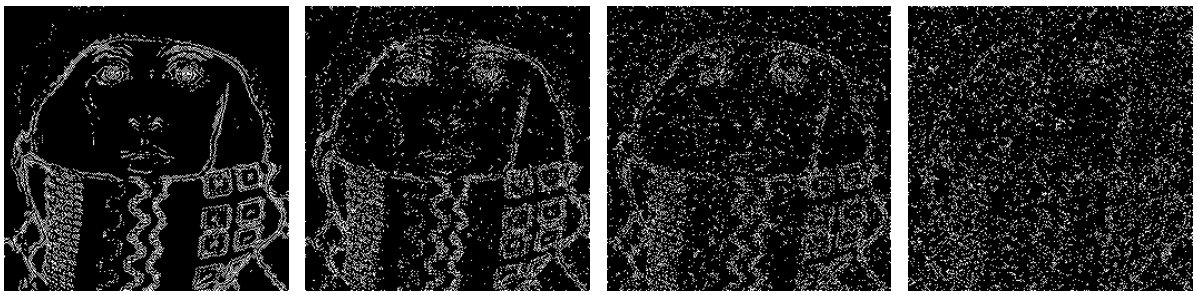
(a) Input ($\sigma = 0$). (b) Input ($\sigma = 0.03$). (c) Input ($\sigma = 0.05$). (d) Input ($\sigma = 0.1$).

(e) *DENS.* (f) *DENS.* (g) *DENS.* (h) *DENS.*

(i) *SPAR.* (j) *SPAR.* (k) *SPAR.* (l) *SPAR.*

Figure 2.6: Reconstruction with the *sparsification* and *densification* methods with 10% of total pixels saved.

(a) Input ($\sigma = 0$). (b) Input ($\sigma = 0.03$). (c) Input ($\sigma = 0.05$). (d) Input ($\sigma = 0.1$).



(e) Mask. (f) Mask. (g) Mask. (h) Mask.

(i) Data u^N. (j) Data u^N. (k) Data u^N. (l) Data u^N.

(m) Reconstruction. (n) Reconstruction. (o) Reconstruction. (p) Reconstruction.

Figure 2.7: Reconstruction with *L2-INC-T-E* method with 10% of total pixels saved.

Chapter 3

Adjoint Method in PDE-based Image Compression

Submitted

Contents

Abstract

We consider a shape optimization based method for finding the best interpolation data in the compression of images with noise. The aim is to reconstruct missing regions by means of minimizing a data fitting term in an L^p-norm between original images and their reconstructed counterparts using linear diffusion PDE inpainting. Reformulating the problem as a constrained optimization over sets (shapes), we derive the topological asymptotic expansion of the considered shape functionals with respect to the insertion of small ball (a single pixel) using the adjoint method. Based on the achieved distributed topological shape derivatives, we propose a numerical approach to determine the optimal set and present numerical experiments showing, the efficiency of our method. Numerical computations are presented that confirm the usefulness of our theoretical findings for non-stationary PDE-based image compression.

keywords image compression – shape optimization – adjoint method – image interpolation – inpainting – PDEs – image denoising

Introduction and Related Works

PDE-based methods have attracted growing interest by researchers and engineers in image analysis field during the last decades [129, 37, 169, 113, 171, 20, 145, 102, 100, 2]. Actually, such methods have reached their maturity both from the point of view of modeling and scientific computing allowing them to be used in modern image technologies and their various applications. Image compression is one of the domain where they appear among the state-of-the-art methods [39, 66, 147, 9, 134, 8, 115]. In fact, the aim for

such problems is to store few pixels of a given image (coding phase) and to recover/restore the missing part in an accurate way (decoding). The PDE-based methods use a diffusion differential operator for the inpainting of missed parts from a available data (boundary or small parts of the initial image) therefore their efficiency for decoding is guaranteed/encoded in the operator without any pre- or post-treatment. The question then is how to ensure with these methods a good choice, if it exists, of the "best" pixels to store for high quality reconstruction of the entire image? An answer to this question is given in [15, 18] for the harmonic or the heat equation, where its reformulation as a constrained (shape) optimisation problem permitted to exhibit an optimal set of pixels to do the job. In addition, analytic selection criteria using topological asymptotics were derived. Due to the simple structure of the shape functionals considered in these previous works, the topological expansion is easily derived (more or less with formal computations) and gives an analytic criterion to characterize the optimal set in compression. The limitation in obtaining the topological expansion this way is twofold : the criterion gives pointwise information on the importance of the location (pixel) to store which results in hard thresholding selection strategy not robust with respect to the noise. Second, the technique is limited to simple functionals, namely an L^2 data-fitting term and a linear diffusion operator.

In this chapter, we consider the compression problem in the same framework but we introduce a new approach to the characterization of the set of pixels to store based on the use of the adjoint method [77, 68, 6, 14]. This approach to obtain the topological expansion is more general than the one previously studied, in the sense that it may be used for other diffusion operators and nonlinear data-fitting term, moreover it allows a better stability with respect to noise for the selection criteria. In particular, when the accuracy of the reconstruction (fidelity term) is measured with an L^p-norm, $p > 1$ and $p \neq 2$, the adjoint problem is still linear and no further complexity is added to the considered problem. Thus, the main results in the chapter includes the rigorous derivation of the topological expansion based on the adjoint problem. We follow mainly the approaches developed by [68] and [6], however, the Dirichlet boundary in the inclusion itself and te form of the elliptic operator, prevents from a direct transposition of their adjoint method based on a local perturbation of the material properties. Therefore, we extend the sensibility analysis to the problem under consideration and we perform the asymptotic expansion of the proposed shape functional using non-standard perturbation techniques combined with a truncation technique. The asymptotic expansion allows us to deduce a gradient algorithm.

Organization of the Chapter

The chapter is organized as follows : in Section 3.1, we introduce the compression problem that takes the form of a constrained optimization problem of finding the best set of pixels to store, denoted K. Section 3.2 is devoted to describe the adjoint method to compute the topological derivative of the cost functionals considered. In Section 3.3, we perform the computations to obtain the topological expansion and the "shape" derivatives which involve the direct and adjoint states. Finally, in Section 3.4, we describe the resulting algorithm and we give some numerical results to confirm the usefulness of the theory. Some of the technical proofs and auxiliary estimates are given in appendices for ease of readability.

3.1 Problem Formulation

Let $D \subset \mathbb{R}^2$ and $f : D \to \mathbb{R}^d$, $d \geq 1$ a given image in some region $K \subset\subset D$. We consider the mixed elliptic boundary problem for a given u_0 in $L^2(D)$,

Problem 3.1.1. *Find u in $H^1(D)$ such that*

$$\begin{cases} u - \alpha\Delta u = u_0, & \text{in } D \setminus K, \\ u = f, & \text{in } K, \\ \frac{\partial u}{\partial \mathbf{n}} = 0, & \text{on } \partial D. \end{cases} \tag{3.1}$$

where the available data f is a Dirichlet "boundary" condition and with homogeneous Neumann boundary condition on ∂D. This PDE corresponds to the first term in the time discretization of the homogeneous

heat equation, where we assume that the initial condition is u_0. For compatibility condition with the "boundary" data on K, we take as u_0 the image $f \in H^1(D)$, with $\Delta f \in L^2(D)$ and such that $\frac{\partial f}{\partial n} = 0$ on ∂D. Setting $v = u - f$, we can write equivalently

Problem 3.1.2. *Find v in $H^1(D)$ such that*

$$\begin{cases} v - \alpha\Delta v = \alpha\Delta f, & \text{in } D \setminus K, \\ v = 0, & \text{in } K, \\ \frac{\partial v}{\partial \mathbf{n}} = 0, & \text{on } \partial D. \end{cases} \tag{3.2}$$

Denoting by $v_K = u_K - f$ the solution of Problem 3.1.2, the question is to identify the region K which gives the "best" approximation u_K, in a suitable sense, that is to say which minimizes some L^p-norm. The constrained optimization problem for the compression reads [15], for $p > 1$,

$$\min_{K \subseteq D,\ m(K) \leq c} \Big\{ \frac{1}{p} \int_D |u_K - f|^p \, dx \ \Big|\ u_K \text{ solution of Problem 3.1.1} \Big\}, \tag{3.3}$$

where m is a "measure". We notice that because of the non differentiability of the functional when $p = 1$, we may apply the following analysis by regularizing in an usual way the function $v \mapsto |v|$. The optimization problem (3.3) is studied in [15] and the existence of an optimal set is established when m is the capacity of sets [176]. To characterize this set K, we aim to compute the topological gradient [38, 6] of the shape functional. Let $x_0 \in D$ and $K_\varepsilon = K \cup \overline{B(x_0, \varepsilon)}$ ($B(x_0, \epsilon)$ denotes the ball centred at x_0 with radius ε), then we look for an expansion of the form

$$J(u_{K_\varepsilon}) - J(u_K) = \rho(\varepsilon)\, G(x_0) + o\big(\rho(\varepsilon)\big).$$

where ρ is a positive function going to zero with ε and G is the so called topological gradient [6, 14, 68]. Therefore, to minimize the cost functional J, one has to create small holes at the locations x where $G(x)$ is the most negative. For the compression problem this amount to select the location where the pixel is the most important to keep.

3.2 Topological Derivative with Adjoint Method

We introduce the following abstract result which describes an adjoint method for the computation of the first variation of a given cost functional (see for instance [6]). Let V be a Hilbert space. We recall the involved norms in Section B.1 in Appendix B. For $\varepsilon \in [0, \zeta]$, $\zeta > 0$, we consider a symmetric bilinear form $a_\varepsilon : V \times V \to \mathbb{R}$ and a linear form $l_\varepsilon : V \to \mathbb{R}$ such that the following assumptions are fulfilled :

- $|a_\varepsilon(v, w)| \leq M_1 \|v\| \|w\|$, $\forall (v, w) \in V \times V$ (**continuity of the bilinear form**),
- $a_\varepsilon(v, v) \geq \alpha \|v\|^2$, $\forall v \in V$ (**uniform coercivity**),
- $|l_\varepsilon(w)| \leq M_2 \|w\|$, $\forall w \in V$ (**continuity of the linear form**),

with $\alpha, M_1, M_2 > 0$ independent of ε. Moreover, we suppose that there exists a continuous bilinear form $\delta a : V \times V \to \mathbb{R}$, a continuous linear form $\delta l : V \to \mathbb{R}$ and a function $\rho : \mathbb{R}_+ \to \mathbb{R}_+$ such that, for all $\varepsilon \geq 0$,

- $\|a_\varepsilon - a_0 - \rho(\varepsilon)\, \delta a\|_{\mathcal{L}_2(V)} = o\big(\rho(\varepsilon)\big)$,
- $\|l_\varepsilon - l_0 - \rho(\varepsilon)\, \delta l\|_{\mathcal{L}(V)} = o\big(\rho(\varepsilon)\big)$,
- $\lim_{\varepsilon \to 0} \rho(\varepsilon) = 0$.

We emphasis that δa and δl do not depend on ε. Finally, for all $\varepsilon \in [0, \zeta]$, consider a functional $J_\varepsilon : V \to \mathbb{R}$, Fréchet-differentiable at the point v_0. Assume further that there exists a number $\delta J(v_0)$ such that

$$J_\varepsilon(w) - J_0(v) = DJ_0(v)(w - v) + \rho(\varepsilon)\, \delta J(v) + o\big(\|w - v\| + \rho(\varepsilon)\big),\ \forall (v, w) \in V \times V.$$

Then we have [6]

Theorem 3.2.1. *Let $v_\varepsilon \in V$ be the solution of the following problem : find $v \in V$ such that,*

$$a_\varepsilon(v, \varphi) = l_\varepsilon(\varphi),\ \forall \varphi \in V.$$

Let w_0 be the solution of the so-called adjoint problem : find $w \in V$ such that

$$a_0(w, \varphi) = -DJ_0(v_0)\varphi,\ \forall \varphi \in V.$$

Then,

$$J_\varepsilon(v_\varepsilon) - J_0(v_0) = \rho(\varepsilon)\big(\delta a(v_0, w_0) - \delta l(w_0) + \delta J(v_0)\big) + o\big(\rho(\varepsilon)\big).$$

To be more specific, for $x_0 \in D$ and $r > 0$, we denote by B_r the open ball centred at x_0 and of radius r. We set

$$V_\varepsilon := \{v \in H^1(D \setminus B_\varepsilon) \mid v = 0 \text{ on } \partial B_\varepsilon\}.$$

Then we consider the boundary value problem :

Problem 3.2.1. *Find $\widetilde{v}_\varepsilon$ in V_ε such that*

$$\begin{cases} -\alpha\Delta\widetilde{v}_\varepsilon + \widetilde{v}_\varepsilon = h, & \text{in } D \setminus B_\varepsilon, \\ \widetilde{v}_\varepsilon = 0, & \text{in } B_\varepsilon, \\ \partial_n\widetilde{v}_\varepsilon = 0, & \text{on } \partial D. \end{cases} \tag{3.4}$$

with $h := \alpha\Delta f$, but h can be any $L^2(D)$ function. We denote $\widetilde{v}_0$ the solution of the problem

Problem 3.2.2. *Find $\widetilde{v}_0$ in $H^1(D)$ such that*

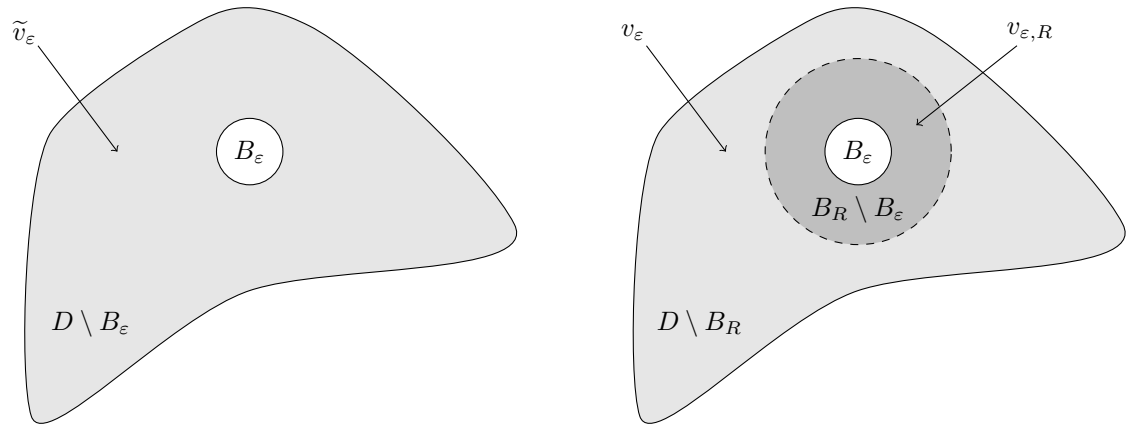
$$\begin{cases} -\alpha\Delta\widetilde{v}_0 + \widetilde{v}_0 = h, & \text{in } D, \\ \partial_n\widetilde{v}_0 = 0, & \text{on } \partial D. \end{cases} \tag{3.5}$$

The dependency of the space V_ε on ε prevents us from using Theorem 3.2.1 directly, therefore, we introduce a truncation technique [77], which consists of inserting a ball B_R, for a fixed $R > \zeta$ and splitting Problem 3.2.1 into two sub-problems that we glue at their common boundary (see Figure 3.1). More precisely, we consider the sub-problems : an *internal* problem

$$\begin{cases} -\alpha\Delta v_{\varepsilon,R} + v_{\varepsilon,R} = h, & \text{in } B_R \setminus B_\varepsilon, \\ v_{\varepsilon,R} = 0, & \text{on } \partial B_\varepsilon, \\ v_{\varepsilon,R} = v_\varepsilon, & \text{on } \partial B_R, \end{cases}$$

and an *external* problem

$$\begin{cases} -\alpha\Delta v_\varepsilon + v_\varepsilon = h, & \text{in } D \setminus B_R, \\ \partial_n v_\varepsilon = \partial_n v_{\varepsilon,R}, & \text{on } \partial B_R, \\ \partial_n v_\varepsilon = 0, & \text{on } \partial D. \end{cases}$$



(a) Before splitting.

(b) After splitting.

Figure 3.1: Illustration of the splitting.

As the two sub-problems transform the initial one into a transmission problem. We have

Proposition 3.2.1. *We have,*

$$\widetilde{v}_\varepsilon = \begin{cases} v_\varepsilon, & \text{in } D \setminus B_R, \\ v_{\varepsilon,R}, & \text{in } B_R \setminus B_\varepsilon. \end{cases}$$

For the *internal* problem, we introduce the notation $v_\varepsilon^{h,\phi}$ instead of $v_{\epsilon,R}$, the solution of the more general problem

Problem 3.2.3. *Find* $v_\varepsilon^{h,\phi}$ *in* $\{v \in H^1(B_R \setminus B_\varepsilon) \mid v = 0 \text{ on } \partial B_\varepsilon\}$ *such that*

$$\begin{cases} -\alpha\Delta v_\varepsilon^{h,\phi} + v_\varepsilon^{h,\phi} = h, & \text{in } B_R \setminus B_\varepsilon, \\ v_\varepsilon^{h,\phi} = 0, & \text{on } \partial B_\varepsilon, \\ v_\varepsilon^{h,\phi} = \phi, & \text{on } \partial B_R. \end{cases} \tag{3.6}$$

Therefore, $v_{\varepsilon,R} = v_\varepsilon^{h,\phi}$, when $\phi = v_\varepsilon$. We also notice that,

$$v_\varepsilon^{h,\phi} = v_\varepsilon^{h,0} + v_\varepsilon^{0,\phi}.$$

We remind the Dirichlet-to-Neumann operator $T_\varepsilon : H^{1/2}(\partial B_R) \to H^{-1/2}(\partial B_R)$ by

$$T_\varepsilon(\phi) := \nabla v_\varepsilon^{0,\phi} \cdot n.$$

and we set

$$h_\varepsilon := -\nabla v_\varepsilon^{h,0} \cdot n \in H^{-1/2}(\partial B_R).$$

Hence, setting $V_R = H^1(D \setminus B_R)$, we can rewrite the *external* problem using this operator as following (we still denote by v_ε the solution) :

Problem 3.2.4. *Find* v_ε *in* V_R *such that*

$$\begin{cases} -\alpha\Delta v_\varepsilon + v_\varepsilon = h, & \text{in } D \setminus B_R, \\ -\partial_n v_\varepsilon + T_\varepsilon v_\varepsilon = h_\varepsilon, & \text{on } \partial B_R, \\ \partial_n v_\varepsilon = 0, & \text{on } \partial D. \end{cases} \tag{3.7}$$

For $\varepsilon \in [0, \zeta]$, $R > \zeta$, and v, φ in $V_R := H^1(D \setminus B_R)$, we define

$$a_\varepsilon(v, \varphi) := \alpha \int_{D \setminus B_R} \nabla v \cdot \nabla \varphi \, dx + \alpha \int_{\partial B_R} T_\varepsilon v \, \varphi \, d\sigma + \int_{D \setminus B_R} v \, \varphi \, dx,$$
$$l_\varepsilon(\varphi) := \int_{D \setminus B_R} h \, \varphi \, dx + \alpha \int_{\partial B_R} h_\varepsilon \, \varphi \, d\sigma.$$

So that the associated variational formulation reads : find $v \in V_R$, such that

$$a_\varepsilon(v, \varphi) = l_\varepsilon(\varphi), \qquad \forall \varphi \in V_R.$$

It is easily checked that

Proposition 3.2.2. *We have,*

- a_ε *is symmetric,*
- l_ε *is continuous.*

In the sequel we write the cost functional in a slightly more general form, in particular to cope with the lack of differentiability for $p = 1$. We take as cost function,

$$\widetilde{J}_\varepsilon(\widetilde{v}) := \int_{D \setminus B_\varepsilon} g(x, \widetilde{v}(x)) \, dx, \ \forall \widetilde{v} \in V_\varepsilon,$$

where g is such that :

(H1) for all $x \in$ D, $s \mapsto g(x, s)$ is $\mathcal{C}^1(\mathbb{R})$ and we denote its derivative by g_s,

(H2) for all $x \in$ D, $s \mapsto g_s(x, s)$ is M-Lipschitz continuous,

(H3) $x \mapsto g_s(x, 0)$ is in $L^2(D)$ and $x \mapsto g(x, 0)$ is in $L^p(D)$, $p > 1$.

Under these assumptions, it is readily checked that

Proposition 3.2.3. *For all $(x, s, t) \in D \times \mathbb{R} \times \mathbb{R}$, we have*

- $|g(x, s)| \leq |g(x, 0)| + |g_s(x, 0)\, s| + \frac{M}{2} s^2$,
- $|g_s(x, s)| \leq |g_s(x, 0)| + M\, |s|$,
- $g(x, t) - g(x, s) \leq g_s(x, s)(t - s) + \theta(x, s, t)(t - s)^2$, *with* $|\theta(x, s, t)| \leq \frac{M}{2}$.

We define now the cost functional on V_R as follows : for $v \in V_R$, we set $\widetilde{v}_\varepsilon \in V_\varepsilon$ the extension of v in $D \setminus B_\varepsilon$ such that,

- $\widetilde{v}_\varepsilon|_{D \setminus B_R} = v$,
- $\widetilde{v}_\varepsilon|_{B_R \setminus B_\varepsilon} = v_\varepsilon^{h,\phi}$, $\phi = v$.

We notice that $\widetilde{v}_\varepsilon$ do not satisfy Problem 3.2.1 except if v is the solution of Problem 3.2.4. Then, we may define the restriction of $\widetilde{J}_\varepsilon$ to V_R by :

$$J_\varepsilon(v) := \widetilde{J}_\varepsilon(\widetilde{v}_\varepsilon), \ \forall v \in V_R.$$

3.2.1 The Adjoint Problem and Related Estimates

We state now the adjoint problem associated to Problem 3.2.4 when $\varepsilon = 0$: we denote by w_0 the weak solution in V_R of

$$a_0(w_0, \varphi) = -DJ_0(v_0)\,\varphi,\ \forall \varphi \in V_R,$$

where v_0 is the solution of Problem 3.2.4. The adjoint state w_0 is then the solution of

Problem 3.2.5. *Find w_0 in V_R such that*

$$\begin{cases} -\alpha\Delta w_0 + w_0 = -g_s\big(\cdot, v_0(\cdot)\big), & \text{in } D \setminus B_R, \\ -\partial_n w_0 + T_0 w_0 = h_0, & \text{on } \partial B_R, \\ \partial_n w_0 = 0, & \text{on } \partial D. \end{cases} \tag{3.8}$$

We aim to find δa, δl and δJ from the adjoint method, Theorem 3.2.1. Let $h \in L^2(D)$ and $\phi \in H^{1/2}(\partial B_R)$. We consider the solution $v_\omega^{h,\phi}$ of the following *exterior* problem,

$$\begin{cases} -\alpha\Delta v_\omega^{h,\phi} + v_\omega^{h,\phi} = 0, & \text{in } \mathbb{R}^2 \setminus B_1, \\ v_\omega^{h,\phi} = v_0^{h,\phi}(x_0), & \text{on } \partial B_1, \\ v_\omega^{h,\phi} = 0, & \text{at } \infty. \end{cases}$$

Since $v_\omega^{h,\phi}$ is a radial function, we can explicitly compute it : for $x \in \mathbb{R}^2 \setminus B_1$,

$$v_\omega^{h,\phi}(x) = \frac{1}{K_0(\alpha^{-1/2})}\, v_0^{h,\phi}(x_0)\, K_0(\alpha^{-1/2}|x - x_0|),$$

where K_0 is the modified Bessel function of the second kind [122]. We extend this solution to $\mathbb{R}^2 \setminus \{x_0\}$ (we still denote by $v_\omega^{h,\phi}$ this extension in the sequel) by setting for $x \in \mathbb{R}^2 \setminus \{x_0\}$,

$$v_\omega^{h,\phi}(x) := \frac{1}{K_0(\alpha^{-1/2})}\, v_0^{h,\phi}(x_0)\, K_0(\alpha^{-1/2}|x - x_0|).$$

3.2.2 Variations of the Bilinear Form

We start by giving the asymptotic development of the variations of the bilinear form in the following proposition :

Proposition 3.2.4. *Let $\delta T : H^{1/2}(\partial B_R) \to H^{-1/2}(\partial B_R)$ such that for all ϕ in $H^{1/2}(\partial B_R)$,*

$$\delta T\phi = -\nabla v_\omega^{0,\phi} \cdot n.$$

Let v, w be in V_R, we set,

$$\delta a(v, w) := \alpha \int_{\partial B_R} \delta T v\, w\, d\sigma.$$

Then, for ε small enough, we have,

$$\|a_\varepsilon - a_0 - \varepsilon^{1/2}\,\delta a\|_{\mathcal{L}_2(V_R)} = O(\varepsilon^{1/2}).$$

Proof. Let $v, w \in V_R$, then,

$$\begin{aligned} a_\varepsilon(v, w) - a_0(v, w) &= \alpha \int_{D\setminus B_R} \nabla v \cdot \nabla w\, dx + \int_{D\setminus B_R} v\, w\, dx + \alpha \int_{\partial B_R} T_\varepsilon v\, w\, d\sigma - \alpha \int_{D\setminus B_R} \nabla v \cdot \nabla w\, dx \\ &\quad - \int_{D\setminus B_R} v\, w\, dx - \alpha \int_{\partial B_R} T_0 v\, w\, d\sigma \\ &= \alpha \int_{\partial B_R} (T_\varepsilon - T_0) v\, w\, d\sigma. \end{aligned}$$

To be able to conclude, we need to use Proposition 3.2.5. □

Proposition 3.2.5. *For ε small enough, we have,*

$$\|T_\varepsilon - T_0 - \varepsilon^{1/2}\delta T\|_{\mathcal{L}\left(H^{1/2}(\partial B_R), H^{-1/2}(\partial B_R)\right)} = O\left(\varepsilon^{1/2}\right).$$

Proof. We set for $x \in \mathbb{R}^2 \setminus B_\varepsilon$,

$$\Psi_\varepsilon^{0,\phi} := v_\varepsilon^{0,\phi} - v_0^{0,\phi} + \varepsilon^{1/2}\, v_\omega^{0,\phi}.$$

Thus, $\Psi_\varepsilon^{0,\phi}$ is the solution of

$$\begin{cases} -\alpha\Delta\Psi_\varepsilon^{0,\phi} + \Psi_\varepsilon^{0,\phi} = 0, & \text{in } B_R \setminus B_\varepsilon, \\ \Psi_\varepsilon^{0,\phi} = \varepsilon^{1/2}\, v_\omega^{0,\phi} - v_0^{0,\phi}, & \text{on } \partial B_\varepsilon, \\ \Psi_\varepsilon^{0,\phi} = \varepsilon^{1/2}\, v_\omega^{0,\phi}, & \text{on } \partial B_R. \end{cases}$$

To apply Proposition B.4.3 in Appendix B, we have to verify that $(\varepsilon^{1/2}\, v_\omega^{0,\phi} - v_0^{0,\phi})$ is in $H^{1/2}(\partial B_\varepsilon)$ and that $\varepsilon^{1/2}\, v_\omega^{0,\phi}$ is in $H^{1/2}(\partial B_R)$.

$$\|\varepsilon^{1/2}\, v_\omega^{0,\phi}\|_{1/2,\partial B_R} = C\,\varepsilon^{1/2}\left|v_0^{0,\phi}(x_0)\right|\left\|K_0\left(\alpha^{-1/2}|\cdot - x_0|\right)\right\|_{1/2,\partial B_R} = C'\,\varepsilon^{1/2}\,|v_0^{0,\phi}(x_0)|.$$

We use the maximum principle for $v_0^{0,\phi}$, we have $|v_0^{0,\phi}(x_0)| \le \|\phi\|_{1/2,\partial B_R}$, and then,

$$\|\varepsilon^{1/2}\, v_\omega^{0,\phi}\|_{1/2,\partial B_R} \le C''\,\varepsilon^{1/2}\,\|\phi\|_{1/2,\partial B_R}.$$

Next,

$$\|\varepsilon^{1/2}\, v_\omega^{0,\phi}(\varepsilon\,\cdot) - v_0^{0,\phi}(\varepsilon\,\cdot)\|_{1/2,\partial B_1} \le \|\varepsilon^{1/2}\, v_\omega^{0,\phi}(\varepsilon\,\cdot) - v_0^{0,\phi}(x_0)\|_{1/2,\partial B_1} + \|v_0^{0,\phi}(x_0) - v_0^{0,\phi}(\varepsilon\,\cdot)\|_{1/2,\partial B_1}.$$

$$\begin{aligned}
\|\varepsilon^{1/2}\, v_\omega^{0,\phi}(\varepsilon\,\cdot) - v_0^{0,\phi}(x_0)\|_{1/2,\partial B_1} &\le \|\varepsilon^{1/2}\, v_\omega^{0,\phi}(\varepsilon\,\cdot)\|_{1/2,\partial B_1} + \|v_0^{0,\phi}(x_0)\|_{1/2,\partial B_1} \\
&\le \varepsilon^{1/2}\|v_\omega^{0,\phi}(\varepsilon\,\cdot)\|_{1/2,\partial B_1} + C|v_0^{0,\phi}(x_0)| \\
&\le \varepsilon^{1/2}\|v_\omega^{0,\phi}(\varepsilon\,\cdot)\|_{1,B_1\setminus B_{1/2}} + C\|\phi\|_{1/2,\partial B_R} \\
&\le \varepsilon^{1/2}\|v_\omega^{0,\phi}(\varepsilon\,\cdot)\|_{1,B_1} + C\|\phi\|_{1/2,\partial B_R} \\
&\le \varepsilon^{1/2}\left(\|v_\omega^{0,\phi}(\varepsilon\,\cdot)\|_{0,B_1} + |v_\omega^{0,\phi}(\varepsilon\,\cdot)|_{1,B_1}\right) + C\|\phi\|_{1/2,\partial B_R} \\
&\le \varepsilon^{1/2}\left(\varepsilon^{-1/2}\|v_\omega^{0,\phi}\|_{0,B_\varepsilon} + \varepsilon^{1/2}|v_\omega^{0,\phi}|_{1,B_\varepsilon}\right) + C\|\phi\|_{1/2,\partial B_R} \\
&\le \|v_\omega^{0,\phi}\|_{0,B_1} + \varepsilon|v_\omega^{0,\phi}|_{1,B_1} + C\|\phi\|_{1/2,\partial B_R} \\
&\le C\|\phi\|_{1/2,\partial B_R} + C\varepsilon\|\phi\|_{1/2,\partial B_R} + C\|\phi\|_{1/2,\partial B_R} \\
&\le C\,\|\phi\|_{1/2,\partial B_R}.
\end{aligned}$$

And

$$\begin{aligned}
\|v_0^{0,\phi}(x_0) - v_0^{0,\phi}(\varepsilon\,\cdot)\|_{1/2,\partial B_1} &= \varepsilon\left\|\varepsilon^{-1}\left(v_0^{0,\phi}(x_0) - v_0^{0,\phi}(\varepsilon\,\cdot)\right)\right\|_{1/2,\partial B_1} \\
&\le C\,\varepsilon\,\|v_0^{0,\phi}\|_{\mathcal{C}^1(B_{R/2})} \\
&\le C\,\varepsilon\,\|\phi\|_{1/2,\partial B_R}.
\end{aligned}$$

We can now use Proposition B.4.3 in Appendix B,

$$\begin{aligned}
\|\Psi_\varepsilon^{0,\phi}\|_{1,B_R\setminus B_{R/2}} &\le C\left(\|\varepsilon^{1/2}\, v_\omega^{0,\phi}\|_{1/2,\partial B_R} + e^{-R/(2\varepsilon\sqrt{\alpha})}\|\varepsilon^{1/2}\, v_\omega^{0,\phi}(\varepsilon\,\cdot) - v_0^{0,\phi}(\varepsilon\,\cdot)\|_{1/2,\partial B_1}\right) \\
&\le C'\,\varepsilon^{1/2}\,\|\phi\|_{1/2,\partial B_R}.
\end{aligned}$$

To finish, Proposition B.1.2 in Appendix B gives,

$$\begin{aligned}\|(T_\varepsilon - T_0 - \varepsilon^{1/2}\,\delta T)\phi\|_{-1/2,\partial B_R} &= \|\nabla\big(v_\varepsilon^{0,\phi} - v_0^{0,\phi} + \varepsilon^{1/2}\,v_\omega^{0,\phi}\big)\cdot n\|_{-1/2,\partial B_R}\\ &= \|\nabla\Psi_\varepsilon^{0,\phi}\cdot n\|_{-1/2,\partial B_R}\\ &\le C\,\|\Psi_\varepsilon^{0,\phi}\|_{1,B_R\setminus B_{R/2}}\\ &\le C'\,\varepsilon^{1/2}\,\|\phi\|_{1/2,\partial B_R}.\end{aligned}$$

□

3.2.3 Variations of the Linear Form

Then, we give the asymptotic development of the variations of the linear form in the following proposition :

Proposition 3.2.6. *We set*

$$\delta h = \nabla v_\omega^{h,0}\cdot n.$$

let w be in V_R, we set,

$$\delta l(w) := \alpha\int_{\partial B_R}\delta h\,w\;d\sigma.$$

Then, for ε small enough, we have,

$$\|l_\varepsilon - l_0 - \varepsilon^{1/2}\,\delta l\|_{\mathcal{L}(V_R)} = O(\varepsilon^{1/2}).$$

Proof. Let $w\in V_R$, then,

$$\begin{aligned}l_\varepsilon(w) - l_0(w) &= \int_{D\setminus B_R} hw\;dx + \alpha\int_{\partial B_R} h_\varepsilon w\;d\sigma - \int_{D\setminus B_R} hw\;dx - \alpha\int_{\partial B_R} h_0 w\;d\sigma\\ &= \alpha\int_{\partial B_R}(h_\varepsilon - h_0)\,w\;d\sigma.\end{aligned}$$

To be able to conclude, we need to use Proposition 3.2.7. □

Proposition 3.2.7. *For ε small enough, we have,*

$$\|h_\varepsilon - h_0 - \varepsilon^{1/2}\delta h\|_{-1/2,\partial B_R} = O\big(\varepsilon^{1/2}\big).$$

Proof. We set for $x\in\mathbb{R}^2\setminus B_\varepsilon$,

$$\Psi_\varepsilon^{h,0} := v_\varepsilon^{h,0} - v_0^{h,0} + \varepsilon^{1/2}\,v_\omega^{h,0}.$$

Thus, $\Psi_\varepsilon^{h,0}$ is the solution of

$$\begin{cases} -\alpha\Delta\Psi_\varepsilon^{h,0} + \Psi_\varepsilon^{h,0} = 0, & \text{in } B_R\setminus B_\varepsilon,\\ \Psi_\varepsilon^{h,0} = \varepsilon^{1/2}v_\omega^{h,0} - v_0^{h,0}, & \text{on } \partial B_\varepsilon,\\ \Psi_\varepsilon^{h,0} = \varepsilon^{1/2}\,v_\omega^{h,0}, & \text{on } \partial B_R.\end{cases}$$

To apply Proposition B.4.3 in Appendix B, we have to verify that $(\varepsilon^{1/2}v_\omega^{h,0} - v_0^{h,0})$ is in $H^{1/2}(\partial B_\varepsilon)$ and that $\varepsilon^{1/2}\,v_\omega^{h,0}$ is in $H^{1/2}(\partial B_R)$.

$$\|\varepsilon^{1/2}\,v_\omega^{h,0}\|_{1/2,\partial B_R} = C\,\varepsilon^{1/2}\,|v_0^{h,0}(x_0)| \le C'\,\varepsilon^{1/2}\,\|h\|_{L^2(D)}.$$

Then,

$$\|\varepsilon^{1/2}v_\omega^{h,0}(\varepsilon\,\cdot) - v_0^{h,0}(\varepsilon\,\cdot)\|_{1/2,\partial B_1} \le \|\varepsilon^{1/2}v_\omega^{h,0}(\varepsilon\,\cdot) - v_0^{h,0}(x_0)\|_{1/2,\partial B_1} + \|v_0^{h,0}(x_0) - v_0^{h,0}(\varepsilon\,\cdot)\|_{1/2,\partial B_1}.$$

$$\begin{aligned}\|\varepsilon^{1/2}v_\omega^{h,0}(\varepsilon\,\cdot) - v_0^{h,0}(x_0)\|_{1/2,\partial B_1} &\leq \varepsilon^{1/2}\|v_\omega^{h,0}(\varepsilon\,\cdot)\|_{1/2,\partial B_1} + \|v_0^{h,0}(x_0)\|_{1/2,\partial B_1}\\ &\leq \varepsilon^{1/2}\|v_\omega^{h,0}(\varepsilon\,\cdot)\|_{0,B_1\setminus B_{1/2}} + \varepsilon^{1/2}|v_\omega^{h,0}(\varepsilon\,\cdot)|_{1,B_1\setminus B_{1/2}} + C\,|v_0^{h,0}(x_0)|\\ &\leq \varepsilon^{1/2}\|v_\omega^{h,0}\|_{0,B_\varepsilon\setminus B_{\varepsilon/2}} + |v_\omega^{h,0}|_{1,B_\varepsilon\setminus B_{\varepsilon/2}} + C\,|v_0^{h,0}(x_0)|\\ &\leq \varepsilon^{1/2}\|v_\omega^{h,0}\|_{0,B_R} + |v_\omega^{h,0}|_{1,B_R} + C\,|v_0^{h,0}(x_0)|\\ &\leq C\,|v_0^{h,0}(x_0)|\\ &\leq C\,\|h\|_{L^2(D)}.\end{aligned}$$

And

$$\begin{aligned}\|v_0^{h,0}(x_0) - v_0^{h,0}(\varepsilon\,\cdot)\|_{1/2,\partial B_1} &= \varepsilon\left\|\varepsilon^{-1}\left(v_0^{h,0}(x_0) - v_0^{h,0}(\varepsilon\,\cdot)\right)\right\|_{1/2,\partial B_1}\\ &\leq C\,\varepsilon\,\|v_0^{h,0}\|_{\mathcal{C}^1(B_{R/2})}\\ &\leq C\,\varepsilon\,\|h\|_{L^2(D)}.\end{aligned}$$

We can now apply Proposition B.4.3 in Appendix B,

$$\begin{aligned}\|\Psi_\varepsilon^{h,0}\|_{1,B_R\setminus B_{R/2}} &\leq C\left(\|\varepsilon^{1/2}\,v_\omega^{h,0}\|_{1/2,\partial B_R} + e^{-R/(2\varepsilon\sqrt{\alpha})}\|\varepsilon^{1/2}v_\omega^{h,0}(\varepsilon\,\cdot) - v_0^{h,0}(\varepsilon\,\cdot)\|_{1/2,\partial B_1}\right)\\ &\leq C\,\varepsilon^{1/2}\,\|h\|_{L^2(D)}.\end{aligned}$$

To conclude, Proposition B.1.2 in Appendix B gives,

$$\begin{aligned}\|h_\varepsilon - h_0 - \varepsilon^{1/2}\delta h\|_{-1/2,\partial B_R} &= \|\nabla\big(v_\varepsilon^{h,0} - v_0^{h,0} + \varepsilon^{1/2}\,v_\omega^{h,0}\big)\cdot n\|_{-1/2,\partial B_R}\\ &= \|\nabla\Psi_\varepsilon^{h,0}\cdot n\|_{-1/2,\partial B_R}\\ &\leq C\,\|\Psi_\varepsilon^{h,0}\|_{1,B_R\setminus B_{R/2}}\\ &\leq C\,\varepsilon^{1/2}\,\|h\|_{L^2(D)}.\end{aligned}$$

□

3.2.4 Variations of the Cost Function

To finish, we give the asymptotic development of the variations of the cost function. We start, by giving an estimate of the variations of the solution with the proposition below :

Proposition 3.2.8. *We set*

$$\delta v^{h,\phi} := -v_\omega^{h,\phi}.$$

Then, for ε small enough, we have,

$$\|v_\varepsilon^{h,\phi} - v_0^{h,\phi} - \varepsilon^{1/2}\,\delta v^{h,\phi}\|_{0,B_R\setminus B_\varepsilon} = o(\varepsilon^{1/2}).$$

Proof. We set for $x \in \mathbb{R}^2 \setminus B_\varepsilon$,

$$\Psi_\varepsilon^{h,\phi} := v_\varepsilon^{h,\phi} - v_0^{h,\phi} + \varepsilon^{1/2}\,v_\omega^{h,\phi}.$$

Thus, $\Psi_\varepsilon^{h,\phi}$ is the solution of

$$\begin{cases} -\alpha\Delta\Psi_\varepsilon^{h,\phi} + \Psi_\varepsilon^{h,\phi} = 0, & \text{in } B_R\setminus B_\varepsilon,\\ \Psi_\varepsilon^{h,\phi} = \varepsilon^{1/2}v_\omega^{h,\phi} - v_0^{h,\phi}, & \text{on } \partial B_\varepsilon,\\ \Psi_\varepsilon^{h,\phi} = \varepsilon^{1/2}\,v_\omega^{h,\phi}, & \text{on } \partial B_R.\end{cases}$$

To apply Proposition B.4.3 in Appendix B, we need to check that $(\varepsilon^{1/2}v_\omega^{h,\phi} - v_0^{h,\phi})$ is in $H^{1/2}(\partial B_\varepsilon)$ and that $\varepsilon^{1/2}\,v_\omega^{h,\phi}$ is in $H^{1/2}(\partial B_R)$.

$$\|\varepsilon^{1/2}\,v_\omega^{h,\phi}\|_{1/2,\partial B_R} \leq C\,\varepsilon^{1/2}\left(\|\phi\|_{1/2,\partial B_R} + \|h\|_{L^2(D)}\right).$$

Then,

$$\|\varepsilon^{1/2} v_\omega^{h,\phi}(\varepsilon\,\cdot) - v_0^{h,\phi}(\varepsilon\,\cdot)\|_{1/2,\partial B_1} \le C\left(\|\phi\|_{1/2,\partial B_R} + \|h\|_{L^2(D)}\right).$$

Therefore, we can use Proposition B.4.3 in Appendix B,

$$\begin{aligned}\|\Psi_\varepsilon^{h,\phi}\|_{0,B_R\setminus B_\varepsilon} &\le C\left(\|\varepsilon^{1/2}\, v_\omega^{h,\phi}\|_{1/2,\partial B_R} + e^{-R/(2\varepsilon\sqrt{\alpha})}\|\varepsilon^{1/2} v_\omega^{h,\phi}(\varepsilon\,\cdot) - v_0^{h,\phi}(\varepsilon\,\cdot)\|_{1/2,\partial B_1}\right)\\ &\le C\,\varepsilon^{1/2}\left(\|\phi\|_{1/2,\partial B_R} + \|h\|_{L^2(D)}\right).\end{aligned}$$

□

Now we can state the estimate of the variations of the cost function.

Proposition 3.2.9. *We set for v in V_R,*

$$\delta J(v) := \int_{B_R} g_s(x,\widetilde{v}_0)\,\delta v^{h,v}\,dx - \alpha\int_{\partial B_R}\partial_n w_0\,\delta v^{h,v}\,d\sigma.$$

Then, for ε small enough, we have,

$$J_\varepsilon(v) - J_0(v) - \varepsilon^{1/2}\delta J(v) = o(\varepsilon^{1/2}).$$

Proof. Let $v \in V_R$, we consider its extension in D, namely $\widetilde{v}_0$, and in $D\setminus B_\varepsilon$, namely $\widetilde{v}_\varepsilon$. We remind that $\widetilde{v}_0$ and $\widetilde{v}_\varepsilon$ are not necessarily a solution of Problem 3.2.2 or of Problem 3.2.1, since v is not necessarily a solution of Problem 3.2.4.

$$\begin{aligned}I_\varepsilon &:= J_\varepsilon(v) - J_0(v) - \varepsilon^{1/2}\delta J(v)\\ &= \widetilde{J}_\varepsilon(\widetilde{v}_\varepsilon) - \widetilde{J}_0(\widetilde{v}_0) - \varepsilon^{1/2}\delta J(v)\\ &= \int_{D\setminus B_\varepsilon} g(x,\widetilde{v}_\varepsilon)\,dx - \int_D g(x,\widetilde{v}_0)\,dx - \varepsilon^{1/2}\int_{B_R} g_s(x,\widetilde{v}_0)\,\delta v^{h,v}\,dx + \varepsilon^{1/2}\alpha\int_{\partial B_R}\partial_n w_0\,\delta v^{h,v}\,d\sigma.\end{aligned}$$

Since $\widetilde{v}_\varepsilon = \widetilde{v}_0$ on $D\setminus B_R$,

$$\begin{aligned}I_\varepsilon = \int_{B_R\setminus B_\varepsilon}\left(g(x,\widetilde{v}_\varepsilon) - g(x,\widetilde{v}_0)\right)dx - \int_{B_\varepsilon} g(x,\widetilde{v}_0)\,dx - \varepsilon^{1/2}\int_{B_R} g_s(x,\widetilde{v}_0)\,\delta v^{h,v}\,dx\\ + \varepsilon^{1/2}\alpha\int_{\partial B_R}\partial_n w_0\,\delta v^{h,v}\,d\sigma.\end{aligned}$$

Using Proposition 3.2.3,

$$\begin{aligned}I_\varepsilon &= \int_{B_R\setminus B_\varepsilon} g_s(x,\widetilde{v}_0)(\widetilde{v}_\varepsilon - \widetilde{v}_0)\,dx + \int_{B_R\setminus B_\varepsilon}\theta(x,\widetilde{v}_0,\widetilde{v}_\varepsilon)(\widetilde{v}_\varepsilon-\widetilde{v}_0)^2\,dx - \int_{B_\varepsilon} g(x,\widetilde{v}_0)\,dx\\ &\qquad - \varepsilon^{1/2}\int_{B_R} g_s(x,\widetilde{v}_0)\,\delta v^{h,v}\,dx + \varepsilon^{1/2}\alpha\int_{\partial B_R}\partial_n w_0\,\delta v^{h,v}\,d\sigma\\ &= \int_{B_R\setminus B_\varepsilon} g_s(x,\widetilde{v}_0)(\widetilde{v}_\varepsilon - \widetilde{v}_0 - \varepsilon^{1/2}\delta v^{h,v})\,dx + \varepsilon^{1/2}\int_{B_R\setminus B_\varepsilon} g_s(x,\widetilde{v}_0)\,\delta v^{h,v}\,dx\\ &\qquad + \int_{B_R\setminus B_\varepsilon}\theta(x,\widetilde{v}_0,\widetilde{v}_\varepsilon)(\widetilde{v}_\varepsilon-\widetilde{v}_0)^2\,dx - \int_{B_\varepsilon} g(x,\widetilde{v}_0)\,dx\\ &\qquad - \varepsilon^{1/2}\int_{B_R} g_s(x,\widetilde{v}_0)\,\delta v^{h,v}\,dx + \varepsilon^{1/2}\alpha\int_{\partial B_R}\partial_n w_0\,\delta v^{h,v}\,d\sigma\\ &= \int_{B_R\setminus B_\varepsilon} g_s(x,\widetilde{v}_0)(\widetilde{v}_\varepsilon - \widetilde{v}_0 - \varepsilon^{1/2}\delta v^{h,v})\,dx - \varepsilon^{1/2}\int_{B_\varepsilon} g_s(x,\widetilde{v}_0)\,\delta v^{h,v}\,dx\\ &\qquad + \int_{B_R\setminus B_\varepsilon}\theta(x,\widetilde{v}_0,\widetilde{v}_\varepsilon)(\widetilde{v}_\varepsilon-\widetilde{v}_0)^2\,dx - \int_{B_\varepsilon} g(x,\widetilde{v}_0)\,dx + \varepsilon^{1/2}\alpha\int_{\partial B_R}\partial_n w_0\,\delta v^{h,v}\,d\sigma.\end{aligned}$$

Thus,

$$|I_\varepsilon| \leq \int_{B_R\setminus B_\varepsilon} \left|g_s(x,\widetilde{v}_0)(\widetilde{v}_\varepsilon - \widetilde{v}_0 - \varepsilon^{1/2}\delta v^{h,v})\right| \, dx + \varepsilon^{1/2}\int_{B_\varepsilon} \left|g_s(x,\widetilde{v}_0)\,\delta v^{h,v}\right| \, dx \\ + \frac{M}{2}\int_{B_R\setminus B_\varepsilon} (\widetilde{v}_\varepsilon - \widetilde{v}_0)^2 \, dx + \int_{B_\varepsilon} |g(x,\widetilde{v}_0)| \, dx + \varepsilon^{1/2}\alpha\int_{\partial B_R} \left|\partial_n w_0\,\delta v^{h,v}\right| \, d\sigma.$$

and

- $\displaystyle\int_{B_R\setminus B_\varepsilon} \left|g_s(x,\widetilde{v}_0)(\widetilde{v}_\varepsilon - \widetilde{v}_0 - \varepsilon^{1/2}\delta v^{h,v})\right| \, dx \leq C\,\varepsilon^{1/2}.$
- $\displaystyle\varepsilon^{1/2}\int_{B_\varepsilon} \left|g_s(x,\widetilde{v}_0)\,\delta v^{h,v}\right| \, dx \leq \varepsilon^{1/2}\int_{B_R} \left|g_s(x,\widetilde{v}_0)\,\delta v^{h,v}\right| \, dx \leq C\,\varepsilon^{1/2}.$
- $\displaystyle\int_{B_R\setminus B_\varepsilon} (\widetilde{v}_\varepsilon - \widetilde{v}_0)^2 \, dx = \int_{B_R\setminus B_\varepsilon} \left(\varepsilon^{1/2}\,\delta v^{h,v} + o(\varepsilon^{1/2})\right)^2 dx = o(\varepsilon^{1/2})\int_{B_R\setminus B_\varepsilon} dx \leq o(\varepsilon^{1/2})\int_{B_R} dx$
  $\leq C\,\varepsilon^{1/2}.$
- $\displaystyle\int_{B_\varepsilon} |g(x,\widetilde{v}_0)| \, dx \leq \left(\int_{B_\varepsilon} \left(g(x,\widetilde{v}_0)\right)^2 dx\right)^{1/2}\left(\int_{B_\varepsilon} 1^2 \, dx\right)^{1/2} \leq \left(\int_{B_R} \left(g(x,\widetilde{v}_0)\right)^2 dx\right)^{1/2}\left(\int_{B_\varepsilon} dx\right)^{1/2}$
  $\leq C\,\varepsilon.$
- $\displaystyle\varepsilon^{1/2}\int_{\partial B_R} \left|\partial_n w_0\,\delta v^{h,v}\right| \, d\sigma = C\,\varepsilon^{1/2}.$

Finally,

$$|I_\varepsilon| \leq C\,\varepsilon^{1/2}.$$

□

Proposition 3.2.10. *We have that J_0 is differentiable on V_R and for v, w in V_R, we have, for ε small enough,*

$$J_\varepsilon(w) - J_0(v) = \varepsilon^{1/2}\delta J(v) + DJ_0(v)(w-v) + o(\varepsilon^{1/2} + \|w-v\|_{V_R}).$$

Proof. By definition, $\widetilde{J_0}$ is differentiable on $H^1(D)$ and for all φ in $H^1(D)$,

$$D\widetilde{J_0}(\widetilde{v}_0)\varphi = \int_D g_s(x,\widetilde{v}_0)\,\varphi \, dx.$$

Then, J_0 is differentiable on V_R and if we note $\widetilde{\varphi}$ the extension on $H^1(D)$ of $\varphi \in V_R$ such that $-\alpha\Delta\widetilde{\varphi} + \widetilde{\varphi} = 0$ in B_R, we have

$$DJ_0(v_0)\,\varphi = D\widetilde{J_0}(\widetilde{v}_0)\,\widetilde{\varphi}.$$

Using the previous proposition, we get for v, w in V_R,

$$\begin{aligned} J_\varepsilon(w) - J_0(v) &= J_\varepsilon(w) - J_0(w) + J_0(w) - J_0(v) \\ &= \varepsilon^{1/2}\delta J(w) + o(\varepsilon^{1/2}) + DJ_0(v)(w-v) + o(\|w-v\|_{V_R}) \\ &= \varepsilon^{1/2}\delta J(w) + \varepsilon^{1/2}\delta J(v) - \varepsilon^{1/2}\delta J(v) + DJ_0(v)(w-v) + o(\varepsilon^{1/2} + \|w-v\|_{V_R}) \\ &= \varepsilon^{1/2}\delta J(v) + DJ_0(v)(w-v) + \varepsilon^{1/2}\big(\delta J(w) - \delta J(v)\big) + o(\varepsilon^{1/2} + \|w-v\|_{V_R}). \end{aligned}$$

To conclude, it remains to show that $\varepsilon^{1/2}\big(\delta J(w) - \delta J(v)\big) = o(\varepsilon^{1/2} + \|w - v\|_{V_R})$.

$$\begin{aligned}
\delta J(w) - \delta J(v) &= -\int_{B_R} g_s(x, \widetilde{w}_0)\, v_\omega^{h,w}\, dx + \alpha \int_{\partial B_R} \partial_n w_0\, v_\omega^{h,w}\, d\sigma + \int_{B_R} g_s(x, \widetilde{v}_0)\, v_\omega^{h,v}\, dx \\
&\qquad - \alpha \int_{\partial B_R} \partial_n w_0\, v_\omega^{h,v}\, d\sigma \\
&= -\int_{B_R} g_s(x, \widetilde{w}_0)\, v_\omega^{h,w}\, dx + \int_{B_R} g_s(x, \widetilde{v}_0)\, v_\omega^{h,w}\, dx - \int_{B_R} g_s(x, \widetilde{v}_0)\, v_\omega^{h,w}\, dx \\
&\qquad + \alpha \int_{\partial B_R} \partial_n w_0\, v_\omega^{h,w}\, d\sigma + \int_{B_R} g_s(x, \widetilde{v}_0)\, v_\omega^{h,v}\, dx - \alpha \int_{\partial B_R} \partial_n w_0\, v_\omega^{h,v}\, d\sigma \\
&= \int_{B_R} \big(g_s(x, \widetilde{v}_0) - g_s(x, \widetilde{w}_0)\big)\, v_\omega^{h,w}\, dx + \int_{B_R} g_s(x, \widetilde{v}_0)\, (v_\omega^{h,v} - v_\omega^{h,w})\, dx \\
&\qquad + \alpha \int_{\partial B_R} \partial_n w_0\, (v_\omega^{h,w} - v_\omega^{h,v})\, d\sigma.
\end{aligned}$$

Then,

$$\begin{aligned}
|\delta J(w) - \delta J(v)| &\le \int_{B_R} |(g_s(x, \widetilde{v}_0) - g_s(x, \widetilde{w}_0))\, v_\omega^{h,w}|\, dx + \int_{B_R} |g_s(x, \widetilde{v}_0)\, (v_\omega^{h,v} - v_\omega^{h,w})|\, dx \\
&\qquad + \alpha \int_{\partial B_R} |\partial_n w_0\, (v_\omega^{h,w} - v_\omega^{h,v})|\, d\sigma \\
&\le \int_{B_R} M|\widetilde{v}_0 - \widetilde{w}_0|\, |v_\omega^{h,w}|\, dx + \int_{B_R} M|\widetilde{v}_0|\, |v_\omega^{h,v} - v_\omega^{h,w}|\, dx \\
&\qquad + \alpha \int_{\partial B_R} |\partial_n w_0\, (v_\omega^{h,w} - v_\omega^{h,v})|\, d\sigma.
\end{aligned}$$

Finally,

- $\displaystyle\int_{B_R} M|\widetilde{v}_0 - \widetilde{w}_0|\, |v_\omega^{h,w}|\, dx \le C\, \|v_\omega^{h,w}\|_{L^2(B_R)}\, \|\widetilde{v}_0 - \widetilde{w}_0\|_{L^2(B_R)} \le C\, \|v - w\|_{V_R}.$
- $\displaystyle\int_{B_R} M|\widetilde{v}_0|\, |v_\omega^{h,v} - v_\omega^{h,w}|\, dx \le C\, \|v_\omega^{h,v} - v_\omega^{h,w}\|_{L^2(B_R)} \le C'\, \|v - w\|_{V_R}.$
- $\displaystyle\int_{\partial B_R} |\partial_n w_0\, (v_\omega^{h,w} - v_\omega^{h,v})|\, d\sigma \le C\, \|v_\omega^{h,w} - v_\omega^{h,v}\|_{L^2(\partial B_R)} \le C'\, \|w - v\|_{V_R}.$

We have the result. □

3.3 Algorithm and Numerical Results

We consider the adjoint problem of Problem 3.2.2 :

Problem 3.3.1. *Find $\widetilde{w}_0$ in $H^1(D)$ such that*

$$\begin{cases} -\alpha\Delta\widetilde{w}_0 + \widetilde{w}_0 = -g_s\big(\cdot, \widetilde{v}_0(\cdot)\big), & \text{in } D, \\ \partial_n \widetilde{w}_0 = 0, & \text{on } \partial D. \end{cases} \tag{3.9}$$

Then, we have the following proposition, proved in Section B.2 in Appendix B,

Proposition 3.3.1. *w_0, solution of Problem 3.2.5, is the restriction of $\widetilde{w}_0$ to $D \setminus B_R$.*

From now, we have everything we need to compute the topological gradient using the adjoint method :

Proposition 3.3.2. *For ε small enough, we have,*

$$j(K_\varepsilon) - j(K) = c\,\varepsilon^{1/2}\,\widetilde{v}_0(x_0)\,\widetilde{w}_0(x_0) + o(\varepsilon^{1/2}),$$

with $\widetilde{v}_0$ solution of Problem 3.2.2 and $\widetilde{w}_0$ solution of Problem 3.3.1.

Proof. Unlike in previous proofs, we have that $\widetilde{v}_0$ and $\widetilde{w}_0$ are the solutions of Problem 3.2.2 and Problem 3.3.1 respectively and are the extensions of v_0, solution of Problem 3.2.4, and w_0, solution of Problem 3.2.5, respectively. Since the conditions for the adjoint method are fulfilled, we have,

$$j(\varepsilon) = j(0) + \big(\delta a(v_0, w_0) - \delta l(w_0) + \delta J(v_0)\big)\,\varepsilon^{1/2} + o(\varepsilon^{1/2}).$$

Using that $v_\omega^{h,v_0} = v_\omega^{h,0} + v_\omega^{0,v_0}$,

$$\begin{aligned}
\delta j(x_0) &= \delta a(v_0, w_0) - \delta l(w_0) + \delta J(v_0) \\
&= -\alpha \int_{\partial B_R} \partial_n v_\omega^{0,v_0}\, w_0 \; d\sigma - \alpha \int_{\partial B_R} \partial_n v_\omega^{h,0}\, w_0 \; d\sigma - \int_{B_R} g_s(x, \widetilde{v}_0)\, v_\omega^{h,v_0} \; dx \\
&\qquad + \alpha \int_{\partial B_R} \partial_n w_0\, v_\omega^{h,v_0}|_{\partial B_R} \; dx \\
&= -\alpha \int_{\partial B_R} \partial_n v_\omega^{h,v_0}\, w_0 \; d\sigma - \int_{B_R} g_s(x, \widetilde{v}_0)\, v_\omega^{h,v_0} \; dx + \alpha \int_{\partial B_R} \partial_n w_0\, v_\omega^{h,v_0} \; dx.
\end{aligned}$$

With Proposition 3.2.1 and Proposition 3.3.1,

$$\delta j(x_0) = -\alpha \int_{\partial B_R} \partial_n v_\omega^{h,v_0}\, \widetilde{w}_0 \; d\sigma - \int_{B_R} g_s(x, \widetilde{v}_0)\, v_\omega^{h,v_0} \; dx + \alpha \int_{\partial B_R} \partial_n \widetilde{w}_0\, v_\omega^{h,v_0} \; dx.$$

Since $-\alpha\Delta\widetilde{w}_0 + \widetilde{w}_0 = -g_s\big(\cdot, \widetilde{v}_0(\cdot)\big)$,

$$\begin{aligned}
\delta j(x_0) &= -\alpha \int_{\partial B_R} \partial_n v_\omega^{h,v_0}\, \widetilde{w}_0 \; d\sigma + \int_{B_R} (-\alpha\Delta\widetilde{w}_0 + \widetilde{w}_0) v_\omega^{h,v_0} \; dx + \alpha \int_{\partial B_R} \partial_n \widetilde{w}_0\, v_\omega^{h,v_0} \; dx \\
&= -\alpha \int_{\partial B_R} \partial_n v_\omega^{h,v_0}\, \widetilde{w}_0 \; d\sigma - \alpha \int_{B_R} \Delta\widetilde{w}_0 v_\omega^{h,v_0} \; dx + \int_{B_R} \widetilde{w}_0 v_\omega^{h,v_0} \; dx + \alpha \int_{\partial B_R} \partial_n \widetilde{w}_0\, v_\omega^{h,v_0} \; dx \\
&= -\alpha \int_{\partial B_R} \partial_n v_\omega^{h,v_0}\, \widetilde{w}_0 \; d\sigma + \alpha \int_{B_R} \nabla\widetilde{w}_0 \cdot \nabla v_\omega^{h,v_0} \; dx - \alpha \int_{\partial B_R} \partial_n \widetilde{w}_0\, v_\omega^{h,v_0} \; d\sigma \\
&\qquad + \int_{B_R} \widetilde{w}_0 v_\omega^{h,v_0} \; dx + \alpha \int_{\partial B_R} \partial_n \widetilde{w}_0\, v_\omega^{h,v_0} \; dx \\
&= -\alpha \int_{\partial B_R} \partial_n v_\omega^{h,v_0}\, \widetilde{w}_0 \; d\sigma + \alpha \int_{B_R} \nabla\widetilde{w}_0 \cdot \nabla v_\omega^{h,v_0} \; dx + \int_{B_R} \widetilde{w}_0 v_\omega^{h,v_0} \; dx \\
&= -\alpha \int_{\partial B_R} \partial_n v_\omega^{h,v_0}\, \widetilde{w}_0 \; d\sigma - \alpha \int_{B_R} \widetilde{w}_0 \Delta v_\omega^{h,v_0} \; dx + \alpha \int_{\partial B_R} \partial_n v_\omega^{h,v_0}\, \widetilde{w}_0 \; d\sigma + \int_{B_R} \widetilde{w}_0 v_\omega^{h,v_0} \; dx \\
&= -\alpha \int_{B_R} \widetilde{w}_0 \Delta v_\omega^{h,v_0} \; dx + \int_{B_R} \widetilde{w}_0 v_\omega^{h,v_0} \; dx \\
&= \int_{B_R} (-\alpha\Delta v_\omega^{h,v_0} + v_\omega^{h,v_0})\, \widetilde{w}_0 \; dx \\
&= \frac{1}{E(1)} v_0^{h,v_0}(x_0) \int_{B_R} \big(-\alpha\Delta E(x - x_0) + E(x - x_0)\big)\, \widetilde{w}_0 \; dx \\
&= \frac{1}{E(1)} v_0^{h,v_0}(x_0) \int_{B_R} \delta_{x_0}\, \widetilde{w}_0 \; dx \\
&= \frac{1}{E(1)}\, v_0^{h,v_0}(x_0)\, \widetilde{w}_0(x_0).
\end{aligned}$$

Finally, using Proposition 3.3.1, we have $v_0^{h,v_0}(x_0) = \widetilde{v}_0(x_0)$, and We get the final result. □

We notice that we this expansion, we get the main theoretical result of the paper which might be summarized as follows: to minimize the L^p-error between an image and its reconstruction by linear diffusion inpainting, we have to keep in the mask the points x_0 which the quantity $\widetilde{v}_0(x_0)\,\widetilde{w}_0(x_0)$ that constitute a soft threshold criterion. The previous computations, does not used directly the L^p-error, since it this one does not satisfy the right hypotheses (for $p = 1$). In the next section, we take specific values of p, depending on the nature of the noise we want to consider, and we use an approximation of the L^p-error that fulfills the assumptions.

3.4 Numerical Results

In this section we present numerical results when the cost functional is the L^1-error and then the L^2-error. We denote by f the original image, f_δ the noised version of f and by u the reconstruction from the inpainting mask K and f_δ. We emphasis that the inpainting masks are built from f_δ. In addition, we remind that, since we are doing compression of noised image, f_δ is available in D during the encoding step and the data are only available in K during the decoding step. Therefore, f is equal to f_δ in D in the inpainting Problem 3.1 during the coding phase, but it is equal to 0 in $D \setminus K$ during the decoding step. We denote by *Lp-ADJ-T* the standard adjoint method by doing a hard-thresholding of $-\widetilde{v}_0\,\widetilde{w}_0$ in order to construct K and by *Lp-ADJ-H* the soft-thresholding (or halftoning [161, 61]) of $-\widetilde{v}_0\,\widetilde{w}_0$, in the same spirit as in [15, 18]. The minus in front of the criterion comes from the fact that we need to take the minimum of $\widetilde{v}_0\,\widetilde{w}_0$, despite thresholding keep the maximum. In this section we use the Floyd-Steinberg dithering [61]. Finally, *H1-T* and *H1-H* correspond to the mask proposed in [15], where we take the hard/soft-thresholding of $|\Delta f_\delta|$.

3.4.1 Salt and Pepper Noise

A common way to deal with impulse noise like salt and pepper noise, is to minimize the L^1-error [119, 120]. However, when $p = 1$ (and more generally, when $p < 2$), the L^p-error do not satisfies (H1) and (H2). Instead, we use approximate the L^1-norm, for $\epsilon > 0$, by

$$t \mapsto g(x,t) = \sqrt{t^2 + \epsilon},$$

which satisfies the hypotheses (H1), (H2) and (H3). We give in Table 3.1, Table 3.2 and Table 3.3 the L^1-error for the methods described above and several amount of salt and/or pepper noise (we take $\epsilon = 0.0001$). By design, the *L1-ADJ-H* give the lower L^1-error. Moreover, we see that most of the corrupted pixels are not present in K for the *L1-ADJ*-methods while they are sectioned in the *H1*-ones. This is not surprising since impulse noises induce a high laplacian at the location of the corrupted pixels. On the other hand, the solution $\widetilde{v}_0$ correspond to a blur of the laplacian of f, while $\widetilde{w}_0$ is close to $-|\widetilde{v}_0|$. Formally, the criterion $-\widetilde{v}_0\,\widetilde{w}_0$ is close to $\Delta f\,|\Delta f|$. Moreover, in the *L1-ADJ*-masks, we can distinguish the edges of the image, while we can not with the *H1-* methods. Interestingly, the *L1-ADJ-H* method gives also better visual results than the *H1-H* method when the image does not contain noise.

In Figure 3.3, Figure 3.4, Figure 3.5 and Figure 3.6, the resulting masks and reconstructions are given for different level of noise.

Noise		**L1-ADJ-T**		**L1-ADJ-H**		**H1-T**	**H1-H**
Salt	Pepper	α	$\|f-u\|_1$	α	$\|f-u\|_1$	$\|f-u\|_1$	$\|f-u\|_1$
0%	0%	0.01	7275.29	3.62	**1942.59**	7158.91	2191.40
2%	0%	2.67	4455.48	1.01	**2027.84**	10971.24	9858.13
0%	2%	1.47	4221.39	0.96	**1947.98**	12634.73	10371.10
1%	1%	0.46	3642.00	0.76	**2310.18**	13241.98	6209.26
4%	0%	2.67	4897.72	0.56	**2257.77**	23308.90	20355.04
0%	4%	1.61	4257.87	1.72	**2104.91**	30196.92	24784.78
2%	2%	0.41	5046.38	0.71	**3444.83**	13241.98	12873.81
10%	0%	3.02	10781.51	0.56	**2858.35**	29785.38	27466.28
0%	10%	5.38	6554.43	0.56	**2711.45**	35098.63	32712.00
5%	5%	0.36	7441.63	0.41	**6122.45**	35256.54	27239.32

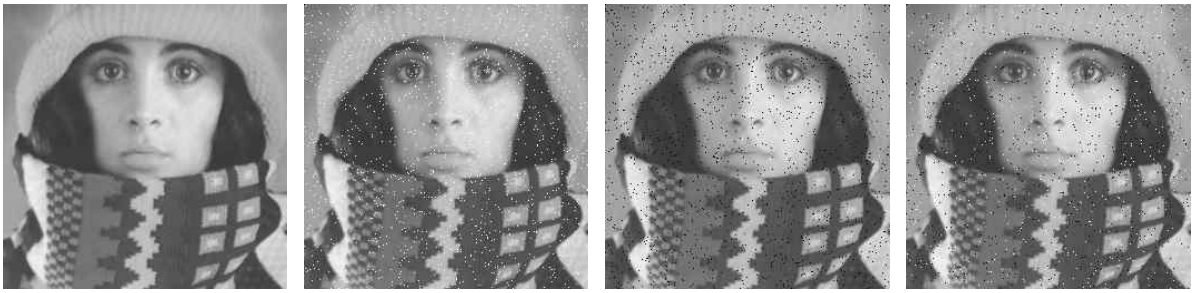
Table 3.1: L^1-error between the original image f and the reconstruction u (build from f_δ) with 5% of total pixels saved.

Noise		**L1-ADJ-T**		**L1-ADJ-H**		**H1-T**	**H1-H**
Salt	Pepper	α	$\|f-u\|_1$	α	$\|f-u\|_1$	$\|f-u\|_1$	$\|f-u\|_1$
0%	0%	0.01	4236.82	2.27	**993.28**	4211.56	1123.36
2%	0%	0.41	2598.67	0.66	**1314.63**	3299.89	3404.45
0%	2%	0.36	2438.89	0.71	**1238.11**	3381.13	3269.42
1%	1%	0.56	2336.34	0.76	**1584.95**	3175.27	3075.79
4%	0%	0.36	3426.40	0.56	**1478.08**	10730.48	8869.47
0%	4%	2.07	2909.84	0.56	**1479.31**	13511.53	10243.84
2%	2%	0.46	3214.27	0.61	**2469.22**	6905.52	6072.67
10%	0%	0.26	7620.77	0.51	**1796.54**	25741.62	22299.13
0%	10%	2.42	4907.27	0.51	**1852.31**	30239.03	27683.34
5%	5%	0.36	6442.23	0.51	**5112.88**	18885.30	15814.61

Table 3.2: L^1-error between the original image f and the reconstruction u (build from f_δ) with 10% of total pixels saved.

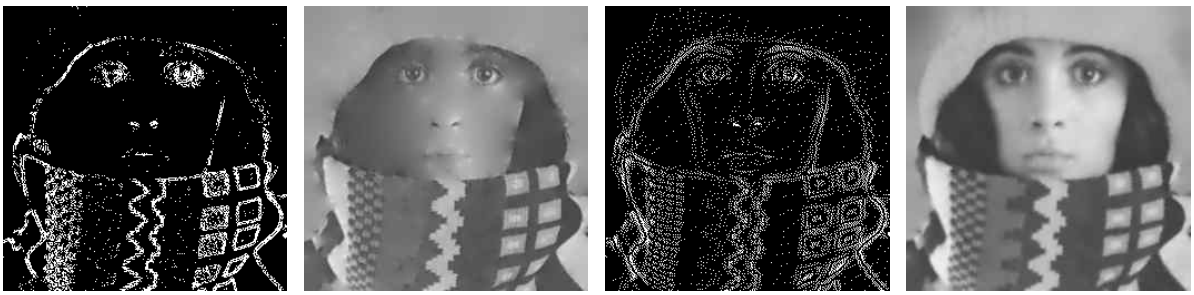
Noise		**L1-ADJ-T**		**L1-ADJ-H**		**H1-T**	**H1-H**
Salt	Pepper	α	$\|f-u\|_1$	α	$\|f-u\|_1$	$\|f-u\|_1$	$\|f-u\|_1$
0%	0%	0.01	2040.62	1.16	**768.31**	1959.49	724.44
2%	0%	0.46	1675.06	0.61	**1061.28**	2183.42	2217.10
0%	2%	0.41	1634.08	0.71	**951.80**	2226.06	2343.02
1%	1%	0.61	1776.52	0.61	**1250.77**	2197.38	2275.70
4%	0%	0.41	2120.61	0.61	**1213.07**	4390.00	4564.17
0%	4%	0.36	2295.56	0.71	**1181.62**	4932.04	4795.51
2%	2%	0.51	2538.10	0.56	**1967.47**	4056.80	4001.66
10%	0%	0.31	5383.94	0.56	**1380.36**	19617.41	16213.35
0%	10%	0.36	3708.22	0.51	**1517.28**	25097.12	20114.90
5%	5%	0.41	5395.05	0.51	**4416.92**	11131.58	10436.74

Table 3.3: L^1-error between the original image f and the reconstruction u (build from f_δ) with 15% of total pixels saved.

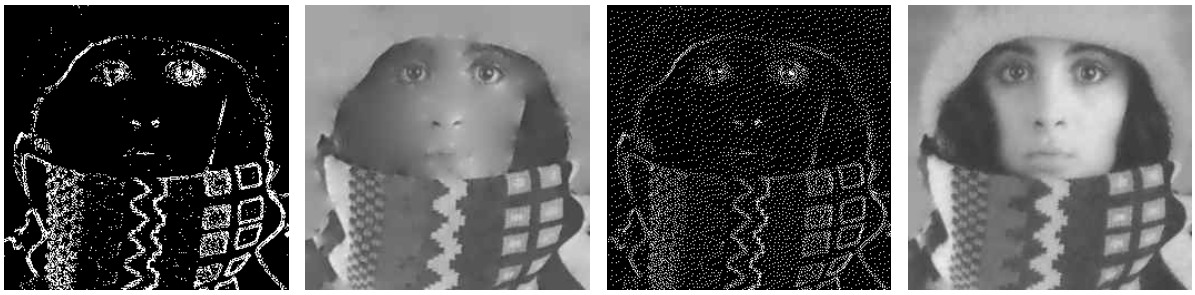


(a) Without noise. (b) With 2% of salt noise. (c) With 2% of pepper noise. (d) With 2% of salt and pepper noise.

Figure 3.2: Input images.



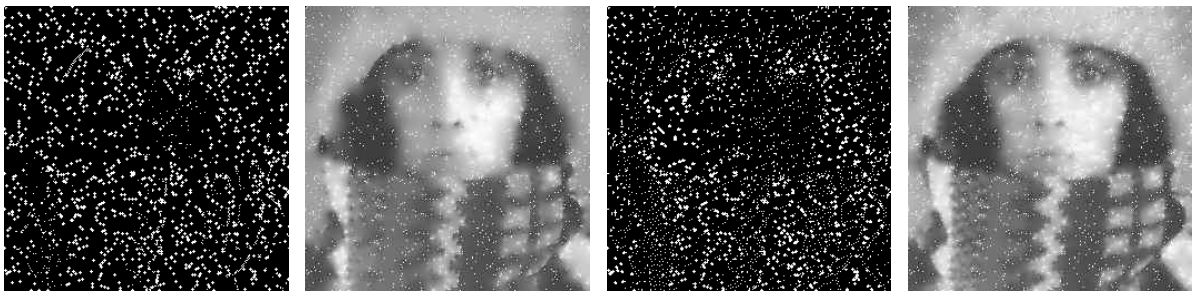
(a) Mask L1-ADJ-T. (b) Reconstruction with L1-ADJ-T. (c) Mask L1-ADJ-H. (d) Reconstruction with L1-ADJ-H.



(e) Mask H1-T. (f) Reconstruction with H1-T. (g) Mask H1-H. (h) Reconstruction with H1-H.

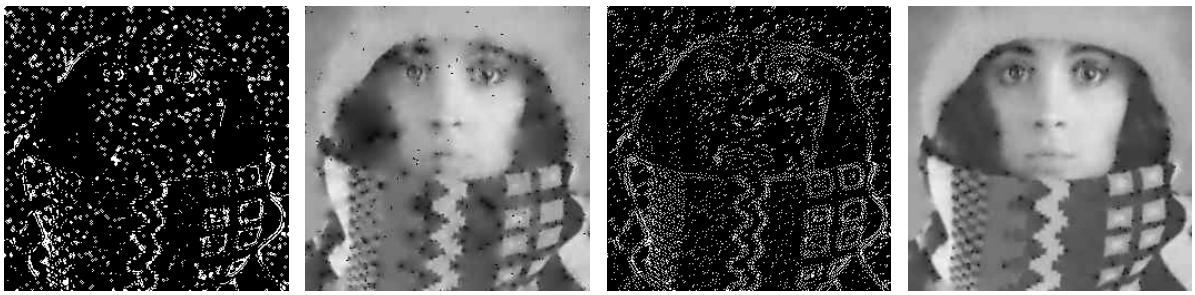
Figure 3.3: Masks and reconstructions from image without noise and with 10% of total pixels saved.

(a) Mask L1-ADJ-T. (b) Reconstruction with L1-ADJ-T. (c) Mask L1-ADJ-H. (d) Reconstruction with L1-ADJ-H.

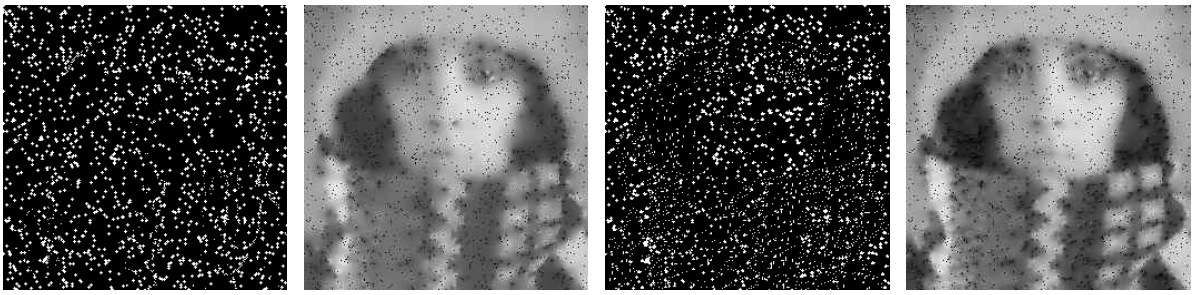


(e) Mask H1-T. (f) Reconstruction with H1-T. (g) Mask H1-H. (h) Reconstruction with H1-H.

Figure 3.4: Masks and reconstructions from image with 2% of salt noise and with 10% of total pixels saved.

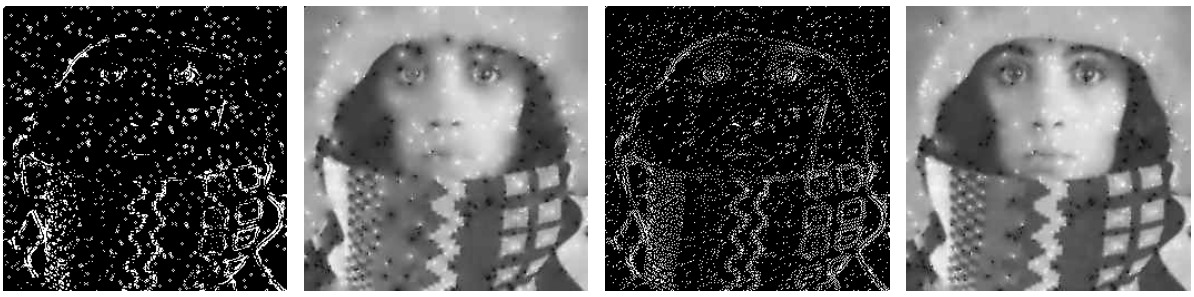


(a) Mask L1-ADJ-T. (b) Reconstruction with L1-ADJ-T. (c) Mask L1-ADJ-H. (d) Reconstruction with L1-ADJ-H.

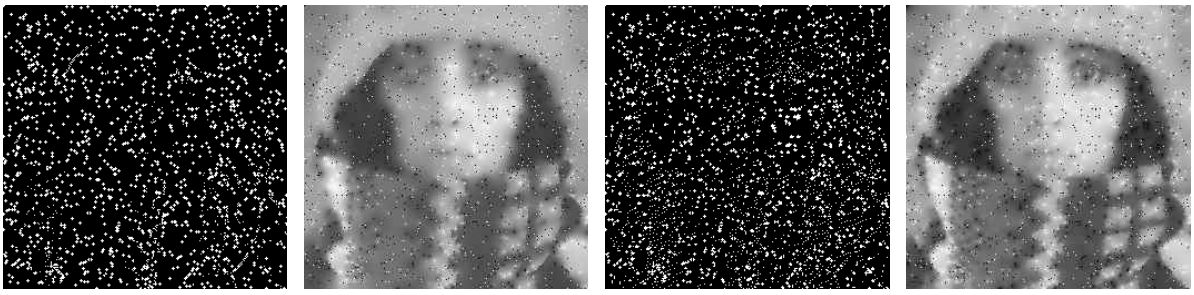


(e) Mask H1-T. (f) Reconstruction with H1-T. (g) Mask H1-H. (h) Reconstruction with H1-H.

Figure 3.5: Masks and reconstructions from image with 2% of pepper noise and with 10% of total pixels saved.



(a) Mask L1-ADJ-T. (b) Reconstruction with L1-ADJ-T. (c) Mask L1-ADJ-H. (d) Reconstruction with L1-ADJ-H.



(e) Mask H1-T. (f) Reconstruction with H1-T. (g) Mask H1-H. (h) Reconstruction with H1-H.

Figure 3.6: Masks and reconstructions from image with 2% of salt and pepper noise and with 10% of total pixels saved.

3.4.2 Gaussian Noise

Now, we consider images with gaussian noise. It is well-known that, a way to remove such a noise, is to minimize the L^2-error and thus we take

$$t \mapsto g(x,t) = \frac{t^2}{2}.$$

We give in Table 3.4, Table 3.5 and Table 3.6 the L^2-error for the methods *L2-ADJ-T*, *L2-ADJ-H*, *H1-T* and *H1-H* with respect to the deviation $\sigma > 0$ of gaussian noise. This time, $\widetilde{v}_0$ is a denoised version of the laplacian of f and $\widetilde{w}_0$ is a smooth version of $-\widetilde{v}_0$. Formally, the criterion $-\widetilde{v}_0\,\widetilde{w}_0$ is close to $|\Delta f|^2$ which is similar to the result found in [18]. We see that for a reasonable level of noise, the *L2-ADJ-H* gives lower L^2-error and that the reconstructed image seems to have less noise than the original one. Like for the

L1-ADJ-methods, we can distinguish the edges of the image in the *L2-ADJ*-masks, while we can not with the *H1*-methods.

Again, we give in Figure 3.8, Figure 3.9 and Figure 3.10 the resulting masks and reconstructions for different level of noise.

Noise	**L2-ADJ-T**		**L2-ADJ-H**		**H1-T**	**H1-H**
σ	α	$\|f-u\|_2$	α	$\|f-u\|_2$	$\|f-u\|_2$	$\|f-u\|_2$
0	0.01	35.02	2.62	16.98	34.46	**11.43**
0.03	0.31	13.94	1.37	**9.62**	16.02	13.95
0.05	0.66	15.88	2.07	**12.56**	19.85	16.64
0.1	1.16	28.58	1.81	**23.40**	30.35	24.24
0.2	0.01	67.07	0.01	52.58	66.36	**41.54**

Table 3.4: L^2-error between the original image f and the reconstruction u (build from f_δ) with 5% of total pixels saved.

Noise	**L2-ADJ-T**		**L2-ADJ-H**		**H1-T**	**H1-H**
σ	α	$\|f-u\|_2$	α	$\|f-u\|_2$	$\|f-u\|_2$	$\|f-u\|_2$
0	0.01	23.08	0.01	9.70	22.99	**6.02**
0.03	0.71	8.88	0.96	**7.76**	11.49	10.38
0.05	0.86	13.42	0.76	**12.50**	15.74	13.85
0.1	0.71	26.81	0.66	24.40	27.01	**23.42**
0.2	0.01	55.74	2.27	47.17	55.32	**42.12**

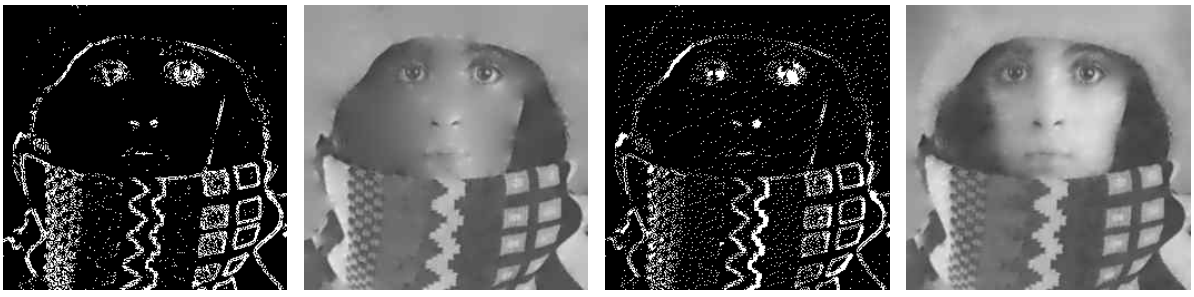
Table 3.5: L^2-error between the original image f and the reconstruction u (build from f_δ) with 10% of total pixels saved.

Noise	**L2-ADJ-T**		**L2-ADJ-H**		**H1-T**	**H1-H**
σ	α	$\|f-u\|_2$	α	$\|f-u\|_2$	$\|f-u\|_2$	$\|f-u\|_2$
0	0.01	11.62	0.01	6.58	11.26	**3.96**
0.03	0.71	7.98	0.56	**7.64**	9.78	8.92
0.05	0.51	12.81	0.66	**12.28**	14.18	12.92
0.1	0.31	25.79	0.76	24.62	25.59	**23.50**
0.2	0.01	50.94	1.11	47.25	50.59	**42.90**

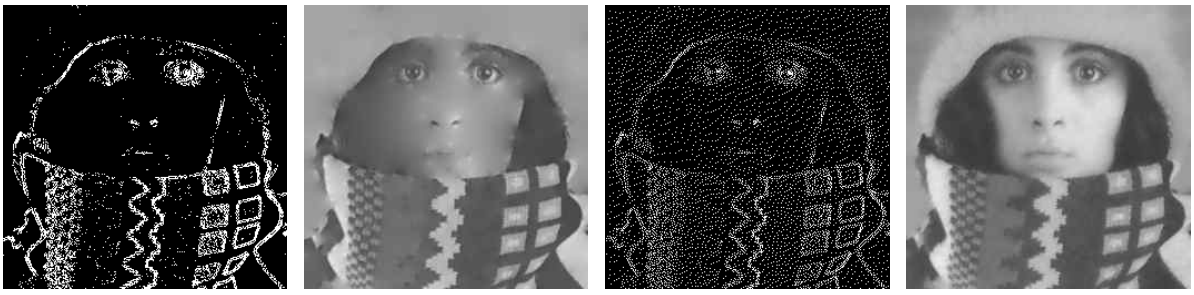
Table 3.6: L^2-error between the original image f and the reconstruction u (build from f_δ) with 15% of total pixels saved.

(a) Without noise. (b) $\sigma = 0.03$. (c) $\sigma = 0.05$. (d) $\sigma = 0.1$.

Figure 3.7: Input images f_δ with gaussian noise of deviation σ.



(a) Mask L2-ADJ-T. (b) Reconstruction with L2-ADJ-T. (c) Mask L2-ADJ-H. (d) Reconstruction with L2-ADJ-H.



(e) Mask H1-T. (f) Reconstruction with H1-T. (g) Mask H1-H. (h) Reconstruction with H1-H.

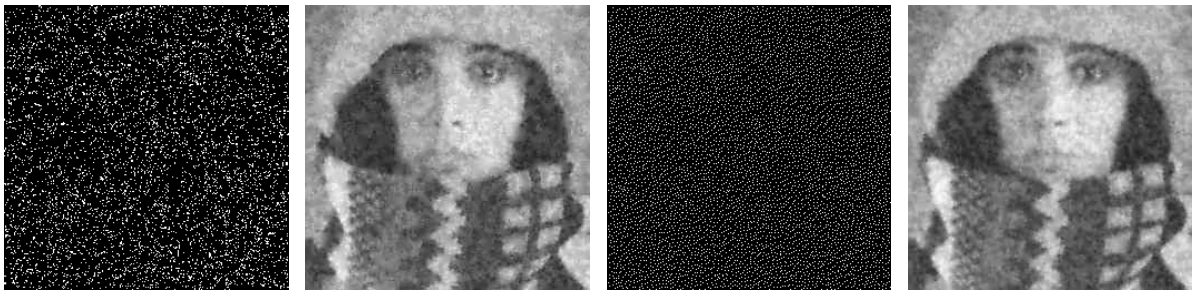
Figure 3.8: Masks and reconstructions from image without noise and with 10% of total pixels saved.

(a) Mask L2-ADJ-T. (b) Reconstruction with L2-ADJ-T. (c) Mask L2-ADJ-H. (d) Reconstruction with L2-ADJ-H.

(e) Mask H1-T. (f) Reconstruction with H1-T. (g) Mask H1-H. (h) Reconstruction with H1-H.

Figure 3.9: Masks and reconstructions from image with gaussian noise of deviation $\sigma = 0.03$ and with 10% of total pixels saved.

(a) Mask L2-ADJ-T. (b) Reconstruction with L2-ADJ-T. (c) Mask L2-ADJ-H. (d) Reconstruction with L2-ADJ-H.



(e) Mask H1-T. (f) Reconstruction with H1-T. (g) Mask H1-H. (h) Reconstruction with H1-H.

Figure 3.10: Masks and reconstructions from image with gaussian noise of deviation $\sigma = 0.1$ and with 10% of total pixels saved.

Conclusion

In this paper, we have formulated the compression problem as a shape optimization one, and we have performed the topological expansion for the optimality condition by the adjoint method.Adapting the approach of [6] and [68], we compute the asymptotic development for variations of the functional which leads to a soft threshold criterion for the choice of the relevant pixels to keep in the mask. Lastly, we present some numerical experiments in the case of the L^2-error and of a regularized L^1-error. In addition, it appears that this new method for selecting the mask outperform the other expansions when the image to compress contains gaussian noise or impulse noise and is easy to implement.

Miscellaneous: A GPU Implementation of PDE-based Image Compression

Contents

In this chapter, we propose an implementation of the compression method described by Belhachmi *et al* together with the homogeneous diffusion inpainting on GPU [15]. Using this technology may allow in the future faster video encoding/decoding with variational methods. Notice that this a work in progress and there is still much to do before achieving a real-time codec by GPU within the framework of PDEs-based methods.

Introduction

Despite inpainting-based compression methods are serious alternatives to the traditional compression standard mainly based on transforms, they suffer from the high computational cost of both the inpainting mask creation step and the inpainting step. More especially, the inpainting mask creation step may involves many inpainting during the process. This limitation prevents them from being applicable to real-time coding/decoding of images.

Several attempts to achieve real-time image inpainting have been proposed over years. Among them, a multigrid solver in case of linear homogeneous diffusion inpaiting implemented on a PlayStation™ 3 have been presented in [99]. To sum up their method, they proposed to solve the inpainting PDE on many coarse grids where the data are smoothed in order to separated the different feature's level, (similar to scale spaces). In 2007, Bornemann and März proposed in [23] a fast inpaiting method base on a non-iterative algorithm that traverses the inpainting domain by the fast marching method while propagating along the way, image values in a coherence direction. The principal common point between these methods is that the algorithm are executed on the CPU.

On the other hand, the use of GPU for scientific computing are becoming more and more popular over years. Indeed, highly parallel solver have been proposed and implemented on GPU for the Navier-Stokes equations or for some optimal transport problem [156, 142]. In particular, Kämper and Weickert proposed in 2022 to use domain decomposition methods to paralleling the solver of the PDE-inpaiting problem. Since GPU have way more processing core than CPU, they proposed to split the problem's domain into many small domains of size 32×32 and to solve each smaller problem in each GPU's core. This way, they achieved to inpaint in real-time 4K images using homogeneous diffusion inpaiting.

The main contribution here is to also use GPU to create the optimal inpainting mask thanks to dithering methods.

Domain Decomposition : The Schwarz's Method

Domain decomposition methods solve boundary value PDE problems by fragmenting the domain Ω into several sub-domains Ω_n, which lead to several sub-problems to solve. To finish, we combine the n solutions of the sub-problems to reconstruct the global solution on Ω. The most known domain decomposition method is the Schwarz's method. For example we consider the following elliptic problem :

$$\begin{cases} Lu = f, & \text{in } \Omega, \\ u = g, & \text{in } \partial\Omega. \end{cases} \tag{3.10}$$

A version said "without overlapping" have been proposed by Lions in 1990 [105]. It is stated as follow : we divide Ω into Ω_1 and Ω_2 such that $\Omega_1 \cap \Omega_2 = \emptyset$ (non overlapping) and we denote by $\Sigma_{1,2}$ is the common boundary between Ω_1 and Ω_2 (see Figure 3.11 (a)). It follows an iterative algorithm to solve the sub-problems which have an additional compatibility condition on the common boundary between Σ.

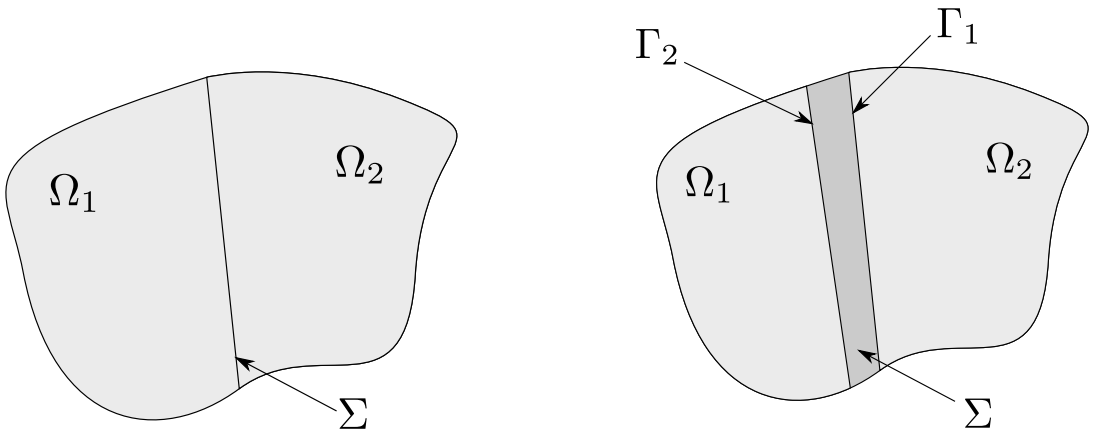
$$\begin{cases} Lu_i^n = f, & \text{in } \Omega_i, \\ u_i^n = g, & \text{on } \partial\Omega_i \setminus \Sigma_{i,j}, \\ \partial_{n_i} u_i^n + \lambda_i u_i^n = \partial_{n_i} u_j^n + \lambda_i u_j^n, & \text{on } \Sigma_{i,j}, \end{cases} \tag{3.11}$$

where $\lambda_i > 0$ is a parameter.

A version said "additive with overlapping" where $\Omega_1 \cap \Omega_2 \neq \emptyset =: \Sigma$ [55]. We denote by Γ_1 (resp. Γ_2) the part of the border of Ω_1 (resp. Ω_2) that is in $\mathring{\Omega}_2$ (resp. $\mathring{\Omega}_1$) (see Figure 3.11 (b)). Unlike the non-overlapping method, the systems of equations have an additional equality condition of the sub-solutions on their common domain Σ. It is possible to solve these non-coupled equations with an iterative algorithm.

$$\begin{cases} Lu_i^n = f, & \text{in } \Omega_i, \\ u_i^n = g, & \text{on } \partial\Omega_i \setminus \Gamma_i, \\ u_i^n = u_j^{n-1}, & \text{in } \Sigma_i, \end{cases} \quad \text{and} \quad \begin{cases} Lu_j^n = f, & \text{in } \Omega_j, \\ u_j^n = g, & \text{on } \partial\Omega_j \setminus \Gamma_j, \\ u_j^n = u_i^{n-1}, & \text{in } \Sigma_j. \end{cases} \tag{3.12}$$

We give in Figure 3.11 an illustration of the non-overlapping and overlapping domain decomposition.



(a) Non-overlapping domains. (b) Overlapping domains.

Figure 3.11: Domain decomposition using the Schwarz's method.

There exist other domain decomposition methods [35, 104, 73].

The main property of these methods is that, every sub-problem can be solved separately and then simultaneously. In order to use parallel computing, the *Message Passing Interface* (MPI) standard have been designed in 1993 for high-performance scientific computing that allows the use of multiple computers. It was made for the passage of messages between remote computers or in a multiprocessor computer. Nowadays, the most used implementation of this standard is the Open MPI project `https://www.open-mpi.org/` which is available on many platforms and programming languages.

GPU Programming

Graphics processing unit (GPGPU) programming is the use of a graphics processing unit (GPU) to perform calculations in applications traditionally handled by the central processing unit (CPU). Essentially, a GPGPU pipeline is a type of parallel processing between one or more GPUs and CPUs that analyzes data as if it were an image. Although GPUs operate at lower frequencies, they typically have several times the number of cores. Thus, GPUs can process much more graphics data per second than a traditional CPU. Migrating the data into graphical form, and then using the GPU to analyze it, can create a significant speedup.

In computing, a compute kernel is a routine compiled for GPUs, separate from but used by a main program running on a CPU.

Some of the operations we use to code our videos can be parallelized and thus programmed as a computational kernel. The computation of the Laplacian, the halftoning method (using dithering rather than error propagation which is by nature sequential [13, 161, 3]) and other elementary operations such as addition or matrix multiplication. We also use methods for numerical solution of linear systems that can be encoded on GPU [126].

Dithering Algorithm

Digital halftoning (or dithering) is a method of rendering the illusion of continuous-tone pictures on displays that are capable of only producing binary picture elements. More precisely, if we denote by $F = [f_{ij}]$ the matrix representing the image and we if give us a dithering mask $M = [f_{kl}]$ which have smaller size than F, then we obtain the binary image (halftoned image) $G = [g_{ij}]$ by

$$g_{ij} := \begin{cases} 1, & \text{if } f_{ij} > m_{kl}, \text{where } k := i \bmod n \text{ and } l := j \bmod m, \\ 0, & \text{otherwise.} \end{cases}$$

We give some examples of dithering mask :

- *Threshold* : M is a 1×1 matrix with the desired threshold $t \geq 0$ i.e. $M = (t)$,
- *Random* : M is a matrix of size $m \times n$ such that each coefficients are randomly chosen between 0 and 1,
- *Bayer matrix* : M is constructed as follow : we first fill each slot with a successive integers, then reorder them such that the average distance between two successive numbers in the map is as large as possible [13].

The dithering technique is known to create periodic pattern in the halftoned image G, also called *dithering artifacts*. To avoid this, dithering masks are generally constructed so that the product $n \times m$ is much larger than the number of distinct threshold levels. Particularly popular dithering masks which respect this rule are blue noise masks [162]. Hereafter is an example of dithering mask :

$$M = \frac{1}{16} \begin{pmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{pmatrix}.$$

We propose to use this halftoning technique instead of the Floyd-Steinberg error propagation dithering algorithm [61] since the two loops over the pixels i and j may be processed by the GPU.

Numerical Results

We propose Figure 3.12 the computational time required to encode/decode *intra frames* as a function of the number of pixels in the image for the Belhachmi method *et al* [15]. For the method *CPU* we use only

one CPU's core for the calculations and for the method *GPU* one will use mainly the *GPU*. We will use mainly the GPU because the solver of linear systems uses the CPU for the calculation of the incomplete LU decomposition of the matrix and uses the GPU for the final resolution of the system of equations. For the GPU programming we use the library *python cupy* using *CUDA*.

The use of the GPU allows to greatly accelerate the computations for the creation of the mask as well as for the resolution of the linear system. On the other hand, in our tests, the creation of the linear system is performed by the CPU for both methods. In the case of the GPU, the A matrix representing the linear system is transferred to the GPU, which explains the difference in computation time for the CPU and GPU methods. Knowing in advance the number of non-zero elements in A, it would be possible to assemble the matrix with the GPU and thus save even more computing time. Here, we have performed tests in the case of *intra frames* but we could also apply it to *inter frames* for the calculation of the optical flow and for the calculation of the prediction.

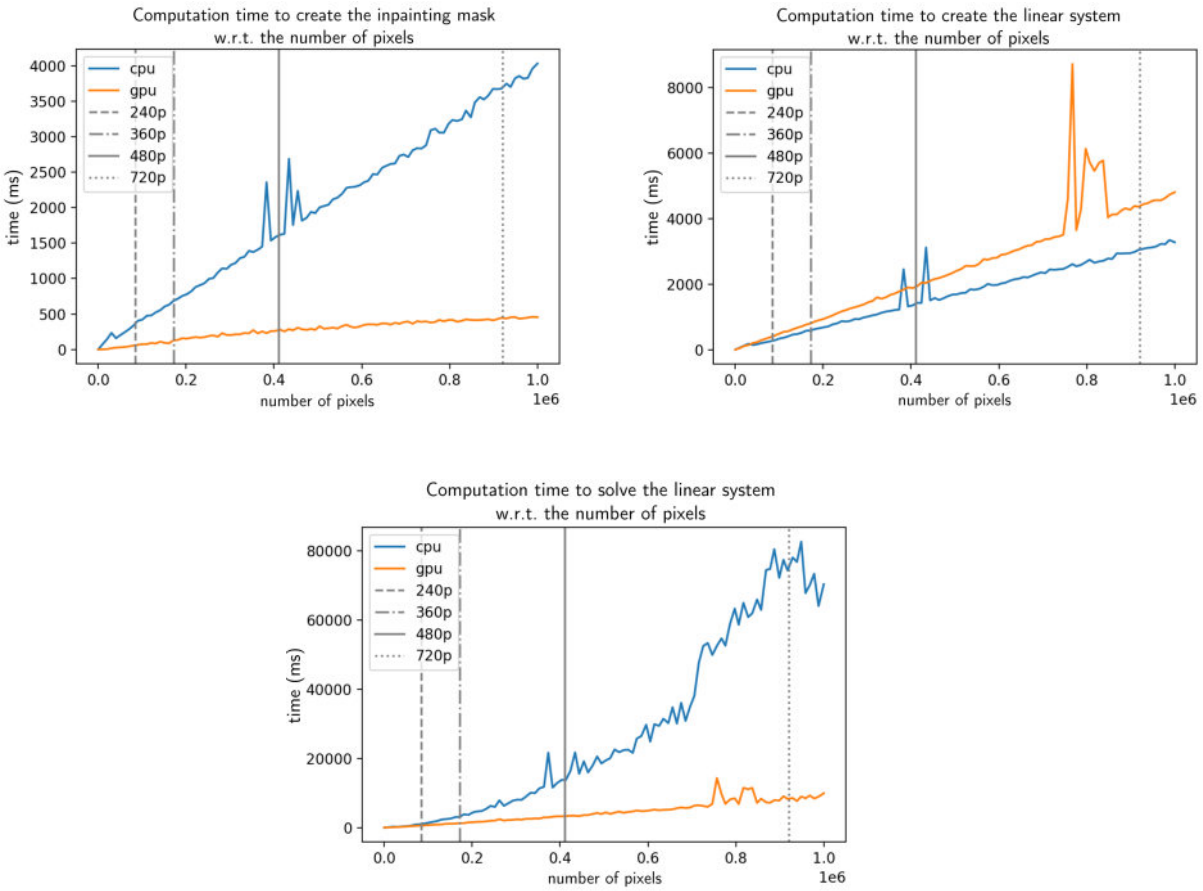


Figure 3.12: Time of the different steps of encoding/decoding of *intra frame* on CPU and GPU.

Conclusion

We proposed an implementation of the compression method based on the soft-thresholding of $|\Delta f|$ as described by Belhachmi *et al* together with the homogeneous diffusion inpainting on GPU [15]. The inpaiting is solved using the CUDA sparse linear solver on GPU proposed by Nvidia while the inpaiting mask is created by the use of GPU kernel. Despite the assembling of the sparse linear system is slower than the CPU-only method due to data migration from CPU to GPU memory, the gain of time computation is significant when the number of pixel in the image increase. The most spectacular gain of computation time is in the mask creation step. Indeed, since this step is fully parallelized, it is processed by the GPU in just one time. To obtain a real-time video codec, it might be interesting to apply this GPU solver on optical flow problems to speed up the motion estimation step of the encoder.

Part II

Motion Estimation by Optical Flow

Classical Formulations of the Optical Flow in Small Displacements

Contents

The optical flow is a field of vectors giving the displacement of the intensity of the pixels of an animated image over time. In the case of a video, when there is no change of scene, two successive images are visually very similar and therefore, the optical flow will be very homogeneous. This is why, rather than saving the two images independently, the authors of [2] propose to compute the optical flow and to sub-sample it. This last step should not have a great influence on the quality of the reconstruction thanks to the regularity of the optical flow.

In this chapter, we will see some classical methods for computing the optical flow.

Fundamental Constraint of the Optical Flow with Constant Luminosity

We define a function f from an image domain $D \subset \mathbb{R}^d$, $d = 2, 3$, to $\mathbb{R}$ representing the intensity of a pixel $\mathbf{x}$ at an instant t by

$$\begin{array}{rccc} f: & D \times [0,1] & \to & \mathbb{R} \\ & (\mathbf{x}, t) & \mapsto & f(\mathbf{x}, t). \end{array}$$

The estimation of the optical flow consists in finding the vector field $\mathbf{u} = (u_i)_{i=1}^d$, called the optic flow, describing the motion of each pixel between two frames of a given sequence. The fundamental constraint of the optical flow is the main assumption for the estimation of the optical flow. We will see only the case where the luminosity is assumed to remain constant over time. The constant luminosity translates into,

$$f(\mathbf{x} + \delta\mathbf{x}, t + \delta t) = f(\mathbf{x}, t).$$

Note that there are other fundamental constraints of the optical flow. Moreover, we only assume the case of small displacements, i.e. when $\delta\mathbf{x}$ is small. We use the Lagrangian coordinates. Since we are in the case of constant luminosity, we assume that the intensity of a point remains constant along its trajectory, i.e. for $(\mathbf{x_0}, t_0)$, there exists a trajectory $t \mapsto (\mathbf{x}(t), t)$, such that $(\mathbf{x}(t_0), t_0) = (\mathbf{x_0}, t_0)$ and for all $t > 0$, $f(\mathbf{x}(t), t) = f(\mathbf{x_0}, t_0)$. In small displacements, it is reasonable to assume that f is differentiable, hence by differentiating with respect to t and taking $t = t_0$, we have,

$$\partial_t f(\mathbf{x_0}, t_0) + \partial_t \mathbf{x}(t_0) \cdot \nabla f(\mathbf{x_0}, t_0) = 0.$$

By setting $\mathbf{u}(\mathbf{x_0}) := \partial_t \mathbf{x}(t_0)$, and since the above equation holds for all $(\mathbf{x_0}, t_0)$, we obtain the *fundamental constraint of the optical flow*,

$$\nabla f(\mathbf{x}, t) \cdot \mathbf{u}(\mathbf{x}, t) + \partial_t f(\mathbf{x}, t) = 0.$$

Variational Formulations

Using only the fundamental constraint of the optical flow is not enough to determine the components of the optical flow $\mathbf{u}$. This problem is called the aperture problem and we have to add a second constraint.

In 1981, Lucas and Kanade [108] proposed the *local method*, which seeks to minimize $\mathbf{u}$, for any $t \in [0,T]$ and $\mathbf{x} \in D$,

$$\mathcal{D}_{\mathrm{l}}(\mathbf{u}) = K_\rho \star \Big(\nabla f(\mathbf{x},t)\cdot\mathbf{u}(\mathbf{x},t) + \partial_t f(\mathbf{x},t)\Big).$$

Here, $\star$ denotes the convolution product and K_ρ denotes the *heat kernel*, fundamental solution of the heat equation, which, when convolved with a function g has the effect of smoothing g. If g is an image, this operation is a *Gaussian blur* of standard deviation $\rho > 0$. The Euler-Lagrange equation gives us in dimension 2,

$$\begin{cases} K_\rho \star (f_x)^2\ u_1 + K_\rho \star (f_x f_y)\ u_2 = -K_\rho \star (f_x f_t), & \text{in } [0,T]\times D,\\ K_\rho \star (f_y f_x)\ u_1 + K_\rho \star (f_y)^2\ u_2 = -K_\rho \star (f_y f_t), & \text{in } [0,T]\times D, \end{cases}$$

The previous linear system can be solved pointwise, which has an advantage in terms of computation time.

The same year, Horn and Schunck proposed the *global method*, which consists in minimizing on $\mathbf{u}$ an energy associated to the *fundamental constraint of the optical flow*,

$$\mathcal{D}_{\mathrm{g}}(u) = \int_D \nabla f(x,t)\cdot\mathbf{u}(\mathbf{x},t) + \partial_t f(\mathbf{x},t)\ dx + \alpha\,\mathcal{R}(\mathbf{u}).$$

If $\mathcal{R}(\mathbf{u}) = \displaystyle\sum_{i=1}^{d}\int_D |\nabla u_i|^2\ dx$, the Euler-Lagrange equation gives us in dimension 2,

$$\begin{cases} -\alpha\Delta u_1 + (f_x)^2\ u_1 + (f_x f_y)\ u_2 = -(f_x f_t), & \text{in } [0,T]\times D,\\ -\alpha\Delta u_2 + (f_y f_x)\ u_1 + (f_y)^2\ u_2 = -(f_y f_t), & \text{in } [0,T]\times D,\\ \frac{\partial u_1}{\partial \mathbf{n}} = \frac{\partial u_2}{\partial \mathbf{n}} = 0, & \text{on } [0,T]\times \partial D. \end{cases}$$

In 2005, Bruhn *et al* [27], proposed the *global-local method*, which is the combination of the two previous methods, and thus minimize $\mathbf{u}$ for all $t \in [0, T-\delta t[$,

$$\mathcal{D}_{\mathrm{gl}}(\mathbf{u}) = \int_D K_\rho(\mathbf{x}) \star \Big(\nabla f(\mathbf{x},t)\cdot\mathbf{u}(\mathbf{x},t) + \partial_t f(\mathbf{x},t)\Big)\ dx + \alpha\,\mathcal{R}(\mathbf{u}).$$

If $\mathcal{R}(\mathbf{u}) = \displaystyle\sum_{i=1}^{N}\int_D |\nabla u_i|^2\ dx$, the Euler-Lagrange equation gives us in dimension 2,

$$\begin{cases} -\alpha\Delta u_1 + K_\rho \star (f_x)^2\ u_1 + K_\rho \star (f_x f_y)\ u_2 = -K_\rho \star (f_x f_t), & \text{in } [0,T]\times D,\\ -\alpha\Delta u_2 + K_\rho \star (f_y f_x)\ u_1 + K_\rho \star (f_y)^2\ u_2 = -K_\rho \star (f_y f_t), & \text{in } [0,T]\times D,\\ \frac{\partial u_1}{\partial \mathbf{n}} = \frac{\partial u_2}{\partial \mathbf{n}} = 0, & \text{on } [0,T]\times \partial D. \end{cases}$$

In this last method, we have two parameters to choose from: ρ and α. The first one is used to make the method more robust to the noise in f, while the second one, associated to the regularization, allows to favor a smooth optical flow. Thus, a high value of α will have the effect of blurring the information close to the edges and a low value of α adds too much useless information in the homogeneous parts. The ideal would be to have α low when we are close to the edges and high in the homogeneous zones.

This is why the adaptive methods of the choice of α have been introduced [16, 17, 72]. In these, we assume that the regularization parameter α is no longer constant but piecewise constant. We use the finite element method, we have α in $\mathbb{P}_0$ and if we note η_K the a-posteriori error on the element K, we set for α^n given,

$$\alpha^{n+1}|_K = \max\left(\frac{\alpha^n|_K}{1+\kappa\max\left(\frac{\eta_K}{\|\eta_K\|_\infty} - \frac{\xi}{100}, 0\right)}, \alpha_{\mathrm{s}}\right).$$

The parameter κ is between 5 and 10 and α_{s} is a threshold parameter to avoid having a too small regularization. With this choice of α^{n+1}, if the relative error is smaller than $\xi\%$ we reduce $\alpha^{n+1}|_K$, otherwise, we keep the same $\alpha^n|_K$.

In practice, we will choose α^0 constant and large since the parameter α can only decrease during the iterations.

Optic Flow Formulation with Varying Illumination

In this section, we recall the classical optic flow formulation with luminosity variations. An approach suggested by Gennert and Negahdaripour [69] for the determination of $\mathbf{u}$ are based on the assumption that the intensity of a pixel is linearly varying between two successive frames. This assumption introduce a new unknown namely the luminosity variation. More precisely, the conservation law is as follows

$$f(\mathbf{x}+\delta\mathbf{x}, t+\delta t) = M(\mathbf{x},t)f(\mathbf{x},t) + C(\mathbf{x},t). \tag{3.13}$$

For small displacements and in many physical situations this assumption is reasonable and coherent with the laws of the optic. Moreover, we can suppose that $C(\mathbf{x},t)$ is close to zero and that $M(\mathbf{x},t)$ is close to the identity. We write $M = I_d + \delta m$ and $C = \delta c$, where I_d is the identity, and δm, δc represent a "small" variation of the illumination. We suppose that $\delta c = 0$. Under these new assumptions, equation (3.13) reads,

$$f(\mathbf{x}+\delta\mathbf{x}, t+\delta t) = \big(I_d + \delta m(\mathbf{x},t)\big) f(\mathbf{x},t).$$

Using Taylor expansion on the left hand side, we obtain

$$\nabla f \cdot \delta\mathbf{x} + \partial_t f \delta t - f\delta m + o(t) = 0.$$

By dividing by δt and let δt tends to zero, we have the optic flow fundamental constraint with luminosity variations for small displacements i.e.

$$\nabla f \cdot \mathbf{u} + \partial_t f - f m_t = 0,$$

with $u_i := \lim_{\delta t\to 0} \frac{\delta x_i}{\delta t}$ and $m_t := \lim_{\delta t\to 0} \frac{\delta m}{\delta t}$. Like without luminosity variations, we need to add constraints to the optic flow fundamental equation by regularizing the unknowns u_1, u_2 and m_t. We use the *global-local* method and we aim to minimize over $(\mathbf{u}, m_t)$ for $t \in [0, 1-\delta t[$,

$$\int_D K_\rho(x) \star \Big(\nabla f \cdot \mathbf{u} + \partial_t f - f m_t\Big)\, dx + \alpha \sum_{i=1}^{d} \int_D |\nabla u_i|^2 \, dx + \lambda \int_D |\nabla m_t|^2 \, dx.$$

In dimension 2, Euler-Lagrange equations leads to the following PDEs system

$$\begin{cases} -\alpha\Delta u_1 + K_\rho \star (f_x)^2\, u_1 + K_\rho \star (f_x f_y)\, u_2 - K_\rho \star (f_x f)\, m_t = -K_\rho \star (f_x f_t), & \text{in } [0,1]\times D,\\ -\alpha\Delta u_2 + K_\rho \star (f_y f_x)\, u_1 + K_\rho \star (f_y)^2\, u_2 - K_\rho \star (f_y f)\, m_t = -K_\rho \star (f_y f_t), & \text{in } [0,1]\times D,\\ -\lambda\Delta m_t + K_\rho \star (f f_x)\, u_1 + K_\rho \star (f f_y)\, u_2 - K_\rho \star (f^2)\, m_t = K_\rho \star (f f_t), & \text{in } [0,1]\times D,\\ \frac{\partial u_1}{\partial \mathbf{n}} = \frac{\partial u_2}{\partial \mathbf{n}} = \frac{\partial m_t}{\partial \mathbf{n}} = 0, & \text{on } [0,1]\times \partial D. \end{cases}$$

This popular method to determine the optic flow uses a smoothing regularization (laplacian) on the components of $\mathbf{u}$ and on m_t along with smoothing the data on f. This leads to homogeneous and dense solutions which may contains too much useless information.

An other way to model the variation of the luminosity is to add a constraint on the constancy of the gradient of luminosity [26],

$$\nabla f(\mathbf{x}+\delta\mathbf{x}, t+\delta t) = \nabla f(\mathbf{x},t).$$

There are other fundamental constraints and regularizations, presented in [124].

As another improvement, we notice that in case of large displacements [72], as can no longer use Taylor expansion, we have to solve a non-linear problem :

$$\min_u \underbrace{\int_D K_\rho \star \big(f(\mathbf{x}+\mathbf{u}, t+\delta t) - f(\mathbf{x},t)\big)^2}_{\text{non-linear}} + \alpha\, \mathcal{R}(\mathbf{u}).$$

Chapter 4

Optimal Transport Model for the Optical Flow Estimation with Varying Illumination

Contents

Abstract

In this chapter, we propose a new model for optical flow computation based on the fluid mechanics using the continuity equation and we make a parallel between the optical flow and optimal transportation problem thanks to the Benamou-Brenier theorem. This new model include brightness variation assumption without the need to suppose other assumptions in order to regularize the optical flow problem since the additional "unknown", namely the brightness variation, is in fact equals to the opposite of the divergence of the optical flow. This parallel between the optical flow and the continuity equation allows us to use the Benamou-Brenier algorithm and we present numerical results to confirm the pertinence of our proposed model.

keywords optical flow – motion estimation – brightness variation – PDEs – optimal transport – Benamou-Brenier formula

Introduction

The optimal mass transport problem seeks the most efficient way of transforming one distribution of mass to another, relative to a given cost function. The first optimal mass transport problem first arose due to Monge [116]. It was later expanded by Kantorovich [95], and found applications in several scientific areas, e.g. operations research and economics.

A significant portion of the theory of the optimal mass transport problem was developed in the Nineties and reaches its maturity for use in a large number of applications. Starting with Brenier's seminal work on characterization, existence, and uniqueness of optimal transport maps [25], followed by Caffarelli's work on regularity conditions of such mappings [34], Gangbo and McCann's work on geometric interpretation of the problem [67], and Evans and Gangbo's work on formulating the problem through partial differential equations and specifically the p-Laplacian equation [58]. A more thorough history and background on the

optimal mass transport problem can be found in Villani's book "Optimal Transport: Old and New" [164], and Santambrogio's book "Optimal transport for applied mathematicians" [144].

The significant contributions in mathematical foundations of the optimal transport problem together with recent advancements in numerical methods [19], have spurred the recent development of numerous data analysis techniques for modern estimation and detection (e.g. classification) problems. In image analysis and vision the application of transport-based techniques to numerous problems including: image retrieval, registration and morphing, color and texture analysis, image denoising and restoration, morphometry, super resolution, and machine learning [123, 22, 144, 158].

In the last decades, the estimation of the *optical flow* gained more and more attention in image processing field [12, 27, 76, 59]. Optical flow may found its application in almost all the movies compression processes, via motion compensation, or for computer vision applications [152, 97, 8]. The problem of motion's detection in an image sequence under constancy assumption on the pixels intensity is an ill-posed problem and involve numerous difficulties like the aperture problem, occlusions or large displacements [170]. So far, there exist efficient methods for the optical flow estimation using Partial Differential Equations (PDEs). Particularly, they offer a complete mathematical framework and a large number of numerical methods [108, 85, 170, 168, 128, 73]. These methods deal with the ill-posedness of the optic flow problem by including regularization techniques like the Tikhonov regularization [157]. The regularization techniques aim to circumvent the ill-posedness by adding priors as assumptions on the solution ensuring for example some smoothness, or sparsity.

In this framework, most of the classical optical flow models assume constant illumination of the scene [108, 85]. The violation of this constraint affect negatively the optic flow estimation by introducing a "false" motion due to the varying illumination. Among the responses to fix this problem, Gennert and Negahdaripour proposed to relax the constancy assumption on the intensity, by adding linear variations of the illumination [69]. In [72], this idea was implemented in the framework of PDEs methods for optic flow estimation by considering small varying illumination as a supplementary variable with the optic flow components.

In this chapter, we introduce a new optimal transport-based model for optical flow estimation with varying illumination. The standard methods in the optic flow estimation field are based on the transport of particles (pixels) along characteristics (loosely speaking a "Boltzmann-Liouville" framework). The constancy assumption is expressed by a non-conservative transport equation where the unknown is the flow vector. The new model that we introduce in this article consists of a "kinetic" formulation where instead of considering a transport of particles, we rather consider a "density" (optimal) transport (loosely speaking a fluid mechanics framework). the motion, i.e. transport, of densities (number of particles and intensities in a small unit volume) are expressed via conservation equations, that is to say, conservative transport equations. The varying illumination is intimately linked to the variation of the considered volume expressed with the (divergence of) flow vector and not as an external variable.

We hope that the new model give us a quite deep understanding of motion analysis in computer vision and the optimal transportation formulation, thanks to Benamou-Brenier's characterisation, offer a complete theoretical to computational framework to solve the optical flow estimation problem. We emphasise that the optimal transport formulation avoid any regularization technique, handles naturally the sparsity constraint (the vector fields is a gradient of convex function given by an eikonal equation), and the variation of the illumination corresponds to a "physical" constraint. In addition, thanks to the rapid development of accurate and low cost numerical methods in the numerical optimal transport theory, many improvements remain open to investigate.

Organisation of the Chapter

In Section 4.1, we propose a new optical flow model based on the continuous fluid mechanics equations. In Section 4.2, we briefly remind the optimal transportation model and the famous Benamou-Brenier formula. In Section 4.3, we recall the Benamou-Brenier algorithm to solve the continuity equation where the periodic boundary conditions are replaced by homogeneous Neumann boundary conditions. Finally, in Section 4.4, we present numerical results to confirm the pertinence of our proposed model.

4.1 Optimal Transport Based Model

Let consider a general flow equation with a velocity $\mathbf{u}$, eventually depending on time, given by

$$\begin{cases} \dot{\chi}(t,\mathbf{x}) = \mathbf{u}(t,\chi(t,\mathbf{x})), \\ \chi(0,\mathbf{x}) = \mathbf{x} \end{cases} \tag{4.1}$$

Instead of taking a constancy assumption on the intensity values of the pixels, let us consider a small volume $\mathcal{U}_t := \chi(t,\mathcal{U}_0)$ containing a number of particles and associate a continuous density function $\rho(t,\mathbf{x}(t))$ representing the mass (i.e. the number of particles) per unit volume at the point $\mathbf{x}(t)$. Then, it is standard to derive a continuity equation for the mass conservation under local form in $[0,1]\times D$,

$$\partial_t \rho + \operatorname{div}(\rho\,\mathbf{u}) = 0. \tag{4.2}$$

We notice that if f is quantity to be preserved in the transport along the characteristics then $\frac{df}{dt} = 0$ along such curves. Then, we set

$$\mathcal{F}(t) := \int_{\mathcal{U}_t} \rho(t,\mathbf{x})\, f(t,\mathbf{x})\; d\mathbf{x} = \int_{\mathcal{U}_0} \rho\big(t,\chi(t,\mathbf{x})\big)\, f\big(t,\chi(t,\mathbf{x})\big) |\det J(t,\mathbf{x})|\; d\mathbf{x},$$

where $J(t,\mathbf{x}) := \frac{\partial \chi(t,\mathbf{x})}{\partial \mathbf{x}}$ is the Jacobian matrix of χ. If we make the assumption that, by following the characteristic given by χ, the value $\rho\, f$ inside the moving small volume remains the same over the time i.e.

$$\frac{d\mathcal{F}(t)}{dt} = 0, \tag{4.3}$$

then we have that,

$$\int_{\mathcal{U}_t} \rho(t,\mathbf{x})\, f(t,\mathbf{x})\; d\mathbf{x} = \int_{\mathcal{U}_0} \rho(0,\mathbf{x})\, f\big(0,\mathbf{x}\big) |\det \underbrace{J(0,\mathbf{x})}_{=I_d}|\; d\mathbf{x} = \int_{\mathcal{U}_0} \rho(0,\mathbf{x})\, f\big(0,\mathbf{x}\big)\; d\mathbf{x}.$$

Thus, the continuity equation holds for $(\rho f, \mathbf{u})$ in $[0,1]\times D$,

$$\partial_t(\rho f) + \operatorname{div}(\rho f\,\mathbf{u}) = 0.$$

Using (4.2), a straightforward calculation and the mass conservation, we get in $[0,1]\times D$,

$$\partial_t f + \mathbf{u}\,\nabla f = 0. \tag{4.4}$$

Now, in the case of varying illumination, mimicking the optic laws we may assume that :

$$\frac{d}{dt} f\big(t,\chi(t,x)\big) := m\big(t,\chi(t,x)\big)\, f\big(t,\chi(t,x)\big),$$

where m acts as an attenuation factor on the intensity f. Now, let

$$\widetilde{f}(t,x) := f(t,x) e^{-\int_0^t m(s,x)\; ds}.$$

Thus $\frac{d\widetilde{f}}{dt} = 0$ along the characteristics, and we may derive that :

$$\widetilde{\mathcal{F}}(t) = \int_{\mathcal{U}_t} \rho(t,\mathbf{x})\, \widetilde{f}(t,\mathbf{x})\; d\mathbf{x} = \int_{\mathcal{U}_0} \rho\big(t,\chi(t,\mathbf{x})\big)\, \widetilde{f}\big(t,\chi(t,\mathbf{x})\big) |\det J(t,\mathbf{x})|\; d\mathbf{x},$$

and $\frac{d\widetilde{\mathcal{F}}}{dt} = 0$, which yields in $[0,1]\times D$,
Thus, back to f, we get the conservative transport equation in $[0,1]\times D$,

$$\partial_t f + \operatorname{div}(f\,\mathbf{u}) = 0. \tag{4.5}$$

by choosing $m = -\operatorname{div}(\mathbf{u})$ and with the "boundary conditions",

$$f(0,\cdot) = f_0, \qquad f(1,\cdot) = f_1. \tag{4.6}$$

We notice that this computations use

$$\begin{cases} \partial_t |\det J(t,\mathbf{x})| = \operatorname{div}\Big(\mathbf{u}\big(t,\chi(t,\mathbf{x})\big)\Big)\,|\det J(t,\mathbf{x})|, \\ |\det J(0,\mathbf{x})| = 1. \end{cases} \tag{4.7}$$

In other words,

$$|\det J(t,\mathbf{x})| = e^{\int_0^t \operatorname{div}(\mathbf{u}(s,\chi(s,\mathbf{x})))\,ds}.$$

Before giving an interpretation to this conservative form, we notice that in case of constant mass density ρ and constant illumination, we retrieve the standard model

$$\frac{\partial f(t,\chi(t,\mathbf{x}))}{\partial t} = 0 \Longleftrightarrow \partial_t f + \mathbf{u}\cdot\nabla f = 0.$$

Thus, the equation (4.4)

$$\partial_t f + \mathbf{u}\cdot\nabla f + \operatorname{div}(\mathbf{u}) f = 0,$$

with $m := -\operatorname{div}(\mathbf{u})$ turns to be an assumption on how the variation of the illumination modify the conservation of the intensity, per unit volume, and may be interpreted as taking into account light pixels entering the volume $\mathcal{U}_t$ instantly and changing the intensity map. That is to say the variation of the illumination acts on the volume as a heating effect on all the pixels of $\mathcal{U}_t$ and not pixel-wise as in [72]. In addition, such an interpretation appears close to the optic laws for light propoagation.

We emphasise that in case $\operatorname{div}(\mathbf{u}) = 0$, then it means that the volume is preserved and in our case the illumination is constant, however, we do not retrieve the standard model in this case, as this one do not assume any divergence free motion.

4.2 The Benamou-Brenier Formula

Let us recall the framework of the Monge-Kantorovich problem which is as follows. Two density functions $\rho_0 > 0$ and $\rho_1 > 0$ in $\mathbb{R}^d$, that we assume to be bounded with total mass one

$$\int_{\mathcal{O}} \rho_0(x)\ dx = \int_{\mathcal{O}} \rho_1(x)\ dx = 1.$$

Let X, Y be two measures spaces. Denoting the measures with densities $\mu = \rho_0\,dx$ and $\nu = \rho_1\,dx$, Monge's optimal transportation problem is to find a measurable map $T : X \to Y$ that transport (pushes) μ onto ν and minimizes the following objective function,

$$C(\mu,\nu) := \inf_{T\in \operatorname{Tr}(\mu,\nu)} \int_X c(x,T(x))\rho_0(x)\ dx,$$

where $c : X \times Y \to \mathbb{R}_+$ is the cost function and $\operatorname{Tr}(\mu,\nu) = \{T : X \to Y \mid T_\sharp\mu = \nu\}$, where $T_\sharp\mu$ represents the pushforward of measure μ and is characterized as,

$$\nu(A) = \mu(T^{-1}(A)), \qquad \text{for any measurable } A \subset Y.$$

The Kantorovich formulation of the transportation problem consists of optimizing over transportation plans, where a transport plan is a probability measure $\gamma \in \mathcal{P}(X \times Y)$ with marginals μ and ν. One can think of γ as the joint distribution of ρ_0 and ρ_1 describing how much "mass" (or number of particles) is being moved to different coordinates. Let γ be a plan with marginals μ and ν, i.e.

$$(\pi_X)_\sharp\gamma = \mu, \qquad (\pi_Y)_\sharp\gamma = \nu,$$

where $\pi_X : X \times Y \to X$, resp $\pi_Y : X \times Y \to Y$ are the canonical projections. Let $\operatorname{Marg}(\mu,\nu)$ be the set of all such plans, then the Kantorovich's formulation can then be written as,

$$C(\mu,\nu) = \min_{\gamma\in\operatorname{Marg}(\mu,\nu)} \int_{X\times Y} c(x,y)\ d\gamma(x,y).$$

For any transport map $T : X \to Y$ there is an associated transport plan, given by

$$\gamma = (I_d \times T)_\sharp \mu,$$

that is to say $\gamma(A, B) = \mu(\{x \in A;\ T(x) \in B\})$. The so-called L^p-Kantorovich (or Wasserstein) distance between μ and ν, which admit the densities ρ_0 and ρ_1 is defined by : $p \geq 1$,

$$W_p(\mu, \nu)^p = \min_{\gamma \in \mathrm{Marg}(\mu,\nu)} \int_{X\times Y} |x - y|^p \ d\gamma(x, y) = \inf_{T \in \mathrm{Tr}(\mu,\nu)} \int_{X\times Y} |x - T(x)|^p \rho_0(x) \ dx. \tag{4.8}$$

In [19, 164] the following characterization of the 2-Wasserstein distance is proved :

Proposition 4.2.1. *(Benamou-Brenier Formula). The square of the L^2-Wasserstein distance is equal to the infimum*

$$\inf_{(\rho,v)} \int_0^1 \int_{\mathcal{O}} \rho(t, x)|v(t, x)|^2 \ d \ xdt, \tag{4.9}$$

among all pairs (ρ, v) satisfying (4.4) and (4.6).

Moreover, formal optimality conditions give :

$$v(t, x) = \nabla\psi, \quad \psi_t + \frac{1}{2}|\nabla\psi|^2 = 0. \tag{4.10}$$

Note. The result of Benamou-Brenier is given for $\mathcal{O} = \mathbb{R}^d$, other versions in bounded domains may be found for example in Villani [164, 60].

Let us normalize our images f_0 and f_1 in the optic flow problem to be probability densities. Then, minimizing (4.8) for $p = 2$, allows us to formulate the optimal transport model for the optic flow estimation with varying illumination as : finding

$$\inf_{(f,\mathbf{u})} \int_0^1 \int_{\mathcal{O}} f(t, x)|\mathbf{u}(t, x)|^2 dx \ dt, \tag{4.11}$$

under the constraints

$$\partial_t f + \mathrm{div}\,(\mathbf{u}\, f) = 0. \tag{4.12}$$

and

$$f(0, \mathbf{x}) = f_0(\mathbf{x}), \qquad f(1, \mathbf{x}) = f_1(\mathbf{x}). \tag{4.13}$$

and the (infinitesimal) illumination $m(t, x) = -\mathrm{div}\,\mathbf{u}(t, x)$.

In this model, the illumination do not appear as supplementary variable (see [72]) and there is no need of regularization as in the standard non conservative transport model. In fact, the transport model gives more than an optic flow estimation that is interpolation functions $f(t, x)$ and $\mathbf{u}(t, x)$ before the final time, i.e. for $t < 1$.

4.3 The Numerical Method

In their paper, Benamou and Brenier proposed to use the augmented Lagrangian method in order to solve (4.11) under constraints (4.12) and (4.13) [19, 63]. They solves the involving equations in $\mathbb{R}^d$ and for numerical constraints they add periodic boundary conditions in space. Here, we work in a bounded domain D and we impose Neumann boundaries condition in space. We gives the modified Benamou-Brenier algorithm which is based on relaxed Uzawa iterations :

For $(\phi^{n-1}, q^{n-1}, \mu^n)$ givens,

- Find ϕ^n, solution of

$$\begin{cases} -r\Delta_{t,\mathbf{x}}\phi^n = \nabla_{t,\mathbf{x}} \cdot (\mu^n - rq^{n-1}), & \text{in } [0, 1] \times D, \\ r\partial_t\phi^n(0, \cdot) = f_0 - f^n(0, \cdot) + ra^{n-1}(0, \cdot), & \text{in } D, \\ r\partial_t\phi^n(1, \cdot) = f_1 - f^n(1, \cdot) + ra^{n-1}(1, \cdot), & \text{in } D, \\ \partial_n\phi^n = 0, & \text{on } [0, 1] \times \partial D, \end{cases} \tag{4.14}$$

with $\mu^n := (f^n, (f\mathbf{u})^n)$ and $q^{n-1} := (a^{n-1}, b^{n-1})$.

- Find $q^n := (a^n, b^n)$ such that

$$\inf_{(a^n,b^n)\in K} (a(t,\mathbf{x}) - \alpha^n(t,\mathbf{x}))^2 + |b(t,\mathbf{x}) - \beta^n(t,\mathbf{x})|^2,$$

with $p^n(t,\mathbf{x}) := (\alpha^n(t,\mathbf{x}), \beta^n(t,\mathbf{x})) = \nabla_{t,\mathbf{x}}\phi^n(t,\mathbf{x}) + \frac{\mu^n(t,\mathbf{x})}{r}$ and

$$K := \Big\{ a : \mathbb{R}\times\mathbb{R}^d \to \mathbb{R},\ b : \mathbb{R}\times\mathbb{R}^d \to \mathbb{R}^d \ \Big|\ a(t,\mathbf{x}) + \frac{|b(t,\mathbf{x})|^2}{2} \le 0,\ \forall (t,\mathbf{x}) \in \mathbb{R}\times\mathbb{R}^d \Big\}.$$

This turns out to be a simple one dimensional projection problem. It is important to notice that this minimization can be performed pointwise in space and time. The analytical solution can be found analytically by using Cardano's formula.

- Do the pointwise update

$$\mu^{n+1} = \mu^n + r(\nabla_{t,\mathbf{x}}\phi^n - q^n),$$

where $r > 0$ is the parameter of the augmented Lagrangian.

The stopping criterium is when the quantity

$$\sqrt{\frac{\int_0^1 \int_D f^n |\mathrm{res}^n|\ dxdt}{\int_0^1 \int_D f^n |\nabla_{\mathbf{x}}\phi^n|^2\ dxdt}},$$

is small enough, where

$$\mathrm{res}^n := \partial_t \phi^n + \frac{|\nabla_{\mathbf{x}}\phi^n|^2}{2}.$$

4.4 Numerical Results

We discretize the time interval $[0,1]$ using a uniform subdivision $t_1 = 0 < t_2 < \ldots < t_N = 1$. We solve the equation on ϕ^n by using finite differences on a regular mesh where every node represents a pixel of the images. We use the same grid for the projection problem and the pointwise update from the Benamou-Brenier algorithm. By solving the continuity equation, we obtain an interpolation in time between f_1 and f_2, an optic flow $\mathbf{u}$ and a luminosity variation m which are time-dependent. However, we wish to remove this time dependency. To this end, we set $\mathbf{u} : \mathbb{R}^d \to \mathbb{R}^d$ and $m : \mathbb{R}^d \to \mathbb{R}$,

$$\mathbf{u}(\mathbf{x}) := \chi(1,\mathbf{x}) - \chi(0,\mathbf{x}) = \int_0^1 \dot{\chi}(s,\mathbf{x})\ ds = \int_0^1 \mathbf{u}(s,\chi(s,\mathbf{x}))\ ds,$$

and

$$m(\mathbf{x}) := -\mathrm{div}\,\mathbf{u}(\mathbf{x}).$$

This add a trajectory reconstruction step to our method in order to retrieve the optic flow. With this optic flow and luminosity variation we have

$$f_2\big(\mathbf{x} + \mathbf{u}(\mathbf{x})\big) = \big(1 + m(\mathbf{x})\big) f_1(\mathbf{x}).$$

We give in Figure 4.1 the starting and ending image along with the true optical flow without luminosity variations for several examples from [10]. In Figure 4.2, we present the numerical illustrations of the method developed in this paper $N_t = 4$ points for the time's discretization and $r = 2$ in (4.14). Since the luminosity variation m is between -1 and 1, we re-scaled it to be between 0 and 1 in Figure 4.2, thus the grey color means that m vanishes.

In Figure 4.3, we give the results for the classical optical flow formulation with varying luminosity assumption and with the global-local method with $\rho = 0.4$, $\alpha = 0.1$ and $\lambda = 0.5$ [72].

The first observation is that the resulting optical flow and the luminosity variations are concentrated on the edges and then are very sparse which is a nice property for compression purpose. It means that, on common homogeneous parts between the images, the is no need to have nor spatial nor luminosity movements. This characteristic is what differs from previous approach where the optic flow vector is smoothed by the regularization, add non-pertinent movement and may introduce error in homogeneous

parts. Here however, there seems to have no reconstruction error in the homogeneous parts. The error is more presents near the edges, where the movements exists. This is due to the regularization effect of r in (4.14). In addition, adding more points in the time discretization gives a more accurate optic flow and luminosity variations.

We give in Table 4.1 the computation time needed to compute the optical flow for both the classical formulation and the optimal transport model.

Video	**Classical**	**Optimal Transport**
1	196.35	6209.77
2	193,55	3854.34
3	209,29	5139.12

Table 4.1: Computation time in seconds.

In fact, the time required to process an iteration of the modified Benamou-Brenier algorithm is almost the same as solving the whole classical optical flow problem. Indeed, the first one need to solve a $N_t \times N_x \times N_y$ linear system and the second solves a $3 \times N_x \times N_y$, where N_t is the number of points for the time discretization and N_x, N_y are the numbers of points for the space discretization. However, the optimal transport approach needs many iterations to converge to the solution (around 20).

Since the continuity equation aims to model fluid motions, we propose in Figure 4.4 an illustration on a video of water falling down from a sink.

(a) First image f_1.

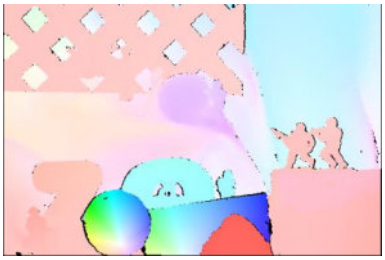
(b) First image f_1.

(c) First image f_1.

(d) Second image f_2.

(e) Second image f_2.

(f) Second image f_2.



(g) Ground truth.

(h) Ground truth.

(i) Ground truth.

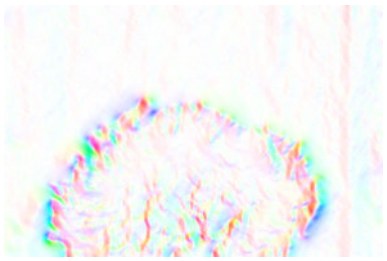
Figure 4.1: Input images and ground truth with constant luminosity assumption [10].

(a) Reconstruction x.

(b) Reconstruction x.

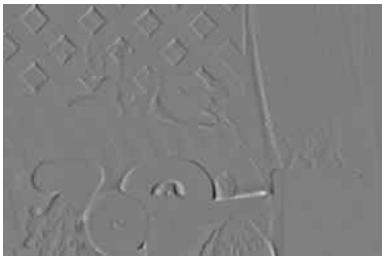
(c) Reconstruction x.

(d) Optic flow $\mathbf{u}$.

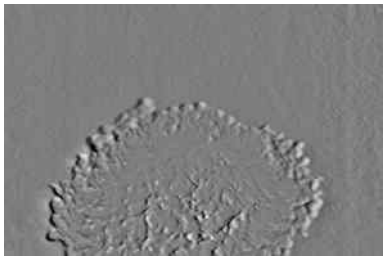


(e) Optic flow $\mathbf{u}$.

(f) Optic flow $\mathbf{u}$.



(g) Luminosity variations m.



(h) Luminosity variations m.

(i) Luminosity variations m.

(j) Error $|x - f_2|$.

(k) Error $|x - f_2|$.

(l) Error $|x - f_2|$.

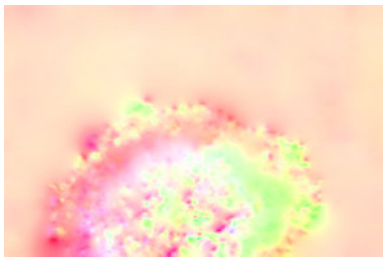
Figure 4.2: Reconstruction x by applying optical flow $\mathbf{u}$ and luminosity variations m on f_2 and its error.

(a) Reconstruction x.

(b) Reconstruction x.

(c) Reconstruction x.

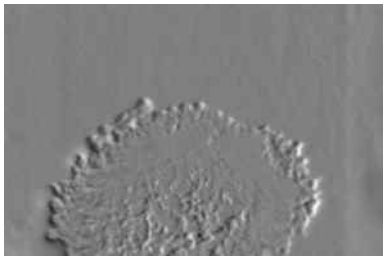
(d) Optic flow $\mathbf{u}$.



(e) Optic flow $\mathbf{u}$.

(f) Optic flow $\mathbf{u}$.

(g) Luminosity variations m.



(h) Luminosity variations m.

(i) Luminosity variations m.

(j) Error $|x - f_2|$. (k) Error $|x - f_2|$. (l) Error $|x - f_2|$.

Figure 4.3: Reconstruction x by applying classical optical flow $\mathbf{u}$ and luminosity variations m on f_2 and its error.

(a) First image f_1. (b) Second image f_2. (c) Reconstruction x.

(d) Optic flow $\mathbf{u}$. (e) Luminosity variations m. (f) Error $|x - f_2|$.

Figure 4.4: Reconstruction x by applying optical flow $\mathbf{u}$ and luminosity variations m on f_2 and its error for video of fluid.

Conclusion

In this chapter, we formulated a new model to handle the optical flow computing problem. This new model deals with brightness variations like several state-of-the-art approaches, but differ from them by not needing any regularization technique to have a well-posed problem. This absence of smoothness assumption leads to a resulting sparse optical flow as well as a sparse brightness variation, where all the information are concentrated around the moving edges. This property may be useful for video compression purpose which use motion compensation methods. To numerically solve our new model, we use optimal transport theory along with the Benamou-Brenier theorem and numerical algorithm. Finally, we present numerical results for determining the optical flow and the brightness variations, for predicting the frames and for comparing this resulting prediction to the original images. It appears that the information given by the optical flow and the error are concentrated near the edges (where the movement belongs) but almost vanishes inside the common homogeneous regions of the images. The counterpart of this property is a longer computation time than classical optical flow model computation.

Part III

Application to Video Compression

Chapter 5

Application to Video Compression

Contents

In this last chapter, we give a complete implementation of a modular codec in which each "block" can be chosen. We propose to compare the different combinations of "blocks".

Introduction

In 2016, Andris *et al* proposed to use a video compression codec using mainly variational methods [8]. As with more traditional codecs, the frames that compose a video are separated into two categories: *intra frames* and *inter frames*. The *intra frames* are decoded using an inpaiting compression method (Part I) while the *inter frames* are decoded using optical flow methods (Part II). Thus, an *inter frame* needs the previous frame (or frames) to be reconstructed. Then, the frames of the video are grouped into *groups of pictures* (GOPs) so that only the first frame of a group is an *intra frames* and the following ones are *inter frames*. The first step, shown in blue on the Figure 5.1, is to determine these GOPs. For a given GOP, the second step, in red on the Figure 5.1, consists in applying the adequate compression method to the type of the considered frame (inpainting or optical flow) and to calculate the *residual data* by subtracting the reconstruction with the original image. Saving the residual data adds the possibility of having a lossless compression codec and avoiding the propagation of errors from one frame to the next. In this chapter, we will not consider the residual data. Finally, the last step, in yellow on the Figure 5.1, saves the data produced by the previous steps and applies a symbol coding in order to eliminate the redundancies due to the coding of the data.

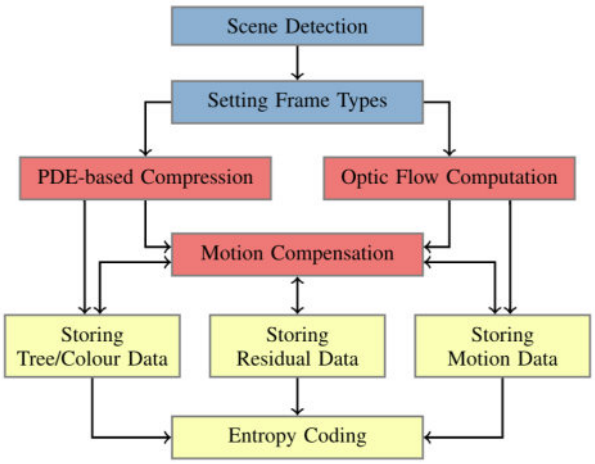


Figure 5.1: Codec structure from [8].

5.1 Choice for the Coding Step

The first step is to determine the type of image being processed. To do this, we calculate the error L^2 between the current image f_n and the previous image f_{n-1}. If this error is less than a certain threshold, we consider that the two images are close enough to be treated as an *inter frame* i.e. by optical flow. Thus, we add f_n to our current GOP. If the error is greater than the threshold, it is considered as an *intra frame* i.e. it is compressed with an inpainting method and a new GOP is started. Moreover, since our codec does not take residual data into account, we want to add a constraint on the size of our GOPs so as not to propagate errors too widely.

For the human visual system, errors in the Cr and Cb chrominance channels (color information) have a smaller impact on perceived quality than errors in the Y luminance component (visual sensation of brightness) [134]. We then choose to convert each frame of the video to the YCrCb format rather than using the RGB (Red, Green and Blue) format directly. Since the Y luminescence channel is the most important, it is this one that will allow us to create an inpainting mask for the *intra frames* and to estimate the optical flow for the *inter frames*. In this way we will obtain a single mask/optical flow for each colored image.

Finally, as in [8], we choose to sub-sample the optical flow when storing it.

5.2 File Format

For a video streaming application, we want to send information about the methods used for each block to each frame. This choice allows to add the possibility to change quality, size or computing power on the fly. For a video storage application, we can put this information once at the beginning of the file since it will not change during decompression.

5.2.1 Coding of an *Intra Frame*

For the coding of an *intra frame*, we choose to be inspired by the Run Length Encoding [140]. We go through the inpainting mask from left to right and from top to bottom and we count the number of pixels located between two consecutive pixels in the mask. Once the data sequence has been constructed, a symbol encoder is applied to remove the encoding redundancy. Here is the format chosen to encode a *intra frame* :

- **Type of frame** (1 byte) : For example 1 for an *intra frame*.
- **Size of the frame** (4 bytes) : 2 bytes for its width and 2 bytes for its height. We can have a maximum resolution of 65536×65536.

- **Data size** (4 bytes or more): The size of this field will depend on the resolution of the image.
- **Data with symbolic coding** (variable size): For each pixel selected in the inpainting mask:
  - **Counter** (variable size) : counter value between two consecutive pixels in the inpainting mask.
  - **Luminance channel** Y (1 byte).
  - **Chrominance channel** Cr (1 byte).
  - **Chrominance channel** Cb (1 byte).

We could reduce the size of the data of a frame by encoding the type of the frame on 1 bit, but for technical constraints due to *python*, we choose here 1 byte.

5.2.2 Coding of an *Inter Frame*

Contrary to the encoding of *intra frames*, it is not necessary to save the dimensions of the image since these must be constant within a GOP. The main difficulty here comes from the fact that the u_1 and u_2 components of the optical flow can be arbitrarily large. However, in the case of small displacements, we can assume that these components are not too large. We can then transform the value of u_1 and u_2 in order to store them on a byte. Moreover, we will not consider the methods taking into account the variations of luminosity and thus we have only u_1 and u_2 to store. Once the data sequence is built, we apply a symbol encoder to remove the encoding redundancy. Here is the format chosen to encode an *inter frame*:

- **Type of frame** (1 byte): For example 0 for an *inter frame*.
- **Data size** (4 bytes): The size of this field will depend on the resolution of the image and the homogeneity of the movement.
- **Data with symbolic coding** (variable size): For each pixel in the optical flow sub-sample:
  - **Component $\mathbf{u_1}$ of the optical flow** (1 signed byte).
  - **Component $\mathbf{u_2}$ of the optical flow** (1 signed byte).

5.3 Numerical Results

We propose to compare the "blocks" of our video codec with existing video compression methods. In this section, we will use finite differences for solving the partial differential equations in the inpainting methods and the optical flow, unless otherwise specified. When we use the finite element method, we will use the *scikit-fem* [78] package of *python* for inpainting and *FreeFem++* [81] for optical flow resolution with adaptive control of the regularization parameter [72].

5.3.1 Comparison of Symbol Encoders

Before we can compare the compression methods for the *inter frames* and the *intra frames*, we start by comparing the coding compression methods. As these are lossless, the use of one method rather than another has no influence on the quality of the decoded data. In this section, we will compare the methods *bzip2*, *LZMA* and *PAQ*. *PAQ* uses a context mixing algorithm which is a type of data compression algorithm in which predictions of the following symbol use different statistical models and are combined to produce a prediction that is often more accurate than any of the individual predictions. *LZMA* uses a dictionary compression algorithm. Finally, the algorithm *bzip2* uses the Burrows-Wheeler transform [30] with the Huffman coding. The most expensive storage in terms of data size in our video codec being the storage of the inpainting masks of the *intra frames*, we choose to make simulations only on the storage of these ones. We choose to do the tests on the following mask creation strategies: random (*RAND*) and mask from Belhachmi *et al* with hard/soft-thresholding (respectively *H1-T* and *H1-H*) [15]. Our tests are carried out on the database of color images from *Kodak* (`http://r0k.us/graphics/kodak/` and Section

5.4) of size 768×512. We give in Figure 5.2 (and Section 5.5) the number of bytes needed to store the inpainting mask compared to the pixel density in the inpainting masks.

It appears that the *PAQ* method allows a better compression of the data of the inpainting mask and this, whatever the strategy of creation of the mask and the density of pixels in it. This is why in the following sections, we will carry out numerical simulations only with the *PAQ* compression.

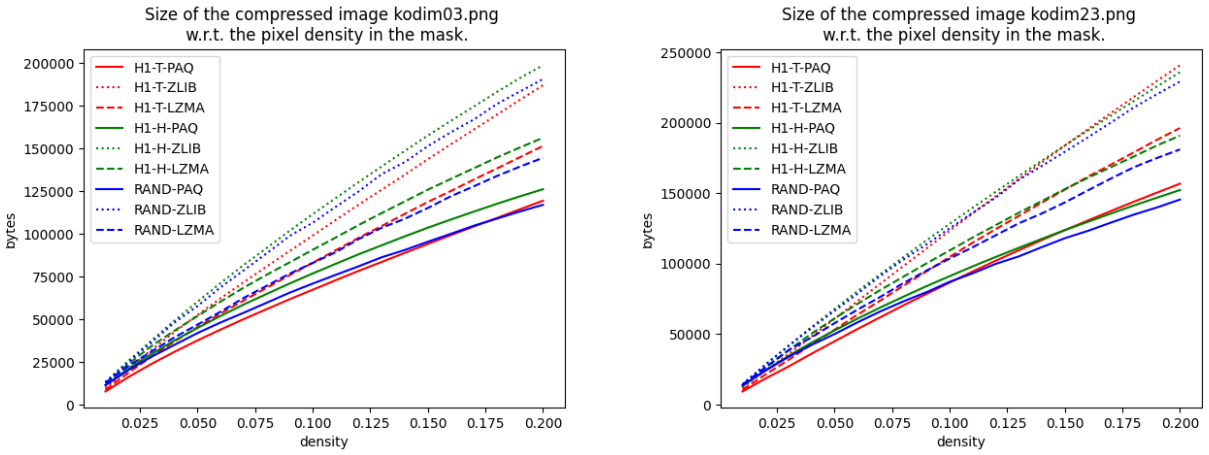


Figure 5.2: Number of bytes needed to store the inpainting mask versus the pixel density in the inpainting mask *H1-T*, *H1-H* and *RAND* for different symbol encoders.

5.3.2 Comparisons of the Blocks of *Intra Frame*

Now, we propose numerical simulations to compare the different combinations of blocks in our codec. As said in the previous section, we will use *PAQ* as a symbol encoder. We propose to compare the following algorithms: *H1-H*, *L2-INC-T* and *L1-ADJ-H*. As for the previous section, we will perform the numerical simulations on the database of color images of *Kodak* (`http://r0k.us/graphics/kodak/` and Section 5.4). In order to make the calculation time reasonable, we choose to resize the images to 384×256. For *L2-INC-T* we choose $\alpha = 0.01$ and $q = 50$, and for *L1-ADJ-H* we choose $\alpha = 1$.

In most cases, the use of finite elements does not improve the quality of the image reconstruction. On the other hand, finite elements allow to significantly reduce the number of unknowns in the linear system and thus to reduce the computation time needed for the reconstruction. Moreover, *L1-ADJ-H* seems to give the best reconstructions for *intra frames* whatever the density of pixels in the inpainting mask. This is the reason why we choose this method in the rest of this chapter.

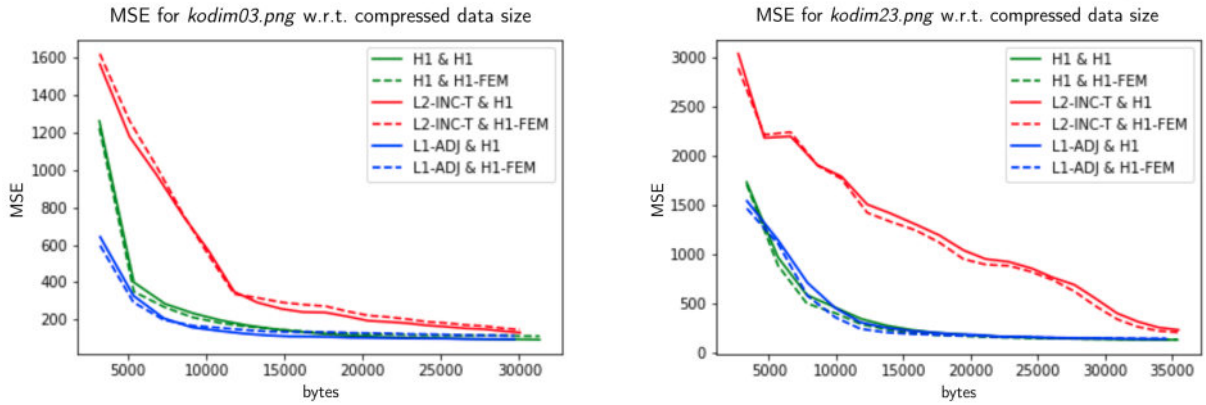


Figure 5.3: Comparison of MSEs as a function of compressed file size for *intra frames*.

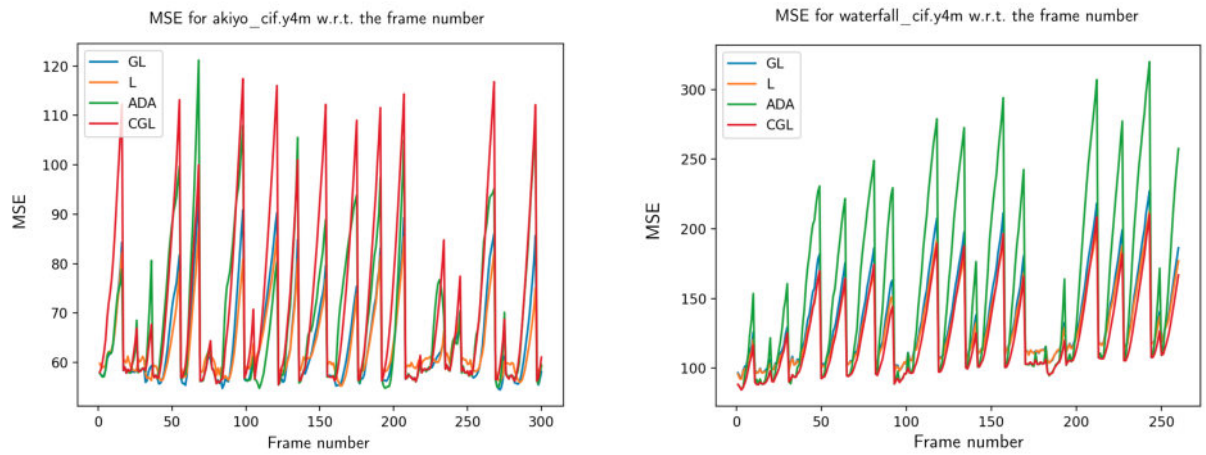
5.3.3 Comparisons of the Blocks of *Inter Frame*

We will perform our tests on the following video database: `https://media.xiph.org/video/derf/`. We choose videos of size 352×288 of different natures (homogeneous zones, textures, camera movements, object movements, ...).

Here and thanks to the numerical results of the previous section, we choose as mask creation strategy *L1-ADJ-H* coupled with linear diffusion inpainting, which offers the best reconstruction quality in terms of MSE. We propose to compare the optical flow computed with the *local* (*L*), *global-local* (*GL*), *adaptive* (*ADA*) and *global-local* with *coarse-to-fine* (*CGL*) method [72]. We take for the *local* method $\rho = 5$, for the *global-local* method $\rho = 5$ and $\alpha = 2$ and for the *coarse-to-fine* we make 5 iterations and we take $\alpha = 0.2$ and $\rho = 1$. The sub-sampling is done with blocks of size 16×16 and we take 8% of the pixels for the *intra frames*.

As expected, the error is propagated between *inter frames* within a GOP. We see on the images given in the Section 5.7 that the images predicted by the optical flow show more blurring effect than the original ones. Moreover, due to the sub-sampling of the optical flow, we have the appearance of visual artifacts similar to those of the block matching methods. Although the *local* (L) optical flow induces a larger error at the beginning of the GOPs, it also seems to propagate the error less. Thus, for the chosen videos, it seems that the optical flow of the *local* (L) method is the most suitable among those tested and with the chosen parameters. Finally, the assumption of constancy of brightness does not seem to be really adapted to real videos because it does not take into account the variations of brightness in the video.

To summarise, we give in the following Figure 5.4 an idea of how the inpainting error are propagated by the optic flow for the methods chosen.



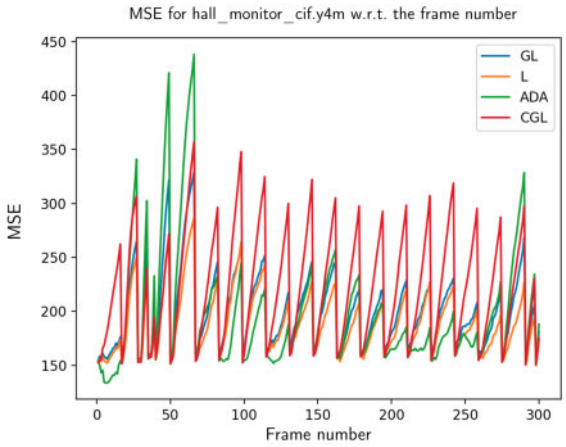
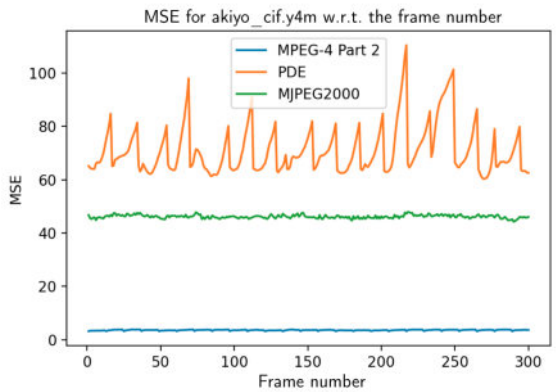


Figure 5.4: Comparison of MSEs as a function of frame number for *inter frames*.

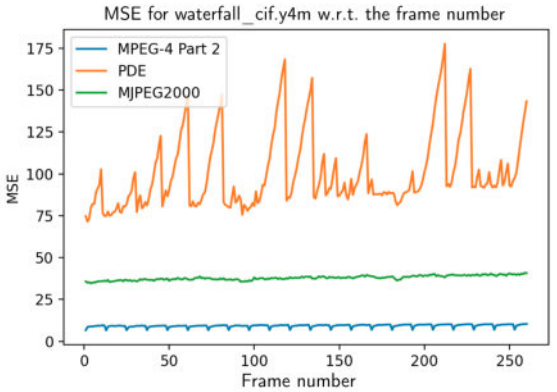
5.3.4 Comparison with Some Existing Standards

We propose to compare one of our variational video compression methods with the video codecs *Motion JPEG2000* and *MPEG-4 Part 2*. Unlike *MPEG-4 Part 2* as well as our codec, *Motion JPEG2000* does not use motion compensation for video encoding. We propose to carry out our tests on the same videos as the preceding section. We use the software *ffmpeg* in its version 3.4.11 for the conversion of the videos in the standard formats. Before that, we choose to modify our codec because *Motion JPEG2000* and *MPEG-4 Part 2* are optimized for the storage of videos. We thus propose to carry out the stage of coding of symbols on the whole of the video rather than for each frame. This has the effect of considerably reducing the size of the final file without affecting the quality of the reconstructions. Our codec will use the *L1-ADJ-H* method with $\alpha = 1$ for the coding of the *intra frames* and the local L method with $\alpha = 5$ for the *inter frames*. Finally, the symbolic coding will be done with *PAQ*.

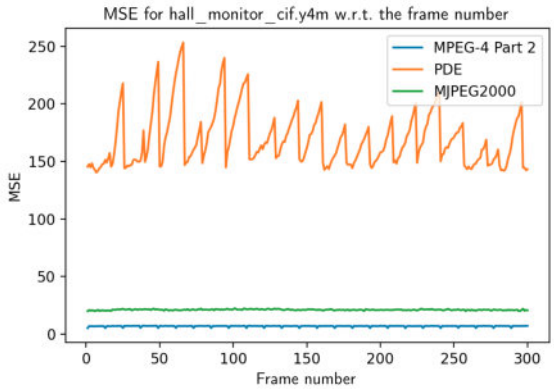
We give Figure 5.5 the MSE error as a function of the image number with as final file size for the three codecs $500Kb$, $2.5Mb$ and $1.5Mb$. For the videos *waterfall_cif.y4m* and *hall_monitor_cif.y4m* we will use blocks of size 8×8 for the sub-sampling of the optical flow and of size 16×16 for the video *akiyo_cif.y4m*. Very clearly, we see that in all cases, the coding of the *intra frames* gives less good reconstruction than for the two other codecs. Moreover, the error propagates very quickly within the GOPs with the optical flow methods implemented in this thesis.



(a) $500Ko$.



(b) $2.5Mo$.



(c) 1.5Mo.

Figure 5.5: Comparison of the MSE error for our codec, *Motion JPEG2000* and *MPEG-4 Part 2*.

Conclusion

In this chapter, we first described a codec using only variational methods for video reconstruction. This codec is composed of "blocks" performing independent compression tasks: a block for encoding *intra frames*, a block for encoding *inter frames* and a block for encoding symbols. We then determined which blocks are the best among the methods studied in this thesis. The combination of *L1-ADJ-H* with the optical flow method and the *PAQ* compression seems to be the best for the tested videos. We then compared our codec to the *Motion JPEG2000* and *MPEG-4 Part 2* codecs. Although a fully variational codec seems promising for video compression, it still lacks maturity. It could be interesting to add residue saving during motion compensation in order to avoid the propagation of errors within a GOP which is very important at the moment. Another way to reduce this propagation would be to use more accurate optical flow calculation methods. Concerning the *intra frames*, we could also add the *tonal optimization* and the *interpolation swapping* to reduce the reconstruction error [147]. Moreover one could reduce the cost of storage of the value of the pixels thanks to a resampling of the data (*brightness rescaling* [147]) and thus to be able to store more pixels without modifying the size of the final file.

5.4 Images Used from the *Kodak* Database

Our numerical simulations are performed on some images from the Kodak color image database (`http://r0k.us/graphics/kodak/`).

(a) *kodim01.png*

(b) *kodim02.png*

(c) *kodim03.png*

(d) *kodim05.png*

(e) *kodim06.png*

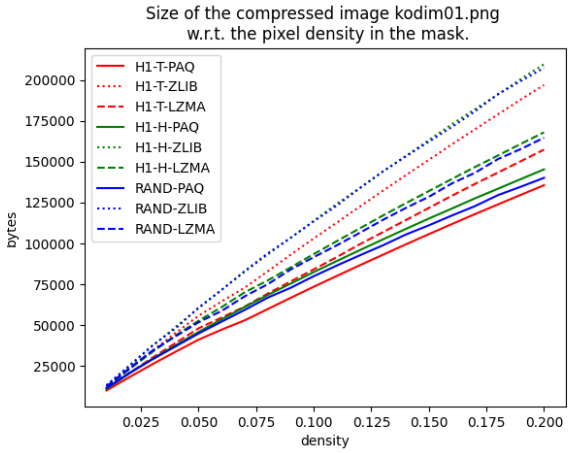
(f) *kodim13.png*

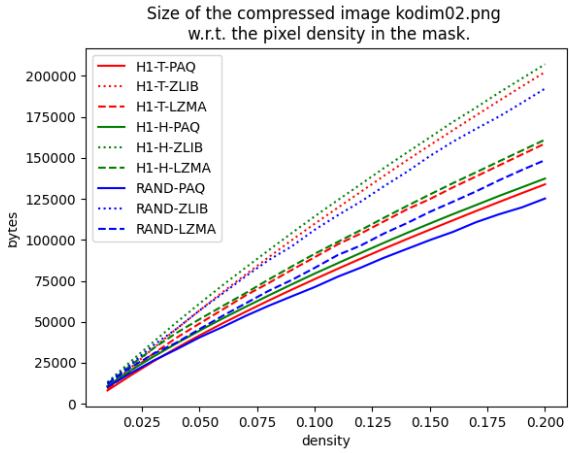
(g) *kodim21.png*

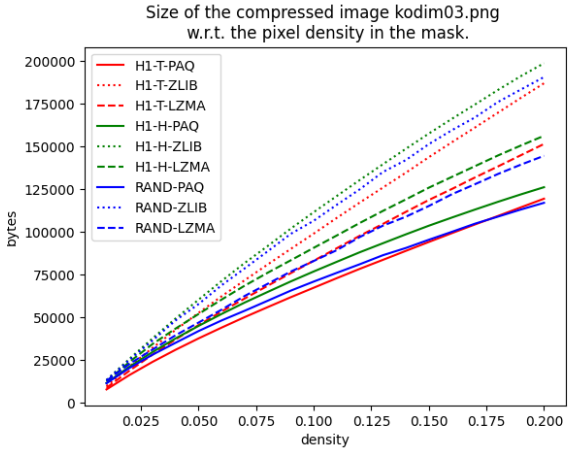
(h) *kodim23.png*

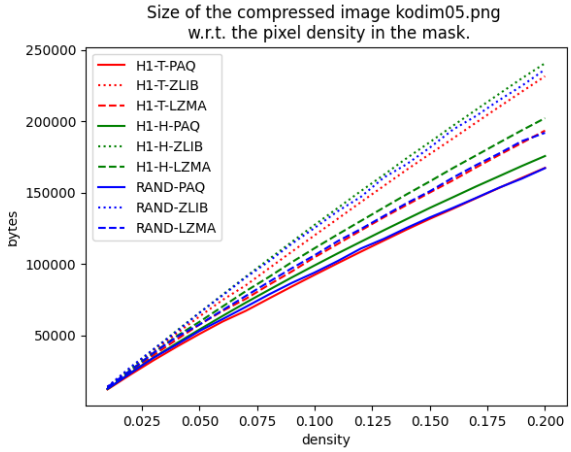
(i) *kodim24.png*

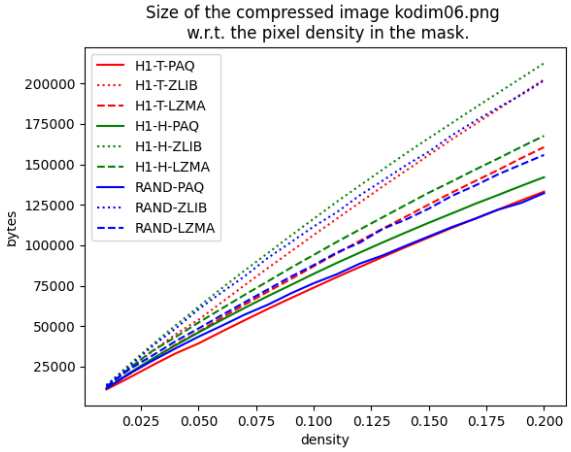
5.5 Symbol Encoder Comparison Graphics

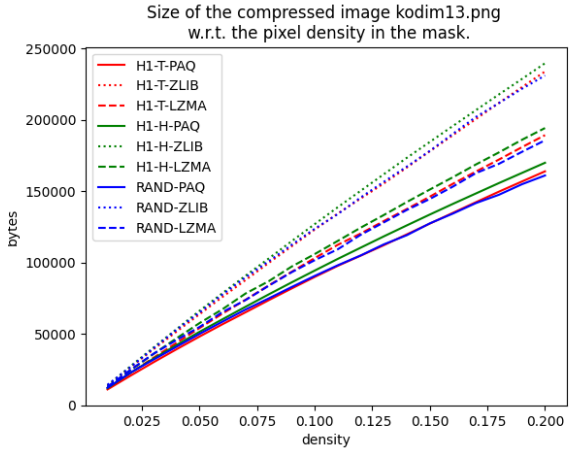












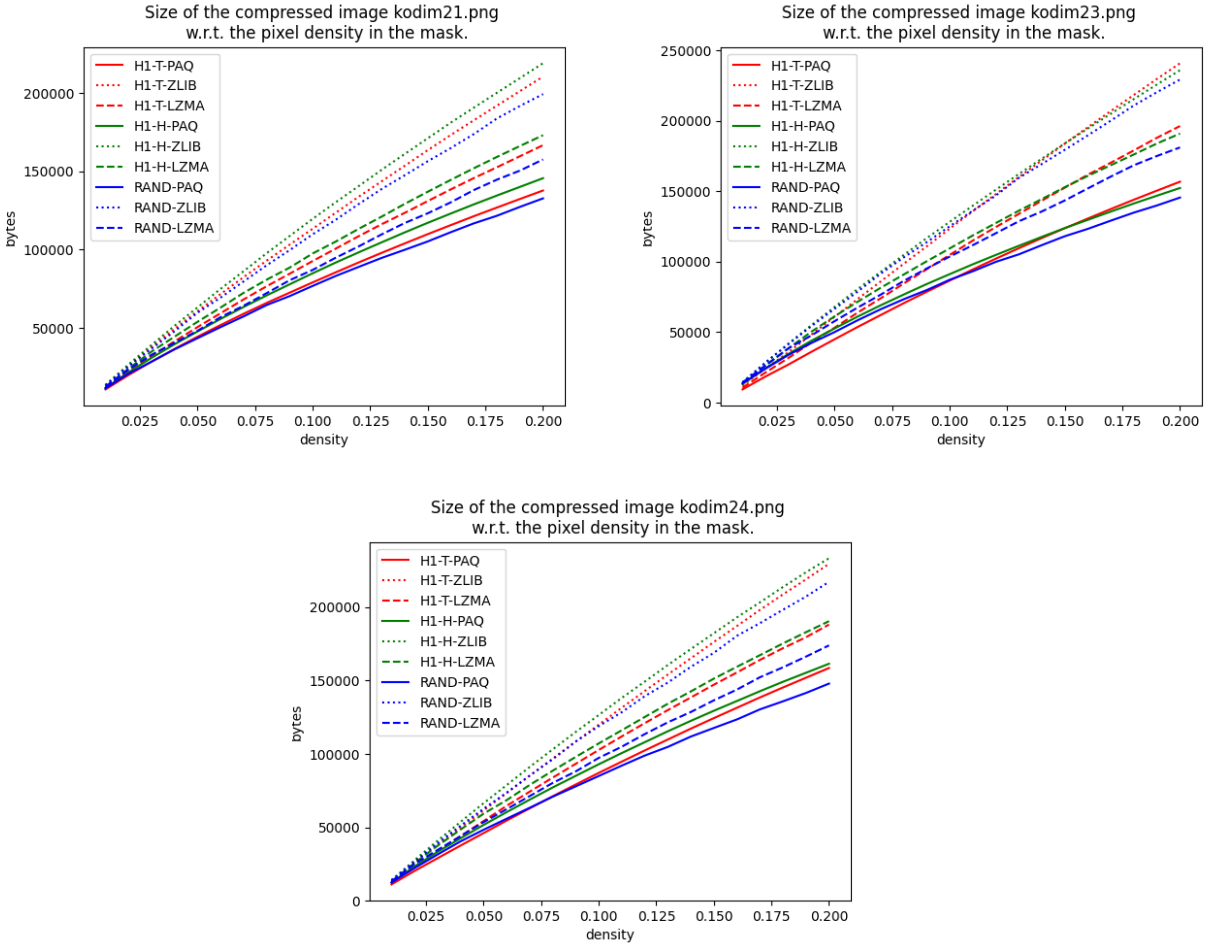
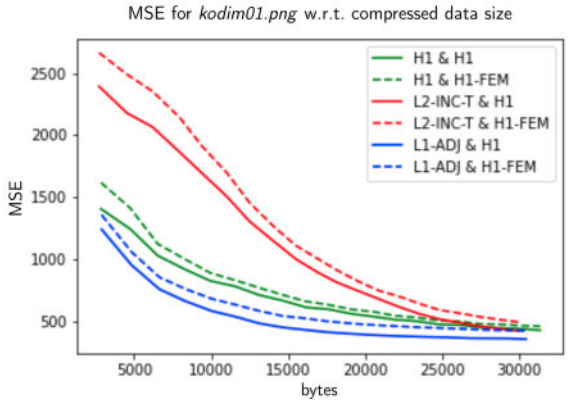
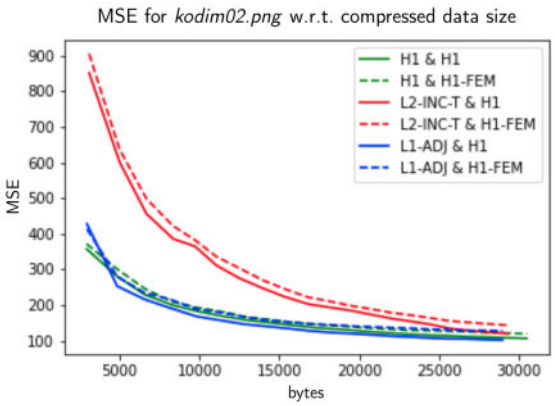
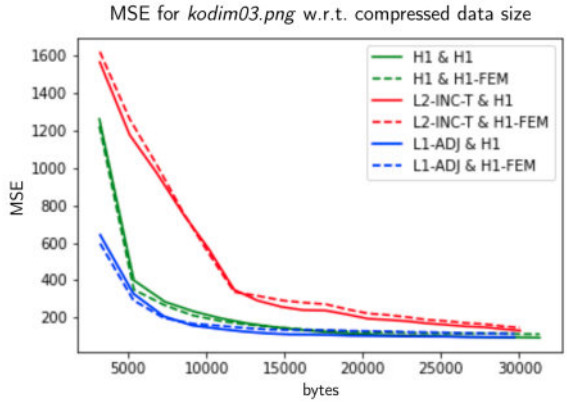


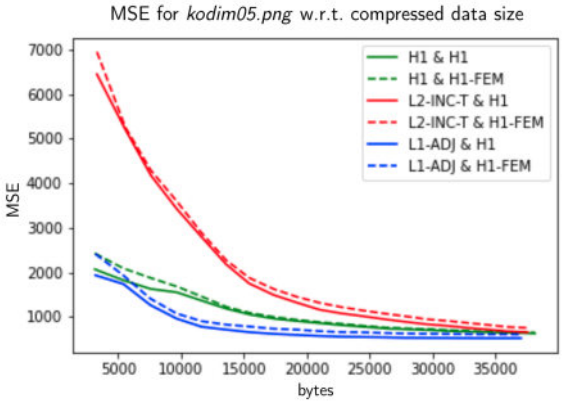
Figure 5.6: Number of bytes needed to store the inpainting mask compared to the pixel density of the inpainting mask *H1-T*, *H1-H* and *RAND* for different symbol coders.

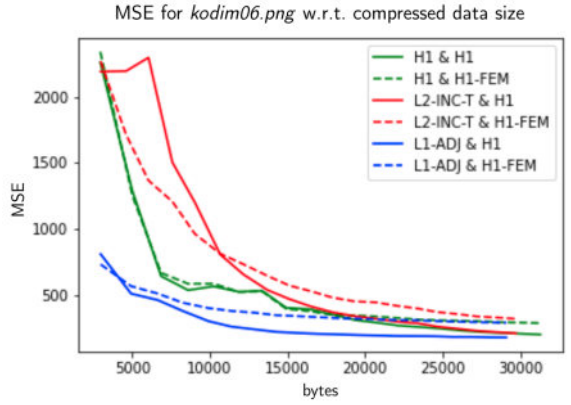
5.6 Comparison Graphs of the Encoders of *Intra Frames*

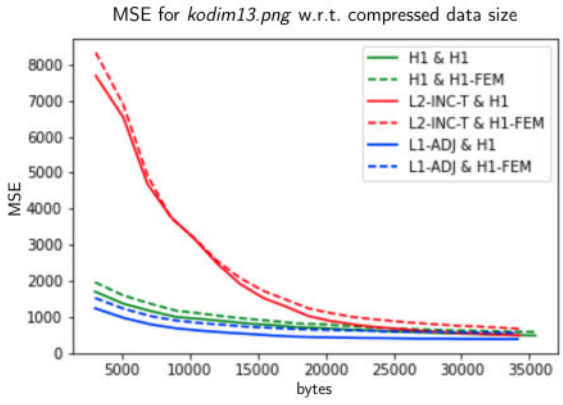












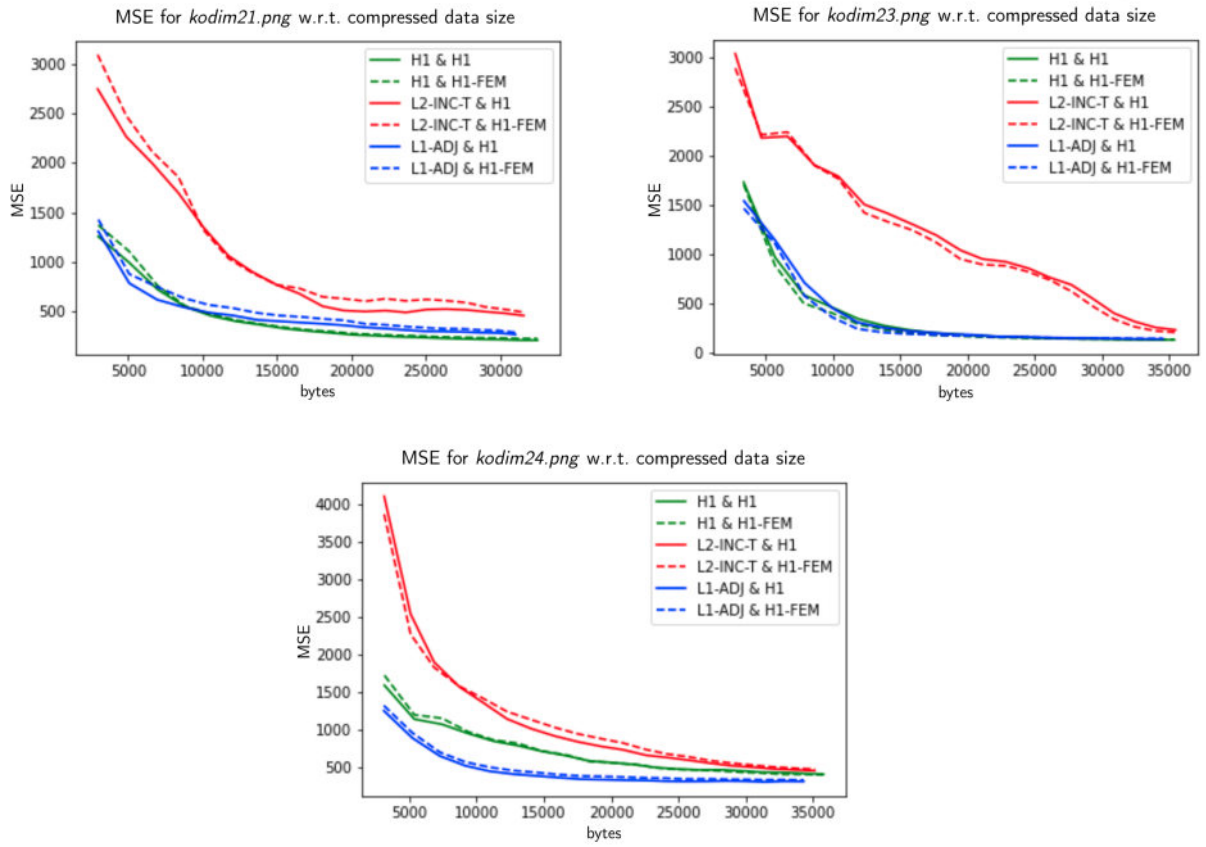
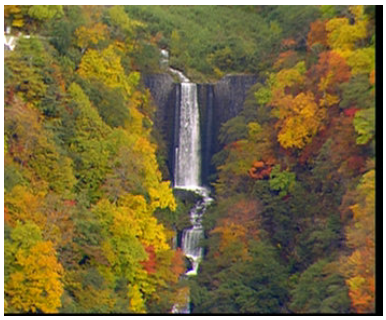


Figure 5.7: MSE of the reconstruction with respect to the final compressed file size for various PDE-based compression methods.

5.7 Reconstruction of *Inter Frames* Using Optical Flow

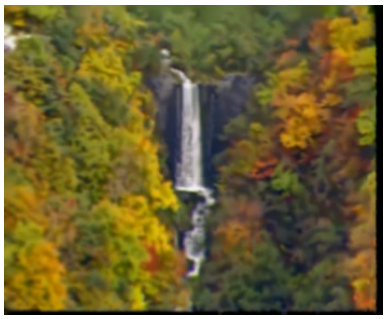
(a) Original.



(b) Original.

(c) Original.

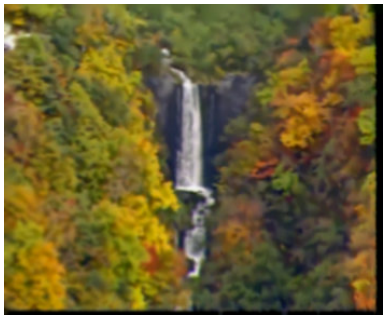
(d) Local method.



(e) Local method.

(f) Local method.

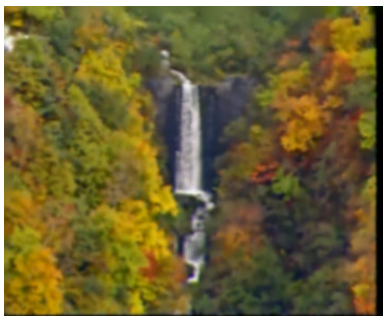
(g) Global-local method.



(h) Global-local method.

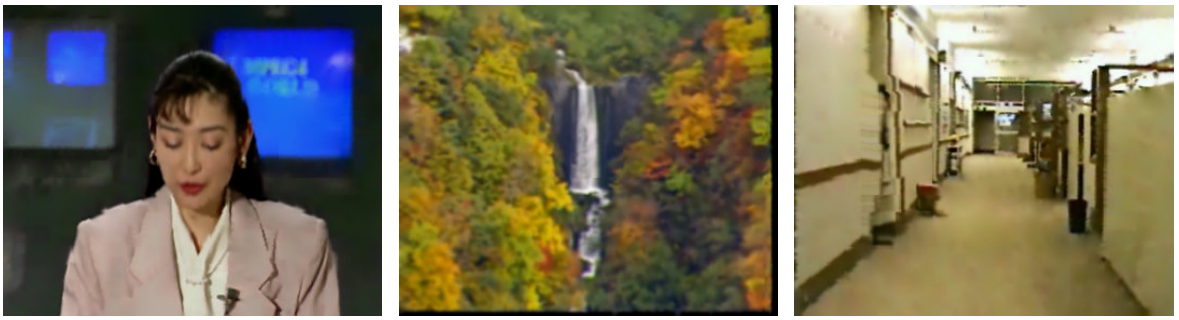
(i) Global-local method.

(j) Adaptive method.



(k) Adaptive method.

(l) Adaptive method.



(m) Coarse-to-fine. (n) Coarse-to-fine. (o) Coarse-to-fine.

Figure 5.8: Predictions of *inter frames* by optical flow for videos, from left to right, *akiyo_cif.y4m*, *waterfall_cif.y4m* and *hall_monitor_cif.y4m*.

Conclusion and Perspectives

In this thesis, we have proposed models and variational methods based on partial differential equations to compress images and videos taking into account the presence of noise in them. The objective is to be able to remove a part of these deteriorations from the image via the action of the compression rather than to carry out a step of pre-filtering on the image to be compressed. For that, this manuscript is cut in three distinct parts. In Part I, we focused on the optimization of the set of pixels to keep in the case where the image to compress is noisy. In Part II, we focused on the estimation of the motion within a video. Finally, in Part III, we implemented a functional codec using the results found in the two previous parts.

In Chapter 1, we considered the problem of finding the best interpolation data in PDE-based compression problems for images with noise. We introduced a geometric variational model to determine a set K that minimizes the distance L^2 between the initial image and its reconstruction from the data in K. In particular, we have studied two points of view to choose an optimal inpainting mask K: a first one using an asymptotic development similar to the topological gradient, and a second one considering sets formed by a finite number of balls called "fat pixels". Both theoretical results underline the importance of the Laplacian of the data and highlight the deep link between the geometric set and the inpainting operator (in our case, the linear diffusion). We have performed several numerical tests and comparisons that demonstrate the effectiveness of the approach in processing images with noise.

In Chapter 2, we studied a new fully parabolic model based on the heat equation for PDE image compression and formulated an associated shape optimization problem for the best choice of data interpolation. Using the results found in Chapter 1, we proved that this model has a relaxed solution under γ-convergence and studied the same two views as in Chapter 1 to choose an optimal inpainting mask. Unlike the stationary case, the iterative approach includes the results of the previous steps in the analytical criterion giving the best choice which now depends on the iterations. Then, we proposed and implemented three algorithms to build the sequence of inpainting masks $(K_n)_n$: *L2-INSTA*, *L2-DEC* and *L2-INC*, and we performed several simulations with different compression rates, different amounts of noise for grayscale images. It appears that in most cases, the masks from the *L2-INC-T* algorithm perform better than the others. Finally, we performed a tonal optimization within the same method by allowing the Dirichlet condition on the mask to change with the iterations. This modification appears to give better results for denoising while compressing an image with noise than the *sparsification* and *densification* algorithms [2, 110] when the noise level is high.

In Chapter 3, we formulated the image compression problem as a shape optimization problem which we reformulated using the so-called *Dirichlet-to-Neumann* operator so that it can be treated by the adjoint method. We have given the adjoint problem and computed the asymptotic development of the variations of the functional. Finally, we presented some numerical results in the case of L^2 error and an approximation of L^1 error. These simulations favored the use of the inpainting mask where the pixel density increases with the analytical criterion obtained with the adjoint method. Moreover, it appeared that our new methods are more efficient in terms of reconstruction quality than the existing ones when the image to be compressed contains Gaussian noise or impulse noise. Moreover, our new methods are easy to implement.

In Chapter 4, we have formulated a new model to deal with the problem of optical flow computation. This new model takes into account luminosity variations like several state-of-the-art approaches, but it differs from them in that it does not require any regularization technique to obtain a well-posed problem. This absence of regularity assumption leads to an optical flow and a luminosity variation that vanish in homogeneous areas and all information is concentrated around the edges. This property can be useful for

video compression that uses motion compensation methods. To solve numerically our new model, we use the optimal transport theory and the Benamou-Brenier theorem and their numerical algorithm. Finally, we present numerical results where we determine the optical flow and brightness variations, reconstruct the images and compare them to the original images. It appears that the error is concentrated near the edges (where there is motion) but vanishes within the homogeneous regions common between the images. The counterpart of this property is a longer computation time than classical optical flow model computation.

Finally, in Chapter 5, we described a variational video codec, composed of "blocks" independent from each other. We have determined which are the best blocks among the methods studied in this thesis and the combination *L1-ADJ-H* with the *local* optical flow method and the compression PAQ seems to be the best for now. We then compared our codec to the JPEG2000 and MPEG-4 Part 2 codecs. Although it seems promising, our codec still lacks maturity. It could be interesting to add residue saving during motion compensation in order to avoid the propagation of errors within a GOP which is very important for the moment.

Among the perspectives considered, we can use the iterative methods proposed in Chapter 2 with the results found in the framework of the adjoint method of Chapter 3. Another direction of research would be to use an inpainting problem minimizing the L^p error (its energy), rather than a model minimizing the L^2 error as developed in this thesis. Finally, we could also look for analytical criteria for inpainting problems using nonlinear PDEs, such as the *edge-enhancing diffusion*. Moreover, there are other types of numerical noises that would be interesting to consider (*speckle* noise, *Poisson* noise, ...). In the same way, it could be interesting to propose an optical flow model more robust to the noise contained in an image sequence.

Finally, we could run the codec in real time by systematically using the finite element method, in order to reduce the number of unknowns when solving the linear system as well as to use domain decomposition methods to be able to process videos in high resolutions. It could also be interesting to make the codec lossless by adding the residuals and to see the impact that this addition has on the final file size.

Bibliography

[1] A. Abdul Halin, R. Yaakob, and A. Aryanfar, *A Comparison of Different Block Matching Algorithms for Motion Estimation*, Procedia Technology, 11 (2013), pp. 199–205.

[2] R. D. Adam, P. Peter, and J. Weickert, *Denoising by Inpainting*, in Scale Space and Variational Methods in Computer Vision, F. Lauze, Y. Dong, and A. B. Dahl, eds., Lecture Notes in Computer Science, Cham, 2017, Springer International Publishing, pp. 121–132.

[3] R. L. Adler, B. P. Kitchens, M. Martens, C. P. Tresser, and C. W. Wu, *The mathematics of halftoning*, IBM Journal of Research and Development, 47 (2003), pp. 5–15. Conference Name: IBM Journal of Research and Development.

[4] N. Ahmed, T. Natarajan, and K. Rao, *Discrete Cosine Transform*, IEEE Transactions on Computers, C-23 (1974), pp. 90–93. Conference Name: IEEE Transactions on Computers.

[5] T. Alt, P. Peter, and J. Weickert, *Learning Sparse Masks for Diffusion-Based Image Inpainting*, in Pattern Recognition and Image Analysis, A. J. Pinho, P. Georgieva, L. F. Teixeira, and J. A. Sánchez, eds., Lecture Notes in Computer Science, Cham, 2022, Springer International Publishing, pp. 528–539.

[6] S. Amstutz, *Sensitivity analysis with respect to a local perturbation of the material property*, Asymptotic Analysis, 49 (2006).

[7] S. Andris, P. Peter, R. M. K. Mohideen, J. Weickert, and S. Hoffmann, *Inpainting-based Video Compression in FullHD*, arXiv:2008.10273 [eess], (2021).

[8] S. Andris, P. Peter, and J. Weickert, *A proof-of-concept framework for PDE-based video compression*, in 2016 Picture Coding Symposium (PCS), Dec. 2016, pp. 1–5. ISSN: 2472-7822.

[9] E. Bae and J. Weickert, *Partial Differential Equations for Interpolation and Compression of Surfaces*, in Mathematical Methods for Curves and Surfaces, M. Dæhlen, M. Floater, T. Lyche, J.-L. Merrien, K. Mørken, and L. L. Schumaker, eds., Lecture Notes in Computer Science, Berlin, Heidelberg, 2010, Springer, pp. 1–14.

[10] S. Baker, S. Roth, D. Scharstein, M. J. Black, J. Lewis, and R. Szeliski, *A Database and Evaluation Methodology for Optical Flow*, in 2007 IEEE 11th International Conference on Computer Vision, Oct. 2007, pp. 1–8. ISSN: 2380-7504.

[11] M. F. Barnsley, S. Demko, and M. J. D. Powell, *Iterated function systems and the global construction of fractals*, Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences, 399 (1985), pp. 243–275. Publisher: Royal Society.

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

[13] B. E. Bayer, *An Optimum Method for Two-Level Rendition of Continuous-Tone Pictures*, Proc.of IEEE Int.conf.on Communications, 1 (1973).

[14] Z. Belhachmi, A. Ben Abda, B. Meftahi, and H. Meftahi, *Topology optimization method with respect to the insertion of small coated inclusion*, Asymptot. Anal., (2018).

[15] Z. Belhachmi, D. Bucur, B. Burgeth, and J. Weickert, *How to Choose Interpolation Data in Images*, SIAM Journal of Applied Mathematics, 70 (2009), pp. 333–352.

[16] Z. Belhachmi and F. Hecht, *Control of the Effects of Regularization on Variational Optic Flow Computations*, Journal of Mathematical Imaging and Vision, 40 (2011). Publisher: Springer Verlag.

[17] ——, *An Adaptive Approach for the Segmentation and the TV-Filtering in the Optic Flow Estimation*, Journal of Mathematical Imaging and Vision, 54 (2016), pp. 358–377.

[18] Z. Belhachmi and T. Jacumin, *Optimal interpolation data for PDE-based compression of images with noise*, Communications in Nonlinear Science and Numerical Simulation, 109 (2022), p. 106278.

[19] J.-D. Benamou and Y. Brenier, *A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem*, Numerische Mathematik, 84 (2000), pp. 375–393.

[20] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, *Image inpainting*, in Proceedings of the 27th annual conference on Computer graphics and interactive techniques, SIGGRAPH '00, USA, July 2000, ACM Press/Addison-Wesley Publishing Co., pp. 417–424.

[21] M. Boix and B. Cantó, *Wavelet Transform application to the compression of images*, Mathematical and Computer Modelling, 52 (2010), pp. 1265–1270.

[22] N. Bonneel, *Optimal Transport for Computer Graphics and Temporal Coherence of Image Processing Algorithms*, habilitation à diriger des recherches, Université Lyon 1 - Claude Bernard, Nov. 2018.

[23] F. Bornemann and T. März, *Fast Image Inpainting Based on Coherence Transport*, Journal of Mathematical Imaging and Vision, 28 (2007), pp. 259–278.

[24] A. Braides, Γ-*Convergence for Beginners*, Oxford University Press, July 2002.

[25] Y. Brenier, *Polar factorization and monotone rearrangement of vector-valued functions*, Communications on Pure and Applied Mathematics, 44 (1991), pp. 375–417.

[26] A. Bruhn, *Variational optic flow computation: accurate modelling and efficient numerics*, PhD thesis, Saarland University, Saarbrucken, 2006.

[27] A. Bruhn, J. Weickert, and C. Schnörr, *Lucas/Kanade Meets Horn/Schunck: Combining Local and Global Optic Flow Methods*, International Journal of Computer Vision, 61 (2005), pp. 211–231.

[28] D. Bucur and G. Buttazzo, *Variational methods in shape optimization problems*, Progress in Nonlinear Differential Equations and their Applications, 65., Birkhäuser, 2005.

[29] D. R. Bull and F. Zhang, *Intelligent image and video compression: communicating pictures*, Elsevier, London San Diego, second edition ed., 2021.

[30] M. Burrows and D. J. Wheeler, *A block-sorting lossless data compression algorithm*, tech. rep., Systems Research Center, 1994.

[31] P. Burt and E. Adelson, *The Laplacian Pyramid as a Compact Image Code*, IEEE Transactions on Communications, 31 (1983), pp. 532–540. Conference Name: IEEE Transactions on Communications.

[32] G. Buttazzo and L. Freddi, *Relaxed optimal control problems and applications to shape optimization*, in Nonlinear Analysis, Differential Equations and Control, F. H. Clarke, R. J. Stern, and G. Sabidussi, eds., NATO Science Series, Springer Netherlands, Dordrecht, 1999, pp. 159–206.

[33] G. Buttazzo, F. Santambrogio, and N. Varchon, *Asymptotics of an optimal compliance-location problem*, ESAIM: Control, Optimisation and Calculus of Variations, 12 (2005).

[34] L. A. Caffarelli, *The regularity of mappings with a convex potential*, Journal of the American Mathematical Society, 5 (1992), pp. 99–104.

[35] X.-C. Cai and M. Sarkis, *A Restricted Additive Schwarz Preconditioner for General Sparse Linear Systems*, SIAM Journal on Scientific Computing, 21 (1999), pp. 792–797.

[36] E. Candes, J. Romberg, and T. Tao, *Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information*, IEEE Transactions on Information Theory, 52 (2006), pp. 489–509. Conference Name: IEEE Transactions on Information Theory.

[37] F. Catté, P.-L. Lions, J.-M. Morel, and T. Coll, *Image Selective Smoothing and Edge Detection by Nonlinear Diffusion*, SIAM Journal on Numerical Analysis, 29 (1992), pp. 182–193. Publisher: Society for Industrial and Applied Mathematics.

[38] J. Cea, A. Gioan, and J. Michel, *Quelques resultats sur l'identification de domaines*, CALCOLO, 10 (1973), pp. 207–232.

[39] T. Chan and J. Shen, *Nontexture Inpainting by Curvature-Driven Diffusions*, Journal of Visual Communication and Image Representation, 12 (2001), pp. 436–449.

[40] T. F. Chan and H.-M. Zhou, *Total Variation Wavelet Thresholding*, Journal of Scientific Computing, 32 (2007), pp. 315–341.

[41] S.-F. Chang, T. Sikora, and A. Purl, *Overview of the MPEG-7 standard*, IEEE Transactions on Circuits and Systems for Video Technology, 11 (2001), pp. 688–695. Conference Name: IEEE Transactions on Circuits and Systems for Video Technology.

[42] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, *Two deterministic half-quadratic regularization algorithms for computed imaging*, in Proceedings of 1st International Conference on Image Processing, vol. 2, Nov. 1994, pp. 168–172 vol.2.

[43] V. Chizhov and J. Weickert, *Efficient Data Optimisation for Harmonic Inpainting with Finite Elements*, arXiv:2105.01586 [eess], (2021). arXiv: 2105.01586.

[44] G. Choquet, *Theory of capacities*, Annales de l'institut Fourier, 5 (1954), pp. 131–295.

[45] ——, *Forme abstraite du théorème de capacitabilité*, Annales de l'institut Fourier, 9 (1959), pp. 83–89.

[46] H. A. Choudhury, N. Sinha, and M. Saikia, *Dynamic Block Matching Algorithm (BMA) Scheduling Algorithm for Fast and Accurate Motion Estimation*, in Proceedings of the International Conference on Computing and Communication Systems, A. K. Maji, G. Saha, S. Das, S. Basu, and J. M. R. S. Tavares, eds., Lecture Notes in Networks and Systems, Singapore, 2021, Springer, pp. 349–357.

[47] S. Curtis, A. Oppenheim, and J. Lim, *Reconstruction of two-dimensional signals from threshold crossings*, in ICASSP '85. IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 10, Apr. 1985, pp. 1057–1060.

[48] G. Dal Maso, *γ-convergence and μ-capacities*, Annali della Scuola Normale Superiore di Pisa - Classe di Scienze, 14 (1987), pp. 423–464.

[49] ——, *An Introduction to Γ-Convergence*, vol. 8, Birkhäuser Boston, 1993. ISSN: 1088-9485.

[50] G. Dal Maso and U. Mosco, *Wiener's criterion and γ-convergence*, Applied Mathematics and Optimization, 15 (1987), pp. 15–63.

[51] L. Davisson, *Data compression using straight line interpolation*, IEEE Transactions on Information Theory, 14 (1968), pp. 390–394. Conference Name: IEEE Transactions on Information Theory.

[52] H. Dell, *Seed Points in PDE-Driven Interpolation*, bachelor's Thesis, University of Saarbrucken, Saarbrucken, 2006.

[53] R. Distasi, M. Nappi, and S. Vitulano, *Image compression by B-tree triangular coding*, IEEE Transactions on Communications, 45 (1997), pp. 1095–1100. Conference Name: IEEE Transactions on Communications.

[54] M. Do and M. Vetterli, *The Finite Ridgelet Transform for Image Representation*, IEEE transactions on image processing : a publication of the IEEE Signal Processing Society, 12 (2003), pp. 16–28.

[55] V. Dolean, S. Lanteri, and F. Nataf, *Convergence analysis of additive Schwarz for the Euler equations*, Applied Numerical Mathematics, 49 (2004), pp. 153–186.

[56] D. G. Duffy, *Green's functions with applications, second edition*, CRC Press, Jan. 2015. Publication Title: Green's Functions with Applications, Second Edition.

[57] L. C. Evans, *Partial Differential Equations: Second Edition*, American Mathematical Society, 2 ed., Mar. 2010.

[58] L. C. Evans and W. Gangbo, *Differential equations methods for the Monge-Kantorovich mass transfer problem*, Memoirs of the American Mathematical Society, 137 (1999), pp. 0–0.

[59] H. Fassold, *A qualitative investigation of optical flow algorithms for video denoising*, Apr. 2022. arXiv:2204.08791 [cs].

[60] A. Figalli and F. Glaudo, *An Invitation to Optimal Transport, Wasserstein Distances, and Gradient Flows*, EMS Press, 1 ed., Aug. 2021.

[61] R. W. Floyd and L. Steinberg, *Adaptive algorithm for spatial greyscale*, Proceedings of the Society for Information Display, 17 (1976), pp. 75–77.

[62] G. B. Folland, *Real Analysis: Modern Techniques and Their Applications*, Wiley, 2 ed., June 2013.

[63] M. Fortin and R. Glowinski, *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems.*, Elsevier Science, Amsterdam, 2000. OCLC: 1049710365.

[64] H. Freeman, *On the Encoding of Arbitrary Geometric Configurations*, IRE Transactions on Electronic Computers, EC-10 (1961), pp. 260–268. Conference Name: IRE Transactions on Electronic Computers.

[65] I. Galić, J. Weickert, M. Welk, A. Bruhn, A. Belyaev, and H.-P. Seidel, *Towards PDE-Based Image Compression*, in Variational, Geometric, and Level Set Methods in Computer Vision, N. Paragios, O. Faugeras, T. Chan, and C. Schnörr, eds., Lecture Notes in Computer Science, Berlin, Heidelberg, 2005, Springer, pp. 37–48.

[66] I. Galić, J. Weickert, M. Welk, A. Bruhn, A. Belyaev, and H.-P. Seidel, *Image Compression with Anisotropic Diffusion*, Journal of Mathematical Imaging and Vision, 31 (2008), pp. 255–269.

[67] W. Gangbo and R. J. McCann, *The geometry of optimal transportation*, Acta Mathematica, 177 (1996), pp. 113–161.

[68] S. Garreau, P. Guillaume, and M. Masmoudi, *The Topological Asymptotic for PDE Systems: The Elasticity Case*, SIAM J. Control and Optimization, 39 (2001), pp. 1756–1778.

[69] M. A. Gennert and S. Negahdaripour, *Relaxing the Brightness Constancy Assumption in Computing Optical Flow*, tech. rep., Massachusetts Institute of Technology, June 1987. Accepted: 2004-10-04T14:57:42Z.

[70] M. Ghanbari, *The cross-search algorithm for motion estimation (image coding)*, IEEE Transactions on Communications, 38 (1990), pp. 950–953. Conference Name: IEEE Transactions on Communications.

[71] A. C. Gilbert, S. Guha, P. Indyk, S. Muthukrishnan, and M. Strauss, *Near-optimal sparse fourier representations via sampling*, in Proceedings of the thiry-fourth annual ACM symposium on Theory of computing, STOC '02, New York, NY, USA, May 2002, Association for Computing Machinery, pp. 152–161.

[72] D. Gilliocq-Hirtz, *Techniques variationnelles et calcul parallèle en imagerie : Estimation du flot optique avec luminosité variable en petits et larges déplacements*, phdthesis, Université de Haute-Alsace, Mulhouse, July 2016.

[73] D. Gilliocq-Hirtz and Z. Belhachmi, *A massively parallel multi-level approach to a domain decomposition method for the optical flow estimation with varying illumination*, Aug. 2015. arXiv:1508.02977 [cs].

[74] W. M. Goodall, *Television by pulse code modulation*, The Bell System Technical Journal, 30 (1951), pp. 33–49. Conference Name: The Bell System Technical Journal.

[75] R. Gray, *Vector quantization*, IEEE ASSP Magazine, 1 (1984), pp. 4–29. Conference Name: IEEE ASSP Magazine.

[76] H. Grossauer, *Inpainting of Movies Using Optical Flow*, in Mathematical Models for Registration and Applications to Medical Imaging, O. Scherzer, ed., Mathematics in industry, Springer, Berlin, Heidelberg, 2006, pp. 151–162.

[77] P. Guillaume and K. Idris, *The Topological Asymptotic Expansion for the Dirichlet Problem*, SIAM J. Control and Optimization, 41 (2002), pp. 1042–1072.

[78] T. Gustafsson and G. D. McBain, *scikit-fem: A Python package for finite element assembly*, Journal of Open Source Software, 5 (2020), p. 2369.

[79] H.-M. Hang, Y.-M. Chou, and S.-C. Cheng, *Motion Estimation for Video Coding Standards*, VLSI Signal Processing, 17 (1997), pp. 113–136.

[80] C. W. Harrison, *Experiments with linear prediction in television*, The Bell System Technical Journal, 31 (1952), pp. 764–783. Conference Name: The Bell System Technical Journal.

[81] F. Hecht, *New development in FreeFem++*, Journal of Numerical Mathematics, 20 (2012), pp. 251–265.

[82] D. Hochman, H. Katzman, and D. Weber, *Application of redundancy reduction to television bandwidth compression*, Proceedings of the IEEE, 55 (1967), pp. 263–266. Conference Name: Proceedings of the IEEE.

[83] L. Hoeltgen, M. Mainberger, S. Hoffmann, J. Weickert, C. H. Tang, S. Setzer, D. Johannsen, F. Neumann, and B. Doerr, *Optimising Spatial and Tonal Data for PDE-based Inpainting*, arXiv:1506.04566 [cs, math], (2015). arXiv: 1506.04566.

[84] L. Hoeltgen, S. Setzer, and J. Weickert, *An Optimal Control Approach to Find Sparse Data for Laplace Interpolation*, in Energy Minimization Methods in Computer Vision and Pattern Recognition, A. Heyden, F. Kahl, C. Olsson, M. Oskarsson, and X.-C. Tai, eds., Lecture Notes in Computer Science, Berlin, Heidelberg, 2013, Springer, pp. 151–164.

[85] B. Horn and B. Schunck, *Determining Optical Flow*, Artificial Intelligence, 17 (1981), pp. 185–203.

[86] D. A. Huffman, *A Method for the Construction of Minimum-Redundancy Codes*, Proceedings of the IRE, 40 (1952), pp. 1098–1101. Conference Name: Proceedings of the IRE.

[87] I. Ismail, A. Hamdy, and R. Frig, *Studying the effect of down sampling and spatial interpolation on fractal image compression*, Proceedings, ICCES'2010 - 2010 International Conference on Computer Engineering and Systems, (2010).

[88] J. Jain and A. Jain, *Displacement Measurement and Its Application in Interframe Image Coding*, IEEE Transactions on Communications, 29 (1981), pp. 1799–1808. Conference Name: IEEE Transactions on Communications.

[89] B. Janssen, F. Kanters, R. Duits, L. Florack, and B. ter Haar Romeny, *A Linear Image Reconstruction Framework Based on Sobolev Type Inner Products*, in Scale Space and PDE Methods in Computer Vision, R. Kimmel, N. A. Sochen, and J. Weickert, eds., Lecture Notes in Computer Science, Berlin, Heidelberg, 2005, Springer, pp. 85–96.

[90] F. A. Jassim and F. H. Altaany, *Image Interpolation Using Kriging Technique for Spatial Data*, Feb. 2013. arXiv:1302.1294 [cs].

[91] H. Jia and L. Zhang, *Directional Cross Diamond Search Algorithm for Fast Block Motion Estimation*, June 2008. arXiv:0806.0689 [cs].

[92] W. Jiang and M. Zhou, *A fast BMA based on combining search candidate subsampling and APDS*, in 2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763), vol. 2, June 2004, pp. 1115–1118 Vol.2.

[93] M. A. Joshi, M. S. Raval, Y. H. Dandawate, K. R. Joshi, and S. P. Metkar, *Image and video compression: fundamentals, techniques, and applications*, Chapman and Hall/CRC, 2015. OCLC: 895660985.

[94] F. Kanters, M. Lillholm, R. Duits, B. Janssen, B. Platel, L. Florack, and B. ter Haar Romeny, *On Image Reconstruction from Multiscale Top Points*, in Scale Space and PDE Methods in Computer Vision, R. Kimmel, N. A. Sochen, and J. Weickert, eds., Lecture Notes in Computer Science, Berlin, Heidelberg, 2005, Springer, pp. 431–442.

[95] L. V. Kantorovich, *On a Problem of Monge*, Journal of Mathematical Sciences, 133 (2006), pp. 1383–1383.

[96] J. Kinnunen and O. Martio, *The Sobolev capacity on metric spaces*, Annales Academiae Scientiarum Fennicae. Mathematica, 21 (1996).

[97] C. Kirisits, L. F. Lang, and O. Scherzer, *Optical Flow on Evolving Surfaces with an Application to the Analysis of 4D Microscopy Data*, in Scale Space and Variational Methods in Computer Vision, A. Kuijper, K. Bredies, T. Pock, and H. Bischof, eds., Berlin, Heidelberg, 2013, Springer, pp. 246–257.

[98] C. Kortman, *Redundancy reduction—A practical method of data compression*, Proceedings of the IEEE, 55 (1967), pp. 253–263. Conference Name: Proceedings of the IEEE.

[99] H. Köstler, M. Stürmer, and U. Rüde, *PDE based Video Compression in Real Time*, 2007.

[100] S. Larnier, J. Fehrenbach, and M. Masmoudi, *The Topological Gradient Method: From Optimal Design to Image Processing*, Milan Journal of Mathematics, 80 (2012).

[101] D. Laurent, N. Dyn, and A. Iske, *Image compression by linear splines over adaptive triangulations*, Signal Processing, 86 (2006), pp. 1604–1616.

[102] F. Lenzen and O. Scherzer, *Partial Differential Equations for Zooming, Deinterlacing and Dejittering*, International Journal of Computer Vision, 92 (2011), pp. 162–176.

[103] W. Li, W. Sun, Y. Zhao, Z. Yuan, and Y. Liu, *Deep Image Compression with Residual Learning*, Applied Sciences, 10 (2020), p. 4023.

[104] J.-L. Lions, Y. Maday, and G. Turinici, *Résolution d'EDP par un schéma en temps < pararéel >*, Comptes rendus de l'Académie des sciences. Série I, Mathématique, 332 (2001), pp. 661–668. Publisher: Elsevier.

[105] P.-L. Lions, *On the Schwarz alternating method. III: a variant for nonoverlapping subdomains*, in Third international symposium on domain decomposition methods for partial differential equations, vol. 6, SIAM Philadelphia, 1990, pp. 202–223.

[106] D. Liu, X. Sun, F. Wu, S. Li, and Y.-Q. Zhang, *Image Compression With Edge-Based Inpainting*, IEEE Transactions on Circuits and Systems for Video Technology, 17 (2007), pp. 1273–1287. Conference Name: IEEE Transactions on Circuits and Systems for Video Technology.

[107] B. F. Logan, *Information in the zero crossings of bandpass signals*, The Bell System Technical Journal, 56 (1977), pp. 487–510. Conference Name: The Bell System Technical Journal.

[108] B. D. Lucas and T. Kanade, *An iterative image registration technique with an application to stereo vision*, in Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2, IJCAI'81, San Francisco, CA, USA, Aug. 1981, Morgan Kaufmann Publishers Inc., pp. 674–679.

[109] M. V. Mahoney, *Adaptive Weighing of Context Models for Lossless Data Compression*, 2005.

[110] M. Mainberger, S. Hoffmann, J. Weickert, C. H. Tang, D. Johannsen, F. Neumann, and B. Doerr, *Optimising Spatial and Tonal Data for Homogeneous Diffusion Inpainting*, in Scale Space and Variational Methods in Computer Vision, A. M. Bruckstein, B. M. ter Haar Romeny, A. M. Bronstein, and M. M. Bronstein, eds., Lecture Notes in Computer Science, Berlin, Heidelberg, 2012, Springer, pp. 26–37.

[111] P. Maragos, R. Schafer, and R. Mersereau, *Two-dimensional linear prediction and its application to adaptive predictive coding of images*, IEEE Transactions on Acoustics, Speech, and Signal Processing, 32 (1984), pp. 1213–1229. Conference Name: IEEE Transactions on Acoustics, Speech, and Signal Processing.

[112] D. Marwood, P. Massimino, M. Covell, and S. Baluja, *Representing Images in 200 Bytes: Compression via Triangulation*, in 2018 25th IEEE International Conference on Image Processing (ICIP), Oct. 2018, pp. 405–409. ISSN: 2381-8549.

[113] S. Masnou and J.-M. Morel, *Level lines based disocclusion*, in Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269), Oct. 1998, pp. 259–263 vol.3.

[114] G. Megala and S. Prabu, *State-Of-The-Art In Video Processing: Compression, Optimization And Retrieval*, Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12 (2021), pp. 1256–1272.

[115] R. M. K. Mohideen, P. Peter, T. Alt, J. Weickert, and A. Scheer, *Compressing Colour Images with Joint Inpainting and Prediction*, arXiv:2010.09866 [eess], (2020). arXiv: 2010.09866.

[116] G. Monge, *Mémoire sur la théorie des déblais et des remblais*, De l'Imprimerie Royale, Paris, 1781. OCLC: 51928110.

[117] A. Netravali and J. Limb, *Picture coding: A review*, Proceedings of the IEEE, 68 (1980), pp. 366–406. Conference Name: Proceedings of the IEEE.

[118] A. N. Netravali and J. D. Robbins, *Motion-compensated television coding: Part I*, The Bell System Technical Journal, 58 (1979), pp. 631–670. Conference Name: The Bell System Technical Journal.

[119] M. Nikolova, *Minimizers of Cost-Functions Involving Nonsmooth Data-Fidelity Terms. Application to the Processing of Outliers*, SIAM J. Numerical Analysis, 40 (2002), pp. 965–994.

[120] ——, *A Variational Approach to Remove Outliers and Impulse Noise*, Journal of Mathematical Imaging and Vision, 20 (2004).

[121] Y. Ninomiya and Y. Ohtsuka, *A Motion-Compensated Interframe Coding Scheme for Television Pictures*, IEEE Transactions on Communications, 30 (1982), pp. 201–211. Conference Name: IEEE Transactions on Communications.

[122] K. Oldham, J. Myland, and J. Spanier, *An Atlas of Functions*, Springer US, 2009. Publication Title: An Atlas of Functions.

[123] N. Papadakis, *Optimal Transport for Image Processing*, habilitation à diriger des recherches, Université de Bordeaux, Bordeaux, Dec. 2015.

[124] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert, *Highly Accurate Optic Flow Computation with Theoretically Justified Warping*, International Journal of Computer Vision, 67 (2006), pp. 141–158.

[125] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros, *Context Encoders: Feature Learning by Inpainting*, in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016, pp. 2536–2544. ISSN: 1063-6919.

[126] S. Patil and G. Kale, *Survey on GPU Based Linear Solver*, International Journal For Science Technology And Engineering, 2 (2016), pp. 409–412.

[127] L. E. Payne and H. F. Weinberger, *An optimal Poincaré inequality for convex domains*, Archive for Rational Mechanics and Analysis, 5 (1960), pp. 286–292.

[128] J. S. Pérez, N. M. López, and A. S. d. l. Nuez, *Robust Optical Flow Estimation*, Image Processing On Line, 3 (2013), pp. 252–270.

[129] P. Perona and J. Malik, *Scale-space and edge detection using anisotropic diffusion*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 12 (1990), pp. 629–639. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[130] P. Peter, J. Contelly, and J. Weickert, *Compressing Audio Signals with Inpainting-Based Sparsification*, in Scale Space and Variational Methods in Computer Vision, Lecture Notes in Computer Science, Cham, June 2019, Springer International Publishing, pp. 92–103.

[131] P. Peter, S. Hoffmann, F. Nedwed, L. Hoeltgen, and J. Weickert, *From Optimised Inpainting with Linear PDEs Towards Competitive Image Compression Codecs*, in Image and Video Technology, T. Bräunl, B. McCane, M. Rivera, and X. Yu, eds., Lecture Notes in Computer Science, Cham, 2016, Springer International Publishing, pp. 63–74.

[132] P. Peter, L. Kaufhold, and J. Weickert, *Turning Diffusion-Based Image Colorization Into Efficient Color Compression*, IEEE Transactions on Image Processing, PP (2016), pp. 1–1.

[133] P. Peter, K. Schrader, T. Alt, and J. Weickert, *Deep Spatial and Tonal Optimisation for Diffusion Inpainting*, Aug. 2022. arXiv:2208.14371 [eess] version: 1.

[134] P. Peter and J. Weickert, *Colour image compression with anisotropic diffusion*, 2014 IEEE International Conference on Image Processing, ICIP 2014, (2015), pp. 4822–4826.

[135] T. Poggio, H. K. Nishihara, and K. R. K. Nielsen, *Zero-Crossings and Spatiotemporal Interpolation in Vision: Aliasing and Electrical Coupling between Sensors.*, tech. rep., MASSACHUSETTS INST OF TECH CAMBRIDGE ARTIFICIAL INTELLIGENCE LAB, May 1982. Section: Technical Reports.

[136] J. Prades-Nebot, M. Morbee, and E. J. Delp, *Generalized PCM Coding of Images*, IEEE Transactions on Image Processing, 21 (2012), pp. 3801–3806. Conference Name: IEEE Transactions on Image Processing.

[137] W. Pratt, J. Kane, and H. Andrews, *Hadamard transform image coding*, Proceedings of the IEEE, 57 (1969), pp. 58–68.

[138] M. Rabbani and P. W. Jones, *Digital image compression techniques*, no. v. TT 7 in Tutorial texts in optical engineering, Spie Optical Engineering Press, Bellingham, Wash., USA, 1991.

[139] J. Rissanen and G. G. Langdon, *Arithmetic Coding*, IBM Journal of Research and Development, 23 (1979), pp. 149–162. Conference Name: IBM Journal of Research and Development.

[140] A. Robinson and C. Cherry, *Results of a prototype television bandwidth compression scheme*, Proceedings of the IEEE, 55 (1967), pp. 356–364. Conference Name: Proceedings of the IEEE.

[141] D. Rotem and Y. Zeevi, *Image Reconstruction from Zero Crossings*, Acoustics, Speech and Signal Processing, IEEE Transactions on, 34 (1986), pp. 1269–1277.

[142] E. Ryu, W. Li, P. Yin, and S. Osher, *Unbalanced and Partial L1 Monge–Kantorovich Problem: A Scalable Parallel First-Order Method*, Journal of Scientific Computing, (2017).

[143] D. Salomon, *Data compression: the complete reference*, Springer, London, 4th ed ed., 2007.

[144] F. Santambrogio, *Optimal transport for applied mathematicians: calculus of variations, PDEs, and modeling*, no. volume 87 in Progress in nonlinear differential equations and their applications, Birkhäuser, Cham Heidelberg New York, 2015.

[145] O. Scherzer, M. Grasmair, H. Grossauer, M. Haltmeier, and F. Lenzen, *Variational Methods in Imaging*, Springer Science & Business Media, Sept. 2008. Google-Books-ID: 6tq9DiTVnfAC.

[146] C. Schmaltz, P. Peter, M. Mainberger, F. Huth, J. Weickert, and A. Bruhn, *Understanding, Optimising, and Extending Data Compression with Anisotropic Diffusion*, International Journal of Computer Vision, 108 (2014).

[147] C. Schmaltz, J. Weickert, and A. Bruhn, *Beating the Quality of JPEG 2000 with Anisotropic Diffusion*, in Pattern Recognition, J. Denzler, G. Notni, and H. Süße, eds., Lecture Notes in Computer Science, Berlin, Heidelberg, 2009, Springer, pp. 452–461.

[148] K. Schrader, T. Alt, J. Weickert, and M. Ertel, *CNN-based Euler's Elastica Inpainting with Deep Energy and Deep Image Prior*, July 2022. arXiv:2207.07921 [cs, eess] version: 1.

[149] A. Secord, *Weighted Voronoi stippling*, in Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering, NPAR '02, New York, NY, USA, June 2002, Association for Computing Machinery, pp. 37–43.

[150] C. E. Shannon, *A mathematical theory of communication*, The Bell System Technical Journal, 27 (1948), pp. 379–423. Conference Name: The Bell System Technical Journal.

[151] Y. Shi, Y. Ge, J. Wang, and J. Mao, *AlphaVC: High-Performance and Efficient Learned Video Compression*, July 2022. arXiv:2207.14678 [cs].

[152] S. M. Smith, *ASSET-2: Real-Time Motion Segmentation and Object Tracking*, Real-Time Imaging, 4 (1998), pp. 21–40.

[153] J. A. Stuller, A. N. Netravali, and J. D. Robbins, *Interframe television coding using gain and displacement compensation*, The Bell System Technical Journal, 59 (1980), pp. 1227–1240. Conference Name: The Bell System Technical Journal.

[154] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, *Overview of the High Efficiency Video Coding (HEVC) Standard*, IEEE Transactions on Circuits and Systems for Video Technology, 22 (2012), pp. 1649–1668. Conference Name: IEEE Transactions on Circuits and Systems for Video Technology.

[155] D. S. Taubman and M. W. Marcellin, *JPEG2000: image compression fundamentals, standards, and practice*, no. 642 in The Kluwer international series in engineering and computer science, Springer Science + Business Media, LLC, New York, softcover reprint of the hardcover 1st edition 2002, third printing ed., 2004.

[156] J. Thibault and I. Senocak, *CUDA Implementation of a Navier-Stokes Solver on Multi-GPU Desktop Platforms for Incompressible Flows*, in 47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition, Orlando, Florida, Jan. 2009, American Institute of Aeronautics and Astronautics.

[157] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems*, SIAM Review, 21 (1979), pp. 266–267. Publisher: Society for Industrial and Applied Mathematics.

[158] L. C. Torres, L. M. Pereira, and M. H. Amini, *A Survey on Optimal Transport for Machine Learning: Theory and Applications*, arXiv:2106.01963 [cs], (2021). arXiv: 2106.01963.

[159] L. Torres-Urgell and R. Lynn Kirlin, *Adaptive image compression using Karhunen-Loeve transform*, Signal Processing, 21 (1990), pp. 303–313.

[160] D. Tschumperle and R. Deriche, *Vector-valued image regularization with PDEs: a common framework for different applications*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 27 (2005), pp. 506–517. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[161] R. Ulichney, *Digital Halftoning*, MIT Press, Cambridge, MA, USA, June 1987.

[162] R. Ulichney, *Dithering with blue noise*, Proceedings of the IEEE, 76 (1988), pp. 56–79. Conference Name: Proceedings of the IEEE.

[163] B. Vidhya and V. Ranganathan, *Kriging Interpolation Technique with Triangulated Irregular Network for Image Compression Using Image Inpainting*, Journal of Computational and Theoretical Nanoscience, 14 (2017), pp. 5756–5760.

[164] C. Villani, *Topics in optimal transportation*, no. v. 58 in Graduate studies in mathematics, American Mathematical Society, Providence, RI, 2003.

[165] G. Wallace, *The JPEG still picture compression standard*, IEEE Transactions on Consumer Electronics, 38 (1992), pp. xviii–xxxiv. Conference Name: IEEE Transactions on Consumer Electronics.

[166] D. Wang, L. Zhang, R. Klepko, and A. Vincent, *A wavelet-based video codec and its performance*, Jan. 2007.

[167] H. Wang, M. Wang, T. Hintz, Q. Wu, and X. He, *VSA-based fractal image compression*, 13th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2005, WSCG'2005 - In Co-operation with EUROGRAPHICS, Full Papers, (2005), pp. 89–96. ISBN: 9788090310070.

[168] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers, *An Improved Algorithm for TV-L1 Optical Flow*, in Statistical and Geometrical Approaches to Visual Motion Analysis, D. Cremers, B. Rosenhahn, A. L. Yuille, and F. R. Schmidt, eds., Lecture Notes in Computer Science, Berlin, Heidelberg, 2009, Springer, pp. 23–45.

[169] J. Weickert, *Theoretical Foundations Of Anisotropic Diffusion In Image Processing*, Computing, Suppl, 11 (1996), pp. 221–236.

[170] J. Weickert, A. Bruhn, N. Papenberg, and T. Brox, *Variational Optic Flow Computation: From Continuous Models to Algorithms*, in International Workshop on Computer Vision and Image Analysis (ed. L. Alvarez), IWCVIA'03, Las Palmas de Gran Canaria, 2003.

[171] J. Weickert, S. Ishikawa, and A. Imiya, *Linear Scale-Space has First been Proposed in Japan*, Journal of Mathematical Imaging and Vision, 10 (1999), pp. 237–252.

[172] Welch, *A Technique for High-Performance Data Compression*, Computer, 17 (1984), pp. 8–19. Conference Name: Computer.

[173] C.-Y. Wu, N. Singhal, and P. Krähenbühl, *Video Compression through Image Interpolation*, Apr. 2018. arXiv:1804.06919 [cs].

[174] J. Xie, L. Xu, and E. Chen, *Image Denoising and Inpainting with Deep Neural Networks*, Advances in Neural Information Processing Systems, 1 (2012).

[175] B. Zalik and N. Lukac, *Chain code lossless compression using move-to-front transform and adaptive run-length encoding*, Signal Processing Image Communication, 29 (2014), pp. 96–106.

[176] W. P. Ziemer, *Extremal length and p-capacity.*, Michigan Mathematical Journal, 16 (1969), pp. 43–51. Publisher: University of Michigan, Department of Mathematics.

Appendix A

Some Proofs for the Stationary Model Chapter 1

For the reader ease, some tools used in this appendix are recalled in Appendix C.

A.1 Proofs for the Analysis of the Model

In this appendix are some of the proofs of theorems used in Section 1.1.

Proof of Theorem 1.1.1. For every K in $\mathcal{K}_\delta(D)$, we have $K \subset D$. Thus ∞_K is in $\mathcal{M}_0^\delta(D)$, and $\mathcal{K}_\delta(D) \subseteq \mathcal{M}_0^\delta(D)$ if we identify a set of $\mathcal{K}_\delta(D)$ to its capacitary measure. Thanks to the γ-compactness, we have $\mathrm{cl}_\gamma\, \mathcal{K}_\delta(D) \subseteq \mathcal{M}_0^\delta(D)$.

Conversely, let μ be in $\mathcal{M}_0^\delta(D)$. We need to prove that μ is in $\mathrm{cl}_\gamma\, \mathcal{K}_\delta(D)$ i.e. μ is the γ-limit of elements of $\mathcal{K}_\delta(D)$. This is to be understood in the sense : there exist elements $(K_n)_n$ of $\mathcal{K}_\delta(D)$ such that, ∞_{K_n} γ-converge to μ. Since $\mathcal{M}_0(D)$ is dense and $\mathcal{M}_0^\delta(D) \subset \mathcal{M}_0(D)$, we obtain that there exists a sequence of subsets of D, $(K_n)_n$, such that, ∞_{K_n} γ-converge to μ. We need to show that K_n are in $\mathcal{K}_\delta(D)$. By the localization argument, we can choose $(K_n)_n$ such that $K_n \subseteq (D^{-\delta})^{1/n}$. Making a homothety $\varepsilon_n K_n$, for $\varepsilon_n > 0$ such that $\varepsilon_n K_n \subseteq D^{-\delta}$, we have $\varepsilon_n K_n \in \mathcal{K}_\delta(D)$. Moreover, we can choose ε_n such that $\varepsilon_n \to 1$, therefore $\varepsilon_n K_n$ γ-converge to μ. □

Proof of Theorem 1.1.3. This proof is similar to the one of Theorem 3.5 in [15], we give it for the sake of completeness. Assume that $(\mu_n)_n$ in $\mathcal{M}_0^\delta(D)$ γ-converges to μ. By γ-compacity, Proposition C.5.4 in Appendix C, μ is in $\mathcal{M}_0^\delta(D)$. Now we prove that F_{μ_n} Γ-converges to F_μ in $L^2(D)$.

- $\liminf$: Let $(u_n)_n$ be a sequence in $H^1(D)$ which converges in $L^2(D)$ to u. Let $\varphi \in C_c^\infty(D)$, $0 \leq \varphi \leq 1$ and $\varphi = 1$ in $D^{-\delta}$. Then $(u_n\varphi)_n$ is a sequence in $H^1(D^{-\delta})$ and $u_n\varphi \to_{n\to+\infty} u\varphi$ in $L^2(D)$. Since $(\mu_n)_n$ $\gamma(F)$-converges to μ, we have,

$$\liminf_{n\to+\infty} F_{\mu_n}(u_n\varphi) \geq F_\mu(u\varphi)$$

i.e.

$$\liminf_{n\to+\infty} \Big(\alpha \int_D |\nabla(u_n\varphi)|^2\ dx + \int_D (u_n\varphi - f)^2\ d\mu_n\Big) \geq \alpha \int_D |\nabla(u\varphi)|^2\ dx + \int_D (u\varphi - f)^2\ d\mu.$$

it follows that,

$$\liminf_{n\to+\infty} \Big(\alpha \int_D |\varphi\nabla u_n|^2\ dx + \alpha \int_D |u_n\nabla\varphi|^2\ dx + 2\alpha \int_D u_n\varphi\nabla u_n \cdot \nabla\varphi\ dx + \int_D (u_n\varphi - f)^2\ d\mu_n\Big)$$

$$\geq \alpha \int_D |\varphi\nabla u|^2\ dx + \alpha \int_D |u\nabla\varphi|^2\ dx + 2\alpha \int_D u\varphi\nabla u \cdot \nabla\varphi\ dx + \int_D (u\varphi - f)^2\ d\mu.$$

Thus

$$\liminf_{n\to+\infty}\Big(\alpha\int_D|\varphi\nabla u_n|^2\,dx+\int_D(u_n\varphi-f)^2\,d\mu_n\Big)\geq\alpha\int_D|\varphi\nabla u|^2\,dx+\int_D(u\varphi-f)^2\,d\mu.$$

We have used $0\leq\varphi\leq 1$, andhave taken the sup over φ,

$$\liminf_{n\to+\infty}\Big(\alpha\int_D|\nabla u_n|^2\,dx+\int_D(u_n-f)^2\,d\mu_n\Big)\geq\sup_{\varphi}\Big(\alpha\int_D|\varphi\nabla u|^2\,dx+\int_D(u\varphi-f)^2\,d\mu\Big).$$

Since μ is equals to 0 in $D\setminus D^{-\delta}$, we have

$$\liminf_{n\to+\infty}\Big(\alpha\int_D|\nabla u_n|^2\,dx+\int_D(u_n-f)^2\,d\mu_n\Big)\geq\sup_{\varphi}\Big(\alpha\int_D|\nabla u|^2\varphi^2\,dx\Big)+\int_D(u-f)^2\,d\mu.$$

We get the Γ-lim inf.

- lim sup : Let u be in $H^1(D)$, such that the maximum principle is fulfilled, i.e. $|u|\leq|f|_\infty$, and let $\tilde{u}$ denotes its extension to $H^1_0(D^\delta)$, where D^δ is the dilatation by a factor $\delta>0$ of D. By the locality property of the γ-convergence, Proposition C.5.5 in Appendix C, we have that μ_n $\gamma(G)$-converges to μ in D^δ, where G is

$$G_\mu(u)=\alpha\int_D|\nabla u|^2\,dx+\varepsilon\alpha\int_{D^\delta\setminus D}|\nabla u|^2\,dx+\int_D(u-f)^2\,d\mu,$$

for $\varepsilon>0$. Hence, there exists a sequence $(u_n^\varepsilon)_n$ of $H^1_0(D^\delta)$ such that u_n^ε converges to $\tilde{u}$ in $L^2(D^\delta)$ and $G_\mu(\tilde{u})\geq\limsup_{n\to+\infty}G_{\mu_n}(u_n^\varepsilon)$ i.e.

$$\alpha\int_D|\nabla\tilde{u}|^2\,dx+\varepsilon\alpha\int_{D^\delta\setminus D}|\nabla\tilde{u}|^2\,dx+\int_D(\tilde{u}-f)^2\,d\mu$$

$$\geq\limsup_{n\to+\infty}\alpha\int_D|\nabla u_n^\varepsilon|^2\,dx+\varepsilon\alpha\int_{D^\delta\setminus D}|\nabla u_n^\varepsilon|^2\,dx+\int_D(u_n^\varepsilon-f)^2\,d\mu_n.$$

Thus, we have

$$\alpha\int_D|\nabla\tilde{u}|^2\,dx+\varepsilon\alpha\int_{D^\delta\setminus D}|\nabla\tilde{u}|^2\,dx+\int_D(\tilde{u}-f)^2\,d\mu$$

$$\geq\limsup_{n\to+\infty}\alpha\int_D|\nabla u_n^\varepsilon|^2\,dx+\int_D(u_n^\varepsilon-f)^2\,d\mu_n.$$

Since $\tilde{u}$ is fixed, we let ε tends to 0 and extract by a diagonal procedure a subsequence $u_n^{\varepsilon_n}$ converging in $L^2(D^\delta)$ to $\tilde{u}$ i.e.

$$\alpha\int_D|\nabla\tilde{u}|^2\,dx+\int_D(\tilde{u}-f)^2\,d\mu\geq\limsup_{n\to+\infty}\alpha\int_D|\nabla u_n^{\varepsilon_n}|^2\,dx+\int_D(u_n^{\varepsilon_n}-f)^2\,d\mu_n.$$

Setting $u_n:=u_n^{\varepsilon_n}|_D\in H^1(D)$, we get (since $u=\tilde{u}|_D$),

$$\alpha\int_D|\nabla u|^2\,dx+\int_D(u-f)^2\,d\mu\geq\limsup_{n\to+\infty}\alpha\int_D|\nabla u_n|^2\,dx+\int_D(u_n-f)^2\,d\mu_n.$$

□

Proof of Theorem 1.1.4. Let $(K_n)_n\subset\mathcal{K}_\delta(D)$ be a maximizing sequence of $E(\infty_K)-\beta\,\mathrm{cap}_\nu(\infty_K)$ i.e.

$$\lim_{n\to+\infty}E(\infty_{K_n})-\beta\,\mathrm{cap}_\nu(\infty_{K_n})=\sup_{K\in\mathcal{K}_\delta(D)}\big(E(\infty_K)-\beta\,\mathrm{cap}_\nu(\infty_K)\big).$$

As $\mathcal{M}_0^\delta(D)$ is a compact with respect to the γ-convergence (Proposition 1.1.2 in Chapter 1), we can extract, from $(\infty_{K_n})_n\subset\mathcal{M}_0^\delta(D)$ a γ-convergent subsequence. We denote by $\mu_{\lim}$ this γ-limit. We denote by $G(\mu_{\lim})$ the value

$$G(\mu_{\lim}):=\lim_{n\to+\infty}E(\infty_{K_n})-\beta\,\mathrm{cap}_\nu(\infty_{K_n}).$$

By definition of the γ-convergence, we have $\Gamma - \lim_{n\to+\infty} F_{\infty_{K_n}} = F_{\mu_{\lim}}$. Since $F_{\infty_{K_n}}$ is equicoercive, we can apply Theorem 7.8 in [49]. Thus

$$E(\mu_{\lim}) := \min_{u\in H^1(D)} F_{\mu_{\lim}}(u) = \lim_{n\to+\infty} \inf_{u\in H^1(D)} F_{\infty_{K_n}}(u) =: \lim_{n\to+\infty} E(\infty_{K_n}).$$

In addition, by Theorem 1.1.3 in Chapter 1 and the uniqueness of the limit, we have

$$\lim_{n\to+\infty} E(\infty_{K_n}) - \beta \operatorname{cap}_\nu(\infty_{K_n}) = G(\mu_{\lim}) = E(\mu_{\lim}) - \beta \operatorname{cap}_\nu(\mu_{\lim}).$$

Finally, we have

$$\begin{aligned}\sup_{K\in\mathcal{K}_\delta(D)} \big(E(\infty_K) - \beta \operatorname{cap}_\nu(\infty_K)\big) &= \lim_{n\to+\infty} E(\infty_{K_n}) - \beta \operatorname{cap}_\nu(\infty_{K_n}) = E(\mu_{\lim}) - \beta \operatorname{cap}_\nu(\mu_{\lim}) \\ &= \max_{\mu\in\mathcal{M}_0^\delta(D)} \big(E(\mu) - \beta \operatorname{cap}_\nu(\mu)\big).\end{aligned}$$

□

A.2 Asymptotic expansion used to compute the topological gradient

Let x_0 be in $\mathbb{R}^2$ and $\varepsilon > 0$. In this section, we aim to find an estimate of

$$\int_{B(x_0,\varepsilon)} w \, dx,$$

where w the solution of the problem below :

Problem A.2.1. *Find w in $H_0^1\big(B(x_0,\varepsilon)\big)$ such that*

$$\begin{cases} w - \alpha\Delta w = g, & \text{in } B(x_0,\varepsilon), \\ w = 0, & \text{on } \partial B(x_0,\varepsilon). \end{cases} \tag{A.1}$$

Without loss of generality, we assume that g is Lipschitz continuous. Although the following result might be known, we did not find it in the literature, therefore, for the sake of completeness, we derive the estimate we need in Chapter 1. To solve Problem A.2.1, we use Green functions $G : B(x_0,\varepsilon) \times B(x_0,\varepsilon)$, corresponding to Problem A.2.1 which are solution to

Problem A.2.2. *Find $G(\cdot, y)$ in $L^2\big(B(x_0,\varepsilon)\big)$ such that*

$$\begin{cases} G(x,y) - \alpha\Delta_x G(x,y) = \delta_y(x), & x \in B(x_0,\varepsilon), \\ G(x,y) = 0, & x \in \partial B(x_0,\varepsilon), \end{cases} \tag{A.2}$$

for y in $B(x_0,\varepsilon)$.

We have easily the following property,

Proposition A.2.1. *Let G be Green functions corresponding to Problem A.2.2. Then, for x in $\overline{B(x_0,\varepsilon)}$,*

$$w(x) := \int_{B(x_0,\varepsilon)} g(y)G(x,y) \, dy,$$

is the solution of Problem A.2.1.

We shall determine explicitly the Green function G in the sequel. To do so, we write G as the sum of a particular solution G_{p} of Problem A.2.2 without the boundary condition, and the general solution G_0 of the homogeneous version of Problem A.2.2 such that $G_0 = -G_{\mathrm{p}}$ on $\partial B(x_0,\varepsilon)$. Below is the main proposition of this section,

Proposition A.2.2. *We have, for $\varepsilon \longrightarrow 0$,*

$$\int_{B(x_0,\varepsilon)} w(x) \, dx = -g(x_0)\pi\varepsilon^2 \ln(\varepsilon) + O(\varepsilon^2).$$

Proof. For ε small enough, we have, using Proposition A.2.1,

$$\int_{B(x_0,\varepsilon)} w(x) \, dx = \int_{B(x_0,\varepsilon)} \int_{B(x_0,\varepsilon)} g(y)G(x,y) \, dy \, dx.$$

Using Fubini, we get

$$\begin{aligned}\int_{B(x_0,\varepsilon)} w(x)\,dx &= \int_{B(x_0,\varepsilon)} g(y) \int_{B(x_0,\varepsilon)} G(x,y)\,dx\,dy \\ &= \int_{B(x_0,\varepsilon)} g(y) \int_{B(x_0,\varepsilon)} G_{\mathrm{p}}(x,y)\,dx\,dy + \int_{B(x_0,\varepsilon)} g(y) \int_{B(x_0,\varepsilon)} G_0(x,y)\,dx\,dy.\end{aligned}$$

Proposition A.2.4 and Proposition A.2.5 give us the result. □

It remains to state and to prove Proposition A.2.4 and Proposition A.2.5. We start by giving an explicit expression for G_{p} with the following proposition from [56].

Proposition A.2.3. *For x and y in $B(x_0,\varepsilon)$ such that $x \neq y$, we have*

$$G_p(x,y) = \frac{1}{2\pi} K_0\Big(\frac{1}{\sqrt{\alpha}}|x-y|\Big),$$

where K_0 is the modified Bessel function of the second kind, see [122].

Then, we compute the first part of Proposition A.2.2.

Proposition A.2.4. *When ε tends to 0, we have,*

$$\int_{B(x_0,\varepsilon)} g(y) \int_{B(x_0,\varepsilon)} G_p(x,y)\,dx\,dy = -g(x_0)\frac{\pi}{2}\varepsilon^2 \ln(\varepsilon) + O(\varepsilon^2).$$

Proof. We begin by setting

$$I_{\mathrm{p}} := \int_{B(x_0,\varepsilon)} g(y) \int_{B(x_0,\varepsilon)} G_{\mathrm{p}}(x,y)\,dx\,dy.$$

According to [122], we have the following asymptotic development

$$K_0(z) = -\ln z + \ln 2 - \gamma + O(z^2|\ln z|),$$

for $z \to 0$, where γ denotes the Euler–Mascheroni constant. Then,

$$G_{\mathrm{p}}(x,y) = -\frac{1}{2\pi}\Big(\ln|x-y| + \frac{1}{2}\ln\alpha + \ln 2 - \gamma\Big) + O(|x-y|^2|\ln|x-y||),$$

for $|x-y| \to 0$. Thus,

$$\begin{aligned}I_{\mathrm{p}} = -\frac{1}{2\pi}\int_{B(x_0,\varepsilon)} g(y) \int_{B(x_0,\varepsilon)} \ln|x-y|\,dx\,dy - \Big(\frac{1}{2}\ln\alpha + \ln 2 - \gamma\Big)\frac{\varepsilon^2}{2}\int_{B(x_0,\varepsilon)} g(y)\,dy \\ + O(1)\int_{B(x_0,\varepsilon)} g(y) \int_{B(x_0,\varepsilon)} |x-y|^2|\ln|x-y||\,dx\,dy.\end{aligned}$$

Using Taylor's formula, we have

$$\begin{aligned}I_{\mathrm{p}} = -\frac{1}{2\pi}g(x_0)\int_{B(x_0,\varepsilon)}\int_{B(x_0,\varepsilon)} \ln|x-y|\,dx\,dy + O(1)\int_{B(x_0,\varepsilon)} \|y-x_0\| \int_{B(x_0,\varepsilon)} \ln|x-y|\,dx\,dy \\ - \Big(\frac{1}{2}\ln\alpha + \ln 2 - \gamma\Big)\frac{\pi}{2}g(x_0)\varepsilon^4 + O(\varepsilon^4) + O(1)\int_{B(x_0,\varepsilon)}\int_{B(x_0,\varepsilon)} |x-y|^2|\ln|x-y||\,dx\,dy \\ + O(1)\int_{B(x_0,\varepsilon)} \|y-x_0\| \int_{B(x_0,\varepsilon)} |x-y|^2|\ln|x-y||\,dx\,dy.\end{aligned}$$

By setting $y = \varepsilon\tilde{y} + x_0$ and $x = \varepsilon\tilde{x} + x_0$, we have

$$\int_{B(x_0,\varepsilon)} \int_{B(x_0,\varepsilon)} \ln|x-y| \; dx \; dy = \varepsilon^2 \int_{B(0,1)} \int_{B(0,1)} \ln(\varepsilon|\tilde{x}-\tilde{y}|) \; d\tilde{x} \; d\tilde{y} = \pi^2\varepsilon^2 \ln\varepsilon + O(\varepsilon^2).$$

Using the same substitution for the remaining integrals, we got the result. □

And we finish by computing the second part of Proposition A.2.2.

Proposition A.2.5. *When ε tends to 0, we have,*

$$\int_{B(x_0,\varepsilon)} g(y) \int_{B(x_0,\varepsilon)} G_0(x,y) \; dx \; dy = O(\varepsilon^2).$$

Proof. We set

$$I_0 := \int_{B(x_0,\varepsilon)} g(y) \int_{B(x_0,\varepsilon)} G_0(x,y) \; dx \; dy.$$

Using Taylor's formula on g around x_0,

$$\begin{aligned} I_0 &= g(x_0) \int_{B(x_0,\varepsilon)} \int_{B(x_0,\varepsilon)} G_0(x,y) \; dx \; dy + O(1) \int_{B(x_0,\varepsilon)} \|y-x_0\| \int_{B(x_0,\varepsilon)} G_0(x,y) \; dx \; dy \\ &\leq g(x_0)\pi\varepsilon^2 \int_{B(x_0,\varepsilon)} \|G_0(\cdot,y)\|_{L^\infty\left(B(x_0,\varepsilon)\right)} \; dy + O(\varepsilon^2) \int_{B(x_0,\varepsilon)} \|y-x_0\| \|G_0(\cdot,y)\|_{L^\infty\left(B(x_0,\varepsilon)\right)} \; dy. \end{aligned}$$

Since G_0 satisfies the maximum principle [57],

$$\begin{aligned} I_0 &\leq g(x_0)\pi\varepsilon^2 \int_{B(x_0,\varepsilon)} \|G_0(\cdot,y)\|_{L^\infty\left(\partial B(x_0,\varepsilon)\right)} \; dy + O(\varepsilon^2) \int_{B(x_0,\varepsilon)} \|y-x_0\| \|G_0(\cdot,y)\|_{L^\infty\left(\partial B(x_0,\varepsilon)\right)} \; dy \\ &= g(x_0)\pi\varepsilon^2 \int_{B(x_0,\varepsilon)} \|G_{\mathrm{p}}(\cdot,y)\|_{L^\infty\left(\partial B(x_0,\varepsilon)\right)} \; dy + O(\varepsilon^2) \int_{B(x_0,\varepsilon)} \|y-x_0\| \|G_{\mathrm{p}}(\cdot,y)\|_{L^\infty\left(\partial B(x_0,\varepsilon)\right)} \; dy \\ &= g(x_0)\frac{1}{2}\varepsilon^2 \int_{B(x_0,\varepsilon)} \Big\| K_0\Big(\frac{1}{\sqrt{\alpha}}|\cdot - y|\Big)\Big\|_{L^\infty\left(\partial B(x_0,\varepsilon)\right)} \; dy \\ &\qquad + O(\varepsilon^2) \int_{B(x_0,\varepsilon)} \|y-x_0\| \Big\| K_0\Big(\frac{1}{\sqrt{\alpha}}|\cdot - y|\Big)\Big\|_{L^\infty\left(\partial B(x_0,\varepsilon)\right)} \; dy. \end{aligned}$$

According to [122], K_0 is an increasing function. Thus, for y in $B(x_0,\varepsilon)$,

$$\Big\| K_0\Big(\frac{1}{\sqrt{\alpha}}|\cdot - y|\Big)\Big\|_{L^\infty\left(\partial B(x_0,\varepsilon)\right)} := \sup_{x\in\partial B(x_0,\varepsilon)} \Big| K_0\Big(\frac{1}{\sqrt{\alpha}}|x-y|\Big)\Big|,$$

is attained where $|x-y| := \sqrt{r_x^2 + r_y^2 - 2r_x r_y \cos(\theta_x - \theta_y)}$ is maximal, i.e. when $\cos(\theta_x - \theta_y) = -1$, i.e. for $\theta_x = \pi + \theta_y$. In that case,

$$|x-y| = \sqrt{\varepsilon^2 + r_y^2 + 2\varepsilon r_y} = \varepsilon + r_y.$$

Thus,

$$\|G_{\mathrm{p}}(\cdot,y)\|_{L^\infty\left(\partial B(x_0,\varepsilon)\right)} = \frac{1}{2\pi} K_0\Big(\frac{1}{\sqrt{\alpha}}(\varepsilon + r_y)\Big).$$

Then, we have

$$\begin{aligned} I_0 &\leq g(x_0)\pi\varepsilon^2 \int_0^\varepsilon r_y K_0\Big(\frac{1}{\sqrt{\alpha}}(\varepsilon + r_y)\Big) \; dy + O(\varepsilon^2) \int_0^\varepsilon r_y^2 K_0\Big(\frac{1}{\sqrt{\alpha}}(\varepsilon + r_y)\Big) \; dy \\ &\leq g(x_0)\frac{\pi}{2}\varepsilon^4 K_0\Big(\frac{2\varepsilon}{\sqrt{\alpha}}\Big) + O(\varepsilon^5) K_0\Big(\frac{2\varepsilon}{\sqrt{\alpha}}\Big). \end{aligned}$$

Again, we use that, when z tends to 0,

$$K_0(z) = -\ln z + \ln 2 - \gamma + O(z^2|\ln z|),$$

and get, since ε tends to 0,

$$K_0\Big(\frac{2\varepsilon}{\sqrt{\alpha}}\Big) = -\ln\varepsilon + \frac{1}{2}\ln\alpha - \gamma + O(\varepsilon^2|\ln\varepsilon|).$$

Therefore,

$$\begin{aligned} I_0 &\leq -g(x_0)\frac{\pi}{2}\varepsilon^4\ln\varepsilon + g(x_0)\frac{\pi}{4}\varepsilon^4\ln\alpha - g(x_0)\frac{\pi}{2}\varepsilon^4\gamma + O(\varepsilon^5\ln\varepsilon) \\ &= O(\varepsilon^2). \end{aligned}$$

□

A.3 Bounds for the density θ in the "Fat Pixels" Approach

We aim to give some estimate of the function θ defined in Theorem 1.3.2 (Chapter 1). In the sequel, we denote by v_K the solution of Problem 1.0.4, $g := \alpha\Delta f$ and $t_1 := \sqrt{2}/2$. We will widely use the following maximum principle (See [57] Theorem 2 in Section 6.4).

Theorem A.3.1 (Weak maximum principle)**.** *We assume that v_K is in $C^2(D) \cap C^0(\bar{D})$. If $g \geq 0$ in $D \setminus K$, then $v_K \geq 0$ in D.*

Moreover, using the weak formulation, Poincaré inequality and Hölder inequality, we have

Lemma A.3.1. *We have*

$$\|v_K\|_{L^2(D)} \leq \big(1 + \alpha C(D)\big)^{-1} \|g\|_{L^2(D)}$$

and

$$\|v_K\|_{L^1(D)} \leq |D|^{1/2} (1 + \alpha C(D))^{-1} \|g\|_{L^2(D)}.$$

Note. If $D \subset \mathbb{R}^2$ is convex we have, according to [127],

$$\|v_K\|_{L^2(D)} \leq \big(1 + \alpha\pi^2 \operatorname{diam}(D)^{-2}\big)^{-1} \|g\|_{L^2(D)}.$$

$$\|v_K\|_{L^1(D)} \leq |D|^{1/2} \big(1 + \alpha\pi^2 \operatorname{diam}(D)^{-2}\big)^{-1} \|g\|_{L^2(D)}.$$

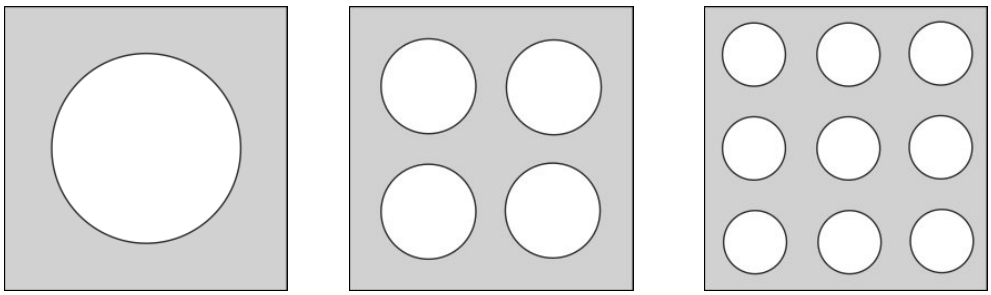
Moreover, if $g = 1$, we have $\|v_K\|_{L^1(D)} \leq \big(1 + \alpha\pi^2 \operatorname{diam}(D)^{-2}\big)^{-1} |D|$.

Lemma A.3.2. *We have, for m in $(0, t_1)$,*

$$\theta(m) \leq C_1(\alpha) \ln m^{-1} + C_2(\alpha),$$

where C_1 and C_2 are constants depending on α.

Proof. We consider a particular family of sets K_n in $\mathcal{A}_{m,n}$. We choose an integer k such that $n = k^2$ and we suppose $K_n \in \mathcal{A}_{m,n}$ are composed of n balls of radius m/k, with their centers superposing the centers of the k^2 squares of side $1/k$ of a regular lattice partitioning the square I^2.



(a) $I^2 \setminus K_1$ (b) $I^2 \setminus K_4$ (c) $I^2 \setminus K_9$

Figure A.1: Drawing of $I^2 \setminus K_n$ with $n := k^2$, for $k = 1, 2, 3$.

Let us denote $v^1_{K_n}$ the solution of Problem 1.0.4 with $g = 1$, $K = K_n$ and $D = I^2$. It holds $\int_I v^1_{K_1}\, dx = n \int_I v^1_{K_n}\, dx$. We recall

$$\theta(m) := \inf_{K_n \in \mathcal{A}_{m,n}} \liminf_n n \int_D g v_{K_n}\, dx.$$

In particular when $g = 1$

$$\theta(m) \leq \liminf_n n \int_D v^1_{K_n}\ dx = \int_I v^1_{K_1}\ dx.$$

We have $I^2 \subset B(x_0, t_1)$. Therefore if we denote by w the solution of Problem 1.0.4 with $g = 1$, $D = B(x_0, t_1)$ and $K = K_1 := B(x_0, m)$, it holds by the maximum principle that $v^1_{K_1} \leq w$. Then we have the following estimate

$$\theta(m) \leq \int_{B(x_0,t_1)} w\ dx.$$

Let us consider the following problem

$$\begin{cases} -\alpha\Delta\tilde{w} = 1 & \text{in } B(x_0, t_1) \setminus \overline{B(x_0, m)}, \\ \tilde{w} = 0 & \text{in } \overline{B(x_0, m)}, \\ \frac{\partial\tilde{w}}{\partial\mathbf{n}} = 0 & \text{on } \partial B(x_0, t_1). \end{cases} \tag{A.3}$$

Now, we set $e := w - \tilde{w}$. Thus we have for all v in $H^1_0(B(x_0, t_1) \setminus \overline{B(x_0, m)})$

$$\alpha \int_B \nabla e \cdot \nabla v\ dx = \alpha \int_B \nabla w \cdot \nabla v\ dx - \alpha \int_B \nabla\tilde{w} \cdot \nabla v\ dx = -\int_B wv\ dx,$$

i.e. e satisfies the problem below

$$\begin{cases} -\alpha\Delta e = -w & \text{in } B(x_0, t_1) \setminus \overline{B(x_0, m)}, \\ e = 0 & \text{in } \overline{B(x_0, m)}, \\ \frac{\partial e}{\partial\mathbf{n}} = 0 & \text{on } \partial B(x_0, t_1). \end{cases} \tag{A.4}$$

Since $w \geq 0$, we have by the maximum principle that $e \leq 0$ i.e. $0 \leq w \leq \tilde{w}$. By consequence

$$\theta(m) \leq \int_{B(x_0,t_1)} \tilde{w}\ dx.$$

The solution $\tilde{w}$ have been computed in [33]. Due to the radial symmetry of $\tilde{w}$, we can get explicitly $\tilde{w}$, solution of :

$$\begin{cases} \tilde{w}''(r) + \frac{1}{r}\tilde{w}'(r) = -\frac{1}{\alpha} & \text{if } m < r < t_1, \\ \tilde{w} = 0 & \text{if } 0 \leq r \leq m, \\ \tilde{w}'(t_1) = 0 & . \end{cases} \tag{A.5}$$

For $r = |x - x_0|$ we have

$$\tilde{w}(x) = \begin{cases} k\ln(\frac{r}{m}) - \frac{1}{4\alpha}(r^2 - m^2) & \text{if } m < r < t_1 \\ 0 & \text{if } 0 \leq r \leq m \end{cases},\ k = \frac{mt_1^2}{2\alpha}.$$

Integrating $\tilde{w}$ over $B(x_0, t_1)$

$$\begin{aligned} \int_{B(x_0,t_1)} \tilde{w}\ dx &= 2\pi \int_m^{t_1} \Big(k\ln(\frac{r}{m}) - \frac{1}{4\alpha}(r^2 - m^2)\Big) r\ dr \\ &= 2\pi k \int_m^{t_1} r\ln(\frac{r}{m})\ dr - \frac{\pi}{2\alpha}\int_m^{t_1} r^3\ dr + \frac{\pi}{2\alpha}m^2\int_m^{t_1} r\ dr \\ &= \pi k t_1^2 \ln\frac{t_1}{m} + \frac{\pi}{2}\underbrace{(\frac{m^2}{2\alpha} - \frac{k}{m})}_{\leq 0}(t_1^2 - m^2) + \frac{\pi}{8\alpha}\underbrace{(m^4 - t_1^4)}_{\leq 0} \\ &\leq \frac{\pi t_1^4}{2\alpha} m\ln\frac{t_1}{m} \leq \frac{\pi t_1^4}{2\alpha}\ln\frac{t_1}{m} = \underbrace{\frac{\pi t_1^4}{2\alpha}}_{=:\ C_1(\alpha)}\ \ln m^{-1} + \underbrace{\frac{\pi t_1^4}{2\alpha}\ln t_1}_{=:\ C_2(\alpha)}. \end{aligned}$$

□

Lemma A.3.3. *We have, for m in $(0, t_1)$,*

$$C_1(\alpha)\ln(m^{-1}) - C_2(\alpha) \leq \theta(m),$$

where C_1 and C_2 are constants depending on α.

Proof. We fix $n \in \mathbb{N}$ and $(x_i)_{i=1,\dots,n} \in I^2$. We consider the sets $K_m := \bigcup_{i=1}^n \overline{B(x_i, mn^{-1/2})}$ and $U_m := I^2 \setminus K_m$. Let us denote u_m^1 the solution of Problem 1.0.4 with $g = 1$, $K = K_m$ and $D = I^2$. Using Holder inequality we have

$$\Big(\int_{\partial U_m} \frac{\partial u_m^1}{\partial n} d\mathcal{H}^1\Big)^2 \leq \int_{\partial U_m} \Big|\frac{\partial u_m^1}{\partial n}\Big|^2 d\mathcal{H}^1 \times \int_{\partial U_m} d\mathcal{H}^1 = \mathcal{H}^1(\partial U_m) \int_{\partial U_m} \Big|\frac{\partial u_m^1}{\partial n}\Big|^2 d\mathcal{H}^1.$$

Moreover, we have thanks to the Green formula

$$\int_{\partial U_m} \frac{\partial u_m^1}{\partial n} d\mathcal{H}^1 = \int_{U_m} \Delta u_m^1 \, dx = \frac{1}{\alpha}\int_{U_m} (u_m^1 - 1)\, dx = \frac{1}{\alpha}\|u_m^1\|_{L^1(U_m)} - \frac{1}{\alpha}|U_m|.$$

Thus

$$\begin{aligned}
&\Big(\frac{1}{\alpha}\|u_m^1\|_{L^1(U_m)} - \frac{1}{\alpha}|U_m|\Big)^2 \leq \mathcal{H}^1(\partial U_m) \int_{\partial U_m} \Big|\frac{\partial u_m^1}{\partial n}\Big|^2 d\mathcal{H}^1 \\
\Leftrightarrow &\frac{1}{\alpha^2}|U_m|^2 - \frac{2}{\alpha^2}\|u_m^1\|_{L^1(U_m)}|U_m| \leq \mathcal{H}^1(\partial U_m) \int_{\partial U_m} \Big|\frac{\partial u_m^1}{\partial n}\Big|^2 d\mathcal{H}^1.
\end{aligned}$$

The fact that $|U_m| \geq 1 - 2\pi m^2$ and Property A.3.1 give us

$$\frac{2\pi^2\alpha - 1}{\alpha^2(1 + 2\pi^2\alpha)} - \frac{2\pi}{\alpha^2} m \leq \mathcal{H}^1(\partial U_m) \int_{\partial U_m} \Big|\frac{\partial u_m^1}{\partial n}\Big|^2 d\mathcal{H}^1.$$

Also, it holds $\mathcal{H}^1(\partial U_m) \leq 2\pi m\sqrt{n}$. Then

$$\begin{aligned}
&\frac{2\pi^2\alpha - 1}{\alpha^2(1 + 2\pi^2\alpha)} - \frac{2\pi}{\alpha^2} m \leq 2\pi m\sqrt{n} \int_{\partial U_m} \Big|\frac{\partial u_m^1}{\partial n}\Big|^2 d\mathcal{H}^1 \\
\Leftrightarrow &\frac{2\pi^2\alpha - 1}{2\pi\alpha^2(1 + 2\pi^2\alpha)} \frac{1}{m} - \frac{1}{\alpha^2} \leq \sqrt{n} \int_{\partial U_m} \Big|\frac{\partial u_m^1}{\partial n}\Big|^2 d\mathcal{H}^1.
\end{aligned}$$

Using that $-\frac{dF}{dm} = \sqrt{n}^{-1} \int_{\partial U_m} \Big|\frac{\partial u_m^1}{\partial n}\Big|^2 d\mathcal{H}^1$, we have

$$\frac{2\pi^2\alpha - 1}{2\pi\alpha^2(1 + 2\pi^2\alpha)} \frac{1}{m} - \frac{1}{\alpha^2} \leq -n\frac{dF}{dm}.$$

Integrating over $[m_1, m_2] \subset (0, t_1)$ yields to

$$\frac{2\pi^2\alpha - 1}{2\pi\alpha^2(1 + 2\pi^2\alpha)} \ln\Big(\frac{m_2}{m_1}\Big) - \frac{m_2 - m_1}{\alpha^2} + nF_{m_2} \leq nF_{m_1}.$$

Taking inf over x_i and passing to lim inf over n when n tends to $+\infty$ leads to

$$\frac{2\pi^2\alpha - 1}{2\pi\alpha^2(1 + 2\pi^2\alpha)} \ln\Big(\frac{m_2}{m_1}\Big) - \frac{m_2 - m_1}{\alpha^2} + \theta(m_2) \leq \theta(m_1).$$

In particular, if $m_2 = t_1$ and $m_1 = m$, $0 < m < t_1 = \frac{\sqrt{2}}{2}$,

$$\begin{aligned}
&\frac{2\pi^2\alpha - 1}{2\pi\alpha^2(1 + 2\pi^2\alpha)} \ln\Big(\frac{t_1}{m}\Big) - \frac{t_1 - m}{\alpha^2} \leq \theta(m) \\
\Leftrightarrow &\frac{2\pi^2\alpha - 1}{2\pi\alpha^2(1 + 2\pi^2\alpha)} \ln\Big(\frac{t_1}{m}\Big) - \frac{t_1}{\alpha^2} \leq \theta(m) \\
\Leftrightarrow &\underbrace{\frac{2\pi^2\alpha - 1}{2\pi\alpha^2(1 + 2\pi^2\alpha)}}_{=:\, C_1(\alpha)} \ln(m^{-1}) - \underbrace{\Big(\frac{2\pi^2\alpha - 1}{2\pi\alpha^2(1 + 2\pi^2\alpha)} \ln(t_1^{-1}) + \frac{t_1}{\alpha^2}\Big)}_{=:\, C_2(\alpha)} \leq \theta(m).
\end{aligned}$$

□

Appendix B

Some Proofs for the Adjoint Method

B.1 Recall on Some Properties of the Sobolev Norms

In this section, we recall the definition of the norms and some of their properties used in the proofs. Let $\mathcal{O}$ be an open subset of $\mathbb{R}^2$. For v in $H^1(\mathcal{O})$, we recall the following Sobolev norms :

$$\begin{aligned}
\|v\|_{0,\mathcal{O}} &:= \left(\int_{\mathcal{O}} v^2 \, dx\right)^{1/2}, \\
|v|_{1,\mathcal{O}} &:= \left(\int_{\mathcal{O}} |\nabla v|^2 \, dx\right)^{1/2}, \\
\|v\|_{1,\mathcal{O}} &:= \left(\int_{\mathcal{O}} v^2 \, dx + \int_{\mathcal{O}} |\nabla v|^2 \, dx\right)^{1/2}, \\
\|v\|_{1,\alpha,\mathcal{O}} &:= \left(\int_{\mathcal{O}} v^2 \, dx + \alpha \int_{\mathcal{O}} |\nabla v|^2 \, dx\right)^{1/2}.
\end{aligned}$$

Now we set

$$\mathcal{O}_\varepsilon := \{x/\varepsilon \mid x \in \mathcal{O}\}.$$

Straightforward computations give :

Proposition B.1.1.

$$\begin{aligned}
\|u(\cdot/\varepsilon)\|_{0,\mathcal{O}} &= \varepsilon \|u\|_{0,\mathcal{O}_\varepsilon}, \\
|u(\cdot/\varepsilon)|_{1,\mathcal{O}} &= |u|_{1,\mathcal{O}_\varepsilon}.
\end{aligned}$$

On the boundary of a ball of radius $r > 0$, namely ∂B_r, we define for ϕ in $H^{1/2}(\partial B_r)$,

$$\|\phi\|_{1/2,\partial B_r} := \inf_{u=\phi \text{ on } \Gamma} \|u\|_{1,B_r \setminus B_{r/2}},$$

and for ψ in $H^{-1/2}(\partial B_r)$, we define the dual norm,

$$\|\psi\|_{-1/2,\partial B_r} := \sup_{v \in H^{1/2}(\partial B_r),\, \|v\|_{1/2,\partial B_r}=1} \int_{\partial B_r} \psi \, v \, d\sigma.$$

Then, it is standard that

Proposition B.1.2. *Let $\psi \in H^1(B_R \setminus B_{R/2})$, with $-\alpha\Delta\psi + \psi = 0$ in $B_R \setminus B_{R/2}$, there exists $C > 0$, such that*

$$\|\nabla\psi \cdot n\|_{-1/2,\partial B_R} \leq C \, \|\psi\|_{1,B_R \setminus B_{R/2}}.$$

B.2 Additional Proofs

Here, we give some proofs of results stated in Chapter 3.

Proof of Proposition 3.2.1. We set

$$v := \begin{cases} v_\varepsilon, & \text{in } D \setminus B_R, \\ v_{\varepsilon,R}, & \text{in } B_R \setminus B_\varepsilon. \end{cases}$$

Let φ be in V_ε, then,

$$\begin{aligned}
\tilde{a}_\varepsilon(v,\varphi) &:= \alpha \int_{D\setminus B_\varepsilon} \nabla v \cdot \nabla \varphi \, dx + \int_{D\setminus B_\varepsilon} v\,\varphi \, dx \\
&= \alpha \int_{D\setminus B_R} \nabla v \cdot \nabla \varphi \, dx + \int_{D\setminus B_R} v\,\varphi \, dx + \alpha \int_{B_R\setminus B_\varepsilon} \nabla v \cdot \nabla \varphi \, dx + \int_{B_R\setminus B_\varepsilon} v\,\varphi \, dx \\
&= \alpha \int_{D\setminus B_R} \nabla v_\varepsilon \cdot \nabla \varphi \, dx + \int_{D\setminus B_R} v_\varepsilon\,\varphi \, dx + \alpha \int_{B_R\setminus B_\varepsilon} \nabla v_{\varepsilon,R} \cdot \nabla \varphi \, dx + \int_{B_R\setminus B_\varepsilon} v_{\varepsilon,R}\,\varphi \, dx \\
&= -\alpha \int_{D\setminus B_R} \Delta v_\varepsilon\,\varphi \, dx + \alpha \underbrace{\int_{\partial D} \partial_{n_{\text{ext}}} v_\varepsilon\,\varphi \, d\sigma}_{=0} + \alpha \int_{\partial B_R} \partial_{n_{\text{int}}} v_\varepsilon\,\varphi \, d\sigma + \int_{D\setminus B_R} v_\varepsilon\,\varphi \, dx \\
&\qquad - \alpha \int_{B_R\setminus B_\varepsilon} \Delta v_{\varepsilon,R}\,\varphi \, dx + \alpha \int_{\partial B_R} \partial_{n_{\text{ext}}} v_{\varepsilon,R}\,\varphi \, d\sigma + \alpha \underbrace{\int_{\partial B_\varepsilon} \partial_{n_{\text{int}}} v_{\varepsilon,R}\,\varphi \, d\sigma}_{=0 \text{ since } \varphi\in V_\varepsilon} \\
&\qquad + \int_{B_R\setminus B_\varepsilon} v_{\varepsilon,R}\,\varphi \, dx \\
&= \int_{D\setminus B_R} (-\alpha\Delta v_\varepsilon + v_\varepsilon)\,\varphi \, dx + \int_{B_R\setminus B_\varepsilon} (-\alpha\Delta v_{\varepsilon,R} + v_{\varepsilon,R})\,\varphi \, dx \\
&\qquad + \alpha \int_{\partial B_R} \partial_{n_{\text{int}}} v_\varepsilon\,\varphi \, d\sigma + \alpha \int_{\partial B_R} \partial_{n_{\text{ext}}} v_{\varepsilon,R}\,\varphi \, d\sigma \\
&= \int_{D\setminus B_\varepsilon} h\,\varphi \, dx + \alpha \int_{\partial B_R} \partial_{n_{\text{int}}} v_\varepsilon\,\varphi \, d\sigma + \alpha \int_{\partial B_R} \partial_{n_{\text{ext}}} v_{\varepsilon,R}\,\varphi \, d\sigma \\
&= \int_{D\setminus B_\varepsilon} h\,\varphi \, dx - \alpha \int_{\partial B_R} \partial_{n_{\text{ext}}} v_{\varepsilon,R}\,\varphi \, d\sigma + \alpha \int_{\partial B_R} \partial_{n_{\text{ext}}} v_{\varepsilon,R}\,\varphi \, d\sigma \\
&= \int_{D\setminus B_\varepsilon} h\,\varphi \, dx = \tilde{l}_0(\varphi).
\end{aligned}$$

By the uniqueness of the solution, we have $v = \widetilde{v}_\varepsilon$. □

Proof of Proposition 3.2.2. • It is sufficient to prove that $\alpha \int_{\partial B_R} T_\varepsilon v\,\varphi \, d\sigma$ is symmetric :

$$\begin{aligned}
\alpha \int_{\partial B_R} T_\varepsilon v\,\varphi \, d\sigma &= \alpha \int_{\partial B_R} \partial_n v_\varepsilon^{0,v}\,\varphi \, d\sigma \\
&= \alpha \int_{\partial B_R} \partial_n v_\varepsilon^{0,v}\, v_\varepsilon^{0,\varphi} \, d\sigma \\
&= \alpha \int_{B_R} \Delta v_\varepsilon^{0,v}\, v_\varepsilon^{0,\varphi} \, dx + \alpha \int_{B_R} \nabla v_\varepsilon^{0,v} \cdot \nabla v_\varepsilon^{0,\varphi} \, dx \\
&= \int_{B_R} v_\varepsilon^{0,v}\, v_\varepsilon^{0,\varphi} \, dx + \alpha \int_{B_R} \nabla v_\varepsilon^{0,v} \cdot \nabla v_\varepsilon^{0,\varphi} \, dx.
\end{aligned}$$

- Let ϕ be in $H^{1/2}(\partial B_R)$,

$$\begin{aligned}
\alpha \int_{\partial B_R} h_\varepsilon\, \phi \, d\sigma &= -\alpha \int_{\partial B_R} \partial_n v_\varepsilon^{h,0}\, v_\varepsilon^{0,\phi} \, d\sigma \\
&= -\alpha \int_{B_R} \Delta v_\varepsilon^{h,0}\, v_\varepsilon^{0,\phi} \, dx - \alpha \int_{B_R} \nabla v_\varepsilon^{h,0} \cdot \nabla v_\varepsilon^{0,\phi} \, dx \\
&= \int_{B_R} h\, v_\varepsilon^{0,\phi} \, dx - \int_{B_R} v_\varepsilon^{h,0}\, v_\varepsilon^{0,\phi} \, dx - \alpha \int_{B_R} \nabla v_\varepsilon^{h,0} \cdot \nabla v_\varepsilon^{0,\phi} \, dx \\
&= \int_{B_R} h\, v_\varepsilon^{0,\phi} \, dx - \int_{B_R} v_\varepsilon^{h,0}\, v_\varepsilon^{0,\phi} \, dx + \alpha \int_{B_R} v_\varepsilon^{h,0} \Delta v_\varepsilon^{0,\phi} \, dx - \alpha \int_{\partial B_R} \partial_n v_\varepsilon^{0,\phi}\, v_\varepsilon^{h,0} \, d\sigma \\
&= \int_{B_R} h\, v_\varepsilon^{0,\phi} \, dx + \alpha \int_{B_R} \underbrace{(\alpha \Delta v_\varepsilon^{0,\phi} - v_\varepsilon^{0,\phi})}_{=0}\, v_\varepsilon^{h,0} \, dx - \alpha \int_{\partial B_R} \partial_n v_\varepsilon^{0,\phi}\, \underbrace{v_\varepsilon^{h,0}}_{=0} \, d\sigma \\
&= \int_{B_R} h\, v_\varepsilon^{0,\phi} \, dx.
\end{aligned}$$

□

Proof of Proposition 3.3.1. We set $w_R := \widetilde{w}_0|_{D\setminus B_R}$. We have to show that $w_R = w_0$ i.e. $a_0(w_R, \varphi_R) = -DJ_0(v_0)\varphi_R$, $\forall \varphi_R \in V_R$. Let $\varphi_R \in V_R$. We denote $\widetilde{\varphi} \in V_0$ the extension of φ_R to V_0 such that $-\alpha\Delta\widetilde{\varphi} + \widetilde{\varphi} = 0$ in B_R. Thus,

$$\begin{aligned}
a_0(w_R, \varphi_R) &= \alpha \int_{D\setminus B_R} \nabla w_R \cdot \nabla \varphi_R \, dx + \alpha \int_{\partial B_R} T_0 w_R\, \varphi_R \, d\sigma + \int_{D\setminus B_R} w_R\, \varphi_R \, dx \\
&= \alpha \int_{D\setminus B_R} \nabla w_R \cdot \nabla \varphi_R \, dx + \alpha \int_{\partial B_R} T_0 w_R\, \varphi_R \, d\sigma + \int_{D\setminus B_R} w_R\, \varphi_R \, dx \\
&\qquad + \int_{B_R} \underbrace{(-\alpha\Delta\widetilde{\varphi} + \widetilde{\varphi})}_{=0}\, \widetilde{w}_0 \, dx \\
&= \alpha \int_{D\setminus B_R} \nabla w_R \cdot \nabla \varphi_R \, dx + \alpha \int_{\partial B_R} T_0 w_R\, \varphi_R \, d\sigma + \int_{D\setminus B_R} w_R\, \varphi_R \, dx - \alpha \int_{B_R} \Delta\widetilde{\varphi}\widetilde{w}_0 \, dx \\
&\qquad + \int_{B_R} \widetilde{\varphi}\, \widetilde{w}_0 \, dx \\
&= \alpha \int_{D\setminus B_R} \nabla w_R \cdot \nabla \varphi_R \, dx + \alpha \int_{\partial B_R} T_0 \varphi_R\, w_R \, d\sigma + \int_{D\setminus B_R} w_R\, \varphi \, dx + \alpha \int_{B_R} \nabla\widetilde{\varphi} \cdot \nabla\widetilde{w}_0 \, dx \\
&\qquad - \alpha \int_{\partial B_R} \partial_n\widetilde{\varphi}\, \widetilde{w}_0 \, d\sigma + \int_{B_R} \widetilde{\varphi}\, \widetilde{w}_0 \, dx \\
&= \alpha \int_{D\setminus B_R} \nabla w_R \cdot \nabla \varphi_R \, dx + \alpha \int_{\partial B_R} T_0 \varphi_R\, w_R \, d\sigma + \int_{D\setminus B_R} w_R \varphi_R \, dx + \alpha \int_{B_R} \nabla\widetilde{\varphi} \cdot \nabla\widetilde{w}_0 \, dx \\
&\qquad - \alpha \int_{\partial B_R} T_0\varphi_R\, w_R \, d\sigma + \int_{B_R} \widetilde{\varphi}\, \widetilde{w}_0 \, dx \\
&= \alpha \int_{D\setminus B_R} \nabla w_R \cdot \nabla \varphi_R \, dx + \int_{D\setminus B_R} w_R\varphi_R \, dx + \alpha \int_{B_R} \nabla\widetilde{\varphi} \cdot \nabla\widetilde{w}_0 \, dx + \int_{B_R} \widetilde{\varphi}\, \widetilde{w}_0 \, dx \\
&= \alpha \int_{D} \nabla\widetilde{w}_0 \cdot \nabla\widetilde{\varphi} \, dx + \int_{D} \widetilde{w}_0\, \widetilde{\varphi} \, dx \\
&= \widetilde{a}_0(\widetilde{w}_0, \widetilde{\varphi}) = -D\widetilde{J}_0(v_D)\widetilde{\varphi}.
\end{aligned}$$

Moreover, by definition $\widetilde{J}_0(v_D) = J_0(v_R)$, thus,

$$D\widetilde{J}_0(v_D)\widetilde{\varphi} = DJ_0(v_0)\varphi_R.$$

By uniqueness of the solution, $w_R = w_0$. □

B.3 Analysis of the Exterior Problem

Now, we give estimates of the solution to the *exterior* problem with the norms defined in Section B.1. For ϕ in $H^{1/2}(\partial B_1)$, we define the exterior problem as the following :

$$\begin{cases} -\alpha\Delta v_\omega + v_\omega = 0, & \text{in } \mathbb{R}^2 \setminus B_1, \\ v_\omega = \phi, & \text{on } \partial B_1, \\ v_\omega = 0, & \text{at } \infty. \end{cases}$$

Then,

Proposition B.3.1. *For y in $\mathbb{R}^2 \setminus \overline{B_1}$, we have*

$$v_\omega(y) = \int_{\partial B_1} E(y-x)p(x)\ d\sigma(x),$$

where E is the fundamental solution (radial) in $\mathbb{R}^2 \setminus \{0\}$ given by

$$E(y) := \frac{1}{2\pi} K_0\left(\frac{1}{\sqrt{\alpha}}|y|\right),$$

where K_0 is the modified Bessel function of the second kind [122] and p is the solution in $H^{-1/2}(\partial B_1)$ of

$$\int_{\partial B_1} E(y-x)p(x)\ d\sigma(x) = \phi(y),\ \forall y \in \partial B_1.$$

Proof. We differentiate and we use [122] : $K_0'(z) = -K_1(z)$ and $K_1'(z) = -K_0(z) - \frac{1}{z}K_1(z)$. □

Proposition B.3.2. *For $|y|$ large enough, it exists C_1 and C_2, only dependant on α, such that,*

$$|v_\omega(y)| \le C_1\,|y|^{-1/2} e^{-|y|/\sqrt{\alpha}}\,\|\phi\|_{1/2,\partial B_1},$$
$$|\nabla v_\omega(y)| \le C_2\,|y|^{-1/2} e^{-|y|/\sqrt{\alpha}}\,\|\phi\|_{1/2,\partial B_1}.$$

Proof. Since K_0 is a positive and decreasing function, we have for $(x,z) \in (\partial B_1)^2$,

$$E(x-z) \ge \frac{1}{2\pi}K_0(2\alpha^{-1/2}) \Leftrightarrow \frac{2\pi}{K_0(2\alpha^{-1/2})}E(x-z) \ge 1.$$

Then, for $z \in \partial B_1$,

$$\left|\int_{\partial B_1} p(x)d\sigma(x)\right| \le \frac{2\pi}{K_0(2\alpha^{-1/2})}\left|\int_{\partial B_1} E(x-z)p(x)\ d\sigma(x)\right| = \frac{2\pi}{K_0(2\alpha^{-1/2})}|\phi(z)|.$$

We integrate on ∂B_1 with respect to z the square of the previous inequality and get,

$$\left|\int_{\partial B_1} p(x)d\sigma(x)\right|^2 \le \frac{2\pi}{K_0(2\alpha^{-1/2})^2}\int_{\partial B_1} |\phi(z)|^2\ d\sigma(z).$$

Let $u = \phi$ on ∂B_1. Then,

$$\left|\int_{\partial B_1} p(x)d\sigma(x)\right|^2 \le \frac{2\pi}{K_0(2\alpha^{-1/2})^2}\int_{B_1\setminus B_{1/2}} |u(z)|^2\ dz,$$

this been true for every u, we take the sup,

$$\left|\int_{\partial B_1} p(x)d\sigma(x)\right|^2 \le \frac{2\pi}{K_0(2\alpha^{-1/2})^2}\|\phi\|^2_{1/2,\partial B_1}.$$

With [122], we have for z big enough,

$$K_0(z) = O(z^{-1/2}e^{-z}),$$

thus, there exists $C > 0$, such that, for $|y|$ large enough,

$$|E(x-y)| \leq C\,|x-y|^{-1/2}e^{-|x-y|/\sqrt{\alpha}}.$$

Moreover, for $|y|$ large enough,

$$|x-y|^{-1/2}e^{-|x-y|/\sqrt{\alpha}} \leq C|y|^{-1/2}e^{-|y|/\sqrt{\alpha}},$$

then,

$$|E(x-y)| \leq C'\,|y|^{-1/2}e^{-|y|/\sqrt{\alpha}}.$$

Therefore

$$|v_\omega(y)| \leq C_1\,|y|^{-1/2}e^{-|y|/\sqrt{\alpha}}\,\|\phi\|_{1/2,\partial B_1}.$$

Next,

$$\partial_{y_i} v_\omega(y) = \partial_{y_i}\int_{\partial B_1} E(y-x)p(x)\;d\sigma(x) = \int_{\partial B_1}\partial_{y_i}E(y-x)p(x)\;d\sigma(x),$$

and

$$\partial_{y_i}E(x-y) = \frac{1}{2\pi}\partial_{y_i}K_0(\alpha^{-1/2}|x-y|) = \frac{-1}{2\pi}\alpha^{-1/2}\,|y-x|^{-1}K_1(\alpha^{-1/2}|y-x|)\,y_i.$$

Thus

$$|\nabla v_\omega(y)| = \frac{1}{2\pi}\alpha^{-1/2}\,|y|\,\left|\int_{\partial B_1}|y-x|^{-1}K_1(\alpha^{-1/2}|y-x|)\,p(x)\;d\sigma(x)\right|,$$

in the same way that before, for $|y|$ big enough, [122],

$$|\nabla v_\omega(y)| \leq C\,|y|^{-1/2}e^{-|y|/\sqrt{\alpha}}\,\|\phi\|_{1/2,\partial B_1}.$$

□

From the above results, we derive easily :

Proposition B.3.3. *There exists C_1, C_2, C_3, C_4, C_5 and C_6 only dependant on α such that, for ε small enough,*

$$\begin{aligned}
\|v_\omega\|_{0,B_{R/\varepsilon}\setminus B_1} &\leq C_1\,\|\phi\|_{1/2,\partial B_1},\\
|v_\omega|_{1,B_{R/\varepsilon}\setminus B_1} &\leq C_2\,\|\phi\|_{1/2,\partial B_1},\\
\|v_\omega\|_{1,B_{R/\varepsilon}\setminus B_1} &\leq C_3\,\|\phi\|_{1/2,\partial B_1},\\
\|v_\omega\|_{0,B_{R/\varepsilon}\setminus B_{R/(2\varepsilon)}} &\leq C_4\,e^{-R/(2\varepsilon\sqrt{\alpha})}\,\|\phi\|_{1/2,\partial B_1},\\
|v_\omega|_{1,B_{R/\varepsilon}\setminus B_{R/(2\varepsilon)}} &\leq C_5\,e^{-R/(2\varepsilon\sqrt{\alpha})}\,\|\phi\|_{1/2,\partial B_1},\\
\|v_\omega\|_{1,B_{R/\varepsilon}\setminus B_{R/(2\varepsilon)}} &\leq C_6\,e^{-R/(2\varepsilon\sqrt{\alpha})}\,\|\phi\|_{1/2,\partial B_1}.
\end{aligned}$$

B.4 Some Estimates for the Various Elliptic Problems in the Previous Sections

In this section, we give the estimates of the solution of the problem below with the norms defined in Section B.1.

Proposition B.4.1. *Let $\phi \in H^{1/2}(\partial B_R)$. Let v_ε be the solution of the following problem :*

$$\begin{cases} -\alpha\Delta v_\varepsilon + v_\varepsilon = 0, & \text{in } B_R \setminus B_\varepsilon, \\ v_\varepsilon = 0, & \text{on } \partial B_\varepsilon, \\ v_\varepsilon = \phi, & \text{on } \partial B_R. \end{cases}$$

Then, it exists $0 < \varepsilon_0 < R$ and $C > 0$ such that, for all $0 < \varepsilon < \varepsilon_0$, we have

$$\|v_\varepsilon\|_{1,B_R\setminus B_\varepsilon} \leq C\|\phi\|_{1/2,\partial B_R}.$$

Proof. Let $R/2 < \varepsilon_0 < R$, them it is readily checked that :

$$\|v_{\varepsilon_0}\|_{1,\alpha,B_R\setminus B_{\varepsilon_0}} \leq C\|v\|_{1,B_R\setminus B_{R/2}}.$$

Next, we take $\varepsilon < \varepsilon_0$. Then, $D_{\varepsilon_0} \subset D_\varepsilon$ and we denote by $\widetilde{v}_{\varepsilon_0}$ the extension by 0 of v_{ε_0} to D_ε. As v_ε is solution of the problem, it follows :

$$\|v_\varepsilon\|_{1,B_R\setminus B_\varepsilon} \leq C\|v_\varepsilon\|_{1,\alpha,B_R\setminus B_\varepsilon}.$$

□

Proposition B.4.2. *Let $\psi \in H^{1/2}(\partial B_\varepsilon)$. Let v_ε be the solution of the following problem :*

$$\begin{cases} -\alpha\Delta v_\varepsilon + v_\varepsilon = 0, & \text{in } B_R \setminus B_\varepsilon, \\ v_\varepsilon = \psi, & \text{on } \partial B_\varepsilon, \\ v_\varepsilon = 0, & \text{on } \partial B_R. \end{cases}$$

Then, for ε small enough,

$$\begin{aligned} \|v_\varepsilon\|_{0,B_R\setminus B_\varepsilon} &\leq C_1\, e^{-R/(2\varepsilon\sqrt{\alpha})}\, \|\psi(\varepsilon\,\cdot)\|_{1/2,\partial B_1}, \\ |v_\varepsilon|_{1,B_R\setminus B_\varepsilon} &\leq C_2\, e^{-R/(2\varepsilon\sqrt{\alpha})}\, \|\psi(\varepsilon\,\cdot)\|_{1/2,\partial B_1}, \\ \|v_\varepsilon\|_{0,B_R\setminus B_{R/2}} &\leq C_3\, e^{-R/(2\varepsilon\sqrt{\alpha})}\, \|\psi(\varepsilon\,\cdot)\|_{1/2,\partial B_1} \\ |v_\varepsilon|_{1,B_R\setminus B_{R/2}} &\leq C_4\, e^{-R/(2\varepsilon\sqrt{\alpha})}\, \|\psi(\varepsilon\,\cdot)\|_{1/2,\partial B_1}. \end{aligned}$$

Proof. We consider the following *exterior* problem :

$$\begin{cases} -\alpha\Delta v_{\omega_\varepsilon} + v_{\omega_\varepsilon} = 0, & \text{in } \mathbb{R}^2 \setminus B_1, \\ v_{\omega_\varepsilon} = \psi(\varepsilon\,\cdot), & \text{on } \partial B_1, \\ v_{\omega_\varepsilon} = 0, & \text{at } \infty. \end{cases}$$

Therefore,

$$v_\varepsilon = v_{\omega_\varepsilon}(\cdot/\varepsilon)|_{B_R\setminus B_\varepsilon} - w_\varepsilon,$$

where w_ε is solution of

$$\begin{cases} -\alpha\Delta w_\varepsilon + w_\varepsilon = 0, & \text{in } B_R \setminus B_\varepsilon, \\ w_\varepsilon = 0, & \text{on } \partial B_\varepsilon, \\ w_\varepsilon = v_{\omega_\varepsilon}(\cdot/\varepsilon), & \text{on } \partial B_R. \end{cases}$$

Using Proposition B.4.1 and Proposition B.3.3, for ε small enough, we have

$$\|w_\varepsilon\|_{1,B_R\setminus B_\varepsilon} \leq C\, e^{-R/(2\varepsilon\sqrt{\alpha})}\, \|\psi(\varepsilon\,\cdot)\|_{1/2,\partial B_1}.$$

It follows from scaling argument,

$$\begin{aligned}\|v_\varepsilon\|_{0,B_R\setminus B_\varepsilon} &\leq \|v_{\omega_\varepsilon}(\cdot/\varepsilon)\|_{0,B_R\setminus B_\varepsilon} + \|w_\varepsilon\|_{0,B_R\setminus B_\varepsilon}\\ &\leq C\, e^{-R/(2\varepsilon\sqrt{\alpha})}\, \|\psi(\varepsilon\,\cdot)\|_{1/2,\partial B_1}.\end{aligned}$$

The other estimations are obtained similarly. □

To summarize the estimations of this section, we have,

Proposition B.4.3. *Let $\psi \in H^{1/2}(\partial B_\varepsilon)$ and $\phi \in H^{1/2}(\partial B_R)$. Let v_ε be the solution of the problem below :*

$$\begin{cases} -\alpha\Delta v_\varepsilon + v_\varepsilon = 0, & \text{in } B_R \setminus B_\varepsilon,\\ v_\varepsilon = \psi, & \text{on } \partial B_\varepsilon,\\ v_\varepsilon = \phi, & \text{on } \partial B_R. \end{cases}$$

Then, for ε small enough,

$$\begin{aligned}\|v_\varepsilon\|_{0,B_R\setminus B_\varepsilon} &\leq C_1\left(\|\phi\|_{1/2,\partial B_R} + e^{-R/(2\varepsilon\sqrt{\alpha})}\, \|\psi(\varepsilon\,\cdot)\|_{1/2,\partial B_1}\right),\\ |v_\varepsilon|_{1,B_R\setminus B_\varepsilon} &\leq C_2\left(\|\phi\|_{1/2,\partial B_R} + e^{-R/(2\varepsilon\sqrt{\alpha})}\, \|\psi(\varepsilon\,\cdot)\|_{1/2,\partial B_1}\right),\\ \|v_\varepsilon\|_{0,B_R\setminus B_{R/2}} &\leq C_3\left(\|\phi\|_{1/2,\partial B_R} + e^{-R/(2\varepsilon\sqrt{\alpha})}\, \|\psi(\varepsilon\,\cdot)\|_{1/2,\partial B_1}\right),\\ |v_\varepsilon|_{1,B_R\setminus B_{R/2}} &\leq C_4\left(\|\phi\|_{1/2,\partial B_R} + e^{-R/(2\varepsilon\sqrt{\alpha})}\, \|\psi(\varepsilon\,\cdot)\|_{1/2,\partial B_1}\right).\end{aligned}$$

Appendix C

Some Tools Used in the Thesis

We will quickly present the tools for the analysis of shape optimization problems. Let Y and U be two spaces. We call Y the state space and U the control space. We give a cost function $J : U \times V \to \bar{\mathbb{R}}$. We will call the set of admissible pairs the set

$$\mathcal{A} := \{(u, y) \in U \times V \mid y \in \operatorname{argmin}_Y G(u, \cdot)\},$$

with $G : U \times Y \to \bar{\mathbb{R}}$. The shape optimisation problem writes

$$\min_{(u,y)\in\mathcal{A}} J(u, y).$$

The question is what conditions on J and on G, and which topologies must be chosen for U and Y in order to have the existence of at least one solution.

C.1 Topology for the State Space

For the state space Y, we will use the topology induced by the Γ-convergence [49, 24]. We start by defining this convergence:

Definition C.1.1 (Γ-convergence)**.** *A sequence of functionals $(F_n)_n$, from Y into $\bar{\mathbb{R}}$, Γ-converges to F in Y if,*

- *for all y in Y, there exists a sequence $(y_n)_n$ in Y such that $y_n \to y$ in Y and $F(y) \geq \limsup_{n\to+\infty} F_n(y_n)$,*
- *for any sequence $(y_n)_n$ in Y such that $y_n \to y$ in Y, we have $F(y) \leq \liminf_{n\to+\infty} F_n(y_n)$.*

We write $\Gamma - \lim_{n\to+\infty} F_n = F$.

Here are some properties of the Γ-convergence.

Proposition C.1.1.

- *The Γ-limit of a sequence $(F_n)_n$ is lower semi-continuous,*
- *Let (F_n) be a sequence which Γ-converges to F and let G be a continuous functional, then $(F_n + G)_n$ Γ-converges to $F + G$.*

In the following, we need the definition of (equi)-coercivity.

Definition C.1.2. *We say that the sequence of functions (F_n) is equi-coercive on Y if, for all $t \in \mathbb{R}$, there exists a compact set $K_t \subset Y$ such that $\{F_n \leq t\} \subset K_t$, for all $n \in \mathbb{N}$.*

Proposition C.1.2. *A sequence of functionals (F_n) is equi-coercive on Y if and only if there exists $\psi : Y \to \bar{\mathbb{R}}$ lower semi-continuous and coercive such that $F_n \geq \psi$ for all $n \in \mathbb{N}$.*

The Γ-convergence is much used in calculus of variations because it ensures the convergence of minimizers under certain conditions. More precisely, we have the following property:

Proposition C.1.3. *If (F_n) is an equi-coercive sequence that Γ-converges to F in Y, then F reaches its minimum and $\min_{y\in Y} F(y) = \lim_{n\to+\infty} \inf_{y\in Y} F_n(y)$. Moreover, if the sequence $(y_n)_n$, $y_n \in \operatorname{argmin}_Y F_n$, converges to y, then $y \in \operatorname{argmin}_Y F$.*

We have then the following theorem, proved in [49] Theorem 10.22.,

Theorem C.1.1. *The Γ-convergence is metrizable on*

$$S_\psi(Y) := \{F : Y \to \bar{\mathbb{R}} \mid G \text{ l.s.c. and } G \geq \psi\},$$

with $\psi : Y \to \bar{\mathbb{R}}$ l.s.c. and coercive.

C.2 Topology for the Control Space

We will use the topology induced by the γ-convergence, defined as follows [48, 32, 28] :

Definition C.2.1. *We say that the sequence $(u_n)_n$ of elements of U $\gamma(G)$-converges to u if $G(u_n, \cdot)$ Γ-converges to $G(u, \cdot)$.*

Note. The $\gamma(G)$-convergence is dependent on G, but when there is no ambiguity, we will simply talk about γ-convergence.

From now on, we assume that :

- for all u in U, $G(u, \cdot)$ is l.s.c.,
- G is equi-coercive, then (Proposition C.1.2) there exists $\psi : Y \to \bar{\mathbb{R}}$ l.s.c. and coercive such that $F_n \geq \psi$ for all $n \in \mathbb{N}$,
- $\Gamma_G : U \to S_\psi(Y)$, $u \mapsto G(u, \cdot)$ is bijective, with $S_\psi(Y)$ defined in Theorem C.1.1.

We then have the following proposition:

Proposition C.2.1.
- *The γ-convergence is metrizable on U,*
- *Γ_G is an isometry.*

C.3 Compactness

We start with a compactness result for the state space.

Proposition C.3.1. *$S_\psi(Y)$ is compact w.r.t. the Γ-convergence.*

Now, we can give examples of control spaces U that are compact for the topology induced by γ-convergence [28].

Definition C.3.1. *We define,*

- $\mathcal{A}_{convex}$*, the class of convex sets included in* D*,*
- $\mathcal{A}_{unif\ cone}$*, the part of the sets satisfying a uniform exterior cone property,*
- $\mathcal{A}_{unif\ flat\ cone}$*, the part of the sets satisfying the uniform flat cone condition.*

We have the following inclusions:

Proposition C.3.2.

$$\mathcal{A}_{convex} \subseteq \mathcal{A}_{unif\ cone} \subseteq \mathcal{A}_{unif\ flat\ cone}.$$

Proposition C.3.3. *The sets* $\mathcal{A}_{convex}$*,* $\mathcal{A}_{unif\ cone}$ *and* $\mathcal{A}_{unif\ flat\ cone}$ *are compact w.r.t. the* γ*-convergence.*

Note that this is a non-exhaustive list, other examples are presented in Section 5.1 of [28]. We then have the existence of solutions for the following shape optimization problem stated below,

Theorem C.3.1. *Let* $j : D \times \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}$ *be a Carathéodory function, i.e. it is measurable in its first argument, and continuous in all others. Then, the optimization problem*

$$\min_{A \in \mathcal{U}_{ad}} \left\{ \int_A j(x, y_A, \nabla y_A) \mid y_A \in \operatorname{argmin}_Y G(A, \cdot) \right\},$$

admits at least one solution for $\mathcal{U}_{ad} = \mathcal{A}_{convex}$*,* $\mathcal{A}_{unif\ cone}$*,* $\mathcal{A}_{unif\ flat\ cone}$*, respectively.*

Even if $S_\phi(Y)$ is compact for Γ-convergence, U may not be compact for the γ-convergence. Indeed, if $G(u_n, \cdot)$ Γ-converges to F, then F may not be of the form $G(u, \cdot)$ for $u \in U$. We then need to take a space larger than U.

C.4 Relaxation

We call the set of relaxed controls the set $\hat{U} := \overline{U}^\gamma$ and we keep noting by γ the convergence on $\hat{U}$. We then have the following proposition:

Proposition C.4.1. *The space* $\hat{U}$ *is compact w.r.t. the* γ*-convergence.*

In order to define the relaxed optimization problem, we need to define a relaxed state functional $\hat{G}$ and a relaxed cost function $\hat{J}$. We define $\hat{G}$ from $\hat{U} \times Y \to \bar{R}$ by

$$\hat{G}(\hat{u}, y) := \Gamma - \lim_{u \to^\gamma \hat{u}} G(u, y),\ \forall (\hat{u}, y) \in \hat{U} \times Y.$$

Then, we set

$$\hat{\mathcal{A}} := \{(\hat{u}, y) \in \hat{U} \times V \mid y \in \operatorname{argmin}_Y \hat{G}(\hat{u}, \cdot)\}.$$

Then, we define $\hat{J}$ from $\hat{\mathcal{A}}$ to $\bar{\mathbb{R}}$ by

$$\hat{J}(\hat{u}, y) = \inf \left\{ \liminf_{n \to +\infty} J(\hat{u_n}, y_n) \mid (\hat{u}_n, y_n) \in \hat{\mathcal{A}},\ \hat{u}_n \xrightarrow{\gamma} \hat{u},\ y_n \xrightarrow{\Gamma} y \right\}.$$

Thus, we define the relaxed optimization problem

$$\min_{(\hat{u}, y) \in \hat{\mathcal{A}}} \hat{J}(\hat{u}, y).$$

We then have the following theorem:

Theorem C.4.1. *The relaxed optimization problem*

$$\min_{(\hat{u},y)\in\hat{\mathcal{A}}} \hat{J}(\hat{u},y),$$

admits at least one solution. Moreover, the infimum of the original optimization problem coincides with the minimum of the relaxed optimization problem. Finally, if $(u_n, y_n) \in \mathcal{A}$ *is a minimizing sequence of the original optimization problem, then there exists a sub-sequence converging to* $(\hat{u}, y) \in \hat{\mathcal{A}}$ *solution of the relaxed optimization problem*

C.5 The case of PDEs with Dirichlet Condition

We may not have a classical solution for PDEs with the Lebesgue measure. On the other hand, Sverák proved that we have the existence of a classical solution in the case of the Hausdorff measure if we add a constraint on the number of related components of our admissible sets [28]. To have the existence of a relaxed solution, we can use the notion of capacity [28, 32, 50, 48]. It is this case that we will detail in the rest of this section.

C.5.1 Capacity

We start by giving an example of Choquet capacity [44, 45], which will have an important role for the shape optimization [48], and some of its properties. Let D be an open of $\mathbb{R}^d$.

Definition C.5.1 (Capacity)**.** *For all compact set* K *of* D*, we define the capacity of* K *w.r.t.* D *by*

$$\mathrm{cap}(E,D) := \inf\Big\{ \int_D |\nabla u|^2 \, dx + \int_D u^2 \, dx \;\Big|\; u \in H_0^1(D),\ u \geq 1 \text{ a.e. in a neighborhood of } E\Big\}.$$

We extend the definition of capacity for any subset E of D.

Definition C.5.2. *We extend the definition of capacity to open sets* U *of* D *by*

$$\mathrm{cap}(U,D) := \sup\big\{ \mathrm{cap}(K,D) \;\big|\; K \subseteq U \text{ compact}\big\}.$$

We extend the definition of capacity for any subset E *of* D *by*

$$\mathrm{cap}(E,D) := \inf\big\{ \mathrm{cap}(U,D) \;\big|\; E \subseteq U \text{ open}\big\}.$$

Now some definitions using the notion of capacity.

Definition C.5.3 (Quasi-everywhere)**.** *If a property is true for all* x *belonging to* $E \setminus Z$ *where* Z *is a subset of* E *with* $\mathrm{cap}(Z) = 0$*, we say that this property is true almost everywhere on* E *and we write q.e.*

Definition C.5.4 (Quasi-open / quasi-close / quasi-compact)**.** *We say that a subset* A *of* D *is quasi-open (respectively quasi-close, quasi-compact) in* D *if for all* $\varepsilon > 0$*, there exists an open set (respectively closed, compact)* U *of* D *such that* $\mathrm{cap}(A \Delta U, D) < \varepsilon$*.*

Note. Δ refers to the symmetric difference, i.e. $E, F \subset D$, $E\Delta F := (E \setminus F) \cup (F \setminus E)$.

We give an equivalent definition [96] :

Proposition C.5.1. *Let E be a subset of D. Then,*

$$\mathrm{cap}(E,D) = \inf_{u\in U'_E} \int_D |\nabla u|^2 + u^2 \, dx,$$

with $U'_E := \{u \in U_E \mid 0 \leq u \leq 1\}$.

Note. Therefore, if E is an open set, the function u from the definition above is the solution of

Problem C.5.1. *Find u in $H^1_0(D)$ such that*

$$\begin{cases} -\Delta u + u = 0, & \text{in } D \setminus E, \\ u = 1, & \text{in } E. \end{cases} \tag{C.1}$$

The capacity is therefore the value of the energy of u.

Then, we state estimates of the capacity with respect to the Lebesgue measure.

Proposition C.5.2. *Let E be subset of D. Then,*

$$\mathcal{L}^d(E) \leq \mathrm{cap}(E).$$

Note. Thus, if $\mathrm{cap}(E,D) = 0$, then $\mathcal{L}^d(E) = 0$. Moreover, we have $0 < \mathrm{cap}\big(\{x_0\}, D\big)$. Indeed, $u \geq 1$ on a neighborhood O of E. Particularly, there exists an open ball $B(x_0, r)$ included in O. Then

$$0 < \mathcal{L}^d\big(B(x_0,r)\big) \leq \mathrm{cap}\big(B(x_0,r), D\big) \leq \mathrm{cap}(O,D) = \mathrm{cap}\big(\{x_0\}, D\big).$$

C.5.2 The Set $\mathcal{M}_0(D)$

In this part, we will extend the capacity of a set to measures of a certain type. We introduce the set $\mathcal{M}_0(D)$, which is denoted $\mathcal{M}_0^*(D)$ in [48].

Definition C.5.5 (The set $\mathcal{M}_0(D)$)**.** *We denote by $\mathcal{M}_0(D)$ the set of all positive Borel measure μ on D, such that*

- $\mu(B) = 0$, *for all Borel set B of D with* $\mathrm{cap}(B,D) = 0$,
- $\mu(B) = \inf\big\{\mu(U) \;\big|\; U \text{ quasi-ouvert}, \; B \subseteq U\big\}$, *for all Borel set B of D.*

Now we give a natural way to identify a set of $\mathbb{R}^d$ with a measure of $\mathcal{M}_0(D)$.

Definition C.5.6. *Let E be a Borel subset of D. We denote by ∞_E the measure in $\mathcal{M}_0(D)$ defined by,*

$$\infty_E(B) := \begin{cases} +\infty & , \text{ if } \mathrm{cap}(B \cap E, D) > 0, \\ 0 & , \text{ otherwise.} \end{cases} \text{, for all } B \text{ Borel subset of } D.$$

Now we can extend the capacity of a set to measures of $\mathcal{M}_0(D)$.

Definition C.5.7 (Measure capacity)**.** *The capacity of a measure μ in $\mathcal{M}_0(D)$ is defined by*

$$\mathrm{cap}(\mu, D) := \inf_{u\in H^1_0(D)} \int_D |\nabla u|^2 \, dx + \int_D u^2 \, dx + \int_D (u-1)^2 \, d\mu.$$

Formally, when $\mu = \infty_A$, we penalize the functional which is then $+\infty$ unless $u = 1$ on A. We have, in a way, encoded the Dirichlet condition by means of this penalization. We can now make the connection between the capacity of sets and the capacity of measures.

Proposition C.5.3. *Let E be a Borel subset of D. Then,* $\mathrm{cap}(\infty_E, D) = \mathrm{cap}(E, D)$.

Finally, we have the following compacity result.

Proposition C.5.4. *The set $\mathcal{M}_0(D)$ is compact w.r.t. the γ-convergence. In addition, measures $\infty_{D\setminus A}$, with A smooth open set of D is dense in $\mathcal{M}_0(D)$.*

The previous result tells us that for any μ in $\mathcal{M}_0(D)$, there exists a family of subsets $(E_n)_n$ of D, such that ∞_{E_n} γ-converges to μ. Thus, in the case of shape optimization, we can take as control space U, measures of the form $\infty_{D\setminus A}$, with A smooth open subset of D and as relaxed control space $\hat{U}$, $\mathcal{M}_0(D)$. Finally, we have a locality result for γ-convergence :

Proposition C.5.5 (Locality of the γ-convergence)**.** *Let $(\mu_n^1)_n$ and $(\mu_n^2)_n$ be two sequences of measures from $\mathcal{M}_0(D)$ which γ-converge to μ^1 and μ^2 respectively. We suppose that μ_n^1 and μ_n^2 coincide q.e. on a subset D' of D, for all $n \in \mathbb{N}$. Then μ^1 and μ^2 coincide q.e. on D'.*