



UNIVERSITÉ de CAEN/BASSE-NORMANDIE

U.F.R. Sciences

ÉCOLE DOCTORALE SIMEM

**T H È S E**

*Présentée par*

**M. Mahmoud GHONIEM**

*et soutenue*

le 14 Décembre 2010

*en vue de l'obtention du*

**DOCTORAT de l'UNIVERSITÉ de CAEN BASSE-NORMANDIE**

**Spécialité : Informatique et applications**

*(Arrêté du 07 août 2006)*

**Régularisation spatio-temporelle :  
applications au débruitage, à l'inpainting  
et à la classification**



**MEMBRES du JURY**

|                           |                      |                            |                      |
|---------------------------|----------------------|----------------------------|----------------------|
| M. Michel DESVIGNES       | Professeur           | Grenoble-INP               | (rapporteur)         |
| M. Abdelmalik TALEB-AHMED | Professeur           | Université de Valenciennes | (rapporteur)         |
| Mme Marinette REVENU      | Professeur           | Université de Caen         | (examinatrice)       |
| M. Mahmoud MELKEMI        | Professeur           | Université de Haute-Alsace | (examineur)          |
| M. Youssef CHAHIR         | Maître de conférence | Université de Caen         | (co-encadrant)       |
| M. Abderrahim ELMOATAZ    | Professeur           | Université de Caen         | (directeur de thèse) |



---

# REMERCIEMENTS

---

Je remercie les membres de mon jury d'avoir bien voulu participer à l'évaluation de mon travail : Michel DESVIGNES, Professeur à Grenoble-INP et Abdelmalik TALEB-AHMED, Professeur à l'Université de Valenciennes, pour avoir bien voulu accepter la charge de rapporteur, ainsi qu'à Marinette REVENU, Professeur à l'Université de Caen, et Mahmoud MELKEMI, Professeur à l'Université de Haute-Alsace, pour leur contribution en tant qu'examineurs.

Je sais gré à Abderrahim ELMOATAZ, Professeur à l'Université de Caen, qui a dirigé ma thèse, et à Youssef CHAHIR, Maître de Conférences à l'Université de Caen, qui a encadré ma recherche, pour leur confiance et leurs conseils durant ces trois années de thèse.

Je tiens à remercier également mes collègues avec qui j'ai partagé des idées et des moments agréables.

Ma gratitude va naturellement à mes parents à qui je dois tout, à mon épouse pour sa patience et son soutien, et à mes frères pour leurs conseils et encouragements.





---

# TABLE DES MATIÈRES

---

|   |            |
|---|------------|
| <b>Introduction</b>   | <b>vii</b> |
| <b>I Restauration : Débruitage et Simplification</b>                                | <b>1</b>   |
| <b>1 Lissage des images et des vidéos</b>   | <b>3</b>   |
| 1.1 Introduction . . . . .  | 3          |
| 1.1.1 Sources du bruit . . . . .  | 4          |
| 1.1.2 Modélisation du bruit . . . . .   | 5          |
| 1.2 Régularisation des images par minimisation d'énergies . . . . .                 | 6          |
| 1.2.1 Régularisation de Tikhonov . . . . .  | 7          |
| 1.2.2 Régularisation VT . . . . .   | 7          |
| 1.2.3 Régularisation $p$ -VT . . . . .  | 8          |
| 1.2.4 Régularisation continue non locale . . . . .                                  | 8          |
| 1.2.5 Régularisation discrète . . . . .   | 10         |
| 1.3 Régularisation des séquences d'images . . . . .                                 | 11         |
| 1.3.1 Méthodes de débruitage locales . . . . .                                      | 13         |
| 1.3.2 Méthodes de débruitage non locales . . . . .                                  | 13         |
| <b>2 Contributions à la régularisation des images et des vidéos</b>                 | <b>15</b>  |
| 2.1 Modèles variationnels avec les $p$ -Laplaciens isotrope et anisotrope . . . . . | 15         |
| 2.1.1 Définition et construction du graphe pondéré . . . . .                        | 15         |
| 2.1.2 Opérateurs différentiels sur graphe . . . . .                                 | 17         |
| 2.1.3 Formulation mathématique et approche proposée . . . . .                       | 20         |
| 2.1.4 Liens avec les méthodes classiques . . . . .                                  | 22         |
| 2.2 Régularisation par $\infty$ -Laplacien . . . . .                                | 24         |
| 2.2.1 $\infty$ -Laplacien sur graphes . . . . .                                     | 25         |
| 2.2.2 Nouvelle interprétation et résolution . . . . .                               | 27         |
| 2.3 Résultats et performances . . . . .   | 28         |
| 2.3.1 Débruitage . . . . .  | 28         |
| 2.3.2 Simplification de vidéos . . . . .  | 36         |
| <b>II Restauration : Inpainting et Interpolation</b>                                | <b>39</b>  |
| <b>3 Inpainting des images et des vidéos</b>  | <b>41</b>  |
| 3.1 Problématique de l'inpainting . . . . .   | 41         |
| 3.2 Structures géométriques vs. textures . . . . .                                  | 42         |
| 3.2.1 Méthodes géométriques . . . . .   | 43         |
| 3.2.2 Méthodes basées sur la synthèse de textures . . . . .                         | 45         |
| 3.2.3 Méthodes non locales . . . . .  | 45         |

|   |            |
|---|------------|
| <b>4 Contributions à l'interpolation et l'inpainting des images et des vidéos</b> | <b>47</b>  |
| 4.1 Inpainting par régularisation   | 47         |
| 4.1.1 Formulation variationnelle du problème d'inpainting                         | 48         |
| 4.1.2 Algorithmes et stratégies d'inpainting                                      | 49         |
| 4.1.3 Résultats expérimentaux   | 53         |
| 4.2 Fonctions $p$ -harmoniques et interpolation sur graphe                        | 61         |
| 4.2.1 Existence et unicité d'une solution   | 61         |
| 4.2.2 Formulation équivalente   | 63         |
| 4.2.3 Cas particulier de $p = \infty$   | 64         |
| 4.2.4 Résultats et applications   | 66         |
| <br>  |            |
| <b>III Régularisation Spectrale</b>   | <b>71</b>  |
| <br>  |            |
| <b>5 Méthodes spectrales pour la classification</b>                               | <b>73</b>  |
| 5.1 Méthodes linéaires  | 75         |
| 5.1.1 ACP   | 75         |
| 5.2 Méthodes basées sur graphe  | 76         |
| 5.2.1 Isomap  | 76         |
| 5.2.2 LLE   | 77         |
| 5.2.3 Laplacian Eigenmaps   | 78         |
| 5.2.4 Cartes de diffusion ( <i>Diffusion Maps</i> )                               | 79         |
| <br>  |            |
| <b>6 Contributions à l'analyse spectrale</b>                                      | <b>81</b>  |
| 6.1 Classification par cartes de diffusion  | 82         |
| 6.1.1 Construction du graphe  | 82         |
| 6.1.2 Laplacien du graphe   | 83         |
| 6.1.3 Cadre de diffusion utilisé  | 83         |
| 6.2 Résultats et applications : segmentation et classification                    | 88         |
| 6.2.1 Segmentation par classification et diffusion                                | 88         |
| 6.2.2 Caractérisation des éléments saillants                                      | 89         |
| 6.2.3 Classification de lettres manuscrites                                       | 94         |
| <br>  |            |
| <b>Conclusion et Perspectives</b>   | <b>103</b> |
| <br>  |            |
| <b>Bibliographie</b>  | <b>107</b> |

---

# INTRODUCTION

---

## Contexte général

La dernière décennie a connu une large prolifération des données numériques de tous types (images, vidéos, maillage, mesures, etc.) grâce aux progrès technologiques. Ces données sont souvent bruitées ou de grandes dimensions. Les traitements de ces données, tels que le débruitage, l'interpolation ou l'organisation, visant à en extraire la connaissance est un vrai défi pour les communautés de traitement d'images et d'apprentissage.

Une large partie de ces données sont directement collectées ou peuvent être représentées sous forme de graphes ou de réseaux complexes qui encapsulent toute l'information. En conséquence, la majorité des traitements que nous avons cités se ramènent à des problématiques d'analyse de graphes. Il existe deux méthodologies envisageables pour étudier les fonctions sur graphes.

La première consiste à travailler directement sur les fonctions définies sur graphes et à formuler directement les problématiques étudiées dessus. Pour les traitements qui nous intéressent, il s'agit de résoudre un certain nombre de problèmes inverses définis sur graphes, tels que la régularisation spatio-temporelle.

La seconde méthodologie consiste à passer d'un espace de « mesure » à un espace d'« observations » plus réduit grâce à des techniques de réduction de dimensionnalité par analyse spectrale. Les traitements souhaités sont alors effectués dans cet espace réduit. C'est ce que nous appelons la régularisation spectrale.

Dans le présent travail, nous nous intéressons aux deux méthodologies ci-dessus. Tout d'abord, nous présentons une approche de régularisation spatio-temporelle sur graphes en utilisant le formalisme des équations aux différences partielles introduites récemment. Nous l'appliquons à la restauration des images et des séquences d'images bruitées ainsi qu'à l'inpainting et l'interpolation des images et des séquences d'images. Ensuite, nous proposons une approche de régularisation spectrale sur un nuage de patches. Nous appliquons nos méthodes pour la segmentation et la caractérisation d'éléments saillants. En outre, nous présentons une application réelle de la visualisation et l'organisation des données qui s'inscrit dans le cadre d'un projet d'apprentissage de l'écriture.

## Contributions et plan du manuscrit

Ce document est organisé en trois parties qui correspondent au lissage, à l'inpainting et à la classification des images et des vidéos. Chaque partie se compose de deux chapitres. Le premier chapitre passe en revue les méthodes proposées dans la littérature pour résoudre la problématique en question. Le second détaille nos contributions sur le sujet et présente les résultats obtenus.

## PARTIE I : DÉBRUITAGE ET SIMPLIFICATION

Dans cette partie, nous abordons la régularisation spatio-temporelle des images et des vidéos et ses applications au débruitage et à la simplification.

### Chapitre 1 : Lissage des images et des vidéos

Ce chapitre aborde la tâche de lissage pour la réduction du bruit dans les images et les vidéos. Il existe de nombreuses approches de lissage d'images provenant des méthodes statistiques, de la théorie de l'information, des transformées dans le domaine fréquentiel, des équations aux dérivées partielles (EDP) et des méthodes variationnelles. Nous nous sommes particulièrement intéressé aux approches *basées sur la régularisation* continue par EDP car notre objectif était de transcrire certaines de ces EDP dans un cadre discret pour différentes applications de la régularisation.

### Chapitre 2 : Contributions à la régularisation des images et des vidéos

Dans ce chapitre, nous proposons l'extension d'un cadre de régularisation discrète aux séquences d'images. Nous avons mis en place des algorithmes de régularisation sur graphes par  $p$ -Laplacien,  $p \in ]0, +\infty[$ , pour le débruitage et la simplification des séquences d'images. Nous proposons une optimisation de l'algorithme non local pour l'accélérer. En outre, nous introduisons un algorithme nouveau de  $p$ -régularisation, avec  $p = \infty$  qui sera comparé aux algorithmes proposés pour  $1 \leq p < \infty$ .

## PARTIE II : INPAINTING ET INTERPOLATION

Cette partie traite le problème de la restauration des données manquantes. Tout d'abord, nous rappelons les approches proposées dans la littérature pour résoudre ce problème. Ensuite, nous présentons l'application de la régularisation à des fins d'inpainting ainsi qu'une seconde approche d'interpolation par fonctions  $p$ -harmoniques.

### Chapitre 3 : Inpainting des images et des vidéos

Dans ce chapitre, nous nous intéressons à la restauration des données manquantes. Contrairement à la restauration des données bruitées, il n'y a pas de bruit à modéliser ni de données initiales auxquelles il faudrait être fidèle. Il s'agit ici d'un problème d'interpolation qui consiste à réparer une scène en construisant un contenu nouveau en harmonie avec les données non perdues de la scène en question. Nous n'abordons pas l'étape préliminaire de sélection des défauts ou des dégradations telles que les déchirures ou les taches. Nous considérons uniquement le remplissage du trou créé par la suppression des taches et nous supposons que la région à combler est déjà définie. Il existe des travaux qui traitent cette étape de prétraitement pour des applications spécifiques telles que la restauration des films anciens.

Ensuite, nous définissons le problème d'inpainting puis nous passons en revue une sélection non exhaustive des solutions présentes dans la littérature.

### Chapitre 4 : Contributions à l'interpolation et l'inpainting des images et des vidéos

Dans ce chapitre, nous considérons deux approches d'interpolation. Premièrement, nous proposons l'extension des méthodes d'interpolation continues au domaine de l'inpainting. Nous traitons le résultat d'une première interpolation par régularisation itérative pour améliorer la qualité de la reconstruction des régions manquantes.

Deuxièmement, nous nous intéressons aux fonctions  $p$ -harmoniques qui constituent un cadre général d'interpolation permettant la réalisation de nombreuses applications dont l'inpainting, l'agrandissement et la segmentation semi-supervisée sur graphes. Nous montrons que ce cadre unifie les approches géométriques et les approches basées sur la synthèse de texture. Ensuite, nous introduisons le cas nouveau de l'interpolation par fonctions  $\infty$ -harmoniques.

## **PARTIE III : RÉGULARISATION SPECTRALE**

### **Chapitre 5 : Méthodes spectrales pour la classification**

Ce chapitre présente un bref état de l'art des méthodes spectrales pour la réduction de la dimensionnalité et la classification des images. Dans un grand nombre d'applications en traitement d'images, nous sommes souvent confrontés à des données massives de grandes dimensions. Afin de faciliter leurs traitements tels que la visualisation ou l'organisation, il est indispensable de réduire leur dimension. Pour ce faire, les méthodes spectrales se sont avérées des outils puissants.

### **Chapitre 6 : Contributions à la classification des images**

Dans ce chapitre, nous formulons une approche de régularisation spectrale sur graphes basée sur les cartes de diffusion. Premièrement, nous proposons un algorithme de segmentation à partir des points caractéristiques de l'image. Ensuite, nous présentons la détection des éléments saillants des images par marches aléatoires sur graphes. Et enfin, nous présentons un projet d'apprentissage d'écriture auquel nous avons contribué en proposant un algorithme de classification de données hétérogènes par régularisation des données dans l'espace spectral réduit.



**PREMIÈRE PARTIE**

---

**RESTAURATION : DÉBRUITAGE ET  
SIMPLIFICATION**

---





---

# LISSAGE DES IMAGES ET DES VIDÉOS

---

## Sommaire

|            |  |           |
|------------|--|-----------|
| <b>1.1</b> | <b>Introduction</b> . . . . .  | <b>3</b>  |
| 1.1.1      | Sources du bruit . . . . .   | 4         |
| 1.1.2      | Modélisation du bruit . . . . .  | 5         |
| <b>1.2</b> | <b>Régularisation des images par minimisation d'énergies</b> . . . . . | <b>6</b>  |
| 1.2.1      | Régularisation de Tikhonov . . . . .                                   | 7         |
| 1.2.2      | Régularisation VT . . . . .  | 7         |
| 1.2.3      | Régularisation $p$ -VT . . . . .                                       | 8         |
| 1.2.4      | Régularisation continue non locale . . . . .                           | 8         |
| 1.2.5      | Régularisation discrète . . . . .                                      | 10        |
| <b>1.3</b> | <b>Régularisation des séquences d'images</b> . . . . .                 | <b>11</b> |
| 1.3.1      | Méthodes de débruitage locales . . . . .                               | 13        |
| 1.3.2      | Méthodes de débruitage non locales . . . . .                           | 13        |

---

## Résumé

Ce chapitre aborde la tâche de lissage pour la réduction du bruit dans les images et les vidéos. Il existe de nombreuses approches de lissage d'images provenant des méthodes statistiques, de la théorie de l'information, des transformées dans le domaine fréquentiel, des équations aux dérivées partielles (EDP) et des méthodes variationnelles. Par souci de cohérence avec notre travail, nous avons choisi de nous limiter aux approches *basées sur la régularisation continue* par EDP uniquement car notre objectif est de transcrire certaines de ces EDP dans un cadre discret pour différentes applications de la régularisation.

## 1.1 Introduction

Le *lissage* est une tâche fondamentale en traitement d'images visant à réduire, voire supprimer, le bruit provenant d'appareils photos numériques ou de scanners et à en améliorer ainsi la qualité visuelle. Il peut être employé à des fins de *débruitage* ou de *simplification*. Le lissage peut être une fin en soi ou être utilisé comme étape de prétraitement préalable à une autre opération de traitement et d'analyse d'images et de vidéos. Ainsi, le lissage des images permet de fournir des images simplifiées permettant l'amélioration d'autres tâches telles que la segmentation, l'extraction de caractéristiques, l'analyse de textures, etc. Nous distinguons la notion de débruitage de la notion d'*amélioration d'images*. En effet, la finalité du débruitage est de restituer le plus fidèlement possible le signal d'origine, tandis que l'enjeu de l'amélioration d'images est la qualité du rendu visuel qui reste une notion subjective. Ceci dit, il est difficile de cerner clairement la frontière entre les

deux notions, vu que le débruitage entraîne obligatoirement une amélioration de l'image. La simplification, elle, offre la possibilité d'analyser des images et des vidéos à différentes échelles.

Il existe de nombreuses approches de lissage d'images provenant des méthodes statistiques, de la théorie de l'information, des transformées dans le domaine fréquentiel, des équations aux dérivées partielles (EDP) et des méthodes variationnelles [Weickert, 1998, Chan et Shen, 2005a, Aubert et Kornprobst, 2006, Winkler, 2006]. L'établissement des équivalences et des relations entre les différentes approches a fait l'objet de nombreux travaux de recherche [Saint-Marc *et al.*, 1991, Scherzer et Weickert, 1998, van den Boomgaard et van de Weijer, 2002, Elad, 2002, Steidl *et al.*, 2004, Barash et Comaniciu, 2004, Didas et Weickert, 2007]. Mrázek *et al.* [Mrázek *et al.*, 2004] ont souligné le lien entre plusieurs méthodes de lissage non linéaires telles que les M-estimateurs [Chu *et al.*, 1998, Winkler *et al.*, 1999], les filtres bilatéraux [Tomasi et Manduchi, 1998], les filtres de diffusion [Perona et Malik, 1990, Weickert, 1997], et la *régularisation* et les techniques bayésiennes [Geman et Geman, 1984, Bertero *et al.*, 1988, Mumford, 1994].

Depuis l'apparition du filtre moyen non local (*NL-means*) [Buades *et al.*, 2005b], de nombreuses approches citées ci-dessus ont été étendues dans un cadre non local. Elles ont montré leur efficacité par rapport aux méthodes locales quant au lissage des images et des vidéos en préservant des caractéristiques comme la texture ou les éléments répétitifs.

Même si ces méthodes semblent très différentes à première vue et proviennent de différentes théories mathématiques, certains travaux récents ont proposé un cadre de régularisation discrète rapprochant et unifiant certaines de ces méthodes locales et non locales [Elmoataz *et al.*, 2008, Pizarro *et al.*, 2010].

Par conséquent, nous avons choisi de travailler dans le cadre de la régularisation discrète proposée par Elmoataz *et al.* [Elmoataz *et al.*, 2008]. Par souci de cohérence avec notre travail, nous avons choisi de nous limiter aux approches *basées sur la régularisation* continue par EDP uniquement. Notre objectif est de transcrire certaines de ces EDP dans un cadre discret pour différentes applications de la régularisation.

Après un rappel des différents types de dégradations auxquelles les images peuvent être exposées, nous présentons dans la section 1.2 les principales classes de régularisation d'images. Ensuite, nous étudions dans la section 1.3 la problématique de la régularisation de vidéos où la dimension temporelle des données modifie significativement le traitement effectué.

### 1.1.1 Sources du bruit

Les images et les vidéos capturées sont des représentations de scènes obtenues à l'aide de dispositifs d'acquisition. Les données acquises subissent une chaîne d'opérations comme la numérisation, l'échantillonnage, la transmission et la compression. À chaque opération, de nouvelles dégradations peuvent avoir lieu. Par conséquent, le résultat final souffre de dégradations de divers types et origines. La restauration d'images a pour objectif principal d'atténuer, voire de supprimer dans les cas les plus favorables, les dégradations et pouvoir ainsi retrouver, à partir de l'observation dégradée, une observation plus proche de l'image telle que l'on devrait l'avoir en situation idéale. En conséquence, la restauration est une étape importante de prétraitement pour de nombreuses problématiques d'analyse d'images telles que la segmentation. Ainsi, la restauration d'images permet d'une part, l'amélioration de la qualité visuelle du contenu et, d'autre part, l'amélioration des performances des traitements qui exploitent le contenu.

Dans ce travail, nous nous intéressons uniquement aux dégradations ponctuelles, i.e., nous considérons le bruit dans les séquences d'images. Les séquences vidéo sont souvent

déformées par le bruit durant l'acquisition, l'enregistrement et/ou la transmission [Amer et Dubois, 2005]. La dégradation de la qualité d'une image peut être de plusieurs types. Dans le langage courant, le terme de « bruit » est implicitement sonore et désigne un signal indésirable audible se superposant à un signal utile tel qu'un morceau de musique, des paroles ou même le silence. Le bruit est donc un signal parasite qui altère la qualité du signal utile. Dans la théorie du signal, la définition du bruit s'étend à tout type de signal, sonore ou pas. Différents types de bruits sont alors distingués et peuvent exister en même temps dans tout système électronique délivrant un signal. Le bruit est une famille de dégradations parmi d'autres. Rappelons les différents types de bruits concernant les signaux que nous traitons à savoir les images et les vidéos numériques :

- bruit d'acquisition
- dégradation de numérisation
- dégradation de transmission
- dégradation de changement de format
- *dirt* et *sparkles*
- artefacts

Les sources de bruit sont difficiles à modéliser. Par conséquent, pour réduire voire supprimer un bruit, il est important de connaître comment ce bruit se mélange aux données utiles (bruit additif, multiplicatif, etc.). Il est également pertinent de connaître la loi de distribution que suit le modèle de bruit (distribution uniforme, gaussienne, de Rayleigh, etc.)

Nous constatons ainsi que la majorité de nos données multimédia peuvent souffrir d'une baisse de qualité en raison des différents types de bruits. Par conséquent, il est évident que le débruitage soit l'un des objectifs majeurs du traitement d'images.

Par nature, la restauration est un problème inverse mal posé. Le principe de la restauration est simple à énoncer : étant donnée une dégradation, il faut la modéliser et appliquer la dégradation inverse à la fonction dégradée afin de retrouver la fonction originale. À ce problème, de nombreuses solutions ont été proposées. Parmi elles, les solutions variationnelles qui peuvent être subdivisées en méthodes locales et méthodes non locales.

### 1.1.2 Modélisation du bruit

Dans de nombreuses applications, une bonne approximation du bruit peut être faite par un modèle de *bruit additif blanc gaussien* [de Haan, 2003]. Dans le présent travail, nous nous plaçons dans le cadre d'un bruit blanc gaussien. Par la suite, nous désignons indifféremment une image ou une vidéo par le terme *fonction*.

Supposons que nous observons une fonction  $f^0 : \Omega \rightarrow \mathbb{R}^2$  (image en niveaux de gris, définie sur un domaine  $\Omega \subset \mathbb{R}^n$ ), et nous cherchons à retrouver l'image « originale »  $f$  dont  $f^0$  est une version dégradée :

$$f^0 = \mathcal{R}f + \eta \quad (1.1)$$

où  $\mathcal{R}$  représente un opérateur linéaire continu de  $\mathcal{L}^2(\Omega)$  dans  $\mathcal{L}^2(\Omega)$  (généralement un opérateur de flou) et  $\eta$  est un bruit gaussien (à moyenne nulle et de variance  $\sigma^2$ ).

Toute image contient du bruit même si visuellement aucun bruit n'est perceptible. La décomposition d'une image bruitée en une scène moins bruitée et une autre contenant le bruit correspond bien à la perception humaine qui sépare la scène du bruit qui s'y mélange (voir figure 1.1). C'est pourquoi nous pouvons distinguer le contenu d'une scène occultée par un bruit très élevé<sup>1</sup>.

---

1. La scène reste visible jusqu'à un rapport signal/bruit,  $SNR = 2/1$ .

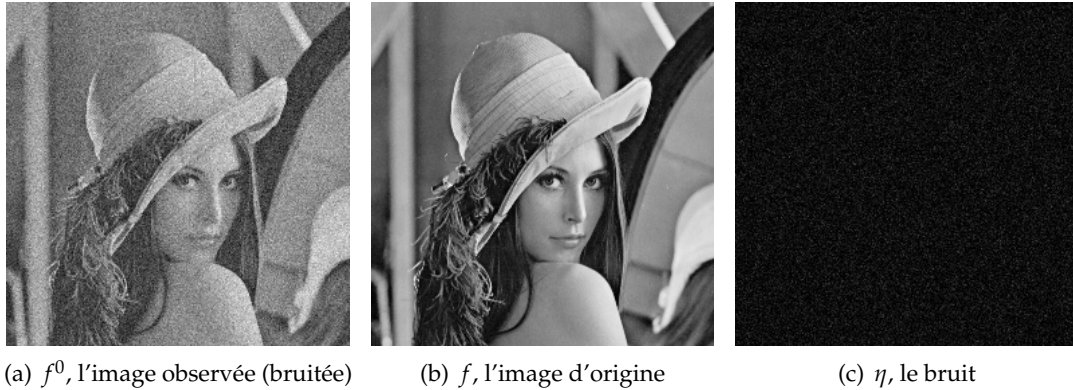


FIGURE 1.1 – Débruiter l'image  $f^0$  1.1(a) consiste à retrouver l'image  $f$  1.1(b) en supprimant le bruit  $\eta$  1.1(c).

La restauration de la fonction  $f^0$  consiste à retrouver une estimation de  $f$  qui représente les données utiles. Les méthodes de restauration d'images sont directement liées aux types de dégradations qu'elles tentent de corriger. Les informations *a priori* comme les défauts de capteur ou les informations *a posteriori* déduites de l'analyse de l'image dégradée sont utilisées pour modéliser au mieux la dégradation.

Une approche classique pour résoudre ce problème utilise la méthode des moindres carrés et consiste à minimiser la fonctionnelle :

$$J(f) = \int_{\Omega} |f_0 - \mathcal{R}f|^2 dx. \quad (1.2)$$

Cependant, l'opérateur  $\mathcal{R}$  n'est pas nécessairement inversible et même lorsqu'il l'est (c'est à dire la transformée de Fourier  $\widehat{\mathcal{R}}$  de  $\mathcal{R}$  ne s'annule pas), la restauration de  $u$  par déconvolution brutale ( $\widehat{\mathcal{R}^{-1}f_0}$ ) est inefficace à cause du terme de bruit. En effet, dans le domaine de Fourier, on a  $\widehat{\mathcal{R}^{-1}f_0} = \widehat{u} + \frac{\widehat{\eta}}{\widehat{\mathcal{R}}}$  et le deuxième terme « explose » aux points où  $\widehat{\mathcal{R}}$  devient très petit, ce qui produit en général des oscillations démesurées qui noient le reste de l'image. Il s'agit d'un problème inverse mal posé. Pour résoudre ce problème, une certaine forme de régularisation raisonnable est nécessaire. Il est désirable que le choix de la régularisation soit fait pour que les bords dans  $f^0$  soient maintenus dans  $f$ .

## 1.2 Régularisation des images par minimisation d'énergies

Il est classique d'aborder le problème inverse par la méthode de régularisation. Il s'agit de minimiser une énergie de deux termes, l'un de régularité  $E_{\text{lissage}}$  et l'autre d'attache de données  $E_{\text{attache}}$  :

$$J(f, f^0, \lambda) = E_{\text{lissage}}(f) + \lambda E_{\text{attache}}(f^0, f). \quad (1.3)$$

$E_{\text{lissage}}$  est un terme de régularisation représentant l'information *a priori* sur la fonction à traiter.  $E_{\text{attache}}$  est un terme de fidélité aux données. Le rapport de force entre ces deux termes est contrôlé par la variable  $\lambda$ . Les différentes méthodes de régularisation diffèrent par le choix des termes  $E_{\text{lissage}}$  et  $E_{\text{attache}}$ .

### 1.2.1 Régularisation de Tikhonov

Parmi les modèles linéaires, mentionnons la *régularisation de Tikhonov* [Tikhonov et Arsenin, 1977]. C'est l'une des méthodes de régularisation les plus utilisées pour résoudre ce type de problèmes mal posés. Elle consiste à minimiser l'énergie

$$J_2(f) = \underbrace{\int_{\Omega} \|\nabla f\|^2 dx}_{E_{\text{lissage}}} + \lambda \underbrace{\int_{\Omega} |f_0 - \mathcal{R}f|^2 dx}_{E_{\text{attache}}}. \quad (1.4)$$

Le premier terme est le terme de régularisation. Le deuxième terme est le terme de moindres carrés d'attache aux données. Le paramètre  $\lambda$  sert à contrôler le processus de régularisation. Un élément clef du problème de minimisation de Tikhonov est le choix approprié du paramètre de régularisation  $\lambda$  : Si  $\lambda$  est choisi trop petit, la reconstruction sera trop proche de celle obtenue dans l'équation (1.2) et ainsi dominée par des composantes à hautes fréquences. En revanche, si  $\lambda$  est choisi trop grand, le terme en  $\lambda$  dominera la solution et les informations importantes dans les données ne seront pas prises en compte.

Une méthode classique de résolution de ce type de problème est de considérer la fonction  $f$  comme solution de l'équation d'Euler-Lagrange

$$\Delta f + \lambda \mathcal{R}^*(\mathcal{R}f - f^0) = 0. \quad (1.5)$$

### 1.2.2 Régularisation VT

Il existe un autre modèle de régularisation classique qui est la variation totale (VT). Son intérêt majeur réside dans le fait qu'elle préserve les discontinuités qui peuvent se voir comme les contours. Grâce à cette qualité, la régularisation VT est largement utilisée dans de nombreuses applications en traitement d'images, telles que la déconvolution aveugle, l'inpainting et la super-résolution. Le terme de régularité dans (1.4) favorise fortement les images très lisses. En particulier, ce terme n'autorise pas les discontinuités. En 1992, Rudin et al. [Rudin et al., 1992] ont proposé d'utiliser, à la place de la norme du gradient dans  $\mathcal{L}_2(\Omega)$ , la variation totale de l'image  $\int_{\Omega} |\nabla f|$  afin de préserver les contours. Ils posent le problème de minimisation :

$$\min \left\{ \int_{\Omega} \|\nabla f\| \right\} \quad (1.6)$$

sous les contraintes

$$\int_{\Omega} |\mathcal{R}f| = \int_{\Omega} f^0 \quad \text{et} \quad \int_{\Omega} |\mathcal{R}f - f^0|^2 = \sigma^2.$$

La première condition exprime le fait que le bruit est à moyenne nulle et la deuxième correspond à la variance  $\sigma^2$  du bruit. L'intérêt d'un tel choix pour la fonction de régularisation provient du fait que la variation totale en dimension 2 autorise les discontinuités sur des courbes, et donc en particulier la présence des bords dans l'image restaurée. En introduisant un multiplicateur de Lagrange  $\lambda > 0$ , [Chambolle et Lions, 1997] ont démontré que le problème (1.6) est équivalent au problème suivant :

$$J_{VT}(f) = \inf_f \int_{\Omega} |\nabla f| + \frac{\lambda}{2} \int_{\Omega} |f_0 - \mathcal{R}f|^2 dx. \quad (1.7)$$

Sous certaines hypothèses, le problème de minimisation (1.7) possède une solution unique. On trouve dans la littérature de nombreuses méthodes rapides de minimisation dont l'une consiste à considérer l'équation d'évolution :

$$\begin{cases} \frac{\partial f}{\partial t} = \operatorname{div} \left( \frac{\nabla f}{\|\nabla f\|} \right) + \lambda \mathcal{R}^*(\mathcal{R}f - f^0) = 0 & \text{sur } \Omega \\ f = f^0 & \text{sur } \Omega \end{cases} \quad (1.8)$$

*Remarque 1.* Lorsque  $\mathcal{R} = I$  (la fonction identité), on retrouve le problème de débruitage.

Ces méthodes linéaires sont très simples et rapides à implémenter. Cependant, ces méthodes basées sur la VT ne préservent pas les structures fines, les détails et les textures.

### 1.2.3 Régularisation $p$ -VT

Il s'agit d'un modèle étendant la régularisation linéaire et la variation totale. Un paramètre  $p$  est introduit à la définition de la variation totale. Il en découle une nouvelle fonctionnelle à minimiser :

$$J(f, f^0) = \frac{1}{p} \inf_f \int_{\Omega} \|\nabla f\|^p + \frac{\lambda}{2} \int_{\Omega} \|f_0 - \mathcal{R}f\|^2 dx, \quad (1.9)$$

où  $\|\nabla f\|$  est la norme du gradient de la fonction  $f$  qui reflète la régularité locale.

Une façon simple de résoudre ce problème est de considérer la fonction d'évolution :

$$\begin{cases} \frac{\partial f}{\partial t} = \operatorname{div} \left( \|\nabla f\|^{p-2} \nabla f \right) + \lambda \mathcal{R}^*(\mathcal{R}f - f^0) = 0 & \text{sur } \Omega \\ f = f^0 & \text{sur } \Omega \end{cases} \quad (1.10)$$

*Remarque 2.* Ce modèle permet de retrouver plusieurs méthodes connues dans la littérature du traitement d'images. Lorsque  $p = 2$ , le problème de minimisation de la fonctionnelle (1.9) correspond à la régularisation de Tikhonov. Pour  $p = 1$ , nous retrouvons la régularisation basée sur la variation totale. Le cas  $p < 1$  a été également utilisé pour la simplification des contours.

Une large famille de modèles variationnels ont utilisé cette fonctionnelle et ses dérivées. Cependant, ces opérateurs sont locaux. Ils ne tiennent donc pas compte des caractéristiques non-locales des données telles que les structures répétitives (comme la texture par exemple).

## 1.2.4 Régularisation continue non locale

### 1.2.4.1 Filtrage non local

Les méthodes non locales en traitement d'images sont généralisées à partir du filtre Yaroslavsky [Yaroslavsky, 1985] et les méthodes à base de patches ont été proposées, à l'origine, pour la synthèse de textures. L'idée est de restaurer un pixel inconnu en utilisant d'autres pixels similaires. La ressemblance est considérée sur un patch centré sur chaque pixel, et pas seulement sur la valeur du pixel lui-même. Afin de débruiter un pixel, il est préférable de moyenner les pixels voisins ayant des structures similaires (patches). Cette idée a donné naissance à un célèbre filtre à voisinage qui a inspiré de nombreux travaux par la suite, le *NL-means* :

$$NL_f(x) = \frac{1}{C(x)} \int_{\Omega} w_f(x, y) f(y) dy, \quad (1.11)$$



où  $f$  est l'image de référence,  $w_f(x, y)$  la fonction de poids et  $C(x)$  est le facteur de normalisation :

$$w_f(x, y) = \exp \left( - \frac{\left( G_a |f(x + \cdot) - f(y + \cdot)|^2 \right) (0)}{h^2} \right) \quad (1.12)$$

et

$$C(x) = \int_{\Omega} w_f(x, y) dy,$$

$G_a$  est le noyau gaussien avec un écart-type  $a$  et  $h$  est un paramètre de filtrage. En général,  $h$  correspond au niveau de bruit. Habituellement,  $h$  est mis à l'écart-type du bruit. Les poids sont significatifs si et seulement si la fenêtre autour de  $y$  ressemble à la fenêtre autour de  $x$ . Ainsi, l'autosimilarité est employée pour réduire le bruit. Les auteurs [Buades et al., 2005b] prennent en compte les données bruitées  $f$  comme image de référence pour construire le poids. Ainsi, ce moyennage pondéré renforce les structures telles que les contours tandis que le bruit est réduit.

#### 1.2.4.2 Énergies de régularisation

Ce filtre peut être interprété dans le cadre de la régularisation non locale. Dans ce cas, il s'agit de remplacer le gradient local par un gradient non local. Nous obtenons :

$$J(f, f^0, \lambda) = E_{\text{lissage}}(f) + \lambda E_{NL}(f, f^0). \quad (1.13)$$

La fonctionnelle de régularisation devient :

$$E_{NL}(f, p) = \frac{1}{2p} \int_{\Omega \times \Omega} w(x, y)^{p/2} |f(y) - f(x)|^p dy dx, \quad (1.14)$$

où la définition du *gradient non local*  $\nabla_w f$  est :  $\Omega \rightarrow \Omega \times \Omega$

$$(\nabla_w f)(x, y) = (f(y) - f(x)) \sqrt{w(x, y)}. \quad (1.15)$$

Pour  $p = 2$ , on retrouve la fonctionnelle non locale isotrope proposée par Gilboa et Osher [Gilboa et Osher, 2007] :

$$E_{NL}(f, 2) = \frac{1}{4} \int_{\Omega \times \Omega} w(x, y) (f(y) - f(x))^2 dy dx. \quad (1.16)$$

Gilboa et Osher ont proposé cette fonctionnelle non locale, qui est en fait une régularisation linéaire, pour le débruitage et la segmentation supervisée.

Pour  $p = 1$ , on obtient la fonctionnelle non locale anisotrope proposée par Gilboa et Osher dans un cadre systématique et cohérent des opérateurs non locaux [Gilboa et Osher, 2009] :

$$E_{NL}(f, 1) = \frac{1}{2} \int_{\Omega \times \Omega} \sqrt{w(x, y)} |f(y) - f(x)| dy dx. \quad (1.17)$$

### 1.2.5 Régularisation discrète

Certains travaux récents ont proposé un cadre de régularisation discrète rapprochant et unifiant certaines de ces méthodes locales et non locales. En particulier, Pizarro et al. [Pizarro et al., 2010] ont formulé le modèle unifié par une fonctionnelle d'énergie avec des données non locales et des termes de lissage. Le terme des données récompense la similitude de l'image filtrée à l'image d'entrée (bruitée), tandis que le terme de lissage pénalise l'importance des écarts par rapport à la régularité de la solution.

Ces termes offrent la possibilité d'envisager aussi bien des informations locales à partir d'une petite région autour du pixel que des informations impliquant des zones de voisinage étendues. Pizarro et al. ont montré que les méthodes classiques sont des cas particuliers, qui peuvent être surpassées en ajustant l'étendue spatiale où les interactions non locales des pixels peuvent avoir lieu. Ces interactions prennent la forme de différences d'intensité non linéaires mesurant la similarité inter-pixels de façon plus précise que les simples différences entre pixels.

Elmoataz et al. [Elmoataz et al., 2008] ont proposé un cadre de régularisation non locale plus général sur graphes qui est le pendant *discret* de l'approche proposée par Gilboa [Gilboa et Osher, 2007]. Ce cadre est basé sur l'introduction d'opérateurs différentiels non locaux qui permet l'adaptation de la régularisation aux graphes et la réunion, dans un même cadre, du traitement de différents types de données. Ainsi, des fonctionnelles discrètes de régularisation ont été proposées :

$$E_{NL}^i(f) = \frac{1}{2p} \sum_{\Omega} \|\nabla_w f\|_p^p \quad \text{et} \quad E_{NL}^a(f) = \frac{1}{2p} \sum_{\Omega} \|\nabla_w f\|_2^p \quad (1.18)$$

où  $\nabla_w f$  est une famille de gradients discrets.

Par ailleurs, grâce à ce cadre, nous obtenons l'unification des méthodes locales et non locales pour  $1 \leq p < \infty$ . Nous présentons un peu plus en détails ce cadre, sur lequel notre contribution se base, dans le chapitre suivant (chapitre 2).

Si la modélisation du problème de débruitage est commune à l'image et à la vidéo, les difficultés rencontrées ne sont pas les mêmes. La plus grande différence entre l'image et la vidéo est l'intégration du temps. C'est pourquoi le mouvement est la difficulté majeure dans le débruitage des vidéos. Nous examinons dans la section 1.3 les méthodes proposées pour le débruitage des vidéos.



## 1.3 Régularisation des séquences d'images

Dans cette section, nous présentons le problème de débruitage des vidéos et un aperçu de l'état de l'art dans ce domaine. La première idée qui viendrait à l'esprit pour débruiter une séquence d'images serait de traiter toutes les images formant la séquence, une par une, en utilisant des méthodes comme celles que nous avons présentées dans la section 1.2. Cette idée est certainement valable mais présente plusieurs inconvénients. D'abord, elle ne tient pas compte des dégradations spécifiques à la vidéo telles que le bougé, la résolution ou encore le zoom. Par ailleurs, elle contraint le traitement à un espace 2D alors que nous disposons d'une dimension supplémentaire, *le temps*. Par conséquent, il est plus avantageux de concevoir un traitement 2D+t directement et de bénéficier des avantages que cela présente, notamment la redondance temporelle.

Dans un premier temps, nous citons les dégradations les plus répandues et leurs sources. Puis, nous passons en revue différentes méthodes proposées pour le débruitage des vidéos. Le mouvement représente la principale difficulté du débruitage des séquences d'images. Par conséquent, il est judicieux de regrouper les méthodes selon le traitement réservé au mouvement.

### Introduction

Tout d'abord, il est important de rappeler que certaines dégradations sont propres aux séquences d'images. Brièvement, nous rappelons la classification des nombreuses dégradations possibles des séquences d'images proposée par De Haan [de Haan, 2003] :

- *dégradations spatiales* : dues aux dispositifs d'acquisition, à la défocalisation et aux limitations de bande passante en électronique. Les effets indésirables qui en proviennent sont souvent le flou, l'écho, etc.
- *dégradations temporelles* : dues au temps d'exposition du matériel photosensible. Ces dégradations incluent le flou de mouvement, le scintillement, les artefacts temporels, etc.
- *dégradations géométriques* : géométrie déformée, dues aux aberrations de lentilles.
- *dégradations ponctuelles* : modifient le niveau de gris sans impact sur le détail ou le mouvement. Ces dégradations sont connues sous le nom de « bruit » qui est une sorte de processus aléatoire dépendant de la source de la dégradation.
- *dégradations chromatiques* : impliquent le changement de teintes.

De nombreuses méthodes ont été proposées pour le débruitage de séquences vidéo. Comme nous l'avons souligné, la prise en compte du mouvement est un enjeu fondamental pour la réduction du bruit en vidéo. Les méthodes proposées peuvent être judicieusement classées selon leur prise en compte du mouvement. Pour ce faire, certaines méthodes impliquent une phase préliminaire d'estimation de mouvement qui est suivie par un processus de réduction de bruit approprié, tandis que d'autres intègrent l'estimation de mouvement dans l'algorithme de débruitage avec ou sans contraintes de lissage. Toutefois, les travaux récents de Buades et al. [Buades et al., 2005a, Buades et al., 2005b] ont démontré que la résolution de l'estimation de mouvement ne peut être conseillée pour le problème de débruitage vidéo.

Les méthodes de la restauration vidéo forment trois familles. La première se base sur l'estimation du mouvement et filtre les trajectoires. La deuxième famille compense le mouvement par une estimation du flot optique, puis filtre les images résultant de la compensation du mouvement. La troisième famille tourne en sa faveur la difficulté du problème bien connu d'ouverture et en tire avantage en évitant l'estimation du mouvement pour le débruitage.

En effet, par le problème d'ouverture, de nombreux pixels dans les images voisines sont similaires au pixel courant à débruiter. Ainsi, la réduction du bruit par un processus de calcul de la moyenne peut utiliser beaucoup plus de pixels que ceux sur une trajectoire unique. Cette observation conduit à utiliser l'algorithme NL-means pour filtrer les séquences d'images. Le NL-means surpasse les algorithmes compensant le mouvement car il implique les isophotes<sup>2</sup> de la séquence entière, y compris l'image courante, et pas seulement une trajectoire.

Dans les méthodes locales de filtrage, la restauration d'un pixel  $x$  ne dépend que de la valeur de ses pixels voisins. L'ensemble des pixels impliqués dans la restauration de  $x$  est appelé le support de  $x$ , et noté  $B(x)$ . Lorsque le support  $B(x)$  n'implique que les pixels appartenant à une même image, nous parlons de filtres spatiaux. Dans ce cas, chaque image de la séquence est filtrée indépendamment du reste. Une telle approche néglige la corrélation temporelle entre images consécutives et n'exploite pas l'importante redondance des séquences d'images. Les filtres temporels exploitent uniquement la corrélation temporelle. Enfin, les filtres spatio-temporels exploitent les deux types de corrélations, spatiales et temporelles. Dans ce cas, le support d'un pixel est constitué de pixels de l'image actuelle, mais aussi des images précédentes et suivantes. Ce fait conduit à un filtrage plus cohérent de la séquence.

Nous présentons ici, les trois catégories de filtres spatio-temporels :

1. Les *filtres statiques* sont des extensions simples de filtres sur l'image. Ces filtres sont obtenus en étendant le support des images 2D à un support 3D espace-temps. Alors même qu'ils sont les plus simples des algorithmes, ils sont indispensables à la compréhension du problème de filtrage de séquences.
2. Les *filtres adaptatifs* sont conçus spécifiquement pour s'accommoder aux séquences d'images et de prendre en compte la possibilité d'un mouvement.
3. Les *filtres à compensation de mouvement* estiment explicitement le mouvement de la séquence à l'aide d'un algorithme d'estimation du mouvement. Les trajectoires obtenues par la précédente estimation fournissent de nouvelles données statiques.

Les travaux les plus récents utilisent la stratégie de compensation de mouvement. Cela semble une condition nécessaire pour obtenir de bons résultats de filtrage de séquences. Toutefois, l'estimation de mouvement est un problème très difficile en soi et il n'existe aucun algorithme capable de donner une solution unique. En fait, la compensation de mouvement peut proposer des trajectoires imprécises impliquant un filtrage flou et une perte d'information. L'objectif de ce chapitre n'est pas de faire une revue complète des filtres de séquences d'images. Pour une description plus complète, voir [Brailean et al., 1995].

Nous soutenons la position selon laquelle l'estimation de mouvement n'est pas indispensable. Le problème d'ouverture, considéré comme un phénomène général dans les vidéos, peut être interprété de façon positive comme suit : il y a des pixels dans l'image

---

2. lignes de même niveau de gris

suivante ou précédente, qui peuvent correspondre au pixel courant. Par la loi des grands nombres, une méthode de débruitage œuvre mieux si l'estimation du pixel courant est basée sur un nombre maximum de pixels. Ainsi, il semble judicieux d'utiliser tous les pixels similaires au pixel courant à travers le temps et l'espace et ne pas se limiter à une seule trajectoire. Cela nous ramène aux filtres de sigma ou de voisinage [Lee, 1983, Yaroslavsky, 1985]. Rappelons que l'idée principale de ce filtre spatial est de débruiter le pixel courant en prenant une moyenne de tous les pixels voisins dont les niveaux de gris sont assez proches. Si l'on applique le filtre sigma sur les données spatio-temporelles, nous moyennons sur les isophotes de tout l'espace-temps.

### 1.3.1 Méthodes de débruitage locales

Généralement, les méthodes de débruitage locales utilisent l'estimation de mouvement. Dans [Protter et Elad, 2009] on trouve une comparaison entre plusieurs méthodes de la littérature telles que [Buades *et al.*, 2005a, Zlokolica *et al.*, 2006, Boulanger *et al.*, 2007, Dabov *et al.*, 2007]. Dans la méthode décrite dans [Zlokolica *et al.*, 2006], les filtres estimant le mouvement et les filtres temporels adaptatifs (le long des trajectoires estimées) sont appliqués de manière récursive, suivi d'un filtre spatial adaptatif intra-image. Deux types de mesures de fiabilité de mouvement sont estimés. La première est une mesure de fiabilité par orientation, appliquée à l'étape de l'estimation de mouvement. L'autre est une mesure de fiabilité par bandes d'ondelettes. Le filtrage spatial subséquent est conçu pour avoir un effet accru où le filtrage temporel avait été moins efficace en raison de la faible fiabilité.

### 1.3.2 Méthodes de débruitage non locales

L'algorithme NL-means ([Buades *et al.*, 2005a] et [Buades *et al.*, 2005b]) adopte une approche alternative au problème de l'estimation du mouvement. Tous les patches du voisinage 3D sont considérés et leurs pixels centraux sont moyennés. Chaque patch est pondéré en fonction de sa similitude avec le patch central, rendant ainsi l'estimation de mouvement transparente et implicite. Au lieu de calculer un vecteur de mouvement unique pour chaque patch, plusieurs vecteurs possibles sont autorisés à coexister, chacun avec une probabilité différente. Cette approche se concentre sur la fusion d'estimations bruitées plus que l'obtention d'informations précises sur l'estimation du mouvement.

[Buades *et al.*, 2005a, Buades *et al.*, 2008] ont travaillé sur le filtrage non local et ont démontré comment le problème d'ouverture peut être mis à profit pour le débruitage vidéo. Une variante de filtrage non local a été présentée [Boulanger *et al.*, 2007] en utilisant des patches de tailles variables. La méthode décrite étend l'approche des NL-means. Les mesures statistiques sont utilisées pour une sélection adaptative optimale de la taille du voisinage pour chaque pixel. En outre, le calcul du poids attribué à chaque pixel dans le voisinage est effectué de manière adaptative. Cet outil puissant entraîne l'amélioration de la performance de débruitage par rapport à l'approche originale du NL-means.

L'algorithme VBM3D ([Dabov *et al.*, 2007]) utilise également une multitude de patches dans le voisinage 3D de chaque pixel pour atténuer le bruit. Les patches les plus semblables dans le voisinage sont collectés et empilés dans un tableau 3D. Une transformée par ondelette 3D est ensuite appliquée avec un seuillage utilisé pour la suppression du bruit. Après l'application de la transformée inverse, les patches sont retournés à leur emplacement d'origine, et moyennés. Une deuxième itération suit, avec un filtrage de Wiener pour améliorer les résultats de débruitage.

Protter et Elad [Protter et Elad, 2007] ont proposé une approche de débruitage des séquences d'images basée sur la diffusion d'un dictionnaire. Dans [Protter *et al.*, 2009], la mé-

thode des NL-means est étendue à la reconstruction de super-résolution. Ces filtres non locaux peuvent être considérés comme une régularisation basée sur des fonctionnelles non locales. Kindermann et al. [Kindermann *et al.*, 2005] ont été les premiers à interpréter les NL-means et les filtres de voisinage comme une régularisation basée sur des fonctionnelles non locales. Plus tard, Gilboa et Osher [Gilboa et Osher, 2007] ont proposé une fonctionnelle quadratique de différences pondérées pour la régularisation d'images et la segmentation semi-supervisée. Elmoataz et al. [Elmoataz *et al.*, 2008] ont récemment mis en place un cadre de régularisation discret non local, qui est l'analogue discret des fonctionnelles continues, euclidiennes et non locales de [Gilboa et Osher, 2007]. Cette méthode est applicable sur des données diverses telles que les images, les vidéos et les maillages à l'aide des graphes pondérés de topologies arbitraires.

Dans le chapitre suivant, nous présentons l'élargissement de ce cadre pour le débruitage et la simplification d'images et de vidéos.

# CONTRIBUTIONS À LA RÉGULARISATION DES IMAGES ET DES VIDÉOS

## Sommaire

|  |           |
|--|-----------|
| <b>2.1 Modèles variationnels avec les <math>p</math>-Laplaciens isotrope et anisotrope</b> . . . | <b>15</b> |
| 2.1.1 Définition et construction du graphe pondéré . . . . .                                     | 15        |
| 2.1.2 Opérateurs différentiels sur graphe . . . . .  | 17        |
| 2.1.3 Formulation mathématique et approche proposée . . . . .                                    | 20        |
| 2.1.4 Liens avec les méthodes classiques . . . . .   | 22        |
| <b>2.2 Régularisation par <math>\infty</math>-Laplacien</b> . . . . .                            | <b>24</b> |
| 2.2.1 $\infty$ -Laplacien sur graphes . . . . .  | 25        |
| 2.2.2 Nouvelle interprétation et résolution . . . . .  | 27        |
| <b>2.3 Résultats et performances</b> . . . . .   | <b>28</b> |
| 2.3.1 Débruitage . . . . .   | 28        |
| 2.3.2 Simplification de vidéos . . . . .   | 36        |

## Résumé

Notre motivation pour ce chapitre est de transcrire les méthodes variationnelles au domaine discret sur graphe. Notre apport réside principalement en deux points. D'une part, nous proposons l'extension du cadre de régularisation discrète [Elmoataz *et al.*, 2008] aux séquences d'images. Nous avons mis en place des algorithmes de régularisation sur graphes par  $p$ -Laplacien, avec  $p \in ]0, +\infty[$ , pour le débruitage et la simplification des séquences d'images. D'autre part, nous introduisons un algorithme nouveau de  $p$ -régularisation, avec  $p = \infty$ , qui sera comparé aux algorithmes proposés pour  $1 \leq p < \infty$ .

## 2.1 Modèles variationnels avec les $p$ -Laplaciens isotrope et anisotrope

Dans cette section, nous proposons deux modèles variationnels basés sur le  $p$ -Laplacien isotrope et le  $p$ -Laplacien anisotrope, avec  $p \in ]0, +\infty[$ , pour la régularisation des séquences d'images. Le cas particulier de  $p = \infty$  fera l'objet de la section 2.2.

### 2.1.1 Définition et construction du graphe pondéré

Dans notre travail, toutes les données traitées sont discrètes. Nous avons retenu la représentation par graphe pondéré car elle facilite la transcription de nos algorithmes à

différents types de données. Ci-après, nous exposons la représentation des données sur graphes que nous utilisons.

### 2.1.1.1 Définition du graphe

Soit  $G = (V, E, w)$  un graphe pondéré composé d'un ensemble fini  $V = \{v_1, v_2, \dots, v_n\}$  de  $n$  sommets, un ensemble fini  $E \subset V \times V$  d'arêtes, et d'une fonction poids  $w : V \times V \rightarrow \mathbb{R}^+$ . Un graphe est pondéré si on l'associe à une fonction poids  $w : V \times V \rightarrow \mathbb{R}^+$ , qui satisfait aux conditions suivantes :

$$\begin{aligned} w(u, v) &= w(v, u) & \forall (u, v) \in E \\ w(u, v) &> 0 & \text{si } u \neq v \\ w(u, u) &= 0 & \text{sinon.} \end{aligned} \quad (2.2)$$

La fonction de poids  $w$  encode les informations locales ou non locales. Elle mesure la dissemblance entre les sommets du graphe. Quand  $w(u, v)$  tend vers 0, les sommets  $u$  et  $v$  sont dissemblables. Inversement,  $w(u, v)$  croît et tend vers 1 lorsque la ressemblance est grande.  $G$  est supposé non orienté, connexe, acyclique, et sans arêtes multiples (pour chaque arête  $(u, v) \in E, (v, u) \in E$ ).

Soit  $\mathcal{H}(V)$  un espace de Hilbert des fonctions à valeurs réelles sur les sommets du graphe  $G = (V, E, w)$ . Une fonction  $f : V \rightarrow \mathbb{R}^n$  dans  $\mathcal{H}(V)$  attribue un vecteur  $f(v)$  à chaque sommet  $v \in V$ . La fonction  $f$  forme un espace fini de dimension  $n$ . Elle peut être considérée comme un vecteur colonne  $\mathbf{f} = [f(v_1), \dots, f(v_N)]^T$ . Par analogie avec l'analyse fonctionnelle dans les espaces continus, l'intégration de  $f$  sur le graphe, est notée  $\int_V f = \sum_V f$ .

De même, soit  $\mathcal{H}(E)$  l'espace de Hilbert des fonctions à valeurs réelles  $F : E \rightarrow \mathbb{R}^n$ , définies sur les sommets du graphe. Ces deux espaces sont munis des produits scalaires habituels :

$$\langle f, g \rangle_{\mathcal{H}}(V) = \sum_{u \in V} f(u)g(u) \quad (2.3)$$

$$\langle F, G \rangle_{\mathcal{H}}(E) = \sum_{u \in V} \sum_{u \sim v} F(u, v)G(u, v) \quad (2.4)$$

où,  $f, g : V \rightarrow \mathbb{R}^n$  et  $F, G : E \rightarrow \mathbb{R}^n$ .

### 2.1.1.2 Construction du graphe

Nous nous intéressons dans le travail présent aux images et aux séquences d'images uniquement. Pour simplifier la lecture, nous considérons par la suite qu'une image est une vidéo de profondeur égale à 1. Nous considérons une séquence vidéo comme une fonction  $f$  définie sur les sommets d'un graphe pondéré  $G_{k_1, k_2, k_3} = (V, E, w)$ , où  $k_1, k_2, k_3 \in \mathbb{N}^3$ . Un sommet  $v$  est défini par un triplet  $(i, j, t)$  où  $(i, j)$  indique la position spatiale du sommet et  $t$ , qui est le numéro de l'image, précise la position temporelle du sommet dans la séquence. Nous désignons par  $u \sim v$  un sommet  $u$  qui appartient au voisinage de  $v$  selon la définition suivante :

$$N_{k_1, k_2, k_3}(v) = \{u = (i', j', t') \in V : |i - i'| \leq k_1, |j - j'| \leq k_2, |t - t'| \leq k_3\}. \quad (2.5)$$

De même, nous étendons la définition des patchs 2D aux vidéos pour obtenir des patchs 3D. Un patch autour d'un sommet  $v$  est une boîte de taille  $r_x \times r_y \times r_t$ , notée  $B(v)$ . Puis, nous associons à ce patch un vecteur caractéristique défini par :

$$F(f^0, v) = f^0(u), \quad u \in B(v).$$

Pour être sûr que chaque voisinage contient plus d'un patch, les relations suivantes doivent être respectées :  $k_1 > \alpha * r_x$ ,  $k_2 > \alpha * r_y$  et  $k_3 \geq r_t$ . En pratique, nous utilisons  $\alpha = 3$ . Il est intéressant de noter que  $r_x$ ,  $r_y$  et  $r_t$  doivent être assez petits par rapport à  $k_1$ ,  $k_2$  et  $k_3$  pour être sûr que le voisinage  $N_{k_1, k_2, k_3}$  contient un nombre significatif de patches.

La fonction poids  $w$  associée à  $G_{k_1, k_2, k_3}$  fournit une mesure de la distance inter-sommets qui peut simplement incorporer des caractéristiques locales, semi-locales ou non locales en fonction de la topologie du graphe. Nous considérons les deux fonctions de poids suivantes. Pour tout  $u, v \in V \times V$  :

$$w_L(u, v) = \exp\left(-\frac{|f(u) - f(v)|^2}{2\sigma_d^2}\right) \quad (2.6)$$

$$w_{NL}(u, v) = w_L(u, v) \cdot \exp\left(-\frac{\|F(f^0, u) - F(f^0, v)\|^2}{h^2}\right), \quad (2.7)$$

où  $\sigma_d^2$  dépend des variations de  $|f(u) - f(v)|$  sur le graphe.  $h$  peut être estimé en utilisant l'écart-type en fonction des variations de  $\|F(f^0, u) - F(f^0, v)\|$  sur le graphe.

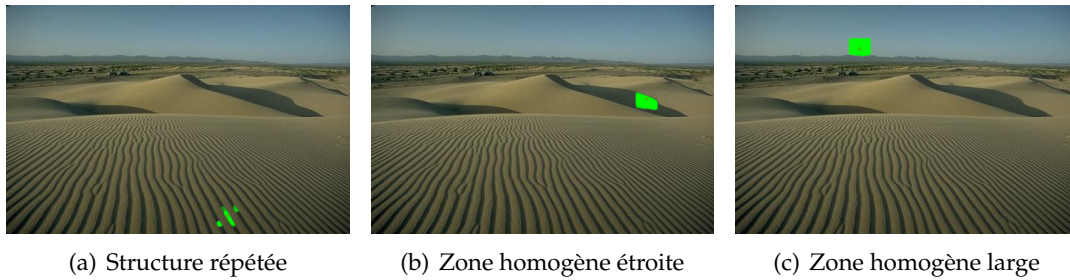


FIGURE 2.1 – Mise en valeur de la pertinence du poids  $w_{NL}$  entre patches dans différentes situations. Les points (en vert) représentent les patches ayant un poids important (supérieur à 0,5) avec le patch central traité. Nous observons que la forme des contours et la texture de la zone traitée sont respectées. Paramètres : Fenêtre  $31 \times 31$  et patch  $5 \times 5$ .

$w_L(u, v)$  est une mesure de la différence entre les valeurs de  $f(u)$  et  $f(v)$ , et est utilisée dans l'approche locale du débruitage. En plus de la différence entre les valeurs,  $w_{NL}(u, v)$  inclut une estimation de la similitude des caractéristiques comparées en mesurant une distance  $\mathcal{L}_2$  entre les patches autour de  $u$  et  $v$ . C'est le principe de l'approche non locale. La figure 2.1 illustre la discrimination des différents patches dans la fenêtre. On voit clairement que pour les zones homogènes, la plupart des patches voisins sont très similaires. Par conséquent, un poids important est attribué à un nombre important de candidats (figures 2.1(b) et 2.1(c)). Au contraire, lorsque le patch est plus « rare » dans la fenêtre de recherche, il y a très peu de voisins ayant un poids important (figure 2.1(a)).

Par la suite, nous utiliserons  $w_{NL}$  car elle est adaptée aux textures. Nous présenterons  $w_L$  uniquement à titre de comparaison pour mieux apprécier le résultat non local.

### 2.1.2 Opérateurs différentiels sur graphe

Ci-après, nous rappelons succinctement le cadre discret ainsi que les outils de travail issus des travaux de Elmoataz et al. [Elmoataz et al., 2008] sur lesquels nous nous basons. Les lecteurs peuvent s'y référer pour plus de détails.



### 2.1.2.1 Différence discrète, dérivée et adjoint

Considérons maintenant les opérateurs de dérivés sur les graphes dont nous avons besoin pour la régularisation.

L'opérateur de *différence* d'une fonction  $f: V \rightarrow \mathbb{R}$ , noté  $d: \mathcal{H}(V) \rightarrow \mathcal{H}(E)$ , est défini sur une arête  $(u, v) \in E$  par

$$(df)(u, v) = \frac{f(v) - f(u)}{\mu_w(u, v)},$$

où  $\mu_w: V \times V \rightarrow \mathbb{R}^+$  est une mesure de similarité qui dépend de la fonction de poids  $w$ . Dans la suite, nous nous limitons à  $\mu_w(u, v) = 1/\sqrt{w(u, v)}$ . Cela induit la définition suivante de l'opérateur de différence

$$(df)(u, v) = \sqrt{w(u, v)} (f(v) - f(u)). \quad (2.8)$$

La *dérivée directionnelle* de la fonction  $f$ , à un sommet  $u$ , et sur une arête  $(u, v)$ , est définie par

$$\partial_v f(u) = (df)(u, v) = \sqrt{w(u, v)} (f(v) - f(u)). \quad (2.9)$$

L'opérateur de différence et la dérivée directionnelle sont antisymétriques, i.e.

$$(df)(u, v) = -(df)(v, u).$$

Ils partagent également la propriété suivante avec la définition continue de la dérivée d'une fonction définie dans l'espace euclidien :

$$f(u) = f(v) \Rightarrow (df)(u, v) = 0. \quad (2.10)$$

L'*opérateur adjoint* de l'opérateur de différence, noté  $d^*: \mathcal{H}(E) \rightarrow \mathcal{H}(V)$ , est défini par

$$\langle df, H \rangle_{\mathcal{H}(E)} = \langle f, d^* H \rangle_{\mathcal{H}(V)}, \forall f \in \mathcal{H}(V), H \in \mathcal{H}(E).$$

Puis, à partir des expressions des produits scalaires dans  $\mathcal{H}(E)$  et  $\mathcal{H}(V)$ , nous pouvons en déduire l'expression de l'opérateur adjoint à un sommet  $u \in V$  :

$$d^* H(u) = \sum_{v \in V} \sqrt{w(u, v)} (H(v, u) - H(u, v)). \quad (2.11)$$

L'adjoint est un opérateur linéaire qui mesure le flux de  $H$  sur le graphe. Par analogie avec les opérateurs différentiels en continu, l'opérateur de divergence de la fonction  $H$  est défini par  $\text{div} H = -d^* H$ . Ensuite, on peut montrer que toute fonction  $H \in \mathcal{H}(E)$  a un flot nul ( théorème de la divergence ) :

$$\sum_{u \in V} (\text{div} H)(u) = \sum_{u \in V} \sum_{v \in V} \sqrt{w(u, v)} (H(u, v) - H(v, u)) = 0. \quad (2.12)$$

*Remarque 3.* Il existe d'autres expressions de l'opérateur de différence en fonction du contexte. Par exemple, l'opérateur de différence défini dans [Zhou et Schölkopf, 2004, Zhou et Schölkopf, 2005] est formulé par

$$(df)(u, v) = \sqrt{\frac{w(u, v)}{\delta(v)}} f(v) - \sqrt{\frac{w(u, v)}{\delta(u)}} f(u), \quad (2.13)$$

où  $\delta(u) = \sum_{v \sim u} w(u, v)$  est le degré du sommet  $u$  dans le graphe. On peut noter que cette formulation ne respecte pas la propriété (2.10). De plus, son opérateur adjoint ne satisfait pas le théorème de la divergence (2.12).



Nous pouvons aussi reprendre les définitions de dérivées suivantes (voir [Elmoataz *et al.*, 2008, Ta *et al.*, 2008b] pour plus de détails) :

$$\partial_v^+ f(u) = \max(\partial_v f(u), 0), \quad (2.14)$$

et,

$$\partial_v^- f(u) = \max(-\partial_v f(u), 0) = -\min(\partial_v f(u), 0).$$

### 2.1.2.2 Gradient discret et $p$ -laplacien

Le *gradient* d'une fonction  $f : V \rightarrow \mathbb{R}$ , à un sommet  $u \in V$ , est le vecteur défini par :

$$\nabla_w f(u) = (\partial_v f(u) : v \sim u)^T = (\partial_{v_1} f(u), \dots, \partial_{v_k} f(u))^T, \quad (2.15)$$

où  $(v_1, \dots, v_k)$  sont les voisins du sommet  $u$ .

Cette définition ne dépend pas de la structure du graphe. L'information relative à la nature du graphe, ou à la fonction  $f$ , peut être incorporée dans la fonction de poids  $w$ .

À partir des définitions 2.14 et de la définition du gradient, on déduit deux nouvelles définitions de gradients :

$$\nabla_w^+ f(u) = (\partial_v^+ f(u) : v \sim u)^T \quad \text{et} \quad \nabla_w^- f(u) = (\partial_v^- f(u) : v \sim u)^T, \quad (2.16)$$

La *variation locale* d'une fonction  $f$  à un sommet  $u$  mesure l'amplitude du gradient en  $u$ . Elle peut être donc être interprétée comme la régularité de la fonction dans le voisinage du sommet en question. Plusieurs normes de  $\nabla_w f(u)$  peuvent être utilisées pour mesurer cette régularité. Considérons la norme  $\mathcal{L}_p$  :

$$\|\nabla_w f(u)\|_p = \left( \sum_{v \sim u} (\partial_v f(u))^p \right)^{1/p} = \left( \sum_{v \sim u} \left( \sqrt{w(u, v)} (f(v) - f(u)) \right)^p \right)^{1/p} \quad (2.17)$$

Lorsque l'ensemble de sommets  $V$  représente un ensemble de données organisées, comme des images ou des vidéos numériques, l'organisation initiale, qui est un graphe par nature, traduit les interactions locales entre les données. Sur ce graphe initial, l'opérateur gradient implique ces interactions locales. Un opérateur de gradient qui implique des interactions non locales peut être obtenu en construisant des graphes de voisinage à partir de l'organisation initiale. Dans les deux cas, l'expression de la variation locale reste inchangée. Du fait que  $w(u, v) = 0$  si  $(u, v) \notin E$ , une expression totalement non locale de la variation locale peut être écrite sous la forme :

$$\|\nabla_w f(u)\|_p = \left( \sum_{v \in V} |\partial_v f(u)|^p \right)^{1/p} = \left( \sum_{v \in V} \left( \sqrt{w(u, v)} |f(v) - f(u)| \right)^p \right)^{1/p} \quad (2.18)$$

En particulier, pour  $p = 2$ , nous avons la norme  $\mathcal{L}_2$  :

$$\|\nabla_w f(u)\|_2 = \sqrt{\sum_{v \in V} (\partial_v f(u))^2} = \sqrt{\sum_{v \in V} \left( \sqrt{w(u, v)} (f(v) - f(u))^2 \right)}, \quad (2.19)$$

et pour  $p = 1$ , la norme  $\mathcal{L}_1$  :

$$\|\nabla_w f(u)\|_1 = \left( \sum_{v \in V} |\partial_v f(u)| \right) = \sum_{v \in V} \left( \sqrt{w(u, v)} |f(v) - f(u)| \right).$$

Nous définissons également les expressions de variations suivantes :

$$\|\nabla_w^+ f(u)\|_p = \left( \sum_{v \in V} (\partial_v^+ f(u))^p \right)^{1/p} = \left( \sum_{v \in V} \max \left( \sqrt{w(u,v)} (f(v) - f(u)), 0 \right)^p \right)^{1/p} \quad (2.20)$$

$$\|\nabla_w^- f(u)\|_p = \left( \sum_{v \in V} (\partial_v^- f(u))^p \right)^{1/p} = \left( \sum_{v \in V} \max \left( -\sqrt{w(u,v)} (f(v) - f(u)), 0 \right)^p \right)^{1/p} \quad (2.21)$$

Le  $p$ -Laplacien isotrope  $\Delta_{p,w}^i: \mathcal{H}(V) \rightarrow \mathcal{H}(V)$ , d'une fonction  $f: V \rightarrow \mathbb{R}$ , à un sommet  $u \in V$ , s'écrit :

$$\Delta_{p,w}^i f(u) = \frac{1}{2} \sum_{v \sim u} w(u,v) \left( \|\nabla_w f(v)\|_2^{p-2} + \|\nabla_w f(u)\|_2^{p-2} \right) (f(u) - f(v)), \quad (2.22)$$

décrivant une famille d'opérateurs du second ordre.

Le  $p$ -Laplacien anisotrope  $\Delta_{p,w}^a: \mathcal{H}(V) \rightarrow \mathcal{H}(V)$ , d'une fonction  $f: V \rightarrow \mathbb{R}$ , à un sommet  $u \in V$ , est :

$$\Delta_{p,w}^a f(u) = \sum_{v \sim u} w(u,v)^{p/2} |f(u) - f(v)|^{p-2} (f(u) - f(v)). \quad (2.23)$$

Cet opérateur est non linéaire si  $p \neq 2$ . Dans ce dernier cas, il correspond au *Laplacien combinatoire du graphe*, comme le 2-Laplacien isotrope. Pour empêcher le dénominateur d'être nul dans (2.23) quand  $p \leq 1$ ,  $|f(u) - f(v)|$  est remplacé par  $|f(u) - f(v)| + \epsilon$ , où  $\epsilon \rightarrow 0$  est une petite constante fixe.

Cette formulation du  $p$ -Laplacien anisotrope permet de récupérer l'expression du pseudo-Laplacien

$$\sum_{i=1}^n \frac{\partial}{\partial x_i} \left( \left| \frac{\partial f}{\partial x_i} \right|^{p-2} \frac{\partial f}{\partial x_i} \right), \quad (2.24)$$

et celle du Laplacien continu non local

$$\int_{\Omega} w(x,y) |f(y) - f(x)|^{p-2} (f(y) - f(x)) dy.$$

Cette dernière correspond exactement au  $p$ -Laplacien anisotrope jusqu'à une puissance  $\frac{p}{2}$  sur les poids.

### 2.1.3 Formulation mathématique et approche proposée

Après avoir introduit les opérateurs différentiels sur graphes, nous présentons la formalisation de la régularisation sur graphes pondérés. En se basant sur les définitions données du gradient et du  $p$ -laplacien, la réduction du bruit et la simplification sont énoncées en termes de régularisation sur graphes.

### 2.1.3.1 Formulation par $p$ -Laplacien

Rappelons que  $f^0$  est une observation d'une fonction d'origine  $f$  altérée par un bruit  $\eta : f^0 = f + \eta$ .

La régularisation discrète de  $f^0 \in \mathcal{H}(V)$  en utilisant le  $p$ -laplacien pondéré consiste à chercher une fonction  $f^* \in \mathcal{H}(V)$  qui est non seulement assez lisse sur  $G_w$ , mais aussi suffisamment proche de  $f^0$ . Deux modèles variationnels de régularisation peuvent être formalisés par la minimisation de deux termes d'énergie en utilisant soit le  $p$ -Laplacien isotrope soit le  $p$ -Laplacien anisotrope.

Le *modèle isotrope* donne la formulation suivante du problème de minimisation :

$$f^* = \min_{f \in \mathcal{H}(V)} \left\{ \frac{1}{p} \sum_{v \in V} \|\nabla f(v)\|_2^p + \frac{\lambda}{2} \|f - f^0\|_{\mathcal{H}(V)}^2 \right\} \quad p \in ]0, +\infty[. \quad (2.25)$$

La première énergie dans (2.25) est le terme de lissage ou le régularisateur, tandis que le second est le terme d'attache aux données. Les deux fonctions d'énergie sont des fonctions strictement convexes de  $f$ . En particulier, par des arguments standard de l'analyse convexe, le problème (2.25) a une solution unique, pour  $p \geq 1$ , qui satisfait :

$$(\Delta_p f(v)) + \lambda(f(v) - f^0(v)) = 0, \forall v \in V.$$

La solution du problème (2.25) conduit à une famille de filtres non-linéaires paramétrés par la fonction de poids, le degré de lissage, et le paramètre de fidélité.

De manière similaire, le *modèle anisotrope* permet de reformuler le problème de minimisation en utilisant le  $p$ -Laplacien anisotrope :

$$f^* = \min_{f \in \mathcal{H}(V)} \left\{ \frac{1}{p} \sum_{v \in V} \|\nabla f(v)\|_p^p + \frac{\lambda}{2} \|f - f^0\|_{\mathcal{H}(V)}^2 \right\} \quad \text{avec } p \in ]0, +\infty[. \quad (2.26)$$

### 2.1.3.2 Résolution du problème de minimisation

Les problèmes de minimisation comme celui de la régularisation peuvent être résolus par plusieurs méthodes numériques qui convergent efficacement vers la solution. Ces méthodes reviennent à résoudre des systèmes linéaires et non-linéaires d'équations. Comme dans le cadre continu, les solutions de ces systèmes se formulent comme des processus de diffusion. Ces mêmes processus peuvent également être utilisés indépendamment de la régularisation. L'avantage du cadre discret est qu'ils ne font pas intervenir d'EDP. Nous présentons la résolution par la méthode linéarisée de Gauss-Jacobi.

Nous nous intéressons uniquement au traitement des séquences vidéo dans le cadre de régularisation que nous venons de présenter. Pour traiter des séquences vidéo, nous pouvons considérer une séquence vidéo comme une simple succession d'images indépendantes et traiter chaque image séparément. Cependant, ceci donne des résultats perfectibles en termes de performance et de qualité. La caractéristique principale des séquences d'images est la redondance du contenu entre les images consécutives. Par conséquent, la dimension temporelle offre une source d'information qui devrait être intégrée dans l'algorithme de régularisation. Ainsi, l'extension à des séquences vidéo est basée sur l'intégration du temps dans le processus de régularisation. Nous profitons de la redondance temporelle des données due au débit d'images élevé pour améliorer la qualité de notre

régularisation. Ainsi, nous développons une régularisation spatio-temporelle sur les graphes pondérés.

Nous utilisons l'algorithme itératif de Gauss-Jacobi pour la résolution du problème de régularisation.

Pour tout  $(u, v)$  dans  $E$  :

$$\begin{cases} f^{(0)} = f^0 \\ \gamma^{(k)}(u, v) = w(u, v) \left( \|\nabla f^{(k)}(v)\|^{p-2} + \|\nabla f^{(k)}(u)\|^{p-2} \right) \\ f^{(k+1)}(v) = \frac{p\lambda f^0(v) + \sum_{u \sim v} \gamma^{(k)}(u, v) f^{(k)}(u)}{p\lambda + \sum_{u \sim v} \gamma^{(k)}(u, v)} \end{cases} \quad (2.27)$$

où  $\gamma^{(k)}$  est la fonction  $\gamma$  à l'étape  $k$ . Les poids  $w(u, v)$  sont calculés à partir de  $f^0$ .

À chaque itération, la nouvelle valeur  $f^{(k+1)}$ , à un sommet  $v$ , dépend de deux quantités, la valeur d'origine  $f^0(v)$  et la moyenne pondérée des valeurs existantes dans le voisinage de  $v$ . Cela montre que le filtre proposé, obtenu par itération, est un filtre passe-bas qui peut accueillir de nombreuses structures de graphes et fonctions de poids. Pour plus de détails sur l'algorithme de Gauss-Jacobi itératif, le lecteur peut se référer à [Elmoataz et al., 2008].

#### 2.1.4 Liens avec les méthodes classiques

Nous remarquons qu'en une itération du processus de diffusion discrète (2.27), avec  $p = 2$ ,  $\lambda = 0$  et des fonctions de poids appropriées, nous retrouvons des filtres bien connus de la littérature. En particulier, nous pouvons récupérer le filtre bilatéral [Smith et Brady, 1997, Tomasi et Manduchi, 1998] en utilisant la fonction de poids suivante :

$$w(u, v) = \exp\left(-\frac{\|u - v\|^2}{\sigma_V^2}\right) \exp\left(-\frac{\|f^0(u) - f^0(v)\|^2}{\sigma_{\mathcal{H}(V)}^2}\right), \quad (2.28)$$

où  $\sigma_V^2$  et  $\sigma_{\mathcal{H}(V)}$  sont respectivement les variances dans le domaine discret  $V$  et dans l'espace de fonctions  $\mathcal{H}(V)$ . De même, l'expression du filtre NL-means peut être obtenu en utilisant la fonction de poids :

$$w(u, v) = \exp\left(-\frac{\rho(F(f^0, u) - F(f^0, v))}{h^2}\right), \quad (2.29)$$

où  $F(f^0, u)$  est un vecteur caractéristique associé au sommet  $u \in V$ , qui est généralement un patch de sommets centré en  $u$ . La fonction  $\rho$  mesure la distance entre les deux vecteurs caractéristiques. Avec ces deux fonctions de poids, le cadre de régularisation anisotrope se comporte comme un filtre bilatéral itératif ou un filtre NL-means itératif, sans mise à jour des poids à chaque itération du processus de diffusion.

#### Optimisation de l'algorithme

L'inconvénient de l'algorithme présenté est son coût de calcul élevé. La méthode non locale compare le patch autour de chaque sommet  $v$  avec tous les patches contenus dans le voisinage non local  $N_{k_1, k_2, k_3}(v)$ . Par conséquent, nous proposons la réduction du contenu

de la fenêtre à un certain pourcentage de sa taille, noté  $X\%$ , en une variante non locale optimisée de l'algorithme. Dans cette version, pour chaque sommet  $v$ , nous choisissons aléatoirement des nœuds dans  $N_{k_1, k_2, k_3}(v)$  pour former le voisinage non local réduit de  $v$  (voir figure 2.2). D'où, l'algorithme non local optimisé suivant :

---

**Algorithme 2.1** Régularisation non locale optimisée
 

---

**ENTRÉES :** Un graphe  $G$ , le nombre d'itérations  $N \in \mathbb{N}^+$

**tantque**  $N \neq 0$  **faire**

*Construire le voisinage non local réduit de  $v$  :* Choisir aléatoirement  $X\%$  des sommets dans  $N_{k_1, k_2, k_3}(v)$  (et hors du patch  $B(v)$ ) en plus du voisinage local.

*Calculer la similarité entre patches entre  $B(v)$  et  $B(u)$  pour tout  $u$  dans le voisinage non local réduit de  $v$ .* Tous les patches  $B(u)$  doivent être entièrement inclus dans  $N_{k_1, k_2, k_3}(v)$ .

*Mettre à jour la valeur  $f(v)$  suivant (2.27)*

$N \leftarrow N - 1$

**fin tantque**

---

L'algorithme 2.1 contient  $N$  itérations, où  $N$  est précisé en entrée. Sinon, nous utilisons un critère de terminaison de sorte que l'algorithme converge en un nombre quelconque d'itérations. En conséquence, la méthode non locale est accélérée, et nous obtenons des résultats tout aussi attrayants comme nous pouvons le constater sur les expérimentations à la section 2.3.

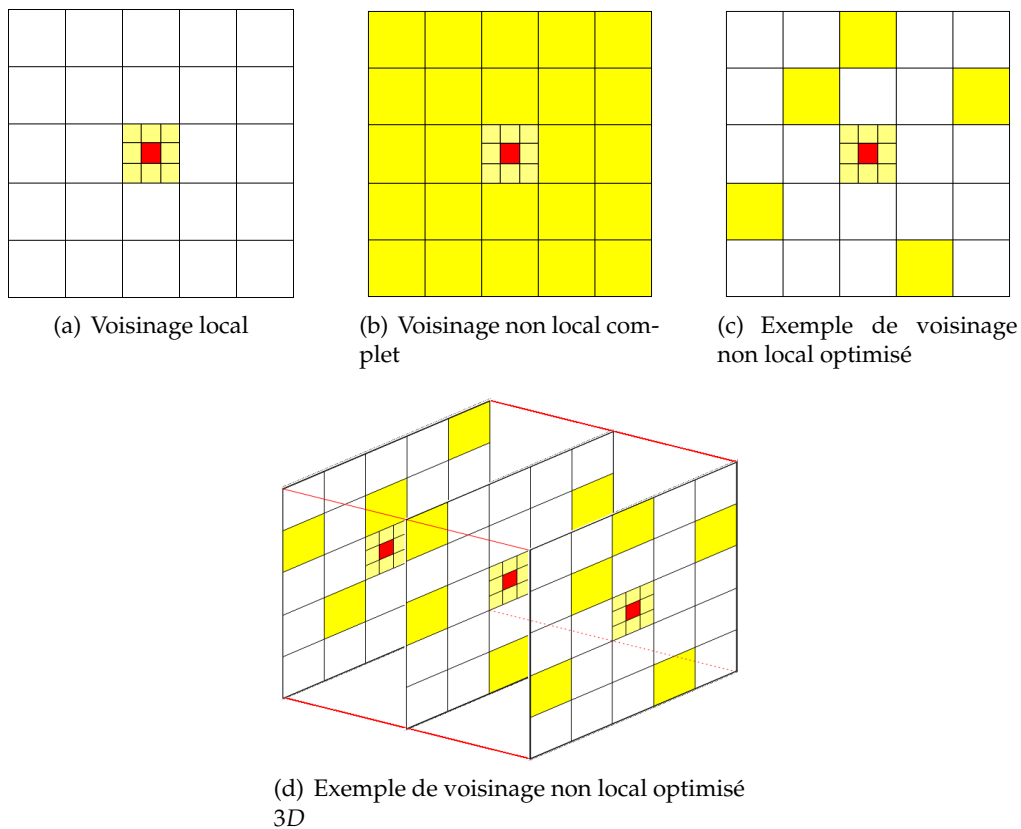


FIGURE 2.2 – Illustration de la définition du voisinage pour les différentes méthodes, locale, non locale et non locale optimisée.

## 2.2 Régularisation par $\infty$ -Laplacien

Dans cette section, nous proposons un nouveau modèle variationnel basé sur l'adaptation de l' $\infty$ -Laplacien (lire l'infini Laplacien) aux graphes pondérés [Elmoataz *et al.*, 2011]. Notre motivation pour utiliser l' $\infty$ -Laplacien émane d'applications d'images et de données. Une introduction détaillée à l' $\infty$ -Laplacien peut être trouvée dans [Crandall, 2008]. L' $\infty$ -Laplacien est également apparu dans plusieurs contextes différents, comme par exemple dans la théorie des jeux pour deux personnes avec un ordre aléatoire du jeu [Barron *et al.*, 2008]. Nous présentons brièvement un aperçu non-exhaustif des travaux antérieurs relatifs à l' $\infty$ -Laplacien dans le domaine du traitement d'images.

Dans la vision par ordinateur, le  $\infty$ -Laplacien a été introduit pour le rehaussement de contours [Torre et Poggio, 1986, Yuille et Poggio, 1986] et a servi de base à la détection de contours [Canny, 1986]. Plus tard, Caselles et al. [Caselles *et al.*, 1998] ont étudié l' $\infty$ -Laplacien pour l'interpolation des données en images. Cong et al. [Cong *et al.*, 2004] ont également proposé un schéma d'interpolation pour la morphose de formes. Elion et Vese [Elion et Vese, 2007] ont proposé un modèle de décomposition de l'image en utilisant la variation totale et l' $\infty$ -Laplacien. Enfin, Guillot et Le Guyader [Guillot et Guyader, 2009] ont récemment proposé d'extrapoler le champ de vecteurs sur le domaine de l'image entière dans un cadre variationnel avec l' $\infty$ -Laplacien.

L' $\infty$ -Laplacien est lié à son équation EDP qui s'écrit :

$$-\Delta_{\infty}f = 0 \quad (2.30)$$

ou sa version dépendant du temps :

$$\frac{\partial f}{\partial t} = -\Delta_{\infty}f \quad (2.31)$$

où  $\Delta_{\infty}f = \sum_{i,j}^n \frac{\partial f}{\partial x_i} \frac{\partial f}{\partial x_j} \frac{\partial f}{\partial x_i \partial x_j}$  désigne la version 1-homogène de l' $\infty$ -Laplacien sur un domaine euclidien pour des fonctions lisses dans un certain ouvert  $\Omega \subseteq \mathbb{R}^n$ . Aronson [Aronson, 1967] interprète formellement l'équation  $\Delta_{\infty}f = 0$ , comme la limite du problème de minimisation  $\int \|\nabla f\|^p dx$  lorsque  $p$  tend vers l'infini, sous des conditions aux limites données.

En effet, l'équation d'Euler-Lagrange de ce problème est l'équation du  $p$ -Laplacien :

$$\Delta_p f = \operatorname{div} \left( \|\nabla f\|_2^{p-2} \nabla f \right) = 0 \quad (2.32)$$

qui peut s'écrire comme [Juutinen et Kawohl, 2006] :

$$(p-2) \|\nabla f\|^{p-4} \left( \frac{\|\nabla f\|_2^2}{(p-2)} \Delta_2 f + \Delta_{\infty}f \right) = 0 \quad (2.33)$$

où  $\Delta_2$  est le Laplacien combinatoire (le 2-Laplacien). Si  $\|\nabla f\| \neq 0$ , alors :

$$\Delta_{\infty}f = -\frac{\|\nabla f\|_2^2}{(p-2)} \Delta_2 f \quad (2.34)$$

et donc, lorsque  $p \rightarrow \infty$ , nous retrouvons l'équation de l' $\infty$ -Laplacien  $\Delta_{\infty}f = 0$ .

Dans un premier temps, nous donnons les définitions des dérivées correspondant au cas particulier de  $p = \infty$ . Puis, nous en déduisons une nouvelle approche pour résoudre le problème de débruitage en utilisant l' $\infty$ -Laplacien.

### 2.2.1 $\infty$ -Laplacien sur graphes

Dans un cadre continu, trouver  $f^* = \min \int \|\nabla f\|_2^p dx$  lorsque  $p \rightarrow \infty$  revient à trouver  $f^*$  solution de  $\Delta_\infty f = 0$ . Dans un cadre discret, nous montrons que trouver  $f_p^* = \min \sum \|\nabla_w f\|_p^p$  quand  $p \rightarrow \infty$  revient à trouver  $f^*$  solution de  $L(f) = \|(\nabla_w^+ f)(u)\|_\infty - \|(\nabla_w^- f)(u)\|_\infty = 0$ .

Rappelons que nous considérons un graphe  $G_w = (V, E)$ , où  $V$  est un ensemble fini de sommets, et  $E \subset V \times V$  un ensemble d'arêtes, et  $w$  une fonction poids de similarité.

**Proposition 1.** Pour  $1 \leq p < +\infty$ , à un sommet  $u \in V$

$$\left(\Delta_{(p+1),w} f\right)(u) = \|(\nabla_w^- f)(u)\|_p^p - \|(\nabla_w^+ f)(u)\|_p^p \quad (2.35)$$

avec  $w'(u, v) = w(u, v)^{p+1/p}$ .

Il existe une relation entre les gradients morphologiques et le Laplacien anisotrope (voir Proposition (2.35)).

*Preuve.* À partir de (2.23), nous avons

$$\left(\Delta_{p,w} f\right)(u) = \sum_{v \sim u} w(u, v)^{p/2} |f(u) - f(v)|^{p-2} (f(u) - f(v)).$$

Comme  $|x| = x^+ + x^-$  et  $x = x^+ - x^-$ , nous obtenons, avec  $A = (f(v) - f(u))$  :

$$\left(\Delta_{(p+1),w} f\right)(u) = \sum_{v \sim u} w(u, v)^{p+1/2} (A^- - A^+) (A^+ + A^-)^{p-1}. \quad (2.36)$$

Puis, en développant  $(A^+ + A^-)^{p-1}$ , puisque  $A^+ A^- = 0$ , il est facile d'arriver à (2.35).  $\square$

Si l'on prend certaines valeurs particulières de  $p$ , nous avons les relations suivantes. Pour  $p = 1$ , l'équation (2.35) donne

$$\left(\Delta_{2,w} f\right)(u) = \|(\nabla_w^- f)(u)\|_1 - \|(\nabla_w^+ f)(u)\|_1. \quad (2.37)$$

L'opérateur Laplacien combinatoire classique peut donc être exprimé comme la différence de deux gradients morphologiques externe et interne.

**Proposition 2.** Une fonction  $f$  est  $\infty$ -harmonique ssi

$$\min_f \{E(f, p) = \sum_{u \in V} \|\nabla_w f(u)\|_p^p\}$$

satisfait l'équation

$$L(f) = 0, \quad (2.38)$$

quand  $p$  tend vers l'infini, avec

$$L(f) = \|(\nabla_w^- f)(u)\|_\infty - \|(\nabla_w^+ f)(u)\|_\infty. \quad (2.39)$$

Par la suite, c'est cette quantité que nous désignerons par  $\Delta_w, \infty = L(f)$ .

*Preuve.* Nous considérons le problème de minimisation suivant :  $f^* = \min \sum \|\nabla_w f\|_p^p$ . Par l'analyse convexe standard, ceci équivaut à trouver  $f^*$  solution de  $\Delta_{p,w}f = 0$ .

À partir de la proposition 2.35, nous avons

$$(\Delta_{p,w}f)(u) = \|(\nabla_w^- f)(u)\|_{p-1}^{p-1} - \|(\nabla_w^+ f)(u)\|_{p-1}^{p-1} = 0$$

avec  $w'(u, v) = w(u, v)^{\frac{p}{p-1}}$  que nous pouvons facilement simplifier à

$$(\Delta_{p,w}f)(u) = \|(\nabla_w^- f)(u)\|_{p-1} - \|(\nabla_w^+ f)(u)\|_{p-1} = 0. \quad (2.40)$$

Nous pouvons en déduire

$$\begin{aligned} & \|(\nabla_w^\pm f)(u)\|_{p-1} \\ &= \left[ \sum_{v \sim u} (w'(u, v)^{\frac{1}{2}} (f(v) - f(u))^\pm)^{p-1} \right]^{\frac{1}{p-1}} \\ &= \left[ \sum_{v \sim u} w(u, v)^{\frac{1}{2}} (w(u, v)^{\frac{1}{2}} (f(v) - f(u))^\pm)^{p-1} \right]^{\frac{1}{p-1}} \\ &= \left[ \sum_{v \sim u} w(u, v)^{\frac{1}{2}} \times \right. \\ & \quad \left. \left( \frac{w(u, v)^{\frac{1}{2}} (f(v) - f(u))^\pm \|(\nabla_w^\pm f)(u)\|_\infty}{\|(\nabla_w^\pm f)(u)\|_\infty} \right)^{p-1} \right]^{\frac{1}{p-1}} \\ &= \|(\nabla_w^\pm f)(u)\|_\infty \left[ w(u, v_v)^{\frac{1}{2}} + \sum_{v \sim u, v \neq v_v} \epsilon(u, v)^{p-1} \right]^{\frac{1}{p-1}} \end{aligned} \quad (2.41)$$

avec  $v_v = \arg \max_{v \sim u} (w(u, v)^{\frac{1}{2}} (f(v) - f(u))^\pm)$  et  $\epsilon(u, v) < 1$ .

Par substitution dans (2.40) et en faisant tendre  $p$  vers l'infini, nous obtenons

$$\begin{aligned} \lim_{p \rightarrow \infty} \left( \|(\nabla_w^- f)(u)\|_{p-1} - \|(\nabla_w^+ f)(u)\|_{p-1} \right) &= \|(\nabla_w^- f)(u)\|_\infty - \|(\nabla_w^+ f)(u)\|_\infty \\ &= 0. \end{aligned} \quad (2.42)$$

□

En se basant sur la proposition 2.35, et sur la limite précédente, nous proposons une nouvelle définition de l' $\infty$ -Laplacien :

$$(\Delta_{\infty,w}f)(u) = \|(\nabla_w^- f)(u)\|_\infty - \|(\nabla_w^+ f)(u)\|_\infty \quad (2.43)$$

où,

$$\|(\nabla_w^- f)(u)\|_\infty = \max_{v \sim u} \left( \sqrt{w(u, v)} \max(f(u) - f(v), 0) \right) \quad (2.44)$$

$$\|(\nabla_w^+ f)(u)\|_\infty = \max_{v \sim u} \left( \sqrt{w(u, v)} \max(f(v) - f(u), 0) \right) \quad (2.45)$$

$\|(\nabla_w^+ f)(u)\|_\infty$  et  $\|(\nabla_w^- f)(u)\|_\infty$  sont les normes infinies de deux familles de gradients discrets sur graphe.

Avec une fonction de poids appropriée ( $w(u, v) = 1, \forall (u, v) \in E$ ), (2.43) peut être reformulée comme dans [Ta et al., 2009, Ta et al., 2011] :

$$-(\Delta_{\infty,1}f)(u) = \min_{v \sim u} f(v) + \max_{v \sim u} f(v) - 2f(u)$$

qui est connu dans la communauté de morphologie mathématique (discrète) comme le Laplacien morphologique.



### 2.2.2 Nouvelle interprétation et résolution

La régularisation discrète par  $\infty$ -Laplacien revient à considérer,  $\forall u \in V$

$$\begin{cases} \frac{\partial f(u,t)}{\partial t} &= -(\Delta_{\infty,w} f)(u,t) \\ f(u,0) &= f^0(u) \end{cases} \quad (2.46)$$

La résolution de ce problème en utilisant (2.43) nous donne :

$$\begin{aligned} f^{n+1}(u) &= f^n(u) + \Delta t [(\|\nabla_w^+ f^n(u)\|_\infty - \|\nabla_w^- f^n(u)\|_\infty)] \\ &= f^n(u) + \Delta t \max_{v \sim u} \left( w(u,v)^{1/2} (f^n(v) - f^n(u))^+ \right) \\ &\quad - \Delta t \max_{v \sim u} \left( w(u,v)^{1/2} (f^n(v) - f^n(u))^- \right) \end{aligned} \quad (2.47)$$

avec  $f(u,0) = f^0(u)$ .

De manière équivalente,

$$\begin{cases} f^{n+1}(u) = \frac{f^n(u) + 2\Delta t \|\nabla_w^+ f^n(u)\|_\infty}{2} + \frac{f^n(u) - 2\Delta t \|\nabla_w^- f^n(u)\|_\infty}{2} \\ f(u,0) = f^0(u) \end{cases} \quad (2.48)$$

ce que nous pouvons réécrire comme :

$$\begin{cases} f^{n+1}(u) = \frac{\delta(f^n(u)) + \mathcal{E}(f^n(u))}{2} \\ f(u,0) = f^0(u) \end{cases} \quad (2.49)$$

avec

$$\delta(f^n(u)) = f^n(u) - 2\Delta t \|\nabla_w^- f^n(u)\|_\infty \quad \text{et} \quad (2.50)$$

$$\mathcal{E}(f^n(u)) = f^n(u) + 2\Delta t \|\nabla_w^+ f^n(u)\|_\infty. \quad (2.51)$$

L'équation (2.48) montre que l'équation (2.47) peut être considérée comme la combinaison d'une érosion et d'une dilatation morphologiques comme c'est le cas pour  $p \neq \infty$  [Ta *et al.*, 2008a]. Par analogie au cas continu, cette équation peut être vue comme une moyenne répétée d'opérations d'érosion et de dilatation.

## 2.3 Résultats et performances

### 2.3.1 Débruitage

Afin de tester nos algorithmes, nous avons dégradé des séquences par un bruit blanc gaussien additif  $\eta$  de variance  $\sigma^2$ .

La Figure 2.3 montre que la méthode non locale fournit le meilleur résultat visuel. Le bruit est fortement réduit alors que les structures fines telles que les textures et les détails fins sont bien conservés. La méthode locale donne des résultats corrects, mais certains détails sont perdus. En particulier, les objets mobiles, tels que la balle de tennis de table deviennent très flous avec la méthode locale alors qu'ils sont nets dans les résultats de la méthode non locale. Cet essai souligne la pertinence de l'approche non locale pour le débruitage des objets en mouvement. La méthode optimisée non locale est très attrayante car elle préserve les détails tout en étant plus rapide et moins coûteuse que la méthode non locale.

Les observations visuelles sont également confirmées par les mesures PSNR rapportées dans le tableau 2.1. Bien que ces résultats sont légèrement en faveur de la méthode non locale, la variante proposée optimisée produit des résultats très compétitifs, en utilisant seulement 30% des données en entrée, et atteint ainsi une performance de calcul plus rapide.

TABLE 2.1 – Comparaison du PSNR (dB), des séquences bruitées (niveau de bruit  $\sigma = 10$ ) débruitées par nos méthodes après une itération.

| Séquence | Taille (px)     | Entrée | Locale | Non locale | Non locale optimisée |
|----------|-----------------|--------|--------|------------|----------------------|
| Flower   | 180 × 144 × 126 | 22.96  | 23.82  | 25.15      | 25.87                |
| Tennis   | 216 × 172 × 126 | 24.68  | 27.12  | 28.65      | 28.18                |
| Football | 180 × 144 × 105 | 24.68  | 25.14  | 26.02      | 26.68                |
| Mobile   | 180 × 144 × 251 | 21.24  | 24.31  | 27.08      | 26.22                |

Puis, nous avons appliqué notre algorithme à une séquence traitée image par image. Nous avons comparé les résultats 2D et 3D. Le PSNR des résultats 3D est systématiquement plus élevé que le PSNR des résultats 2D comme le montre la figure 2.4. On constate que le traitement vidéo optimisé avec la méthode non locale fournit un débruitage de qualité supérieure à celle de traitement d'images.

Cette expérience confirme et évalue la contribution importante de la redondance temporelle pour le débruitage vidéo.

Il est également intéressant d'évaluer l'efficacité de nos méthodes pour différents niveaux de bruit. Nous avons effectué une série de tests sur des séquences corrompues par des bruits ayant un  $\sigma$  de 5 à 50. Les résultats sont regroupés dans la figure 2.5. Ces résultats montrent comment le PSNR d'entrée diminue considérablement lorsque le niveau de bruit augmente, contrairement aux résultats de nos trois méthodes. Plus le bruit est élevé et plus l'écart entre le PSNR d'entrée et ceux de nos résultats est grand.

Ces mesures confirment également que la méthode non locale optimisée donne un PSNR similaire à celui de la méthode non locale. En conséquence, nous considérons que la méthode non locale optimisée est un bon compromis entre la méthode entièrement non locale et la méthode locale tant du point de vue de la performance et que celui de la qualité.

Nous pouvons observer sur la figure 2.6 que pour un nombre d'itérations fixe, le processus de lissage est plus avancé pour  $p = \infty$  que pour  $p = 2$ , lui-même plus avancé que pour  $p = 1$

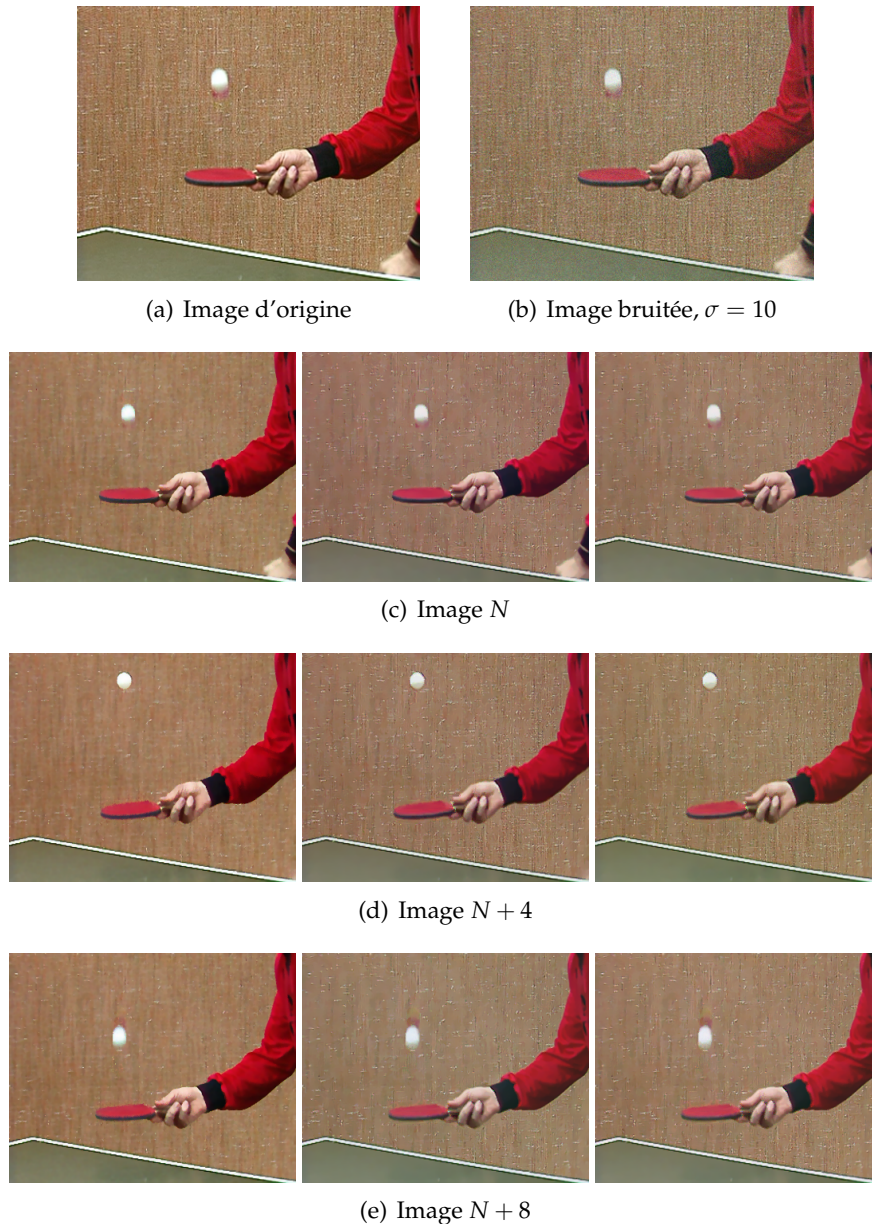


FIGURE 2.3 – Illustration de nos trois méthodes de débruitage avec  $p = 1$ ,  $\lambda = 0,5$  et  $h = 30$  après 3 itérations. (c), (d) et (e) : de gauche à droite : le résultat local, le non local et le non local optimisé.

Une autre façon de considérer le problème est de voir en combien d'itérations la convergence est atteinte ( $MSE < 10^{-3}$  entre deux itérations successives) en utilisant les mêmes paramètres. Le cas  $p = 1$  est connu et sa convergence est lente. Par conséquent, nous comparons la convergence de  $p = \infty$  au cas classique de  $p = 2$ . La figure 2.7 synthétise le résultat obtenu.

Sur les figures 2.10 et 2.11, nous constatons la suppression du bruit des séquences vidéo tout en conservant la texture.

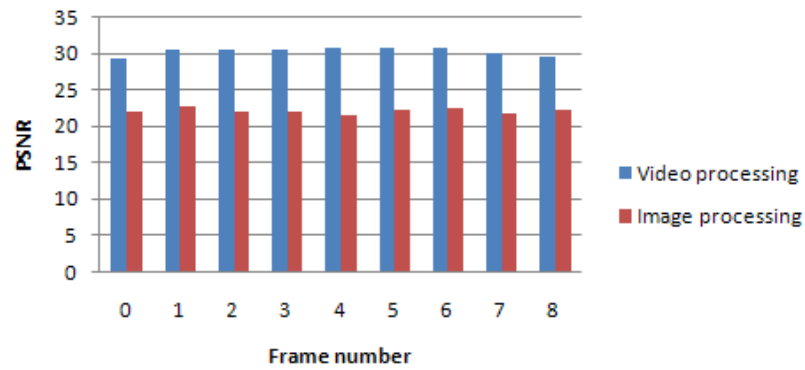


FIGURE 2.4 – Comparaison du PSNR entre résultats du débruitage vidéo et du débruitage image.

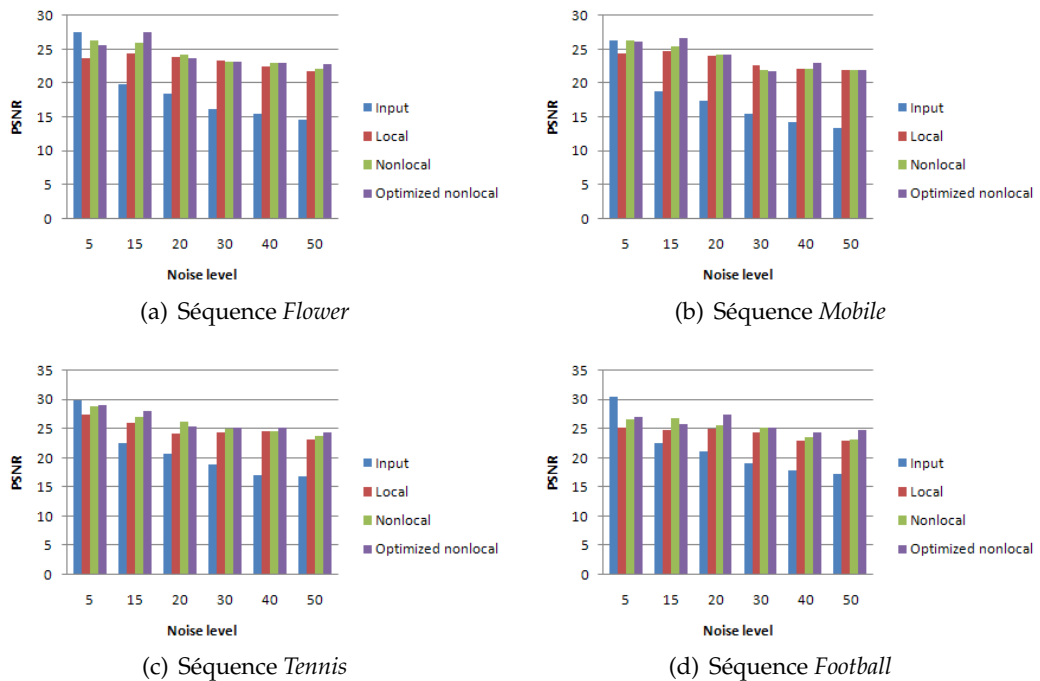


FIGURE 2.5 – Comparaison de PSNR entre nos différents résultats de débruitage et le PSNR d'entrée.



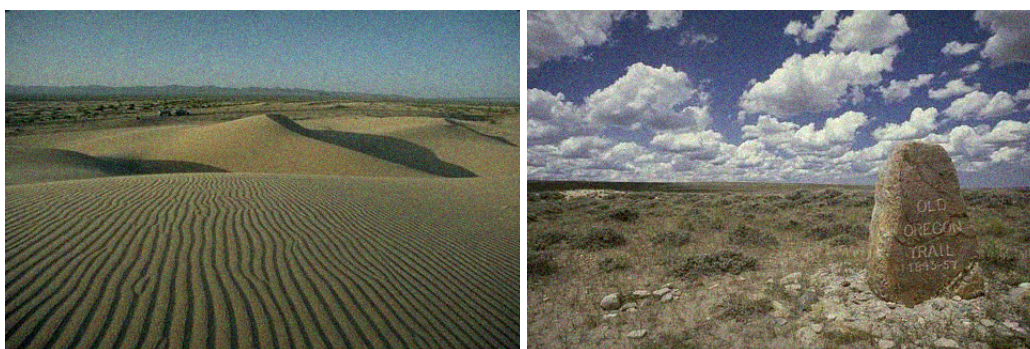
(a) Images bruitées,  $\sigma = 15$ (b)  $p = 1$ (c)  $p = 2$ (d)  $p = \infty$ 

FIGURE 2.6 – Comparaison des résultats obtenus par l’algorithme non local avec  $p = 1$ ,  $p = 2$  et  $p = \infty$  pour les mêmes paramètres et un nombre d’itérations fixé à 5.



(a)  $p = 2$ , convergence après 6 itérations



(b)  $p = \infty$ , convergence après 4 itérations

FIGURE 2.7 – Comparaison des résultats obtenus par l’algorithme non local avec  $p = 2$  et  $p = \infty$  pour les mêmes paramètres : Fenêtre  $9 \times 9$ , patch  $3 \times 3$ ,  $h = 3$  et  $\lambda = 5$ .



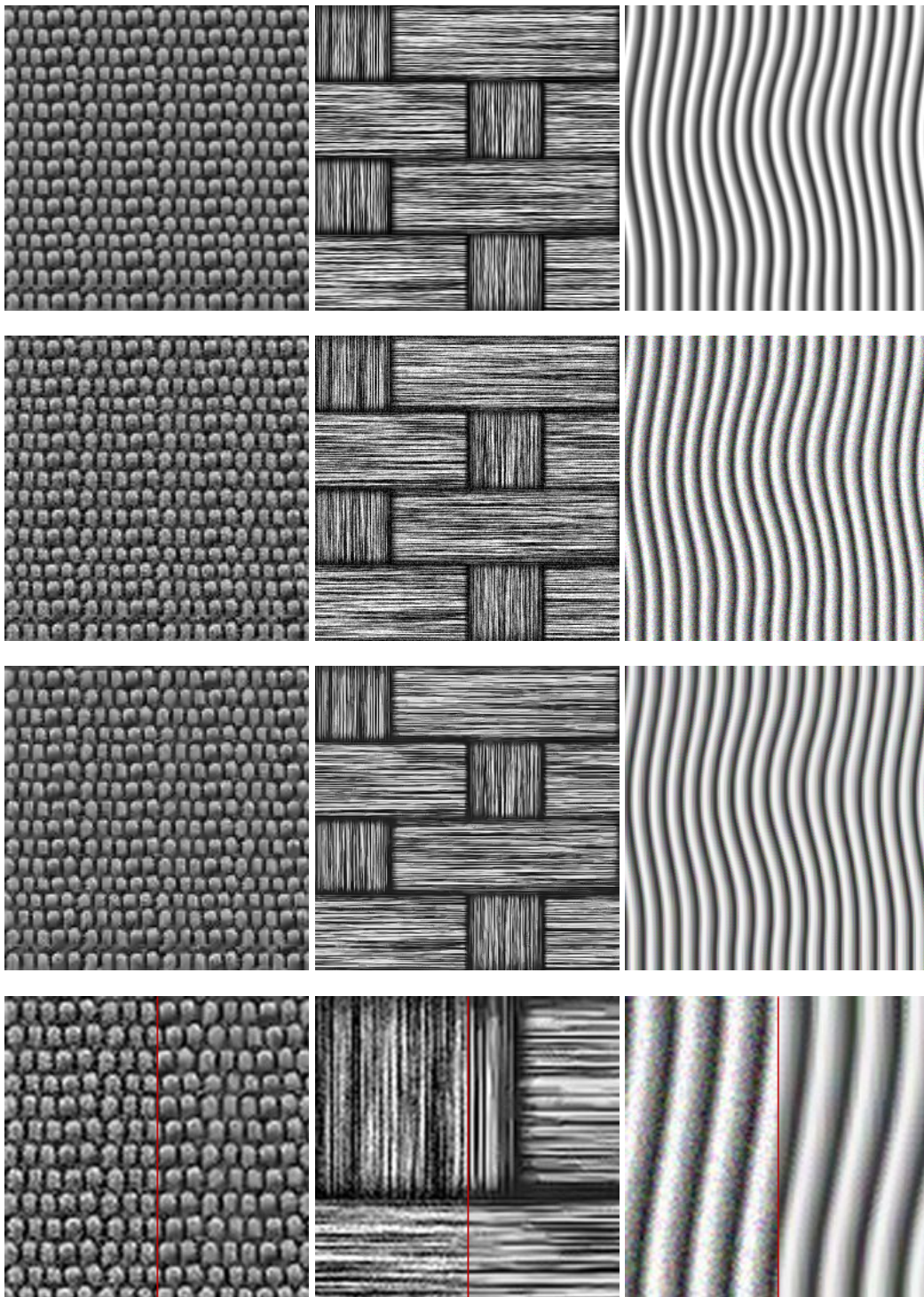


FIGURE 2.8 – Débruitage de textures avec  $p = \infty$ . Chaque colonne de haut en bas : l'image initiale, l'image bruitée à  $\sigma = 20$ , l'image restaurée et un agrandissement sur une région dont la moitié de gauche est bruitée et celle de droite est débruitée.

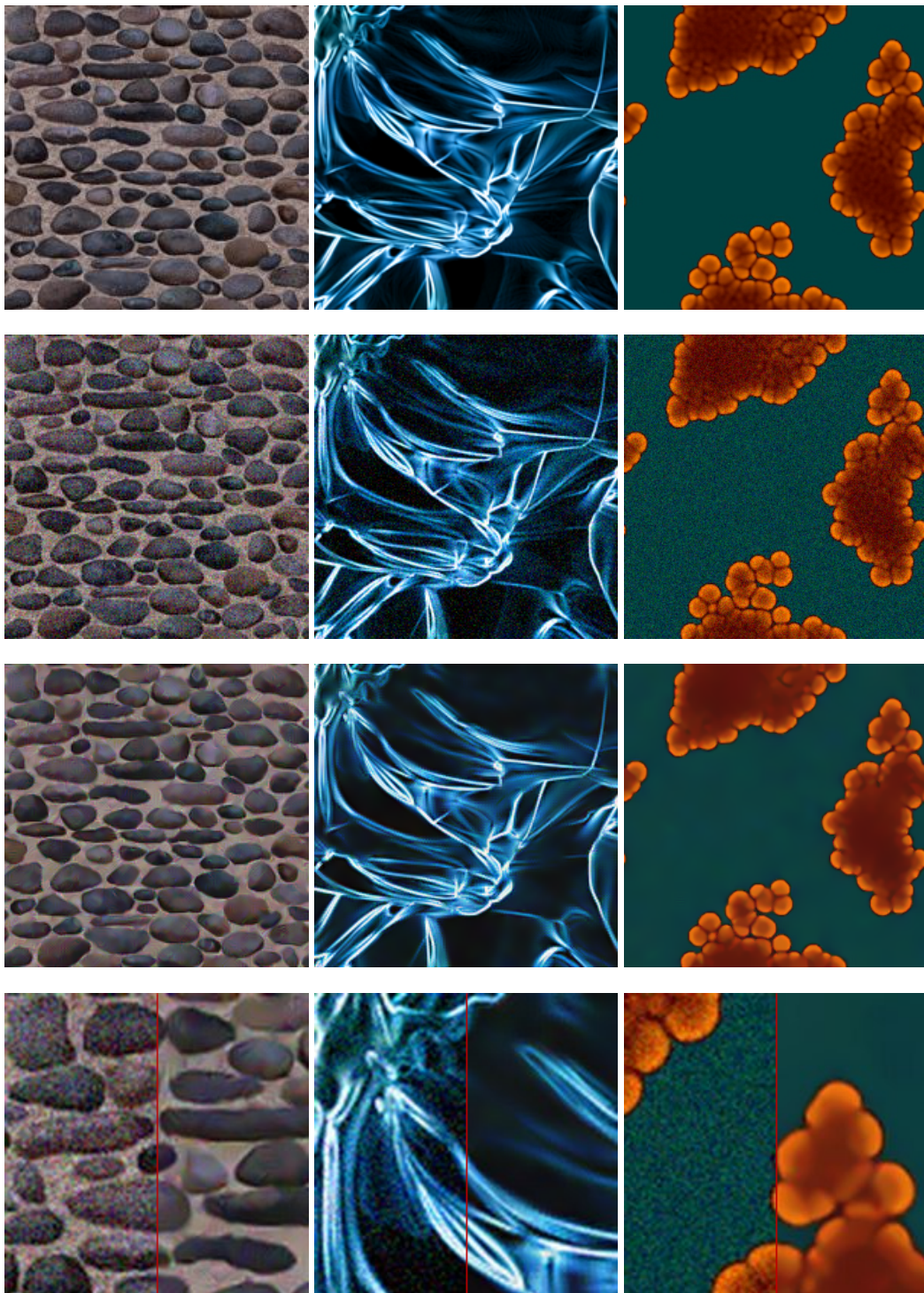
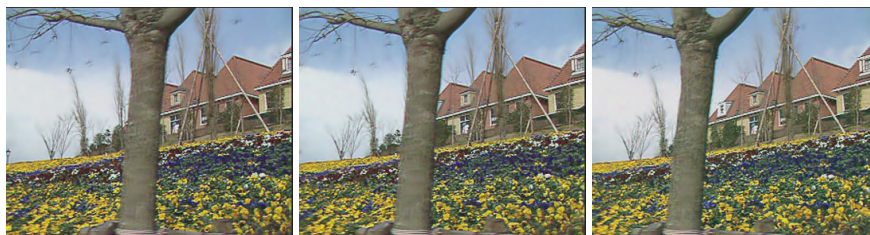


FIGURE 2.9 – Débruitage de textures en couleur avec  $p = \infty$ . Chaque colonne de haut en bas : l'image initiale, l'image bruitée à  $\sigma = 20$  et l'image débruitée et un agrandissement sur une région dont la moitié de gauche est bruitée et celle de droite est débruitée.



(a) Séquence bruitée,  $\sigma = 15$ 

(b) Séquence débruitée

FIGURE 2.10 – Débruitage de séquence vidéo *Flower* par la méthode non locale avec  $p = \infty$ , une fenêtre  $9 \times 9$ , un patch  $3 \times 3$  et après 20 itérations.

(a) Séquence bruitée,  $\sigma = 15$ 

(b) Séquence débruitée

FIGURE 2.11 – Débruitage de la séquence vidéo *Bus* par la méthode non locale avec  $p = \infty$ , une fenêtre  $9 \times 9$ , un patch  $3 \times 3$  et après 20 itérations.

### 2.3.2 Simplification de vidéos

Contrairement au débruitage, le but de la simplification n'est pas la fidélité de l'image débruitée à la scène qu'elle représente mais l'obtention d'une représentation simplifiée voire schématique de la scène. Il est clair que les applications du débruitage diffèrent de celles de la simplification, notamment dans les applications médicales où l'image obtenue doit être la plus fidèle possible à la scène étudiée afin de ne pas fausser le pronostic des spécialistes utilisant ces images.

La simplification de vidéos peut être interprétée comme un cas particulier de notre algorithme avec  $p < 1$ . Pour mettre en évidence l'influence de  $p$  sur les résultats, nous avons utilisé une fonction de poids constante  $w = 1$ . Nous obtenons des partitions homogènes du contenu vidéo. En comparaison avec le résultat de  $p = 2$ , nous observons que, pour  $p = 0,5$  et  $p = 0,1$ , des régions similaires s'unissent et forment des blocs plus grands et les contours sont plus nets. De plus, nous obtenons une simplification plus grossière de la vidéo quand  $p$  décroît (voir figure 2.12). Ces résultats peuvent être mis à profit pour la segmentation vidéo et la détection d'objets visuels.

## Conclusion

Dans ce chapitre, nous avons présenté une approche variationnelle par  $p$ -Laplacien pour le lissage. Nous nous sommes intéressés aux méthodes proposant une solution par régularisation. Notre apport réside principalement en deux points. D'une part, nous avons mis en place des algorithmes de débruitage de séquences d'images basés sur une régularisation non locale sur graphes par  $p$ -Laplacien, pour  $p \in ]0, \infty[$ . D'autre part, nous étendons ce cadre variationnel au cas particulier de  $p = \infty$  et nous l'appliquons à plusieurs exemples d'images et de vidéos.

Le choix des paramètres de régularisation doit être fait par l'utilisateur en fonction de ses besoins. La fonction poids  $w_{NL}$  capte les interactions entre les éléments comparés et permet d'obtenir de bons résultats. Par conséquent, le choix de la méthode non locale s'impose face à la méthode locale.

Les paramètres  $h$  et les dimensions de la fenêtre de recherche et du patch sont dépendants du contenu à traiter. Les variantes multi-résolutions permettent une certaine automatisation mais le réglage manuel d'un expert reste la meilleure solution afin de ne pas alourdir le processus.

Pour augmenter la rapidité davantage, nous avons proposé une variante optimisée qui permet d'accélérer le traitement. Par ailleurs, nous étudions la combinaison des traitements avec  $p = \infty$  et  $p = 2$  afin d'améliorer davantage la rapidité de  $p = 2$  tout en conservant une qualité élevée du résultat.

Au chapitre 3, nous nous intéressons à un deuxième type de restauration. Il s'agit de l'inpainting qui consiste en l'*interpolation* de données manquantes.

### Publications associées

- GHONIEM, M., CHAHIR, Y. et ELMOATAZ, A. Video denoising via discrete regularization on graphs. In *ICPR'08*, pages 1 – 4, 2008. [Ghoniem et al., 2008b]
- GHONIEM, M., CHAHIR, Y. et ELMOATAZ, A. Video denoising and simplification via discrete regularization on graphs. In *ACIVS*, pages 380 – 389, 2008. [Ghoniem et al., 2008a]
- GHONIEM, M., CHAHIR, Y. et ELMOATAZ, A. Nonlocal video denoising, simplification and inpainting using discrete regularization on graphs. *Signal Processing*, 90(8) : 2445 – 2455, 2010. Special Section on Processing and Analysis of High-Dimensional Masses of Image and Signal Data. [Ghoniem et al., 2010]

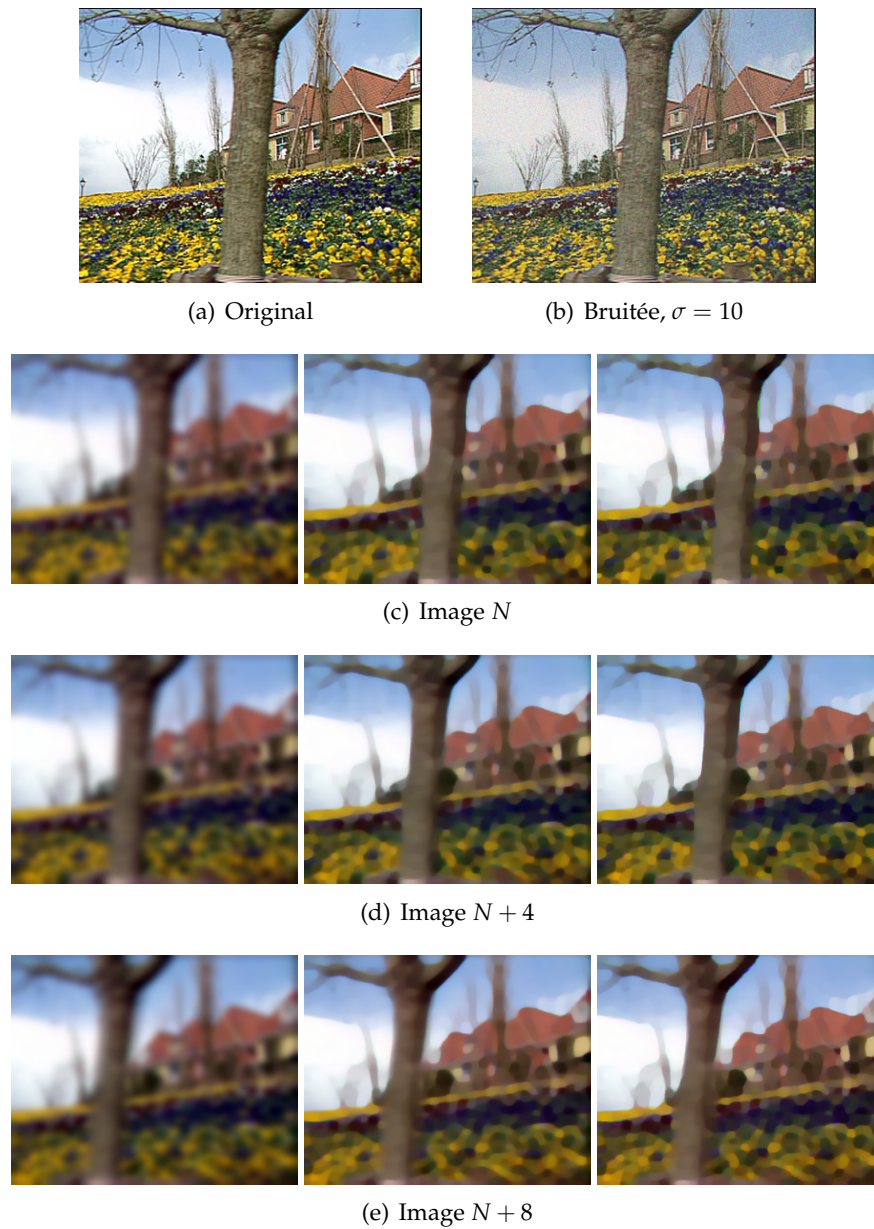


FIGURE 2.12 – Simplification de vidéos par la méthode non locale optimisée, avec  $\lambda = 0$  et  $w = 1$  après 5 itérations. De gauche à droite :  $p = 2$ ,  $p = 0,5$  et  $p = 0,1$ .



**DEUXIÈME PARTIE**

---

**RESTAURATION : INPAINTING ET  
INTERPOLATION**

---





# INPAINTING DES IMAGES ET DES VIDÉOS

## Sommaire

|   |           |
|---|-----------|
| <b>3.1 Problématique de l’inpainting</b> . . . . .          | <b>41</b> |
| <b>3.2 Structures géométriques vs. textures</b> . . . . .   | <b>42</b> |
| 3.2.1 Méthodes géométriques . . . . .                       | 43        |
| 3.2.2 Méthodes basées sur la synthèse de textures . . . . . | 45        |
| 3.2.3 Méthodes non locales . . . . .                        | 45        |

## Résumé

Dans ce chapitre, nous nous intéressons à la restauration des données manquantes. Contrairement à la restauration des données bruitées, il n’y a pas de bruit à modéliser ni de données initiales auxquelles il faudrait être fidèle. Il s’agit ici d’un problème d’interpolation qui consiste à réparer une scène en construisant un contenu nouveau en harmonie avec les données non perdues de la scène en question. Nous n’abordons pas l’étape préliminaire de sélection des défauts ou des dégradations telles que les déchirures ou les taches. Nous considérons uniquement le remplissage du trou créé par la suppression des taches et supposons que la région à combler est déjà définie. Il existe des travaux qui traitent cette étape de prétraitement pour des applications spécifiques telles que la restauration des films anciens.

Dans la section 3.1, nous définirons le problème d’inpainting. Différentes solutions performantes feront l’objet de la section 3.2. Une sélection non exhaustive de méthodes est fournie afin de présenter les différentes classes de solutions proposées.

## 3.1 Problématique de l’inpainting

L’inpainting consiste à remplir des zones manquantes d’une image ou d’une vidéo avec un contenu approprié calculé à partir des données préservées. L’objectif est de combler les trous par un contenu en harmonie avec le reste de l’image au point de rendre les zones reconstruites non décelables ou du moins difficilement.

De nombreuses applications se basent sur l’inpainting. À titre non exhaustif, nous pouvons citer la restauration de photographies, l’agrandissement et l’interpolation d’images, la compression d’images, la correction des zones perdues en cas d’erreur de transmission, la suppression d’objets ou de textes pour des effets spéciaux, la restauration de tableaux artistiques de façon automatique et sans altération du tableau réel pendant les essais, etc.

La première étape de l’inpainting consiste à délimiter la région à reconstruire. Cette étape correspond à la construction d’un *masque* binaire où les régions manquantes sont

identifiées. Par exemple si l'on souhaite effacer un texte superposé à une image, le masque correspondra aux lettres qui forment le texte à effacer. En superposant le masque sur l'image de départ, nous obtenons l'image masquée qui devra être traitée comme le montre la figure 3.1.

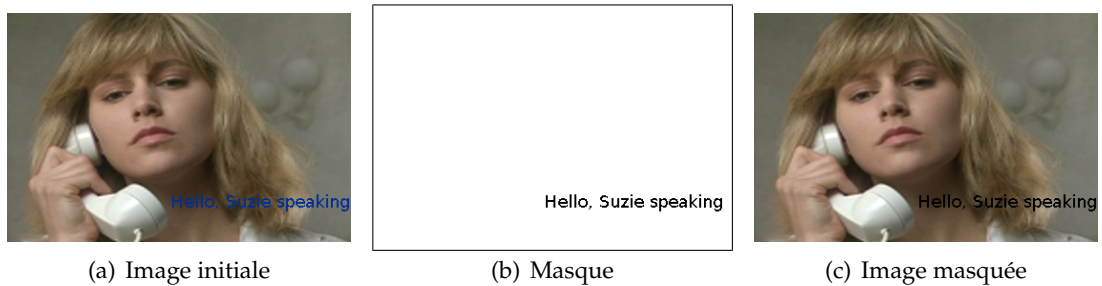


FIGURE 3.1 – Construction de l'image donnée en entrée au processus d'inpainting.

Il est important de noter que l'objectif de l'inpainting est de construire un contenu plausible. Par conséquent, il peut y avoir plusieurs résultats différents acceptables. La figure 3.2 illustre un cas où le résultat de l'inpainting est vraisemblable mais diffère de l'image initiale volontairement altérée. Aucune information n'est disponible concernant l'intersection des deux bandes. En conséquence, le résultat obtenu est tout à fait satisfaisant.

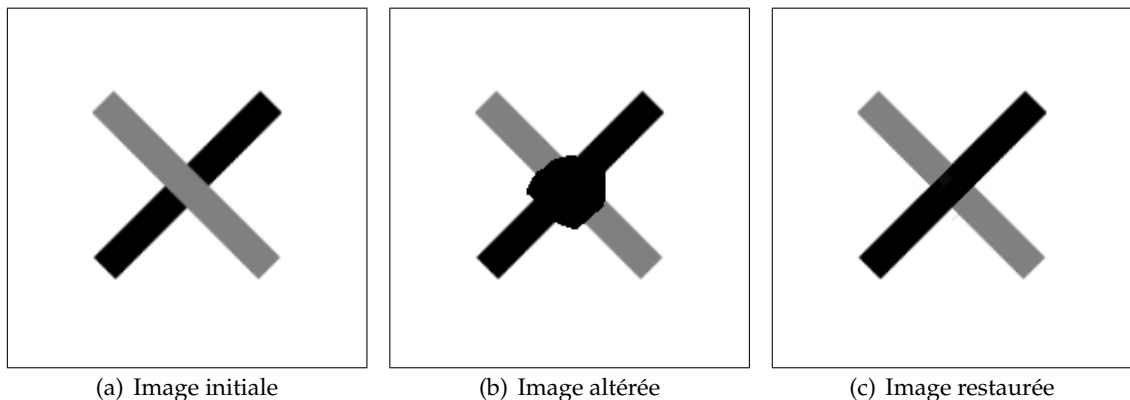


FIGURE 3.2 – Après l'inpainting, la bande noire passe au-dessus de la bande grise (3.2(c)) alors qu'à l'origine, c'est l'inverse (3.2(a)).

Cet exemple souligne l'influence que peut avoir la construction du masque. Par conséquent, le soin apporté à la construction du masque sur des images complexes peut améliorer le résultat du processus d'inpainting. Pour toutes les expérimentations présentées ici, le masque a été construit manuellement. Pour une application donnée, il est envisageable d'automatiser l'étape de la construction du masque. Par la suite, nous considérons directement l'image masquée sans nous préoccuper de sa construction. Ci-après, nous nous intéressons aux méthodes proposées pour l'inpainting des images et des vidéos.

## 3.2 Structures géométriques vs. textures

L'inpainting a fait l'objet d'un grand nombre de travaux. Différentes stratégies ont été envisagées pour proposer une solution à ce problème. Deux grandes classes de mé-



thodes ont émergé. La première classe s'est focalisée sur la reconstruction des structures géométriques. Si ces méthodes réussissent à reconstruire correctement la géométrie des structures, elles fournissent des résultats insatisfaisants au niveau des détails de nature stochastique tels que les textures. La deuxième classe a proposé des solutions plus efficaces pour la reconstruction des textures mais qui sont inadaptées aux structures géométriques. Subséquemment, une troisième classe de travaux est apparue tirant profit des deux premières en essayant d'assurer aussi bien la reconstruction de la composante géométrique que celle de la composante texturée.

À la lumière de cette distinction, nous catégorisons les méthodes d'inpainting en trois groupes : les méthodes dites géométriques, les méthodes basées sur la synthèse de textures et les méthodes non locales. Pour plus de clareté, nous désignons respectivement par « structures géométriques » et « structures stochastiques », les contours et les textures.

### 3.2.1 Méthodes géométriques

Parmi les méthodes d'interpolation qui ont été proposées, un certain nombre de méthodes sont variationnelles ou basées sur les EDP. L'emploi de ces méthodes est corroboré par un constat crucial en vision humaine qui stipule que les contours jouent un rôle important dans la perception visuelle.

Les méthodes variationnelles et les méthodes basées sur les EDP résolvent les problèmes d'optimisation avec la contrainte des données connues sur le bord du trou. Ces méthodes ont prouvé leur efficacité à reconstruire les structures géométriques. Le problème est défini à l'aide d'une fonctionnelle de l'image d'origine et de l'image restaurée. Les images sont modélisées comme une fonction ayant un certain degré de lissage basé sur le calcul d'une énergie *elastica* généralisée. Différents modèles d'images sont utilisés tels que le modèle VT [Rudin *et al.*, 1992, Chan et Shen, 2005b], le modèle Mumford Shah [Mumford et Shah, 1989], ou une fonctionnelle *elastica* d'Euler [Masnou et Morel, 1998, Masnou, 2002, Chan *et al.*, 2002b]. Une solution à un problème d'inpainting peut être obtenue en trouvant l'image restaurée qui minimise cette fonctionnelle. Ce problème d'optimisation peut être résolu par le calcul des variations (EDP) par la méthode d'Euler-Lagrange. Un aperçu de ces méthodes peut être trouvé dans les travaux de Ballester [Ballester *et al.*, 2001] et al. et Chan et al. [Chan et Shen, 2001, Chan *et al.*, 2002a]. Dans ces méthodes, les données connues sont diffusées vers les régions manquantes, pixel par pixel.

Il existe également des méthodes basées sur les EDP dans lesquelles une EDP est directement définie comme dans [Bertalmio *et al.*, 2000, Bertalmio *et al.*, 2001, Grossauer et Scherzer, 2003], et qui sont capables de reconstruire des régions contenant des structures géométriques prononcées aux bords du trou. Dans ce cadre, [Bertalmio *et al.*, 2001] ont proposé quelques travaux préliminaires sur l'inpainting d'images où les zones manquantes sont remplies par propagation de l'information (structures linéaires) vers l'intérieur de la région manquante le long des isophotes.

Inspirés par l'approche par EDP de Bertalmio et al. [Bertalmio *et al.*, 2000], Chan et Shen [Chan et Shen, 2001] proposent un modèle d'inpainting basé sur la VT, fondé sur les principes Bayésiens et variationnels.

$$\min_f E_{tv}(f) = \int_{\Omega} |\nabla f| \quad (3.1)$$

avec la contrainte :

$$f = g \quad \text{sur } \partial A \quad (3.2)$$

où  $A$  est la zone à combler et  $\partial A$  sa frontière.

Minimiser cette fonctionnelle d'énergie revient à relier les lignes de niveau de part et d'autre de la région à combler en suivant la plus petite distance. Ceci peut être exprimé par la formule co-aire :

$$\int |\nabla f| dx = \int_0^1 \int_{\Gamma_\lambda} ds d\lambda \quad (3.3)$$

où  $\Gamma_\lambda = \{u: f(u) = \lambda\}$  est la ligne de niveau et  $ds$  est la longueur de l'arc des lignes de niveau. La variation totale de  $u$  est la longueur totale de toutes les lignes de niveau. Par conséquent, avec la contrainte (3.2), les contours nets sont reliés suivant les lignes de niveau dans la région connue.

L'inpainting par VT interpole des images dans toutes les régions manquantes tout en préservant les contours nets. Toutefois, il ne relie pas toujours les contours correctement.

Chan et al. [Chan et al., 2002b] ont proposé de minimiser l'énergie de l'elastica d'Euler au lieu de la VT. L'inpainting par l'elastica d'Euler améliore l'inpainting VT parce que la courbure des lignes de niveau est davantage pénalisée. Ceci permet d'éviter les entortillements aux bords du trou en raison de leur courbure infinie. L'inpainting par l'elastica d'Euler répare les zones manquantes selon la minimization d'énergie suivante :

$$\min_f E_{elastica}(f) = \int (a + b\kappa^2) |\nabla f| \quad (3.4)$$

avec la contrainte :

$$f = g \quad \text{sur } \partial A \quad (3.5)$$

Dans cette fonctionnelle,  $a$  et  $b$  sont des constantes positives et  $\kappa = \nabla \cdot \left( \frac{\nabla f}{|\nabla f|} \right)$  est la courbure de  $f$ . Minimiser cette fonctionnelle d'énergie revient à relier des contours nets en fonction de la courbure des lignes de niveau dans la région connue. Ceci peut être expliqué par la formule de la co-aire :

$$\int (a + b\kappa^2) |\nabla f| dx = \int_0^1 \int_{\Gamma_\lambda} (a + b\kappa^2) ds d\lambda \quad (3.6)$$

Il existe d'autres méthodes qui impliquent des EDP rigides du quatrième ordre dans le problème d'inpainting. Bertozzi et al. [Bertozzi et al., 2007a, Bertozzi et al., 2007b] ont proposé un modèle pour l'inpainting des images binaires en modifiant l'équation de Cahn-Hilliard. Ils ont établi que le flux de quatrième ordre gradient du modèle de l'inpainting Esedoglu-Shen [Esedoglu et Shen, 2002], basé sur le modèle d'images de Mumford-Shah-Euler, a quelques points communs avec l'équation de Cahn-Hilliard, pour laquelle des solutions rapides sont disponibles. Le modèle à deux échelles de Cahn-Hilliard réussit à relier les contours à travers de larges régions. Récemment, Dobrosotskaya et Bertozzi [Dobrosotskaya et Bertozzi, 2008] ont proposé un inpainting par ondelettes qui adapte l'équation du second ordre d'Allen-Cahn et maintient des vitesses d'inpainting comparables. En outre, leur modèle a été étendu aux images en niveaux de gris.

Généralement, ces méthodes offrent les meilleurs résultats sur les régions de petites tailles. Inversement, lorsque les alentours du trou sont formés de structures stochastiques comme une texture, ce type de méthodes tend à fournir un contenu dégradé inapproprié dans une zone texturée. Ce défaut est dû à la nature locale de ces méthodes, inadaptées à la reconstruction de structures stochastiques.

### 3.2.2 Méthodes basées sur la synthèse de textures

Le deuxième groupe d'algorithmes d'inpainting est basé sur la reconstruction des régions manquantes par *synthèse de textures* grâce à des modèles statistiques du contenu de l'image. L'échantillonnage non-paramétrique [Efros et Leung, 1999, Efros et Freeman, 2001] est utilisé dans les techniques basées textures. Introduites à l'origine dans le contexte de la synthèse de textures, ces méthodes dites de *copier-coller* sont désormais couramment utilisées pour la reconstruction de parties manquantes d'images ou de vidéos.

Les méthodes de rééchantillonnage non-paramétrique, dites également méthodes *par l'exemple* [Zhu et al., 1997, Zhu et al., 1998, Criminisi et al., 2003, Criminisi et al., 2004, Do et al., 2006], ont prouvé leur capacité à reconstruire des structures stochastiques dans les zones occultées, mais aussi (dans une certaine mesure) la géométrie. Ces méthodes sont basées sur la recherche et la copie de patchs similaires dans le voisinage du trou. Les données se reconstruisent du bord extérieur au centre du trou. En général, une stratégie de remplissage est employée. Celle-ci influence le résultat obtenu. Criminisi et al. [Criminisi et al., 2004] se base sur une mesure de gradient au bord du trou afin de déterminer l'ordre de remplissage. Les patchs des zones à fort gradient sont traités en premier.

Toutefois, la reconstruction des structures géométriques par l'exemple est limitée à des régions de taille relativement modeste et à des bords rectilignes. Les méthodes par l'exemple, par le caractère local de leurs définitions, sont limitées dans la restauration de la géométrie à grande échelle. Des approches multi-échelles peuvent minimiser cet inconvénient. Une caractéristique des travaux de Criminisi et al. [Criminisi et al., 2004] et de Efros et al. [Efros et Leung, 1999, Efros et Freeman, 2001] est le remplissage d'un trou en une seule passe sans modification ultérieure de la valeur mise en place. L'inconvénient de cette démarche est de ne pas pouvoir rectifier le remplissage dans le cas où un patch non optimal aurait été choisi. Ceci peut notamment arriver lorsqu'un trou coupe un contour. Le contour sera reconstruit de part et d'autre du trou sans aucune certitude de joindre les deux extrémités au centre. Ceci rend d'autant plus délicat la stratégie de remplissage du trou, qui est une difficulté en soi.

### 3.2.3 Méthodes non locales

Il est important de rappeler que les images naturelles comportent aussi bien des structures géométriques (comme les contours, les angles, etc.) que des structures stochastiques (motifs répétés, contenu statistique, etc.). Par conséquent, une solution d'inpainting doit, dans l'idéal, pouvoir reconstruire aussi bien ces deux composantes. C'est le but de cette classe de méthodes qui tendent à unifier les deux approches *géométriques* et *basées textures* dans un même cadre afin de tirer profit des points forts de ces deux classes.

Les travaux récents sur la non localité ont mis en évidence que la multitude des candidats utilisés pour le calcul de la valeur d'un nœud est un avantage à exploiter pour obtenir des reconstructions plus rigoureuses. Depuis les travaux sur le filtrage NL-means, de nombreuses méthodes variationnelles et basées sur les EDP ont été proposées. Elles ont révélé leur efficacité à préserver les structures stochastiques. Kindermann, Osher et Jones [Kindermann et al., 2005] ont été les premiers à interpréter le NL-means comme une régularisation basée sur des fonctions non locales. Le NL-means a inspiré plusieurs contributions dans le domaine de l'inpainting, la colorisation et la segmentation semi-supervisée tels que la reconstruction super-résolution [Protter et al., 2009] ou l'adaptation du NL-means à l'inpainting par l'exemple [Wong et Orchard, 2008].

Des travaux récents tendent à unifier les approches locales et non locales. La notion des EDP non locales a émergé. Par exemple, Gilboa et Osher [Gilboa et Osher, 2007] ont proposé une approche de régularisation non locale continue et ses applications au filtrage et à l'inpainting. Un cadre de travail variationnel d'inpainting non local a été aussi présenté dans [Arias *et al.*, 2009]. Un cadre discret de régularisation non locale d'images et de variétés est proposé dans [Elmoataz *et al.*, 2008]. Dans ce cadre, nous présentons une approche d'inpainting d'images et de vidéos unifiant les méthodes locales et non locales [Ghoniem *et al.*, 2010] comme nous le verrons dans le chapitre suivant.

# CONTRIBUTIONS À L'INTERPOLATION ET L'INPAINTING DES IMAGES ET DES VIDÉOS

## Sommaire

|   |    |
|---|----|
| <b>4.1 Inpainting par régularisation</b> . . . . .                                    | 47 |
| 4.1.1 Formulation variationnelle du problème d'inpainting . . . . .                   | 48 |
| 4.1.2 Algorithmes et stratégies d'inpainting . . . . .                                | 49 |
| 4.1.3 Résultats expérimentaux . . . . .   | 53 |
| <b>4.2 Fonctions <math>p</math>-harmoniques et interpolation sur graphe</b> . . . . . | 61 |
| 4.2.1 Existence et unicité d'une solution . . . . .                                   | 61 |
| 4.2.2 Formulation équivalente . . . . .   | 63 |
| 4.2.3 Cas particulier de $p = \infty$ . . . . .                                       | 64 |
| 4.2.4 Résultats et applications . . . . .   | 66 |

## Résumé

Dans ce chapitre, nous proposons deux approches d'interpolation. Dans un premier temps, à la section 4.1, nous présentons l'extension des méthodes d'interpolation continues [Chan et Shen, 2005b] au domaine de l'inpainting. Nous traitons le résultat d'une première interpolation par régularisation itérative afin d'améliorer la qualité de l'interpolation. À la section 4.2, nous nous intéressons aux fonctions  $p$ -harmoniques qui constituent un cadre général d'interpolation. Ce dernier permet la réalisation de nombreuses applications dont l'inpainting sur lequel nous allons nous focaliser. Nous montrons que ce cadre permet d'unifier les approches géométriques et les approches basées sur la synthèse de texture. Nous évoquerons en particulier le cas  $p = 2$  et nous introduirons le cas nouveau de  $p = \infty$ .

## 4.1 Inpainting par régularisation

Cette section présente l'extension de notre cadre de régularisation discrète à l'inpainting des images et des vidéos. Le processus d'inpainting consiste à remplir les parties manquantes avec les données les plus appropriées en vue d'obtenir des zones reconstruites harmonieuses et difficilement détectables qui se fondent dans le reste de la scène. Nous proposons de restaurer les données manquantes en utilisant l'approche de régularisation.

Cette approche présente l'avantage de compléter les parties manquantes en harmonie avec les données dans le voisinage de telle sorte qu'il n'y a pas de frontière abrupte entre

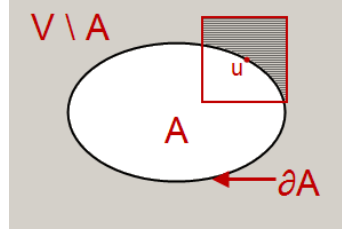


FIGURE 4.1 – Illustration de l'inpainting de la région  $A$  à partir des données connues ( $V \setminus A$ ) en comparant le patch autour de  $u$  (la zone hachurée) avec les patches de même forme.

les données existantes et la partie reconstruite. En outre, alors que certaines méthodes de l'état de l'art impliquent un surlissage de la texture, cet écueil est évité dans notre travail en utilisant une méthode de régularisation non locale. Par conséquent, nous observons empiriquement que nos résultats restituent aussi bien les contours que la texture.

#### 4.1.1 Formulation variationnelle du problème d'inpainting

La première approche que nous proposons pour l'inpainting peut être considérée comme une spécialisation du cadre de régularisation présentée au chapitre 2. Au démarrage du processus d'inpainting, de nouvelles valeurs sont calculées pour les parties manquantes. Dans ces trous, il n'y a pas de valeur initiale et, par conséquent, le paramètre de fidélité  $\lambda$  est fixé à 0. Soit  $A$  l'ensemble des nœuds correspondant aux trous à remplir (voir figure 4.1).

La formulation de régularisation discrète générale (2.25) peut être écrite :

$$f^* = \min_{f \in \mathcal{H}(V)} \left\{ \frac{1}{p} \sum_{v \in V} \|\nabla f(v)\|^p + \frac{\lambda(v)}{2} \|f - f^0\|_{\mathcal{H}(V)}^2 \right\} \quad (4.1)$$

où,

$$\lambda(v) = \begin{cases} \lambda = \text{constante}, & \text{si } v \in V \setminus A \\ 0, & \text{sinon.} \end{cases}$$

Pour l'estimation des premières valeurs, la formulation de régularisation discrète (4.1) se réduit à :

$$f^* = \min_{f \in \mathcal{H}(V)} \left\{ \frac{1}{p} \sum_{v \in V} \|\nabla f(v)\|^p \right\}.$$

Le système d'équations (2.27) devient pour ce cas particulier :  $\forall (u, v) \in A^2$ ,

$$\begin{cases} f^{(0)} = f^0 \\ \gamma^{(k)}(u, v) = w(u, v) \left( \|\nabla f^{(k)}(v)\|^{p-2} + \|\nabla f^{(k)}(u)\|^{p-2} \right) \\ f^{(k+1)}(v) = \frac{\sum_{u \sim v} \gamma^{(k)}(u, v) f^{(k)}(u)}{\sum_{u \sim v} \gamma^{(k)}(u, v)} \end{cases} \quad (4.2)$$

où  $\gamma^{(k)}$  est la fonction  $\gamma$  à l'étape  $k$ .

(4.3)

Lorsque chaque élément de  $A$  a été évalué une première fois, le processus de régularisation est itéré avec  $\lambda \geq 0$  afin de raffiner les valeurs obtenues par l'interpolation.

## 4.1.2 Algorithmes et stratégies d'inpainting

### 4.1.2.1 Construction du graphe

Nous considérons une image ou une séquence vidéo comme une fonction  $f$  définie sur les sommets d'un graphe pondéré  $G_{k_1, k_2, k_3} = (V, E, w)$ , où  $k_1, k_2, k_3 \in \mathbb{N}^3$ . Un sommet  $v$  est défini par un triplet  $(i, j, t)$  où  $(i, j)$  indique la position spatiale du sommet et  $t$ , qui est le numéro de l'image, précise la position temporelle du sommet dans la séquence. Nous désignons par  $u \sim v$  un sommet  $u$  qui appartient au voisinage de  $v$  selon la définition suivante :

$$N_{k_1, k_2, k_3}(u) = \{v = (i', j', t') \in V : |i - i'| \leq k_1, |j - j'| \leq k_2, |t - t'| \leq k_3\}.$$

De même, nous étendons la définition du patch aux vidéos pour obtenir des patches 3D. Un patch autour d'un sommet  $v$  est une boîte de taille  $r_x \times r_y \times r_t$ , notée  $B(v)$ . Puis, nous associons à ce patch un vecteur caractéristique défini par :

$$F(f^0, v) = f^0(u), \quad u \in B(v).$$

Pour être sûr que chaque voisinage contient plus d'un patch, les relations suivantes doivent être respectées :  $k_1 > \alpha_1 * r_x$ ,  $k_2 > \alpha_2 * r_y$  et  $k_3 \geq r_t$ . En pratique, nous utilisons  $\alpha_1 = \alpha_2 = 3$ . En effet, il est indispensable d'avoir  $r_x$ ,  $r_y$  et  $r_t$  assez petits par rapport à  $k_1$ ,  $k_2$  et  $k_3$  pour être sûr que le voisinage non local contient un nombre significatif de patches.

La fonction poids  $w$  associée à  $G_{k_1, k_2, k_3}$  fournit une mesure de la distance inter-sommets qui peut simplement incorporer des caractéristiques locales, semi-locales ou non locales en fonction de la topologie du graphe. Nous considérons les deux fonctions de poids :

$$w_L(u, v) = \exp\left(-\frac{|f(u) - f(v)|^2}{2\sigma_d^2}\right), \text{ et}$$

$$w_{NL}(u, v) = w_L(u, v) \cdot \exp\left(-\frac{\|F(f^0, u) - F(f^0, v)\|^2}{h^2}\right),$$

où  $\sigma_d^2$  dépend des variations de  $|f(u) - f(v)|$  sur le graphe. La constante  $h$  peut être estimée en utilisant l'écart-type en fonction des variations de  $\|F(f^0, u) - F(f^0, v)\|$  sur le graphe.

$w_L(u, v)$  est une mesure de la différence entre les valeurs de  $f(u)$  et  $f(v)$ , et est utilisée dans l'*approche locale* du débruitage. En plus de la différence entre les valeurs,  $w_{NL}(u, v)$  inclut une estimation de la similitude des caractéristiques comparées en mesurant une distance  $\mathcal{L}^2$  entre les patches autour des nœuds  $u$  et  $v$ . C'est l'*approche non locale*.

Une fois le graphe est construit, le problème de minimisation est considéré.

### 4.1.2.2 Algorithmes et stratégies

Notre méthode consiste à remplir un trou en partant de son contour extérieur jusqu'à son centre de manière récursive. Ainsi, les régions manquantes sont reconstituées en remplissant une série de contours imbriqués. Pour un pixel donné du contour extérieur, nous calculons une nouvelle valeur à partir des données connues dans un volume centré sur le pixel courant, que nous appelons la *fenêtre de recherche*. Lorsque tout le contour extérieur est traité, il est enlevé de  $A$ .

Cela signifie que nous ne mettons pas à jour l'ensemble  $A$  jusqu'à ce qu'un contour ait été complètement traité. En d'autres termes, nous n'incluons pas la valeur calculée d'un pixel dans l'estimation des autres pixels appartenant au même contour. Ceci permet de réduire le risque de propagation d'erreurs dans nos calculs. Lorsqu'un contour est traité, il n'appartient plus à  $A$  et il est considéré comme données connues pouvant être prises en compte pour traiter les autres nœuds manquants. Quand le processus d'inpainting progresse, le masque se réduit dynamiquement petit à petit et devient par la suite vide. À ce stade, tous les trous sont remplis.

Plusieurs stratégies peuvent être mises en place.

**Première stratégie** La première stratégie consiste à remplir complètement les trous puis à régulariser les données reconstruites de manière itérative (voir algorithme 4.1.2.2).

---

**Algorithme 4.1** Inpainting non local - stratégie 1
 

---

ENTRÉES : Un graphe  $G$ , le nombre d'itérations  $N \in \mathbb{N}^+$

**tantque**  $N \neq 0$  **faire**

**pour tout**  $o$ , contour imbriqué du trou **faire**

**pour tout**  $p \in o$  **faire**

*Attribuer une valeur à  $p$  :*

        Construire le patch et le voisinage de  $p$

        Comparer le patch centré en  $p$  avec tous les patchs de la fenêtre de recherche.

        Calculer une valeur suivant la formule (4.2) et l'attribuer à  $p$ .

**fin pour**

*Mettre à jour le graphe et le masque : Supprimer  $o$  du masque.*

**fin pour**

$N \leftarrow N - 1$

**fin tantque**

---

**Seconde stratégie** La seconde approche consiste à itérer la régularisation de chaque contour imbriqué dans le trou avant de commencer le remplissage du contour imbriqué suivant (voir algorithme 4.1.2.2).

---

**Algorithme 4.2** Inpainting non local - stratégie 2
 

---

ENTRÉES : Un graphe  $G$ , le nombre d'itérations  $N \in \mathbb{N}^+$

**pour tout**  $o$ , contour imbriqué du trou **faire**

**tantque**  $N \neq 0$  **faire**

**pour tout**  $p \in o$  **faire**

*Attribuer une valeur à  $p$  :*

        Construire le patch et le voisinage de  $p$

        Comparer le patch centré en  $p$  avec tous les patchs de la fenêtre de recherche.

        Calculer une valeur suivant la formule (4.2) et l'attribuer à  $p$ .

**fin pour**

**fin tantque**

*Mettre à jour le graphe et le masque : Supprimer  $o$  du masque.*

**fin pour**

$N \leftarrow N - 1$

---

Les algorithmes itératifs 4.1.2.2 ou 4.1.2.2 sont appliqués un nombre fixe de fois ou arrêtés par un critère de convergence. Nous considérons que la convergence est atteinte



quand l'Erreur Quadratique Moyenne (EQM ou  $MSE$ ) entre  $f^{k+1}$  et  $f^k$  est inférieure à un seuil défini par l'utilisateur.

#### 4.1.2.3 Comparaison et distance entre patches

Cette section détaille la façon de calculer une nouvelle valeur pour un nœud dans une zone manquante, en harmonie avec les données avoisinantes connues. À cette fin, nous considérons la distance entre patches. Cependant, à cause de la forme du trou, un patch peut prendre différentes formes. Une illustration de différents masques est donnée à la figure 4.2.

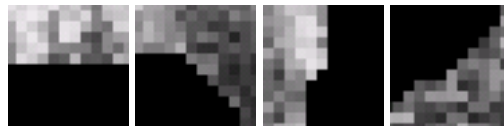


FIGURE 4.2 – Illustration de patches de formes différentes.

La distance entre patches définie pour mesurer la similitude entre les patches de référence et les autres patches de la fenêtre de recherche tient compte de la forme du patch traité et ne considère que les patches de même forme. Cette mesure constitue un moyen pratique de décrire les caractéristiques importantes des vecteurs pertinents pour l'application en question, tels que l'intensité ou la texture. La figure 4.3 illustre la discrimination opérée entre patches par la fonction poids.

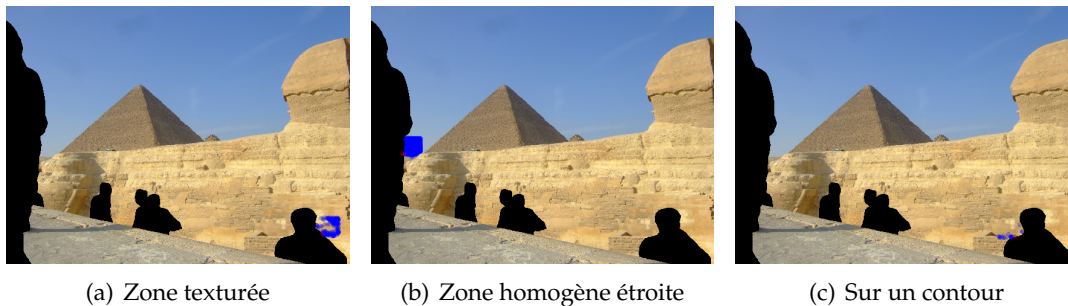


FIGURE 4.3 – Mise en valeur de la pertinence du poids non local entre patches masqués dans différentes situations. Les points (en bleu) ayant un poids important (supérieur à 0,5) avec le point traité (en rouge) respectent la forme des contours de la zone traitée et des zones manquantes. Paramètres : Fenêtre  $61 \times 61$  et patch  $13 \times 13$ .

Le principal défi est de respecter l'aspect général et de préserver les structures importantes et les détails. Ainsi, le choix des paramètres de notre processus doit être fait en fonction du contenu à traiter.

En effet, il est à noter que les valeurs des paramètres choisies judicieusement, en particulier la taille de la fenêtre de recherche et la taille du patch, sont essentielles pour la qualité des résultats. Des valeurs inappropriées fournissent des résultats de piètre qualité. Pour surmonter cette difficulté, nous proposons une variante *multi-résolution* de l'inpainting.

**Stratégies multi-résolution** Nous introduisons l'utilisation des patches de différentes tailles pour qu'ils s'adaptent mieux à l'homogénéité de la zone de données à l'étude. De larges patches sont adéquats pour les zones homogènes, tandis que les détails et la texture

ont tendance à exiger un patch de taille variable en fonction du contenu représenté par ces données. L'utilisation de patches à différentes échelles pour sélectionner les candidats les plus proches améliore la robustesse de notre méthode. Cette variante a l'avantage de sélectionner automatiquement la taille appropriée du patch, et est particulièrement appropriée pour les vidéos ayant des variations de zoom entre les images. La multi-résolution peut opérer à différents niveaux :

1. inclure la multi-résolution dans le calcul des poids pour améliorer la discrimination des candidats,
2. traiter le trou entièrement, de la résolution la plus fine à la résolution la plus grossière.

### 4.1.2.4 Analyse des stratégies et choix des paramètres

**Parcours et ordre de visite** L'approche de remplissage consistant à remplir le trou ligne par ligne (ou colonne par colonne) est moins intéressante que l'approche de remplissage par contours imbriqués pour les raisons suivantes.

- L'ordre de visite influence le résultat et favorise un côté du bord par rapport à l'autre. Par exemple, lorsque le remplissage se fait ligne par ligne et de gauche à droite, le front gauche va évoluer rapidement vers le centre et influencera le contenu du trou tandis que le front droit « subira » l'influence du front gauche. Ceci est dû au fait que chaque pixel restauré est utilisé pour le calcul de la valeur de ces voisins. Par conséquent, les premiers pixels restaurés orienteront le remplissage du trou en fonction de leur contenu. Si l'ordre de remplissage est modifié, alors le résultat de l'inpainting sera certainement différent.
- Il existe un risque plus important d'avoir un résultat visuellement aberrant sous forme d'étirement d'un côté et aboutissant à une discontinuité avec le bord en face. Ceci est une conséquence de l'ordre de visite expliqué au point précédent.
- La convergence est beaucoup plus lente. En effet, l'approche concentrique permet d'avancer symétriquement de part et d'autre du trou pour se joindre au centre. Au contraire, l'approche point par point parcourt le trou d'un bord à l'autre.

**Formes des patches** L'exclusion des patches trop « vides » améliore la qualité du résultat. En effet, si un grand patch est quasi vide, cela veut dire qu'on ne compare que quelques pixels uniquement, ce qui est en contradiction avec l'approche non locale dans laquelle des blocs représentatifs sont comparés. Par conséquent, si un patch est trop creux, alors il est rejeté.

**Influence du nombre d'itérations** Le remplissage itératif du trou présente le grand avantage de pouvoir corriger un premier choix inapproprié ou d'améliorer un choix parfait, par régularisation itérée. Contrairement aux travaux en une seule passe que nous avons évoqués au chapitre 3, un faux positif ou un faux négatif ne condamne pas le résultat. Nous observons ainsi l'évolution de la région reconstruite au fil des itérations jusqu'à l'arrivée à un état stable. Dans nos expériences, nous utilisons la EQM entre deux itérations consécutives afin de détecter l'absence de modification significative du résultat.

La figure 4.4 montre la reconstruction d'un trait. La première itération n'est pas satisfaisante. Néanmoins, lors des itérations suivantes, la partie manquante du trait est reconstruite petit à petit.

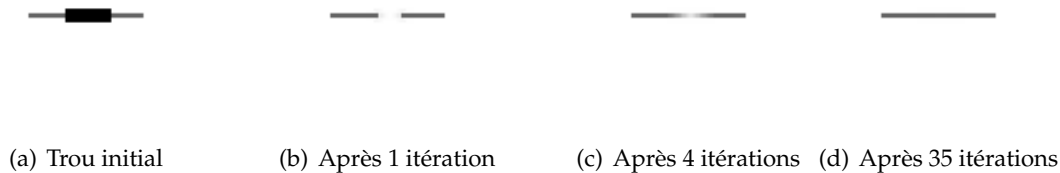


FIGURE 4.4 – Le fait d’itérer corrige un premier remplissage inexact.

**Influence de  $h$**   $C'$  est un paramètre d'échelle comme dans toutes les méthodes à noyau. Si  $h$  est trop grand, le résultat est flou. Il faut donc l'ajuster comme si l'on ajuste la focale d'une lentille afin d'obtenir une image nette. Différentes heuristiques ont été proposées et la question reste ouverte. Par conséquent, nous avons préféré rester à un réglage manuel de ce paramètre.

**Influence de  $p$**  Le paramètre  $p$  est le degré de lissage. Le cas  $p = 1$  correspond à la médiane,  $p = 2$  correspond à la moyenne et  $p = \infty$  correspond à  $\frac{\min + \max}{2}$ . Nous donnons plusieurs exemples des différentes valeurs de  $p$  dans les résultats expérimentaux, ci-après.

*Remarque 4.* L'optimisation de l'algorithme de régularisation par sélection aléatoire de patchs en plus du voisinage local est également valable pour l'inpainting. Cependant nous ne l'utilisons pas pour les expérimentations présentées dans ce chapitre.

#### 4.1.2.5 Liens avec des travaux connus

Le cadre ci-dessus unifie et englobe plusieurs techniques spécifiques présentes dans la littérature. En fait, en considérant des valeurs particulières des paramètres, nous retrouvons les résultats qui ont été établis dans le traitement de l'image. Pour  $p = 2$  et une itération, la méthode non locale est équivalente au filtre NL-means [Buades *et al.*, 2008] qui a été adapté à l'inpainting [Wong et Orchard, 2008]. Avec  $p = 1$  et  $w = 1$ , on obtient l'inpainting par variation totale (VT) locale anisotrope [Chan et Shen, 2005b]. Avec  $p = 1$ , c'est l'inpainting VT anisotrope non local. Notre méthode peut être considérée comme une extension de [Efros et Leung, 1999]. En fait, si nous construisons le graphe des  $k$ -plus proches voisins avec  $k = 1$  et une distance de patchs entre les nœuds, nous obtenons la même approche. Toutefois, dans notre algorithme, nous pouvons considérer des valeurs différentes pour  $k$ .

#### 4.1.3 Résultats expérimentaux

Nous présentons maintenant quelques tests qui démontrent l'efficacité de notre méthode d'inpainting. À cette fin, plusieurs exemples ont été sélectionnés, et la distance entre patchs a été déterminée en fonction de l'intensité. Elle consiste à remplir les parties manquantes d'une image ou d'une vidéo avec les données les plus appropriées en vue d'obtenir des zones reconstruites harmonieuses et difficilement détectables.

## 4.1.3.1 Images de synthèse en niveaux de gris

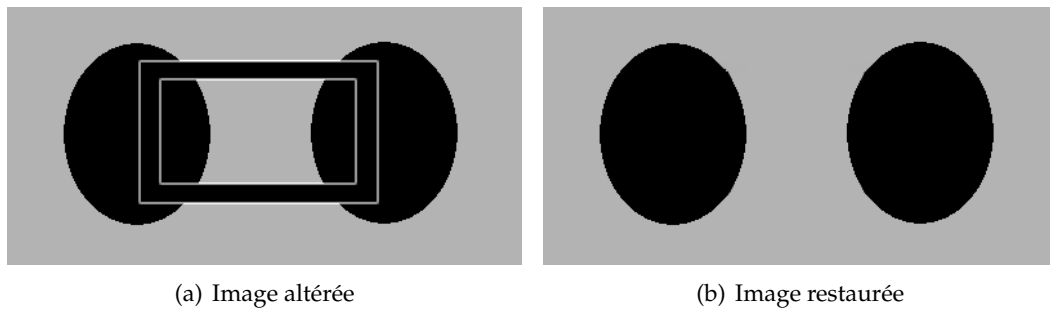


FIGURE 4.5 – Restauration de la géométrie avec  $p = 2$ , une fenêtre de recherche  $15 \times 15$  et un patch  $3 \times 3$ .

La figure 4.5 illustre l'efficacité de notre algorithme à la reconstruction de la forme géométrique des deux ovales et de leurs niveaux de gris.

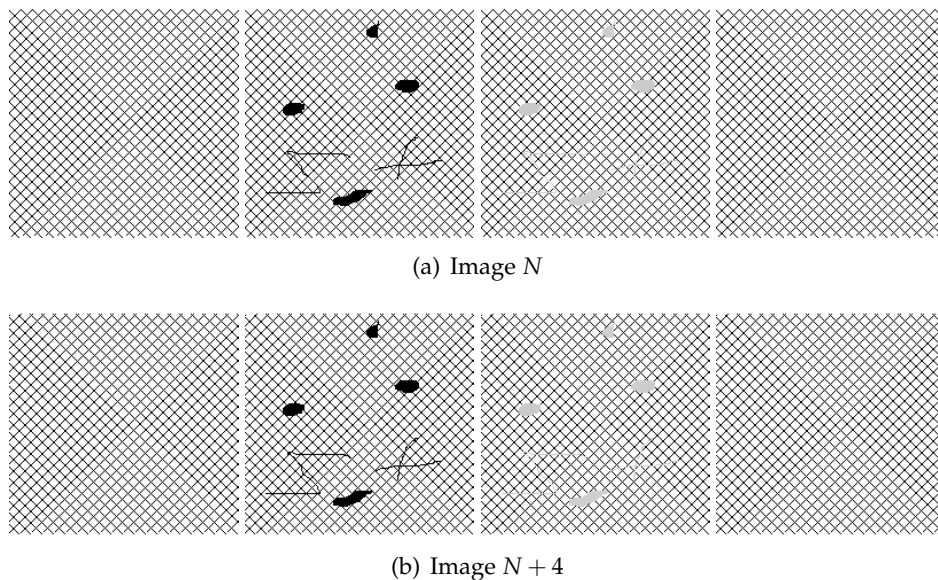


FIGURE 4.6 – Inpainting d'une vidéo synthétique. Chaque image de cette vidéo est une translation de l'image précédente. De gauche à droite : la vidéo d'origine, la vidéo altérée, le résultat de l'inpainting local et celui de l'inpainting non local.

Dans le premier exemple, notre algorithme a été appliqué à une image de synthèse. La figure 4.6 montre que l'image est parfaitement reconstruite en utilisant notre algorithme non local, alors que l'approche d'inpainting local a donné des résultats médiocres. Ce test offre une vérification de base de l'algorithme non local.

## 4.1.3.2 Images de texture en niveaux de gris

Les résultats sur des images de textures sont rapportés sur les figures 4.7, 4.8, et 4.9. Notez que les zones reconstruites ne sont pas détectables et se fondent harmonieusement avec les données non corrompues.

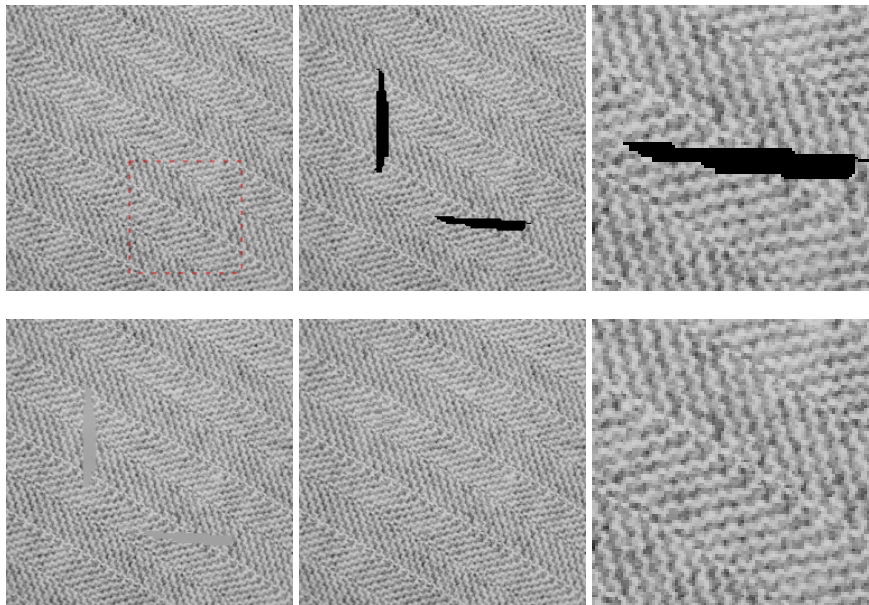


FIGURE 4.7 – Inpainting d’une image texturée (*weave*) avec une fenêtre de recherche  $61 \times 61$  et un patch  $31 \times 31$ . De gauche à droite : sur la première ligne, l’image d’origine, l’image corrompue et une zone agrandie de l’image ; sur la seconde ligne, le résultat de l’inpainting local, celui de la méthode non locale avec  $p = 2$  et une région agrandie de ce dernier.

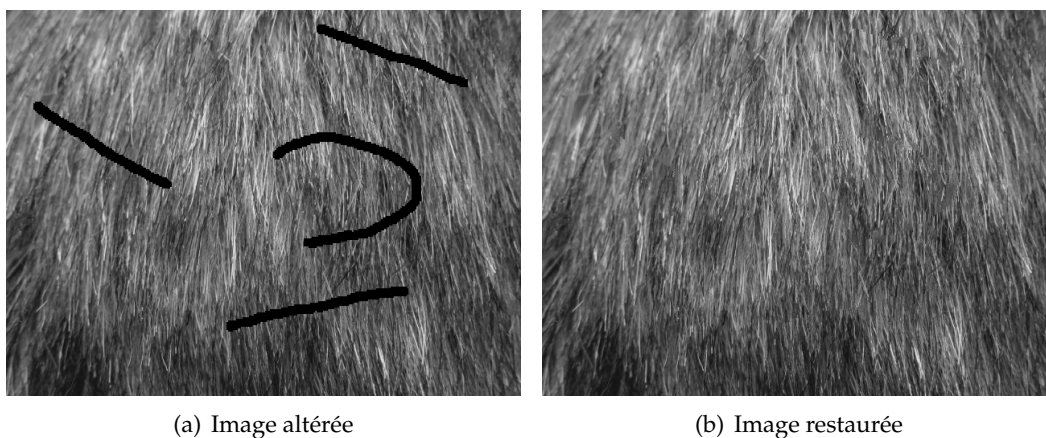


FIGURE 4.8 – Inpainting de textures avec  $p = 2$ , une fenêtre de recherche  $17 \times 17$  et un patch  $7 \times 7$ .

#### 4.1.3.3 Inpainting unifié de la géométrie et de la texture en niveaux de gris

Nous pouvons voir sur la figure 4.10 que notre algorithme réussit à reconstruire simultanément la forme géométrique et le contenu texturé des deux ovales.

Les tests de l'inpainting sur des images complexes contenant des zones d'homogénéité variable fournissent également des résultats très encourageants. La figure 4.11 expose la restauration de l'image connue de Barbara. Nous avons volontairement retiré des zones dans des régions homogènes, des régions de texture, et des zones d'homogénéité variable, pour tester la robustesse de notre méthode.

#### 4.1.3.4 Inpainting d'images en couleur

Nous présentons dans cette sous-section le résultat de notre algorithme pour l'inpainting des images en couleurs. Nous avons choisi de traiter chacune des trois composantes RGB indépendamment des deux autres. Nous obtenons des résultats satisfaisants de cette façon. Quelques aberrations chromatiques peuvent apparaître dans certains cas. Pour y remédier, une solution consiste à lier les trois composantes par un modèle couleur [Bertalmio *et al.*, 2000].

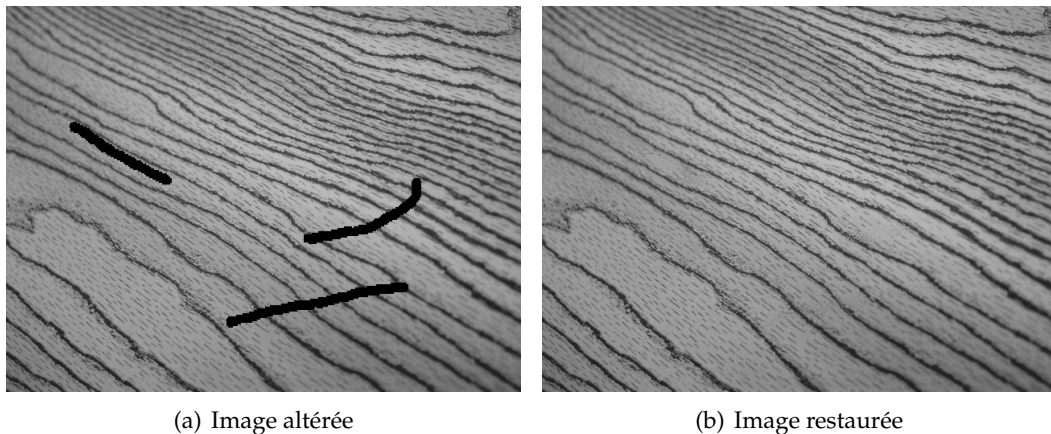


FIGURE 4.9 – Inpainting de texture avec  $p = 2$ , une fenêtre de recherche  $27 \times 27$  et un patch  $11 \times 11$ .

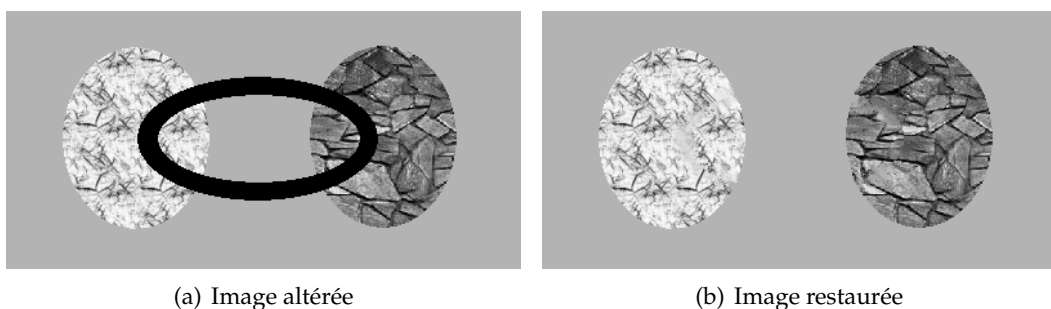


FIGURE 4.10 – Restauration unifiée de la géométrie et de deux textures différentes avec  $p = 2$ , une fenêtre de recherche  $41 \times 41$  et un patch  $3 \times 3$ .



À la figure 4.12, le grillage de protection, indispensable pour la sécurité des visiteurs, est indésirable sur le cliché souvenir de l'ours. En appliquant notre algorithme, les zones correspondants au grillage sont reconstruites et une image plus agréable est obtenue.

À la figure 4.13, nous avons enlevé les personnes du cliché afin d'avoir une vue dégagée du site des pyramides. Les zones supprimées sont très larges. Nous pouvons noter que la texture « briques » de pierre est correctement étendue (en bas à droite de l'image) au trou adjacent mais également le gradient de bleu du ciel est harmonieusement reconstruit (en haut à gauche). Néanmoins, nous pouvons observer deux défauts dans le résultat. Le premier est l'aberration chromatique que nous avons évoquée ci-dessus. Le second est l'aspect peu naturel près de la bordure de l'image. Ceci est dû au manque de candidat dans ces zones périphériques combiné à la taille importante du trou. Les zones reconstruites reposent uniquement sur des valeurs estimées et non pas des données préservées de l'image.



FIGURE 4.11 – Inpainting de l'image de *Barbara* avec  $p = 2$ , une fenêtre de recherche  $21 \times 21$  et un patch  $5 \times 5$ . De gauche à droite : l'image d'origine, l'image corrompue, le résultat de l'inpainting local et celui de l'inpainting non local.

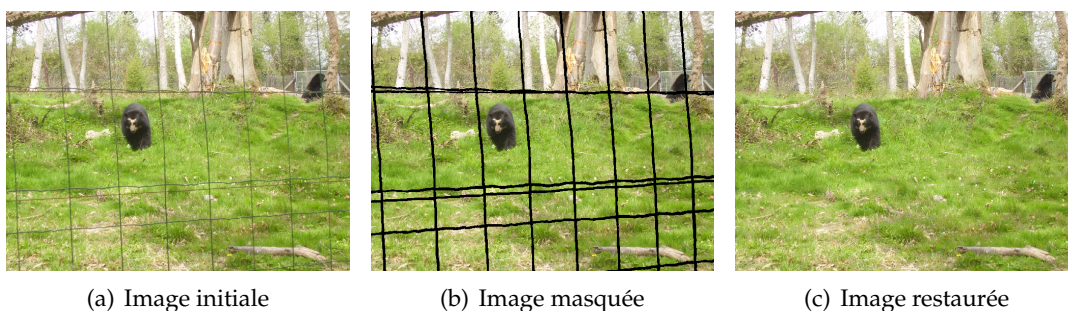


FIGURE 4.12 – Restauration d'une image couleur réelle avec  $p = 2$ , une fenêtre de recherche  $21 \times 21$  et un patch  $11 \times 11$ .

#### 4.1.3.5 Inpainting des séquences d'images

À la figure 4.14, il est possible d'observer la contribution de la redondance temporelle dans le processus d'inpainting. Nous avons altéré une image de la vidéo de *Suzie*. Les résultats démontrent la grande efficacité de notre approche non locale à reconstruire la texture et les structures géométriques de la vidéo. Par exemple, des structures fines telles que les cheveux de Suzie ou des zones homogènes comme le fond sont aussi bien récupérés par l'approche non locale tandis que l'approche locale montre ses limites.

Pour tester notre algorithme sur la suppression de texte sur des vidéos, nous avons ajouté un texte statique à la scène *foreman*. Nous pouvons observer dans la figure 4.15 que le texte est supprimé correctement et le résultat est visuellement correct. Par conséquent, notre algorithme peut être utilisé pour la suppression des sous-titres inscrustés ou tout texte indésirable. De même, notre méthode peut être avantageusement utilisée pour enlever des objets sur des séquences vidéo. Dans la figure 4.16, nous avons retiré la balle de la vidéo.

Nos tests sur plusieurs corpi vidéo montrent que la suppression de plusieurs petites zones est plus facile que la suppression d'une grande zone unique de même surface. En effet, dans le cas où les petites régions sont dispersées partout dans la vidéo, de nombreuses parties connues existent tout autour et pourraient être utilisées pour reconstituer les parties manquantes. Au contraire, quand une grande zone compacte manque, la plupart des candidats utilisés pour remplir la partie intérieure du trou seront des valeurs reconstruites, ce qui conduit à une moindre qualité. Une façon de surmonter cette difficulté est de considérer une fenêtre de recherche importante, mais au prix fort d'un coût de calcul plus élevé.

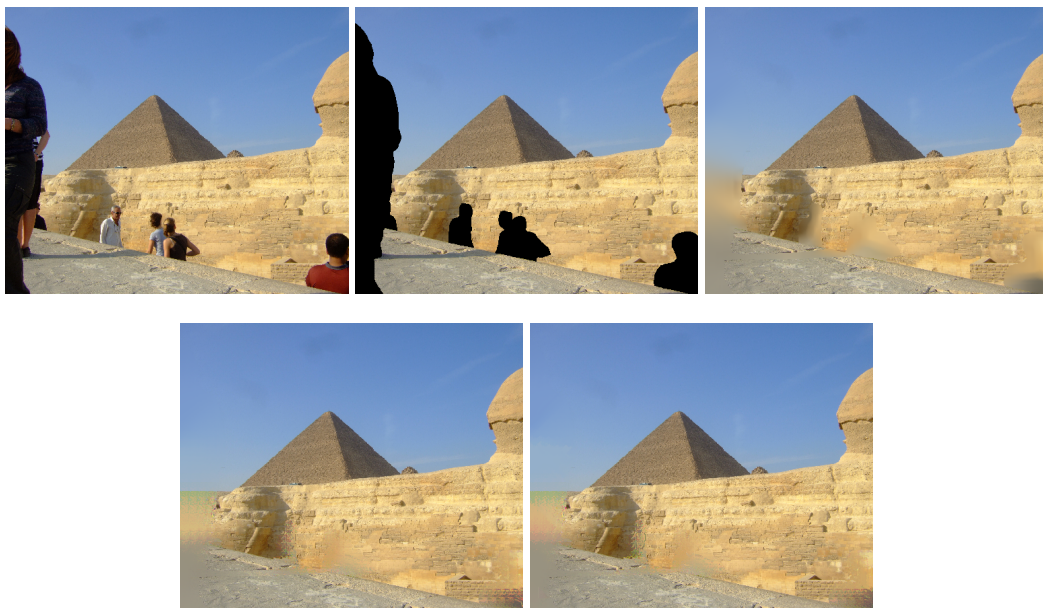


FIGURE 4.13 – Inpainting de larges surfaces, sur une image de la pyramide de Gizeh, avec une fenêtre de recherche  $61 \times 61$  et un patch  $13 \times 13$  et après 10 itérations. De gauche à droite, l'image d'origine, l'image corrompue, le résultat de l'inpainting local et celui de l'inpainting non local pour  $p = 2$  et  $p = 1$ .





FIGURE 4.14 – Sur la première ligne, les images d’origine. Sur les deuxième et troisième lignes, de gauche à droite, l’image altérée, le résultat de l’inpainting local et celui de l’inpainting non local avec une fenêtre de recherche  $21 \times 21$  et un patch  $6 \times 6$ .

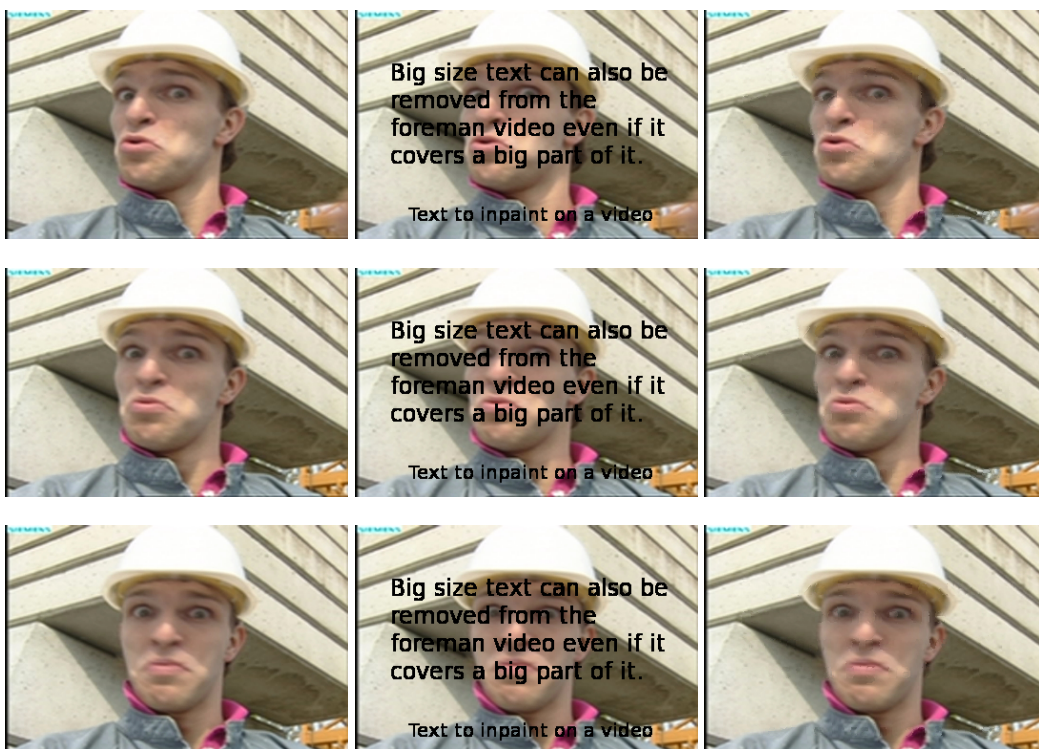


FIGURE 4.15 – Suppression de texte dans une vidéo couleur (*foreman*). De gauche à droite, les images de la vidéo d’origine, celles de la vidéo corrompue et le résultat de notre méthode d’inpainting non locale.

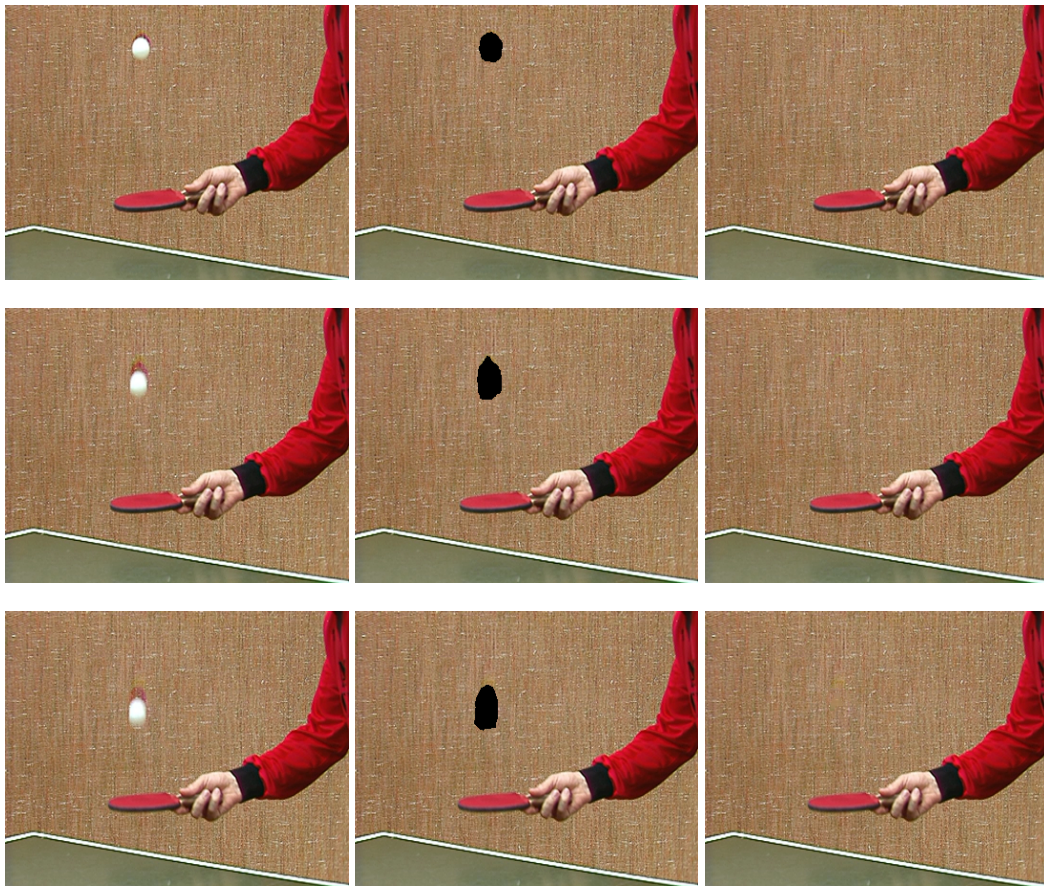


FIGURE 4.16 – Suppression d’objets dans une vidéo couleur (*tennis*). De gauche à droite, les images de la vidéo d’origine, la vidéo corrompue et le résultat de notre méthode d’inpainting non local (la vidéo sans la balle).

## 4.2 Fonctions $p$ -harmoniques et interpolation sur graphe

À la section précédente, nous avons présenté l'extension des méthodes d'interpolation continues au domaine discret. Dans cette section, nous nous intéressons aux fonctions harmoniques qui constituent un cadre d'interpolation général permettant la résolution d'un certain nombre d'applications dont l'inpainting.

Cette section aborde le problème de l'interpolation des données sur les graphes de topologie arbitraire et montre certaines applications sur l'image. Notre contribution réside principalement en deux points.

Premièrement, nous fournissons une nouvelle méthode pour résoudre le problème  $p$ -Dirichlet avec des fonctions  $p$ -harmoniques sur les graphes de topologie arbitraire. Des algorithmes robustes sont présentés pour les applications d'interpolation et d'inpainting d'images. Un des principaux avantages de notre approche est l'unification des approches locales et non locales de ces traitements dans un cadre discret qui le rend applicable à toutes les données représentées sur graphes. En plus de l'unification du traitement des structures géométriques et stochastiques, elle offre un cadre discret qui dépasse l'inpainting pour englober les problématiques d'inpainting, d'interpolation et de segmentation semi-supervisée sur graphes. En effet, notre approche par fonctions  $p$ -harmoniques inclut les méthodes par coupes de graphes (*graph cuts*) ou marches aléatoires (*random walks*).

Deuxièmement, nous introduisons, pour la première fois, le cas spécifique de  $p = \infty$  sur les graphes de topologie arbitraire. Dans les domaines continus, les fonctions  $\infty$ -harmoniques ont de bonnes propriétés et ont montré leur efficacité. Par conséquent, notre extension de l' $\infty$ -Laplacien ouvre de nombreuses pistes pour les méthodes de traitement d'images dans les domaines discrets.  $p = \infty$  fournit des résultats similaires à  $p = 2$  tout en étant plus rapide que les autres valeurs en termes de calculs.

### 4.2.1 Existence et unicité d'une solution

Mathématiquement, les problèmes d'interpolation consistent à trouver une fonction  $p$ -harmonique  $f$  qui minimise la  $p$ -énergie avec des conditions de bord données (des valeurs spécifiques sur la frontière  $\partial A$ ). Considérons  $1 \leq p < \infty$ .

**Définition 1.** Soit  $A \subset V$ . Une fonction  $f$  est  $p$ -harmonique sur  $A$  si elle minimise l'énergie :

$$\mathcal{R}_w^p(f, A) = \frac{1}{2p} \sum_{\substack{u \sim v \\ u \in A}} \|\nabla_w f(u)\|_p^p = \frac{1}{2p} \sum_{u \sim v} w(u, v)^{p/2} (|f(v) - f(u)|^p)^p, \quad (4.4)$$

pour toute fonction  $f$  dans  $A \cup \partial A$ , avec la contrainte de bord  $f = g$  sur  $\partial A$ .

Donc,

$$\mathcal{R}_w^p(f, A) \leq \mathcal{R}_w^p(g, A) \quad \text{et} \quad f = g \quad \text{sur} \quad \partial A. \quad (4.5)$$

Pour le cas spécifique de  $p = \infty$ , il s'agit de considérer l'extension de  $p$  à l'infini,  $\mathcal{R}_w^\infty(f, A)$ . C'est l'objet de la sous-section 4.2.3.

**Définition 2.** Soit  $S = \left\{ X/x_i \in [m, M], i \in \mathbb{N}, \text{ avec } m = \min_{\partial A} \text{ et } M = \max_{\partial A} \right\}$ .

**Proposition 3.** Pour toute fonction  $g$  dans  $\partial A$ , il existe une fonction unique  $f$  définie sur  $A \cup \partial A$  qui est  $p$ -harmonique sur  $A$  et qui vérifie  $f = g$  sur  $\partial A$ .

#### 4.2.1.1 Existence

On peut noter que s'il existe une solution  $f$ , alors le vecteur  $(f(u_1), \dots, f(u_n)) \in S$ .

En effet, si des nœuds  $u$  et  $v$  vérifient  $f(u) < m$  et  $f(v) > M$ , alors nous pouvons créer une nouvelle fonction  $f_{max}$  ayant comme valeur  $m$  et  $M$  en ces points.

Notons que l'énergie  $\mathcal{R}_w^p(f_{max}, A)$  décroît, ce qui contredit la définition.

Nous retrouvons un problème d'optimisation. Par des arguments classiques d'analyse, nous pouvons confirmer l'existence d'une solution  $f: S \subset \mathbb{R}^n \rightarrow \mathbb{R}$  continue sur un compact.

#### 4.2.1.2 Unicité

**Lemme 1.**

$$\forall a, b \in \mathbb{R}^2 \quad |a + b|^p \leq 2^{p-1} (|a|^p + |b|^p). \quad \left| \frac{a + b}{2} \right|^p \leq \frac{|a|^p + |b|^p}{2} \quad (4.6)$$

Il y a égalité si  $a = b$ .

Soit  $f(x) = |x|^p$ . L'inégalité (4.6) se réécrit :

$$f\left(\frac{a + b}{2}\right) \leq \frac{f(a) + f(b)}{2} \quad (4.7)$$

$f$  est strictement convexe et strictement croissante.

On a  $f(at + (1-t)b) < tf(a) + (1-t)f(b) \quad \forall t \in [0, 1]$ .

Pour  $t = \frac{1}{2}$ ,

$$f\left(\frac{a + b}{2}\right) < \frac{f(a) + f(b)}{2}$$

Quand  $a = b$ ,

$$f\left(\frac{a + a}{2}\right) = f(a) = \frac{f(a) + f(a)}{2}$$

*Preuve.* Supposons l'existence de deux solutions  $f$  et  $g: \mathcal{R}_w^p(f, A) = \mathcal{R}_w^p(g, A) = \bar{f}$ . Soit  $h = \frac{f+g}{2}$ .

$$\begin{aligned} \mathcal{R}_w^p(h, A) &= \sum_{u \sim v} w(u, v)^{p/2} \left( \left| \frac{f(v) + g(v)}{2} - \frac{f(u) + g(u)}{2} \right|^p \right)^p \\ &= \sum_{u \sim v} \frac{1}{2^p} w(u, v)^{p/2} (|f(v) - f(u) + g(v) - g(u)|^p)^p \\ &\leq \frac{1}{2^p} \sum_{u \sim v} w(u, v)^{p/2} (|f(v) - f(u)|^p + |g(v) - g(u)|^p)^p \quad \text{par le lemme (4.6)} \\ &\leq \frac{1}{2} \sum_{u \sim v} w(u, v)^{p/2} (|f(v) - f(u)|^p + |g(v) - g(u)|^p) = \frac{(\mathcal{R}_w^p(f, A) + \mathcal{R}_w^p(g, A))}{2} = \bar{f} \end{aligned} \quad (4.8)$$

Si  $\bar{f}$  est minimal, alors  $\mathcal{R}_w^p(h, A) = \bar{f}$ . Par le lemme 4.6, l'inégalité devient égalité.

$$\begin{aligned} f(u) - f(v) &= g(u) - g(v) \quad \forall u, v. \\ f(u) &= g(u) \quad \forall u, v \in \partial A. \end{aligned} \quad (4.9)$$

Comme le graphe est connexe, nous pouvons trouver pour chaque nœud  $u_0 \in A$  un chemin  $u_1, \dots, u_k \in (\partial A)^k$  tel que  $f(u_k) = g(u_k)$ .  $\square$

### 4.2.2 Formulation équivalente

**Proposition 4.** Une fonction  $f$  est  $p$ -harmonique ssi

$$\sum_{u \in V} \sum_{v \sim u} \sqrt{w(u, v)} |f(v) - f(u)|^{p-2} (f(v) - f(u)) (h(v) - h(u)) = 0, \quad (4.10)$$

pour toute fonction  $h/h(u) = 0 \quad \forall u \in \partial A$ .

L'idée de la preuve est inspirée de [Holopainen et Soardi, 1997].

*Preuve.* Soit  $f$  une fonction  $p$ -harmonique. On définit  $f_t = f + th \quad t \in \mathbb{R}$  et  $h(u) = 0 \forall u \in \partial A$ .  $f$  minimise l'énergie  $\mathcal{R}_w^p$  :

$$0 = \frac{d}{dt} (\mathcal{R}_w^p(f_t, A))_{t=0} = \frac{d}{dt} \left( \frac{1}{2p} \sum_{u \in V} \sum_{v \sim u} w(u, v)^{p/2} |f_t(v) - f_t(u)|^p \right)_{t=0} \quad (4.11)$$

$$= p \sum_{v \sim u} w(u, v)^{p/2} |f(v) - f(u)|^{p-2} (f(v) - f(u)) (h(v) - h(u)) \quad (4.12)$$

Supposons que  $f$  satisfait (4.10). Soit  $g$  une fonction définie sur  $A \cup \partial A$  et  $g = f$  sur  $\partial A$ . En appliquant l'équation (4.10) à  $h = f - g$  et en utilisant l'inégalité de Hölder, on obtient :

$$\mathcal{R}_w^p(f, A) = p \sum_{u \in V} \sum_{v \sim u} w(u, v)^{p/2} |f(v) - f(u)|^{p-2} (f(v) - f(u)) (g(v) - g(u)) \quad (4.13)$$

$$\leq \sum_{u \in V} \sum_{v \sim u} w(u, v)^{p/2} |f(v) - f(u)|^{p-1} |g(v) - g(u)| \quad (4.14)$$

$$\leq \sum_{u \in V} \left( \sum_{v \sim u} w(u, v)^{p/2} |f(v) - f(u)|^p \right)^{(p-1)/p} \left( \sum_{u \sim v} w(u, v)^{p/2} |g(v) - g(u)|^p \right)^{1/p} \quad (4.15)$$

On obtient donc  $\mathcal{R}_w^p(f, A) \leq \mathcal{R}_w^p(g, A)$  pour tout  $g$ . Et donc,  $f$  est  $p$ -harmonique.  $\square$

**Proposition 5.** Une fonction  $f$  est  $p$ -harmonique ssi

$$\Delta_{p,w} f(u) = \sum_{u \in V} \sum_{v \sim u} w(u, v)^{p/2} |f(v) - f(u)|^{p-2} (f(v) - f(u)) = 0 \quad \forall u \in \partial A, \quad (4.16)$$

La preuve suivante est similaire à celle présentée dans [Holopainen et Soardi, 1997].

*Preuve.* Soit  $h = \sum_i h_i$ , où chaque  $h_i(v) = 0$  pour tout  $v \in A \cup \partial A$ . On peut montrer que

$$\sum_{u \in V} \sum_{v \sim u} w(u, v)^{p/2} |f(v) - f(u)|^{p-2} (f(v) - f(u)) (h(v) - h(u)) = -2 \sum_i h_i(v) \Delta_{p,w} f(v). \quad (4.17)$$

Par (4.10), on a

$$-2 \sum_i h_i(v) \Delta_{p,w} f(v) = \sum_{u \in V} \sum_{v \sim u} w(u, v)^{p/2} |f(v) - f(u)|^{p-2} (f(v) - f(u)) (h(v) - h(u)) = 0.$$

$\square$



Une manière de résoudre le problème (4.4) est de considérer l'équation suivante. Pour  $0 \leq p < \infty$  :

$$\begin{cases} \Delta_{p,w} f(u) = 0 & \forall u \in A \\ f(u) = g(u) & \forall u \in \partial A \end{cases} \quad (4.18)$$

Ceci nous amène à :

$$\begin{cases} \sum_{v \sim u} w(u,v)^{p/2} |f(u) - f(v)|^{p-2} (f(u) - f(v)) = 0 \\ f(u) = g(u) & \forall u \in \partial A. \end{cases} \quad (4.19)$$

Définissons  $\gamma(u,v) = w(u,v)^{p/2} |f(u) - f(v)|^{p-2}$ . L'équation (4.18) peut être formulée

$$\sum_{v \sim u} \gamma(u,v) (f(u) - f(v)) = 0. \quad (4.20)$$

Nous obtenons comme point fixe

$$f(u) = \frac{\sum_{v \sim u} \gamma(u,v) (f(v))}{\sum_{v \sim u} \gamma(u,v)}. \quad (4.21)$$

Pour obtenir la solution, plusieurs méthodes peuvent être utilisées. Nous utilisons la méthode itérative de Gauss-Jacobi pour résoudre l'itération du point fixe. Cela conduit à une méthode itérative robuste pour résoudre le problème de  $p$ -Dirichlet avec des fonctions  $p$ -harmoniques sur graphes comme celles que nous avons présentées dans (4.2).

### 4.2.3 Cas particulier de $p = \infty$

Nous proposons une nouvelle définition d'une fonction  $\infty$ -harmonique. Considérons  $w = 1$ . Pour simplifier la notation, nous notons  $\Delta_p f$  pour  $\Delta_{p,w=1} f$ .

**Définition 3.** Une fonction  $f$  est  $\infty$ -harmonique ssi :

$$f(u) = \lim_{p \rightarrow \infty} f_p(u), \quad \forall u \in A \setminus \partial A \quad (4.22)$$

où  $f_p$  est la fonction qui satisfait  $\Delta_p f_p(u) = 0$ .

**Proposition 6.** Une fonction  $f$  est  $\infty$ -harmonique ssi :

$$f(u) = \frac{\max_{v \sim u} (f(v)) + \min_{v \sim u} (f(v))}{2} \quad (4.23)$$

*Preuve.* Il est facile de démontrer que pour une fonction  $p$ -harmonique  $f$ , nous pouvons changer l'échelle de la valeur des nœuds à n'importe quel intervalle souhaité avant le calcul de la  $p$ -énergie ou  $\Delta_{p,w}$  et revenir à l'échelle initiale par la suite. Ainsi, nous pouvons considérer la nouvelle échelle  $[0, 2]$  pour tout sommet  $u$ .

Considérons la fonction  $f^+ : f^+(u) = 1 + \epsilon$  pour un certain  $\epsilon > 0$ . Nous calculons :

$$\Delta_p f^+(u) = \sum_{v \sim u} |f^+(u) - f^+(v)|^{p-2} (f^+(u) - f^+(v)) \quad (4.24)$$

$$= A(1 + \epsilon)^{p-1} + \sum_i \alpha_i^{p-1} - \sum_j \beta_j^{p-1} - B(1 - \epsilon)^{p-1} \quad (4.25)$$

où  $A$  et  $B$  sont positifs et  $\alpha_i < 1, \beta_j < 1, \forall i, j$ .

Ce qui ramène à :

$$(\Delta_p f^+)(u) > 0, \text{ quand } p > P^+, \text{ pour un réel } P^+ \quad (4.26)$$

On note  $f^-(u) = 1 - \epsilon$ , et nous montrons que

$$(\Delta_p f^-)(u) < 0, \text{ quand } p > P^-, \text{ pour un réel } P^- \quad (4.27)$$

Comme  $\Delta_p f_p(u) = 0$ , alors :

$$\Delta_p f^-(u) < \Delta_p f_p(u) < (\Delta_p f^+)(u) \text{ quand } p > \max(P^-, P^+), \quad (4.28)$$

car  $\Delta_p$  est strictement croissant par rapport à  $f(u)$ , ce qui signifie que :

$$1 - \epsilon < f_p(u) < 1 + \epsilon, \text{ quand } p > \max(P^-, P^+), \quad (4.29)$$

i.e.

$$|f_p(u) - 1| < \epsilon, \text{ quand } p > \max(P^-, P^+), \quad (4.30)$$

Ceci montre que  $\lim_{p \rightarrow \infty} f_p(u) = f = 1$ .

1 est le milieu de l'échelle actuelle. En revenant à l'échelle initiale, le milieu de l'intervalle d'origine  $[\min(f), \max(f)]$  est :

$$f(u) = \frac{\max_{v \sim u}(f(v)) + \min_{v \sim u}(f(v))}{2} \quad (4.31)$$

Ce que l'on peut réécrire en utilisant l'*inf*ty-Laplacien :

$$\max_{v \sim u}(f(v)) = \max\left(\max_{v \sim u}(f(v) - f(u), 0)\right) = f(u) + \|\nabla_w^+ f\|_\infty + f(u), \quad (4.32)$$

et

$$\min_{v \sim u}(f(v)) = \max\left(\max_{v \sim u}(f(u) - f(v), 0)\right) = f(u) - \|\nabla_w^- f\|_\infty + f(u) \quad (4.33)$$

Soit,

$$f(u) = \frac{2f(u) + \|\nabla_w^+ f\|_\infty + \|\nabla_w^- f\|_\infty}{2} \quad (4.34)$$

□

Ce résultat porte à penser que le même résultat est obtenu pour  $w \neq 1$ . Ainsi nous retrouvons la définition donnée à la section 2.2. Par analogie à ce qui a été présenté pour  $1 \leq p < \infty$ , nous obtenons l'algorithme suivant. Pour tout  $(u, v)$  avec  $u$  et  $v$  dans  $A$  :

$$f^{(0)} = f^0, \quad \forall v \in A \quad (4.35)$$

$$f^{(k)} = f^0, \quad \forall v \in \partial A \quad (4.36)$$

$$f^{(k+1)}(v) = f^{(k)}(u) + \frac{1}{2} \left( \max_{v \sim u} \left( \sqrt{w(u, v)} \max(f(v) - f(u), 0) \right) \right) \quad (4.37)$$

$$- \frac{1}{2} \times \left( \max_{v \sim u} \left( \sqrt{w(u, v)} \max(f(u) - f(v), 0) \right) \right), \quad \forall v \in A \quad (4.38)$$

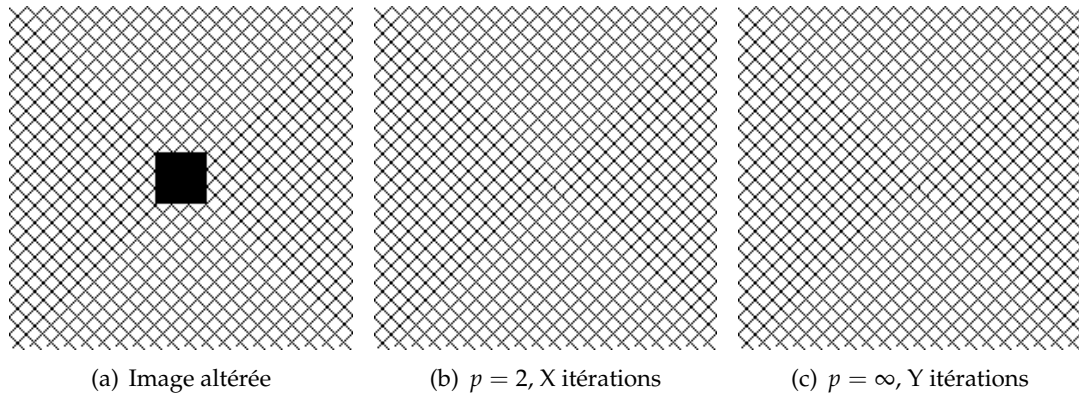


FIGURE 4.17 – Inpainting d'une image de synthèse. Résultats obtenus en utilisant  $p = 2$  et  $p = \infty$  sont visuellement similaires, pourtant, en termes de calcul,  $p = \infty$  est beaucoup plus rapide.

## 4.2.4 Résultats et applications

### 4.2.4.1 Inpainting avec $p = \infty$

La figure 4.17 fournit un exemple synthétique simple pour montrer l'avantage en termes de calcul, d'utiliser  $p = \infty$  sans aucune différence visuelle par rapport à  $p = 2$ . Nous tenons à souligner que ce résultat est obtenu à un coût de calcul beaucoup plus faible avec  $p = \infty$ . En fait, par définition, l'algorithme avec  $p = \infty$  est basé sur les fonctions  $\max()$  alors qu'avec  $p \in [1, \infty[$ , il est basé sur une moyenne pondérée. Par conséquent, pour le même nombre d'itérations, le temps de traitement est plus court pour  $p = \infty$  que pour  $p = 2$  tout en donnant des résultats tout aussi compétitifs visuellement. Nous remarquons que l'algorithme est encore plus lent pour  $p = 1$  que pour  $p = 2$ . En outre, nos expériences révèlent que dans les conditions de convergence identiques,  $p = \infty$  nécessite moins d'itérations que  $p = 2$  et beaucoup moins que  $p = 1$  pour atteindre la convergence. En résumé, pour  $p = \infty$  les vitesses de calcul et de convergence sont plus élevées de toute autre valeur de  $p$  pour peu ou pas de différence visuelle. Par conséquent, les progrès réels en utilisant  $p = \infty$  résident dans les économies de calcul.

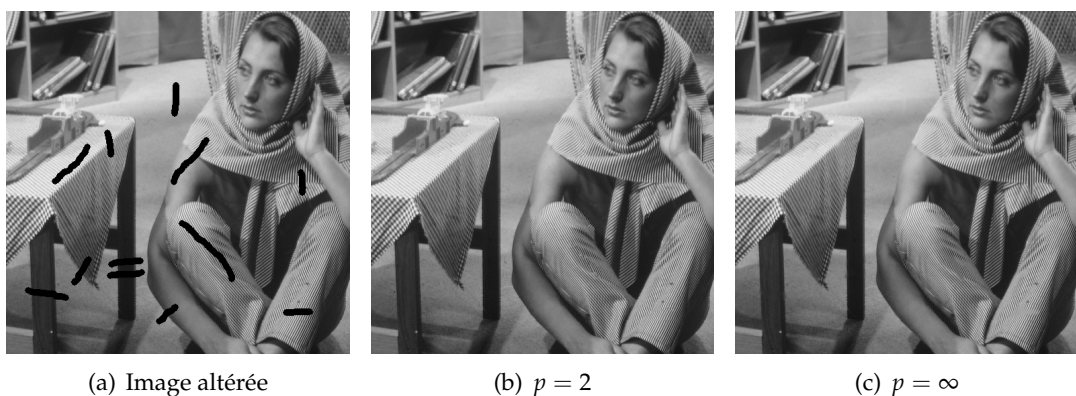


FIGURE 4.18 – Comparaison entre les résultats d'inpainting avec  $p = 2$  et  $p = \infty$ .

Dans la figure 4.18, nous pouvons observer que le résultat avec  $p = \infty$  est semblable à celui avec  $p = 2$ .



Le résultat sur une image de texture est rapporté dans la figure 4.19. Nous pouvons observer que les zones reconstruites ne sont pas détectables et fusionnent harmonieusement avec les données conservées dans les deux résultats, avec  $p = 2$  et  $p = \infty$ .

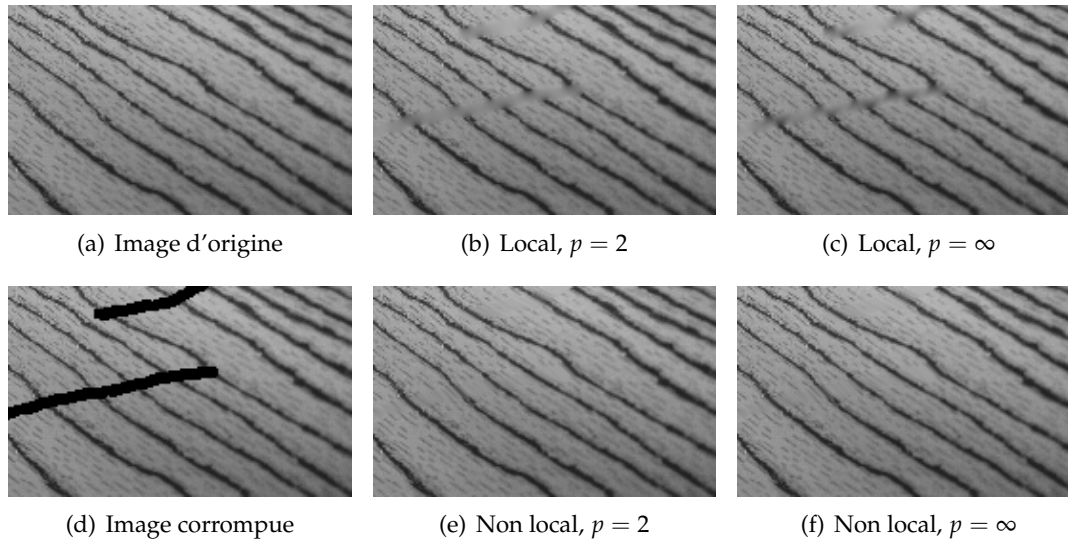


FIGURE 4.19 – Inpainting d’une image de texture avec une fenêtre de recherche  $31 \times 31$  et un patch  $15 \times 15$ . Sur la première ligne, de gauche à droite, l’image d’origine, le résultat de l’inpainting local avec  $p = 2$  et  $p = \infty$ . Sur la seconde ligne, l’image corrompue et le résultat de notre méthode d’inpainting non local avec  $p = 2$  et  $p = \infty$ .

#### 4.2.4.2 L’agrandissement

La tâche consiste à agrandir une image initiale en reconstruisant une version de la taille voulue (supérieure à la taille initiale) à partir d’un contenu en basse résolution. Pour ce faire, les pixels d’origine sont dispatchés sur l’image de grande taille, puis l’espace vide entre eux est comblé par inpainting. Cette méthode a de toute évidence des applications directes dans la compression d’images.

La figure 4.20 montre l’image de *Barbara* élargie par un facteur 8, tout en préservant la texture et des bords nets. En comparaison avec l’agrandissement bicubique, nous pouvons observer que notre résultat est très similaire, mais moins lissé.

#### 4.2.4.3 Segmentation semi-supervisée

Dans cette sous-section, nous montrons qu’une grande partie des algorithmes de segmentation supervisée peuvent être ramenés aux algorithmes basés sur les fonctions  $p$ -harmoniques. Les algorithmes de segmentation basés sur les graphes se sont répandus ces dernières années. Les variantes modernes des algorithmes de segmentation sur graphes sont principalement construites à partir d’un petit ensemble d’algorithmes de base : les *graph cuts*, la marche aléatoire et la méthode des plus courts chemins. Récemment, ces trois algorithmes ont tous été placés dans un cadre commun qui leur permet d’être considérés comme des cas particulier d’un algorithme de segmentation semi-supervisée plus général avec les différents choix d’un paramètre  $q$  [Sinop et Grady, 2007]. Dans ce cadre, une image est représentée par un graphe  $G = (V, E)$ .

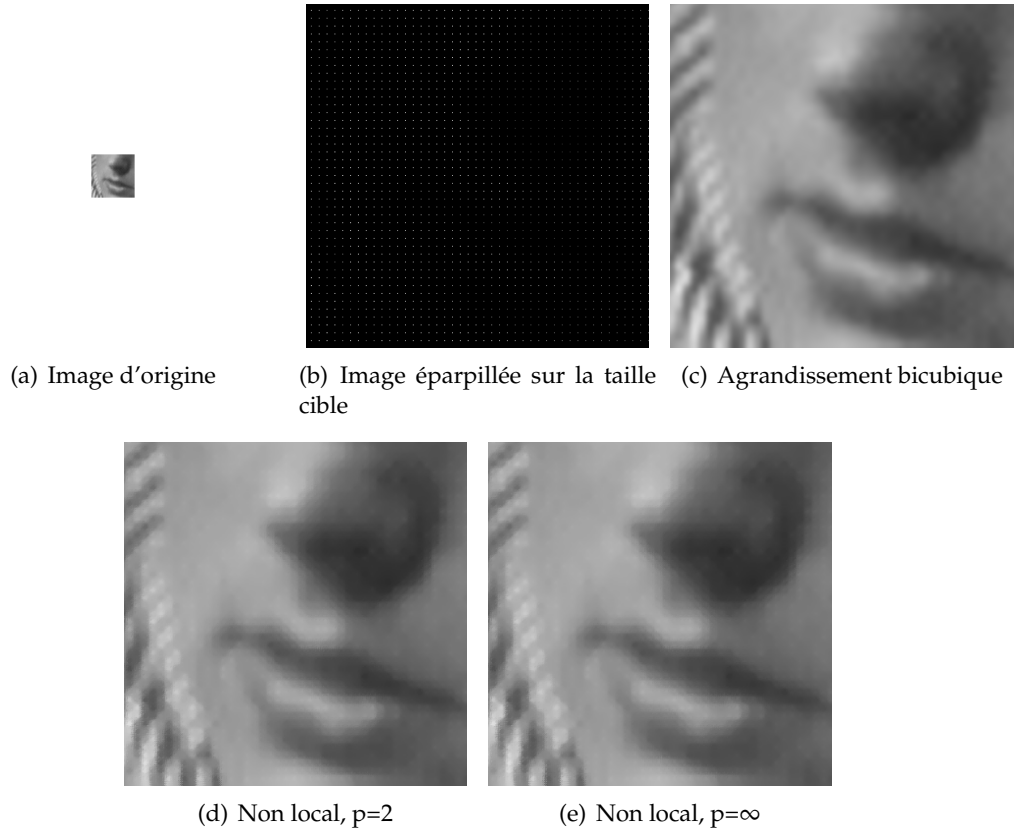


FIGURE 4.20 – L'agrandissement  $\times 8$  d'une partie de l'image de *Barbara*.

Le principe de la segmentation en deux classes consiste à :

**Étape 1 :** Choisir deux classes de germes l'un marquant le premier plan ( $f(F) = 1$ ) et l'autre marquant l'arrière-plan ( $f(B) = 0$ ).

**Étape 2 :** Trouver un optimum de l'équation

$$f = \arg \min_f \sum_{(u,v) \in E} (w(u,v)^p |f(u) - f(v)|)^q \quad (4.39)$$

**Étape 3 :** Diffuser les germes initiales

$$f(u) = \begin{cases} 1 & \text{si } f(u) \geq 0,5 \\ 0 & \text{si } f(u) < 0,5 \end{cases} \quad (4.40)$$

Cela revient à chercher un optimum de l'équation (4.39) qui peut être interprété comme la probabilité pour un pixel donné d'appartenir à l'un ou l'autre du premier plan ou l'arrière-plan.

En considérant,  $q = p$ , on peut réécrire l'algorithme de segmentation comme suit :

$$\begin{cases} f = \arg \min_f \sum_{u \in V} \|\nabla_{w'} f(u)\|_p^p & \text{sur } V \setminus (F \cup B) \\ f = \begin{cases} 1 & \text{si } f(u) \geq 0,5 \\ 0 & \text{si } f(u) < 0,5 \end{cases} \end{cases} \quad (4.41)$$

où  $w' = w^{\frac{2p}{q}}$ .

Nous retrouvons donc notre algorithme basé sur les fonctions  $p$ -harmoniques qui est directement applicable à la segmentation semi-supervisée. L'ensemble  $A$  de notre formalisme correspond à  $V \setminus (F \cup B)$  et  $\partial A$  correspond à  $F \cup B$ .

Les figures 4.21 et 4.22 sont des exemples de la segmentation d'une zone cible souhaitée. Les étiquettes marquant la cible et celles marquant l'arrière-plan sont interpolées aux régions similaires dans toute l'image en utilisant les poids calculés à partir de l'image à segmenter. À la convergence de ce processus, l'image est segmentée en deux. Dans les figures 4.21 et 4.22, la limite entre les régions segmentées est marquée en rouge. Cette application peut être utilisée dans des algorithmes de *matting* (voir figure 4.23).

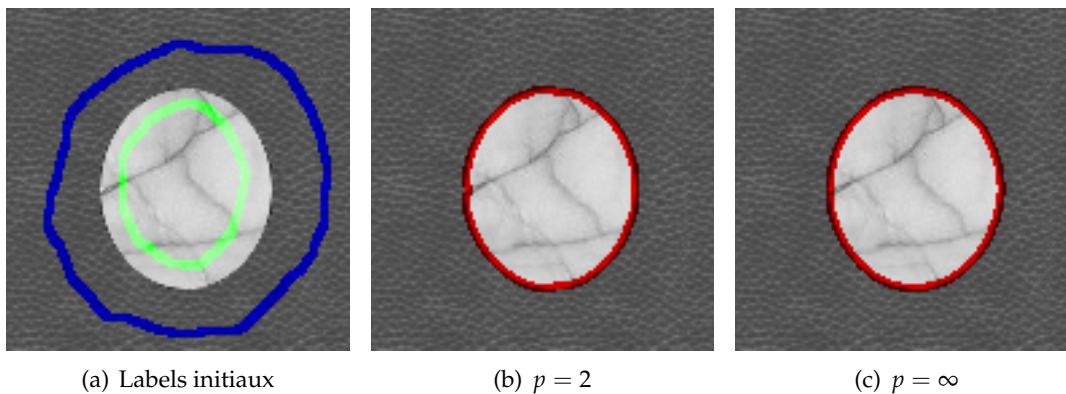


FIGURE 4.21 – De gauche à droite : Une image de textures de cuir et de marbre avec des labels initiaux définis par l'utilisateur marquant la région cible et le fond, les résultats de la segmentation avec  $p = 2$  et  $p = \infty$ .

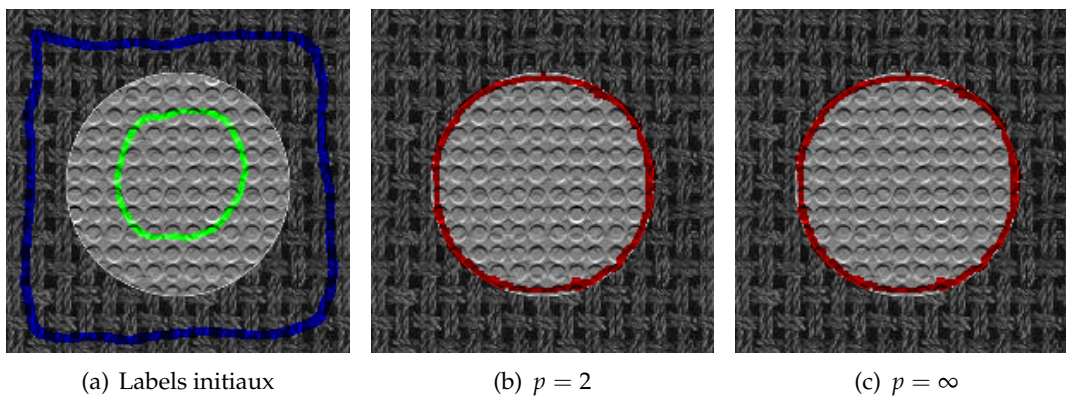


FIGURE 4.22 – De gauche à droite : Une image de deux textures différentes avec des labels initiaux définis par l'utilisateur marquant la région cible et le fond, les résultats de la segmentation avec  $p = 2$  et  $p = \infty$ .

## Conclusion

Dans cette section, nous avons présenté notre apport à la problématique de l'inpainting et l'interpolation des images et des vidéos. Nous avons présenté deux approches d'inpainting. La première est l'extension de l'interpolation continue au domaine discret. Nous traitons le résultat d'une première interpolation par régularisation itérative afin

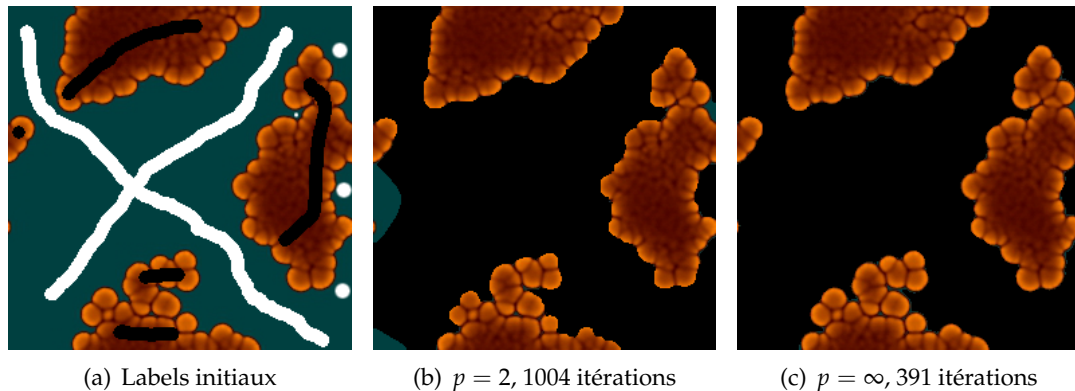


FIGURE 4.23 – Exemple de matting sur l'image *algae*. L'algorithme avec  $p = \infty$  est beaucoup plus rapide qu'avec  $p = 2$  et donne un meilleur résultat.

d'améliorer la qualité de l'interpolation. La seconde approche se base sur les fonctions  $p$ -harmoniques afin de proposer des algorithmes d'interpolation et d'inpainting. Ensuite, nous avons introduit le cas  $p = \infty$  et avons montré son intérêt en termes de vitesses de calcul et de convergence. De nombreuses applications en découlent : la suppression de textes et d'objets, ou au contraire l'incrustation d'objets ou encore la segmentation semi-supervisée.

### Publications associées

- GHONIEM, M., CHAHIR, Y. et ELMOATAZ, A. Geometric and texture inpainting based on discrete regularization on graphs. *In ICIP'09*, pages 1349 – 1352, 2009. [Ghoniem et al., 2009]
- GHONIEM, M., CHAHIR, Y. et ELMOATAZ, A. Nonlocal video denoising, simplification and inpainting using discrete regularization on graphs. *Signal Processing*, 90(8) : 2445 – 2455, 2010. Special Section on Processing and Analysis of High-Dimensional Masses of Image and Signal Data. [Ghoniem et al., 2010]
- ELMOATAZ, A., GHONIEM, M., LÉZORAY, O. et TA, V.T. Nonlocal Infinity Laplacian on Graphs with Applications to Image and Data Processing. *Submitted to PAMI* (preprint), 2010. [Elmoataz et al., 2011]

**TROISIÈME PARTIE**

---

**RÉGULARISATION SPECTRALE**

---



# MÉTHODES SPECTRALES POUR LA CLASSIFICATION

## Sommaire

|   |    |
|---|----|
| <b>5.1 Méthodes linéaires</b> . . . . .                       | 75 |
| 5.1.1 ACP . . . . .   | 75 |
| <b>5.2 Méthodes basées sur graphe</b> . . . . .               | 76 |
| 5.2.1 Isomap . . . . .  | 76 |
| 5.2.2 LLE . . . . .   | 77 |
| 5.2.3 Laplacian Eigenmaps . . . . .                           | 78 |
| 5.2.4 Cartes de diffusion ( <i>Diffusion Maps</i> ) . . . . . | 79 |

## Résumé

Dans un grand nombre d'applications en traitement d'images, nous sommes souvent confrontés à des données massives de grandes dimensions. Afin de faciliter leurs traitements tels que la visualisation ou l'organisation, il est indispensable de réduire leur dimension. Pour ce faire, les méthodes spectrales se sont avérées des outils puissants. Dans ce chapitre, nous présentons quelques méthodes spectrales pour la classification des images.

## Introduction

Les données du monde réel telles que les signaux sonores, les photographies numériques, ou les vidéos ont généralement une haute dimensionnalité due à un grand nombre de descripteurs. Cette grande dimensionnalité représente une entrave pour le traitement, l'organisation, la recherche, l'analyse ou la visualisation de ces données. Ces problèmes sont classiquement abordés par les techniques de sélection de variables et de réduction de dimension, qui visent à trouver des structures intrinsèques de dimension réduite. La réduction de dimension est la transformation de données de grande dimension dans une représentation significative de dimension réduite. Idéalement, la représentation réduite devrait avoir une dimensionnalité qui correspond à la dimensionnalité intrinsèque des données qui est le nombre minimum de paramètres nécessaires pour tenir compte des propriétés observées [Fukunaga, 1990]. Les données du monde réel telles que les signaux sonores, les photographies numériques, ou les vidéos ont généralement une haute dimensionnalité due à un grand nombre de descripteurs. Cette grande dimensionnalité représente une entrave pour le traitement, l'organisation, la recherche, l'analyse ou la visualisation de ces données. Ces problèmes sont classiquement abordés par les techniques de sélection de variables et de réduction de dimension, qui visent à trouver des structures intrinsèques de dimension réduite. La réduction de dimension est la transformation de données de grande dimension dans une représentation significative de dimension réduite.

Idéalement, la représentation réduite devrait avoir une dimensionnalité qui correspond à la dimensionnalité intrinsèque des données qui est le nombre minimum de paramètres nécessaires pour tenir compte des propriétés observées [Fukunaga, 1990].

En conséquence, la réduction de la dimensionnalité facilite, entre autres, la classification, la visualisation, et la compression de données de grande dimension. Traditionnellement, la réduction de la dimensionnalité est réalisée par des techniques linéaires telles que l'Analyse en Composantes Principales (ACP) [Pearson, 1901], l'analyse factorielle [Spearman, 1904], et la mise à l'échelle classique [Torgerson, 1952]. Cependant, ces techniques linéaires ne peuvent pas traiter correctement des données complexes non linéaires.

Dans la dernière décennie, un grand nombre de techniques non linéaires de réduction de dimensionnalité ont été proposées. Pour un aperçu, le lecteur est référé, par exemple, à [Burgess, 2005, Saul *et al.*, 2006, Lee et Verleysen, 2007, Venna, 2007]. Contrairement aux techniques linéaires traditionnelles, les techniques non linéaires ont la capacité de traiter des données complexes non linéaires. En particulier, les techniques non linéaires sont plus performantes sur les données réelles susceptibles de former une variété fortement non linéaires. Par exemple, l'ensemble de données *Swiss roll* comprend un ensemble de points qui se trouvent sur une variété (*manifold*) en forme de spirale à deux dimensions qui est incorporé dans un espace à trois dimensions. Un grand nombre de techniques non linéaires sont parfaitement capables de trouver cet emboîtement, contrairement aux techniques linéaires.

Le problème de la réduction de dimensionnalité non linéaire peut être défini comme suit. Supposons que nous avons un ensemble de données représentées dans une matrice de taille  $n \times D$  de  $\mathbf{X}$  comprenant  $n$  vecteurs  $\mathbf{x}_i$  ( $i \in \{1, 2, \dots, n\}$ ) avec une dimensionnalité  $D$  (où  $d < D$ , et souvent  $d \ll D$ ). Mathématiquement, la dimensionnalité intrinsèque signifie que les points de données  $\mathbf{X}$  sont situés sur ou près d'une variété avec une dimensionnalité  $d$  qui est incorporée dans l'espace  $D$ -dimensionnelle. Notons qu'aucune hypothèse n'est faite sur la structure de cette variété. Elle peut être non-riemannienne en raison des discontinuités (par exemple, si elle est constituée d'un certain nombre de sous-variétés déconnectées).

Les techniques de réduction de dimensionnalité transforment les ensembles de données  $\mathbf{X}$  dont la dimensionnalité  $D$  en un nouvel ensemble de données  $\mathbf{Y}$  avec une dimensionnalité  $d$ , tout en conservant la géométrie des données autant que possible. En général, ni la géométrie de la variété de données, ni la dimension intrinsèque  $d$  de l'ensemble de données  $\mathbf{X}$  ne sont connues. Par conséquent, la réduction de dimensionnalité est un problème mal posé qui ne peut être résolu qu'en supposant certaines propriétés des données (telles que la dimension intrinsèque). Notons un point de données de grande dimensionnalité par  $\mathbf{x}_i$ , où  $\mathbf{x}_i$  est la  $i^{\text{e}}$  ligne de la matrice  $D$ -dimensionnelle  $\mathbf{X}$ . Le correspondant de basse dimensionnalité de  $\mathbf{x}_i$  est noté  $\mathbf{y}_i$ , où  $\mathbf{y}_i$  est la  $i^{\text{e}}$  ligne de la matrice  $d$ -dimensionnelle  $\mathbf{Y}$ .

La figure 5.1 montre une taxonomie des techniques de réduction de la dimensionnalité. Toutes ces méthodes de réduction de données exploitent le contenu spectral d'une matrice de similarité mesurant la distance entre les données deux à deux. Plus spécifiquement, les vecteurs propres sont un outil fournissant une représentation des données dans un espace où les données sont mieux différenciées et peuvent être facilement classées en groupes de données similaires. Les valeurs propres offrent la possibilité de capturer les caractéristiques discriminantes d'un ensemble de données.

Sans chercher l'exhaustivité, nous analysons uniquement les méthodes ACP, Isomap, Diffusion maps, LLE et Laplacian Eigenmaps.



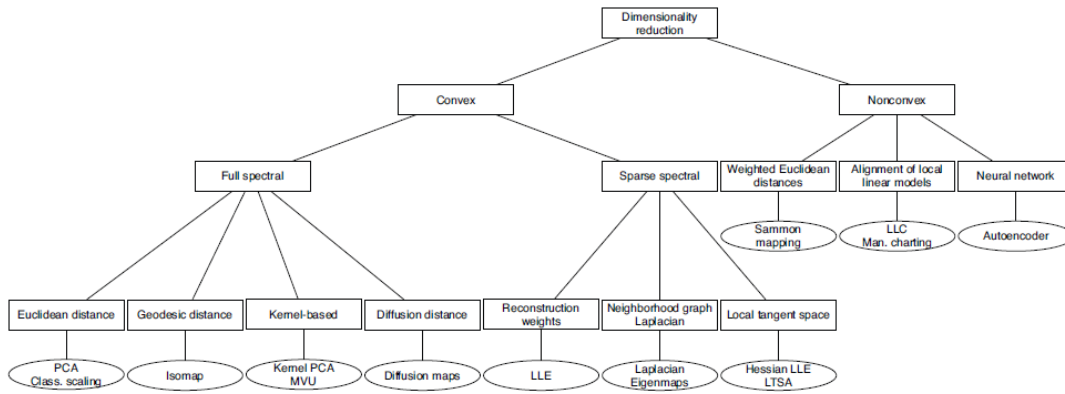


FIGURE 5.1 – Taxonomie des méthodes de réduction de données [van der Maaten *et al.*, 2009].

## 5.1 Méthodes linéaires

### 5.1.1 ACP

L'Analyse en Composantes Principales (ACP) [Pearson, 1901, Hotelling, 1933] est basée sur le calcul d'une représentation de faible dimensionnalité d'un ensemble de données de haute dimensionnalité qui conserve le plus fidèlement possible sa structure de covariance. Dans l'ACP, les motifs d'entrée  $\mathbf{x}_i \in \mathbb{R}^d$  sont projetés dans le sous-espace  $m$ -dimensionnel qui minimise l'erreur de reconstruction,

$$\mathcal{E}_{ACP} = \sum_i \left\| \mathbf{x}_i - \sum_{\alpha=1}^m (\mathbf{x}_i \cdot \mathbf{e}_{ff}) \mathbf{e}_{ff} \right\|^2, \tag{5.1}$$

où les vecteurs  $\{\mathbf{e}_{ff}\}_{\alpha=1}^m$  définissent une base partielle orthonormée de l'espace d'entrée. De l'équation (5.1), nous pouvons facilement montrer que le sous-espace avec l'erreur de reconstruction minimum est également le sous-espace avec une variance maximum. Les vecteurs de base de ce sous-espace sont donnés par les  $m$  premiers vecteurs propres de la matrice  $d \times d$  de covariance,

$$\mathbf{C} = \frac{1}{n} \sum_i \mathbf{x}_i \mathbf{x}_i^\top, \tag{5.2}$$

en supposant que les modèles d'entrée  $\mathbf{x}_i$  sont centrés sur l'origine. Les sorties des ACP sont tout simplement les coordonnées des motifs d'entrée dans ce sous-espace dont les axes principaux sont indiqués par les vecteurs propres. Les valeurs propres de la matrice de covariance dans l'équation (5.2) mesurent la variance prévue des données de grande dimension placées le long des axes principaux. Ainsi, le nombre de valeurs propres significatives mesure la dimensionnalité du sous-espace qui contient la plupart de la variance des données. Un écart important dans le spectre des valeurs propres indique que les données se limitent principalement à un sous-espace de moindre dimensionnalité. Plus de détails sur l'ACP peuvent être trouvés dans [Jolliffe, 2002].

Malgré l'existence de différentes techniques, l'ACP est de loin la technique linéaire (non supervisée) la plus populaire.

## 5.2 Méthodes basées sur graphe

Les méthodes linéaires génèrent des représentations fidèles de faible dimension lorsque les motifs d'entrée de grande dimension sont principalement confinés dans un sous-espace de faible dimension. Si les motifs d'entrée sont éparpillés un peu partout dans ce sous-espace, les spectres de valeurs propres de ces méthodes révèlent également la dimension intrinsèque de l'ensemble de données, c'est-à-dire le nombre des modes de variabilité sous-jacents. Un cas plus intéressant se pose, lorsque les motifs d'entrée se trouvent dans une sous-variété de faible dimension de l'espace d'entrée. Dans ce cas, la structure de l'ensemble des données peut être fortement non linéaire, et les méthodes linéaires sont vouées à l'échec. Les méthodes basées sur graphes ont récemment émergé comme un outil puissant pour l'analyse des données de grande dimension qui ont été échantillonnées à partir d'une sous-variété de faible dimension. Ces méthodes commencent par la construction d'un graphe peu dense dans lequel les nœuds représentent des modèles d'entrée et les arêtes représentent les relations de voisinage. Le graphe résultant (supposé connecté, pour simplifier) peut être considéré comme une approximation discrète de la sous-variété échantillonnée par les modèles d'entrée. À partir de ces graphes, nous pouvons alors construire des matrices dont les décompositions spectrales révèlent la structure de faible dimension de la sous-variété (et parfois même la dimensionnalité elle-même). Bien que capables de révéler la structure fortement non linéaire, les méthodes sur graphes pour l'apprentissage de variétés sont basées sur des problèmes d'optimisation facilement solubles tels que les problèmes des plus courts chemins, les moindres carrés, et la diagonalisation des matrices. Dans ce qui suit, nous examinons quatre exemples représentatifs des algorithmes de graphes à base d'apprentissage de variétés : Isomap [Tenenbaum *et al.*, 2000], *maximum variance unfolding* (MVU) [Weinberger *et al.*, 2005, Sun *et al.*, 2006], *locally linear embedding* (LLE) [Roweis et Saul, 2000, Saul *et al.*, 2003], et le Laplacien Eigenmap [Belkin et Niyogi, 2001, Belkin et Niyogi, 2003].

### 5.2.1 Isomap

La mise à l'échelle classique a donné de bons résultats dans de nombreuses applications, mais elle souffre du fait qu'elle vise essentiellement à conserver des distances euclidiennes, et ne tient pas compte de la répartition des points de données voisins. Si les données de grande dimension se trouvent sur ou près d'une variété courbée, comme dans l'ensemble de données *Swiss roll*<sup>1</sup>, la mise à l'échelle classique peut considérer que deux points sont proches, alors qu'ils sont éloignés sur la variété (voir figure 5.2). L'Isomap [Tenenbaum *et al.*, 2000] est une technique qui résout ce problème en tentant de préserver les distances géodésiques (ou curvilignes) entre les points de données. La distance géodésique est la distance entre deux points mesurée sur la variété. Dans l'Isomap, les distances géodésiques entre les points de données  $x_i$  ( $i = 1, 2, \dots, n$ ) sont calculées en construisant un graphe  $G$  de voisinage, dans lequel chaque point de données  $x_i$  est lié à ses  $k$ -plus proches voisins  $x_{ij}$  ( $j = 1, 2, \dots, k$ ) dans l'ensemble de données  $X$ . Le plus court chemin entre deux points dans le graphe forme une estimation de la distance géodésique entre ces deux points, et peut facilement être calculé à l'aide de l'algorithme du plus court chemin de Dijkstra ou celui de Floyd [Dijkstra, 1959, Floyd, 1962]. Les distances géodésiques entre les points de données dans  $X$  sont calculées, de manière à former une matrice de distances géodésiques. Les représentations de faible dimensionnalité  $y_i$  des points de données  $x_i$  dans l'espace de faible dimension  $Y$  sont calculées en appliquant la mise à l'échelle classique sur la matrice de distances géodésiques résultante.

1. Source de l'image *Swiss roll* : <http://c13s.wordpress.com>

Une faiblesse importante de l'algorithme Isomap est son instabilité topologique [Balasubramanian et Schwartz, 2002]. Isomap peut construire des connexions erronées dans le graphe de voisinage  $G$ . Ces court-circuits peuvent gravement nuire à la performance des Isomap [Lee et Verleysen, 2005]. Plusieurs approches ont été proposées pour résoudre le problème de court-circuit, par exemple, en supprimant les points de données ayant d'importants flux dans l'algorithme du plus court chemin [Choi et Choi, 2007] ou en supprimant les voisins les plus proches qui violent la linéarité locale du graphe de voisinage [Saxena et al., 2004]. Quand il réussit, l'Isomap conduit à une représentation de faible dimension, dans laquelle les distances euclidiennes entre les éléments de sortie correspondent aux distances géodésiques entre les éléments d'entrée de la sous-variété à partir desquels ils ont été échantillonnés. L'Isomap a été appliqué avec succès à des tâches telles que l'inspection du bois [Niskanen et Silvén, 2003], la visualisation de données biomédicales [Lim et al., 2003], et l'estimation de la pose de la tête [Raytchev et al., 2004].

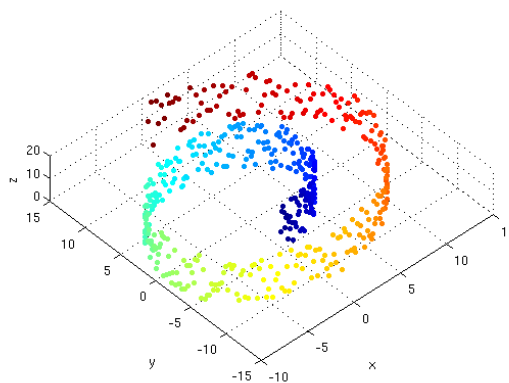


FIGURE 5.2 – Données en *swiss roll*.

### 5.2.2 LLE

*Local Linear Embedding* (LLE) [Roweis et Saul, 2000] est une technique qui est similaire à l'Isomap par la construction d'un graphe représentant les données. Contrairement à l'Isomap, il tente de préserver uniquement les propriétés locales des données. En conséquence, le LLE est moins sensible à un court-circuit que l'Isomap, parce que seul un petit nombre de propriétés locales sont touchées si un court-circuit a lieu. En outre, la préservation des propriétés locales permet une intégration réussie des variétés non-convexes. Dans le LLE, les propriétés locales des données de la variété sont construites en écrivant les points de données de grande dimension comme une combinaison linéaire de leurs voisins les plus proches. Dans la représentation de faible dimensionnalité, le LLE tente de conserver le poids de reconstruction dans les combinaisons linéaires aussi bien que possible.

LLE décrit les propriétés locales de la variété autour d'un point de donnée  $x_i$  en écrivant le point comme une combinaison linéaire  $w_i$  (le poids de reconstruction susmentionné) de ses  $k$ -plus proches voisins  $x_{i_j}$ . Par conséquent, le LLE correspond à un hyperplan à travers le point de donnée  $x_i$  et ses plus proches voisins, ce qui suppose que la variété est localement linéaire. L'hypothèse de linéarité locale implique que le poids de reconstruction  $w_i$  du point  $x_i$  est invariant par translation, rotation, et changement d'échelle. En raison de l'invariance de ces transformations, une application linéaire de l'hyperplan à un espace de dimension inférieure préserve le poids de reconstruction en l'espace de plus faible dimensionnalité. En d'autres termes, si la représentation de faible dimensionnalité

des données conserve la géométrie locale de la variété, le poids de reconstruction  $w_i$  qui reconstruit les points  $\mathbf{x}_i$  à partir de leurs voisins dans la représentation des données de grande dimension, reconstruit également les points  $\mathbf{y}_i$  à partir de leurs voisins dans la représentation de faible dimensionnalité. En conséquence, trouver la représentation des données  $\mathbf{Y}$  de dimension  $d$  équivaut à minimiser la fonction de coût.

$$\phi(\mathbf{Y}) = \sum_i \left\| \mathbf{y}_i - \sum_{j=1}^k w_{ij} \mathbf{y}_j \right\|^2 \quad \text{sous la condition} \quad \left\| \mathbf{y}^{(k)} \right\|^2 = 1 \quad \forall k, \quad (5.3)$$

où  $\mathbf{y}^{(k)}$  représente la  $k^e$  colonne de la matrice solution  $\mathbf{Y}$ . La contrainte sur la covariance des colonnes de  $\mathbf{Y}$  est nécessaire pour exclure la solution triviale  $\mathbf{Y} = \mathbf{0}$ . Roweis et Saul [Roweis et Saul, 2000] ont montré que les coordonnées des représentations de faible dimensionnalité  $\mathbf{y}_i$  qui minimisent cette fonction de coût sont obtenues en calculant les vecteurs propres correspondants aux  $d$  plus petites valeurs propres non nulles du produit  $(\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W})$  où  $\mathbf{W}$  est une matrice creuse  $n \times n$ . Les entrées de cette matrice valent 0 si  $i$  et  $j$  ne sont pas connectés dans le graphe de voisinage, sinon, ils valent le poids de reconstruction correspondant. Dans cette formule,  $\mathbf{I}$  est la matrice identité de taille  $n \times n$ .

La popularité du LLE a conduit à la proposition de variantes linéaires de l'algorithme [He et al., 2005, Kokiopoulou et Saad, 2005] et à des applications réussies, par exemple, la super-résolution [Chang et al., 2004] et la localisation de source sonore [Duraiswami et Raykar, 2005]. Toutefois, il existe aussi des études expérimentales qui rapportent une faible performance du LLE [Jenkins et Mataric, 2002, Lim et al., 2003]. Une explication possible réside dans les difficultés que LLE a face aux variétés contenant des trous [Roweis et Saul, 2000].

En outre, le LLE a tendance à regrouper de larges portions de données très proches dans l'espace de faible dimension, car la contrainte de covariance sur la solution est trop simple [van der Maaten et al., 2009]. En outre, la contrainte de covariance peut donner lieu à des changements d'échelle indésirables des données dans la variété [Goldberg et al., 2008].

### 5.2.3 Laplacian Eigenmaps

Similaire à la technique LLE, le Laplacien Eigenmap trouve une représentation de faible dimensionnalité des données en préservant les propriétés locales de la variété [Belkin et Niyogi, 2001]. Dans le Laplacien Eigenmap, les propriétés locales sont basées sur les distances entre les voisins proches. Il calcule une représentation de basse dimensionnalité des données dans laquelle les distances entre un point de données et ses  $k$ -plus proches voisins sont réduites au minimum. Cela se fait d'une manière pondérée, à savoir, la distance de la représentation de basse dimensionnalité des données entre un point de données et son voisin le plus proche contribue davantage à la fonction de coût que la distance entre le point de données et son deuxième plus proche voisin. En utilisant la théorie spectrale des graphes, la minimisation de la fonction de coût est définie comme un problème aux valeurs propres.

L'algorithme du Laplacien Eigenmap construit d'abord un graphe de voisinage  $G$  dans lequel chaque point de données  $x_i$  est relié à ses  $k$ -plus proches voisins. Pour tous les  $\mathbf{x}_i$  et  $\mathbf{x}_j$  dans le graphe  $G$  qui sont reliés par une arête, le poids de l'arête est calculé en utilisant la fonction noyau gaussien, conduisant à une matrice d'adjacence creuse  $\mathbf{W}$ . Dans le calcul

des représentations de basse dimensionnalité  $\mathbf{y}_i$ , la fonction de coût qui est réduite au minimum, est donnée par

$$\phi(\mathbf{Y}) = \sum_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|^2 w_{ij}. \quad (5.4)$$

#### 5.2.4 Cartes de diffusion (*Diffusion Maps*)

Le cadre des cartes de diffusion (*diffusion maps* (DM) [Nadler *et al.*, 2005, Lafon et Lee, 2006] provient du domaine des systèmes dynamiques. Le principe des cartes de diffusion est basé sur la définition d'une marche aléatoire de Markov sur le graphe des données. En effectuant la marche aléatoire pour un certain nombre de pas, une mesure de la proximité des points de données est obtenue. En utilisant cette mesure, la distance de diffusion est définie. Dans la représentation de basse dimensionnalité des données, les distances de diffusion sont conservées du mieux possible. L'idée clé sous-jacente à la distance de diffusion, c'est qu'elle est basée sur l'intégration sur tous les chemins sur le graphe. Cela rend la distance de diffusion plus robuste aux court-circuits que la distance géodésique, par exemple, qui est employée dans Isomap. Les cartes de diffusion ont été appliquées avec succès à la mise en correspondance des formes [Rajpoot *et al.*, 2007] et de nombreuses autres applications.

Au chapitre suivant, notre attention porte essentiellement sur les cartes de diffusion que nous utilisons à des fins de régularisation spectrale pour la classification des images.



# CONTRIBUTIONS À L'ANALYSE SPECTRALE

## Sommaire

|   |           |
|---|-----------|
| <b>6.1 Classification par cartes de diffusion</b> . . . . .                     | <b>82</b> |
| 6.1.1 Construction du graphe . . . . .  | 82        |
| 6.1.2 Laplacien du graphe . . . . .   | 83        |
| 6.1.3 Cadre de diffusion utilisé . . . . .                                      | 83        |
| <b>6.2 Résultats et applications : segmentation et classification</b> . . . . . | <b>88</b> |
| 6.2.1 Segmentation par classification et diffusion . . . . .                    | 88        |
| 6.2.2 Caractérisation des éléments saillants . . . . .                          | 89        |
| 6.2.3 Classification de lettres manuscrites . . . . .                           | 94        |

## Résumé

Dans ce chapitre, nous formulons une approche de régularisation spectrale sur graphes basée sur les cartes de diffusion. Premièrement, nous proposons un algorithme de classification et de segmentation à partir des points caractéristiques de l'image. Ensuite, nous présentons une approche de détection d'éléments saillants par marches aléatoires sur graphes. Et enfin, nous présentons un projet d'apprentissage d'écriture auquel nous avons contribué en fournissant un algorithme de classification de données hétérogènes par régularisation des données dans l'espace spectral réduit.

## Introduction

L'utilisation des méthodes spectrales pour des tâches de classification est relativement récente et leurs résultats sont très intéressants car ils sont proches de ceux obtenus à la main par un humain. Ces méthodes sont capables de définir des classes qui ne constituent pas des régions convexes dans l'espace. Différents algorithmes de classification spectrale ont été proposés utilisant les vecteurs propres de diverses manières.

En général, l'objectif de ces algorithmes est d'identifier, dans une distribution de données, les sous-ensembles qui manifestent un « comportement » similaire. Ces méthodes de regroupement spectral sont devenues célèbres après les travaux de Shi et Malik [Shi et Malik, 2000, Ng *et al.*, 2001].

L'idée intuitive de la classification non-supervisée est de diviser les vecteurs en plusieurs groupes de telle manière que les vecteurs d'un même groupe sont considérés « similaires » tandis que les vecteurs situés dans des groupes distincts sont définis comme



« différents ». Le succès de ces algorithmes est fondé sur le fait qu'ils ne font aucune supposition sur la forme des groupes à former (*clusters*).

## 6.1 Classification par cartes de diffusion

Dans cette section, nous expliquons notre approche de classification spectrale utilisant les cartes de diffusion (*diffusion maps*) qui est une méthode basée sur le Laplacien. Les Laplaciens dérivés de graphes ne sont pas utilisés seulement en tâches de regroupement, ils sont aussi utiles en apprentissage semi-supervisé et en reconstruction de variétés.

Soit  $(X, \mu)$  un espace d'ensembles finis d'objets, où  $X = \{x_1, \dots, x_n\}$  un ensemble de jeu de  $m$  points de données tel que  $\forall x_i \in X, x_i \in \mathbb{R}^n$ , ( $x_i$  est un élément de dimension  $n$ ) et  $\mu$  est la mesure qui représente la distribution de ces points dans le jeu de données. Pour simplifier, nous supposons que  $\mu$  est finie. Dans un premier temps, nous rappelons la structure de diffusion décrite dans [Coifman *et al.*, 2005], en commençant par la construction du noyau de diffusion, puis par la présentation de ses propriétés spectrales pour analyser la géométrie des données. Ensuite, nous présentons l'implémentation de cette approche dans le cas discret avec des applications pour la classification et la segmentation de données en imagerie.

### 6.1.1 Construction du graphe

L'idée de base, issue de la théorie des graphes, est de représenter l'ensemble de données  $X = \{x_1, \dots, x_n\}$  par un graphe pondéré  $G = (V, E, w)$ . Deux sommets  $v_i$  et  $v_j$  sont adjacents si l'arête  $(v_i, v_j) \in E$ . On dit que les deux sommets sont voisins (notation  $v_i \sim v_j$ ).

La matrice des poids obtenus  $\mathbf{W}$  est appelée matrice d'affinité ou matrice de similarité. Elle est définie par :  $\mathbf{W} = w(v_i, v_j)$  si les sommets  $v_i$  et  $v_j$  sont voisins, et  $w(v_i, v_j) = 0$  dans le cas contraire. Les sommets étant liés à eux-mêmes, nous avons pour tout sommet  $v_i$ ,  $w(v_i, v_i) = 1$ .

Lafon, Keller et Coifman [Lafon *et al.*, 2006] ont montré que l'utilisation d'un noyau gaussien  $w_\varepsilon$  produit une représentation des données qui est fortement corrélée à la distribution des échantillons de données. Ainsi, la connectivité est exprimée en termes de mesure de similarité par le noyau gaussien suivant :

$$k(v_i, v_j) = w_\varepsilon(v_i, v_j) = e^{-\frac{\|v_i - v_j\|^2}{\varepsilon}}$$

où  $\varepsilon$  est un paramètre d'échelle et  $\|\cdot\|$  désigne la distance euclidienne standard.

Le degré d'un sommet  $v_i$  est défini par :  $d(v_i) = \sum_{v_j \in V} w(v_i, v_j)$ .

La matrice de degré  $\mathbf{D}$  est une matrice diagonale contenant les degrés  $d_1, d_2, \dots, d_n$ . Ainsi, nous définissons la matrice des degrés des sommets  $\mathbf{D}$  avec :  $\mathbf{D}_{ii} = D(v_i, v_i) = d(v_i)$  et  $D(v_i, v_j) = 0$  pour  $v_i \neq v_j$ .

Le volume  $\text{Vol}(A)$  est une mesure de la « taille » d'un sous-ensemble  $A \subset V$  définie par :  $\text{vol}(A) = \sum_{v_i \in A} d(v_i, v_j)$ .

## 6.1.2 Laplacien du graphe

### 6.1.2.1 Version non normalisée

Soit le Laplacien non normalisé, connu aussi sous le nom du Laplacien combinatoire. Il est défini par :

$$\mathbf{L}_{\text{nn}} = \mathbf{L}_{\text{nn}}(v_i, v_j) = \begin{cases} d(v_i), & \text{si } v_i = v_j \\ -w(v_i, v_j), & \text{si } v_i \sim v_j \\ 0, & \text{sinon.} \end{cases}$$

Cette matrice peut aussi s'écrire  $\mathbf{L}_{\text{nn}} = \mathbf{D} - \mathbf{W}$ . Cette matrice est symétrique et semi-définie positive, et possède des valeurs propres positives.

### 6.1.2.2 Version normalisée

On trouve dans la littérature deux versions normalisées du Laplacien. La version symétrique est :

$$\mathbf{L}_{\text{nij}} = \mathbf{L}_{\text{n}}(v_i, v_j) = \begin{cases} 1 - \frac{w(v_i, v_j)}{d(v_j)}, & \text{si } v_i = v_j \\ -\frac{w(v_i, v_j)}{\sqrt{d(v_i)d(v_j)}}, & \text{si } v_i \sim v_j \\ 0, & \text{sinon.} \end{cases}$$

La matrice  $\mathbf{L}_{\text{nn}}$  peut s'écrire aussi :

$$\mathbf{L}_{\text{n}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}^{\frac{1}{2}} \quad (6.1)$$

$$= \mathbf{D}^{-\frac{1}{2}} \mathbf{L}_{\text{nn}} \mathbf{D}^{\frac{1}{2}} \quad (6.2)$$

La version asymétrique s'écrit :

$$\mathbf{L}_{\text{n}} = \mathbf{I} - \mathbf{D}^{-1} \mathbf{W} = \mathbf{I} - \mathbf{P}$$

Cette dernière a l'avantage de montrer clairement le lien entre le Laplacien et les marches aléatoires.

## 6.1.3 Cadre de diffusion utilisé

### 6.1.3.1 Marches aléatoires sur graphes

Nous allons nous intéresser à un processus de marche aléatoire (ou de diffusion dans le graphe  $G$ ). Le temps est discrétisé  $t = (0, 1, 2, \dots)$ . À chaque instant, un marcheur est localisé sur un sommet et se déplace à l'instant suivant vers un sommet choisi aléatoirement et uniformément parmi les sommets voisins. La suite des sommets visités est alors une marche aléatoire, et la probabilité de transition du sommet  $v_i$  au sommet  $v_j$  est définie à chaque étape par :

$$p_{ij} = p(v_i, v_j) = \frac{w(v_i, v_j)}{d(v_i)}$$

Ceci définit la matrice de transition  $\mathbf{P} = p(v_i, v_j)$ , de la chaîne de Markov correspondant à la marche aléatoire. La matrice  $\mathbf{P}$  est stochastique ( $\forall v_i, \forall v_j, 0 \leq p(v_i, v_j) \leq 1$  et  $\sum_{v_j \in V} p(v_i, v_j) = 1$ ). La matrice de transition peut ainsi être définie par  $\mathbf{P} = \mathbf{D}^{-1} \mathbf{W}$ .

Considérons  $p^t(v_i, v_j)$ , le noyau correspondant à la  $t^e$  puissance de  $\mathbf{P}$ ,  $\mathbf{P}^t$ , qui peut être interprété comme la probabilité pour un marcheur d'atteindre le sommet  $v_j$  en partant du sommet  $v_i$  en passant par  $t$  étapes. L'intérêt d'introduire cette matrice de transition est que l'exploration du graphe, par la marche aléatoire qu'elle engendre, permet de déterminer des propriétés topologiques du graphe reliées aux propriétés spectrales de  $\mathbf{P}$ .

$\mathbf{P}$  est généralement symétrique et pour chaque colonne la somme des éléments est égale à 1. Cette matrice est intéressante car elle reflète la géométrie intrinsèque des données. Une marche aléatoire correspond à une chaîne de Markov homogène puisque les probabilités de transition restent les mêmes à chaque fois que l'on revient sur un nœud du graphe. Les chaînes de Markov sont définies en termes d'états et de transitions entre ces derniers. Les états sont, dans notre cas, les nœuds du graphe.

Dans une chaîne de Markov, deux états  $i$  et  $j$  sont dits communicants si l'on peut atteindre l'un à partir de l'autre avec une probabilité finie, ce qui signifie que le graphe est connexe. Si l'on veut décrire la probabilité de transition  $p^t(v_i, v_j)$  d'un nœud  $v_i$  à un nœud  $v_j$  en  $t$  étapes, il suffit de considérer des voisinages plus larges, ce qui correspond à élever la matrice  $\mathbf{P}$  à la puissance  $t$ .

Si le graphe est connexe et non bipartite, alors la marche aléatoire converge vers une distribution unique stationnaire  $\pi = [\pi_1, \dots, \pi_n]$  satisfaisant  $\mathbf{P}^t \pi = \pi$ .

$\pi$  est un vecteur qui traduit une distribution statistique de l'occupation des états de la chaîne de Markov correspondant à la marche aléatoire. De plus, pour une distribution initiale  $\pi_0$  quelconque,  $\lim_{t \rightarrow \infty} \mathbf{P}^t(x, y) = \pi_0 = \phi_0(y)$  avec  $\phi_0(x) = \frac{d(x)}{\text{vol}(V)}$ .

La distribution stationnaire  $\phi_0$  est un vecteur propre de  $\mathbf{P}$  tel que :

$$\phi^T P = \phi_0^T$$

Coifman et al. montrent que la décomposition spectrale du noyau  $p^t$  est

$$p^t(v_i, v_j) = \sum_{l \geq 0}^{N-1} \lambda_l^t \psi_l^t(v_i) \phi_l^t(v_j), \quad (6.3)$$

avec  $\{\lambda_l^t\}$ , l'ensemble des valeurs propres de  $\mathbf{P}^t$  tel que :

$$1 = |\lambda_0^t| \geq |\lambda_1^t| \geq \dots \geq 0$$

$\{\phi_l^t(v_j)\}$  et  $\{\psi_l^t(v_i)\}$  représentent respectivement les vecteurs propres gauche et droite qui vérifient l'équation suivante :

$$\phi_0(x) \psi_l(x) = \phi_l(x)$$

### 6.1.3.2 Distance de diffusion

La distance de diffusion  $D_t(x_i, x_j)$  entre deux points  $x_i$  et  $x_j$  peut être calculée par :

$$D_t^2(v_i, v_j) = \sum_{v_l \in V} \frac{(p^t(v_i, v_l) - p^t(v_j, v_l))^2}{\phi_0(v_l)} \quad (6.4)$$

Une simple distance  $L_2 \|\cdot\|$  entre les deux probabilités  $p^t(v_i, \cdot)$  et  $p^t(v_j, \cdot)$  peut mesurer la connectivité entre les deux points  $x_i$  et  $x_j$  en  $t$  étapes (ou chemins de longueur  $t$ ). En combinant (6.3) et (6.4), on obtient :

$$D_t^2(v_i, v_j) = \sum_{l \geq 0} (\lambda_l^t \psi_l^t(v_i) - \lambda_l^t \psi_l^t(v_j))^2, \quad (6.5)$$

car  $\phi_0$  est un vecteur constant.

L'équation (6.5) montre que les vecteurs propres droite de  $\mathbf{P}^t$  peuvent être utilisés pour exprimer la distance de diffusion. Ainsi, on a introduit une famille de cartes de diffusion indexée par le paramètre  $t$  :

$$\boldsymbol{\psi}^t(x) = \begin{pmatrix} \lambda_0^t \boldsymbol{\psi}_0^t(x) \\ \lambda_1^t \boldsymbol{\psi}_1^t(x) \\ \vdots \\ \lambda_{N-1}^t \boldsymbol{\psi}_{N-1}^t(x) \end{pmatrix} \quad (6.6)$$

Par la suite, nous omettons le paramètre  $t$  qui sera fixé comme paramètre d'entrée. À partir de l'équation (6.5), nous pouvons remarquer que la distance de diffusion génère une cartographie quasi isométrique tant qu'elle est calculée par une distance  $L_2$  dans l'espace réduit en utilisant les premiers vecteurs propres. Notons aussi que la chaîne de Markov (à densité invariable) peut être construite comme suit :

$$\tilde{w}(v_i, v_j) = \frac{w(v_i, v_j)}{d(v_i)d(v_j)} \quad (6.7)$$

Ainsi, la nouvelle probabilité de transition devient :

$$p(v_i, v_j) = \frac{\tilde{w}(v_i, v_j)}{\tilde{d}(v_i)}, \quad (6.8)$$

avec  $\tilde{d}(v_i) = \sum_{v_j \in V} \tilde{w}(v_i, v_j)$ .

De cette manière, nous pouvons recouvrir la géométrie de la variété indépendamment de la densité des données. L'ensemble des données  $\{x_i\}_{i=1..N}$  peut être représenté par :

$$y_i = \boldsymbol{\psi}_t(x_i) = \begin{pmatrix} \lambda_0 \boldsymbol{\psi}_0(x_i) \\ \lambda_1 \boldsymbol{\psi}_1(x_i) \\ \vdots \\ \lambda_{d-1} \boldsymbol{\psi}_{d-1}(x_i) \end{pmatrix} \quad (6.9)$$

avec  $d \ll N$ .

Le but est de représenter chaque  $x_i \in \mathbb{R}^N$  par un point  $y_i \in \mathbb{R}^d$ , avec  $d \ll N$ , de sorte que l'ensemble  $Y = \{y_1, y_2, \dots, y_d\}$  capture toute l'information géométrique intrinsèque de l'ensemble d'origine. Pour cela, Belkin et Niyogi [Belkin et Niyogi, 2003] ont montré que l'utilisation de l'opérateur de Laplace-Beltrami constitue un choix intéressant pour la prise en compte de l'information géométrique. En particulier, l'information locale est bien préservée par l'utilisation de ces fonctions propres (*eigenmaps*).

Le calcul du graphe Laplacien pondéré et l'approximation de l'opérateur Laplace-Beltrami sont obtenus par l'algorithme 6.1.

*Remarque 5.* Nous retrouvons des cas classiques pour certaines valeurs de  $\alpha$ .

- $\alpha = 0$  : Graphe de Laplacien.
- $\alpha = 0,5$  : le propagateur Fokker-Plank.
- $\alpha = 1$  : l'opérateur de Laplace-Beltrami.

---

**Algorithme 6.1** Algorithme de graphe Laplacien pondéré
 

---

**ENTRÉES :**  $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^N, d, \varepsilon, \alpha$

**SORTIES :**  $Y = \{y_1, \dots, y_n\} \subset \mathbb{R}^d$

- 1: Construction de la matrice de similarité  $W$  en utilisant le noyau gaussien :  $k(x_i, x_j) = w_\varepsilon(x_i, x_j)$
- 2: Normalisation du noyau par la densité :

$$\widetilde{w}(x_i, x_j) = \frac{w(x_i, x_j)}{(d(x_i)d(x_j))^\alpha}$$

- 3: Construction de la matrice de transition  $\mathbf{P}$  avec le noyau :

$$\widetilde{p}(x_i, x_j) = \frac{\widetilde{w}(x_i, x_j)}{\widetilde{d}(x_i)}$$

avec  $\widetilde{d}(x_i) = \sum_j w'(x_i, x_j)$ .

- 4: Projection dans l'espace réduit et exploitation des distances de diffusion.
- 5: **Espace de diffusion**  $\{\lambda_i \psi_i\}$

$$x \rightarrow y = (\lambda_0 \psi_0(x_0), \lambda_1 \psi_1(x_1), \dots, \lambda_{d-1} \psi_{d-1}(x_{d-1}))$$

- 6: Distance de diffusion :

$$D_t^2(x_i, x_j) = \sum_{l \geq 0} \left( \lambda_l^2 (\psi_l(x_i) - \psi_l(x_j))^2 \right) = \sum_{x_l \in V} (p(x_i, x_l) - p(x_j, x_l))^2$$

avec  $1 = |\lambda_0| \geq |\lambda_1| \geq \dots \geq 0$

- 7: Utilisation d'un classifieur comme les *k-means*.
-

La première normalisation de la matrice de similarité permet de trouver une représentation indépendante de la distribution. La seconde sert à rendre le noyau stochastique. Les vecteurs propres de la matrice de transition  $\mathbf{P}^t$  peuvent être interprétés comme la généralisation des fonctions de Fourier sur un graphe. Ainsi les valeurs propres de faibles valeurs correspondent aux vecteurs propres de hautes fréquences, et celles de fortes valeurs correspondent aux vecteurs de basses fréquences. Notons que l'on ne considère pas le premier vecteur propre car il ne porte aucune information. Notons également, que le signe de  $\psi_1$  permet de catégoriser l'ensemble des données  $X$  en deux ou plusieurs classes et que toute distance entre deux points dans l'espace de mesures sera transformée en une distance euclidienne dans l'espace de représentation, facilitant ainsi la comparaison de données et leur fusion. Ce second vecteur propre  $\psi_1$  est connu comme le vecteur de Fiedler et peut être utilisé pour ordonner l'ensemble des données  $X$ .

La distance de diffusion a une interprétation fonctionnelle. D'abord, elle peut être considérée comme une distance de diffusion entre  $x_i$  et  $x_j$  : elle mesure le taux de connectivité entre les points du jeu de données. Moins il y aura de chemins entre ces deux points, plus le taux sera petit. Inversement, plus il y aura de chemins entre ces deux points, plus le taux sera grand.

## 6.2 Résultats et applications : segmentation et classification

Dans cette partie, nous proposons divers applications de segmentation basées sur la régularisation spectrale sur graphes. Ensuite, nous appliquons les marches aléatoires à la caractérisation visuelle des objets, en particulier, à la texture. Cette approche, qui est une transformation non linéaire (transformation markovienne), est inspirée de la caractérisation linéaire (*eigen-transform*).

### 6.2.1 Segmentation par classification et diffusion

L'algorithme de segmentation que nous présentons est basé sur la classification et la diffusion des points caractéristiques de l'image. Il peut être décomposé en quatre étapes :

1. Extraction des points caractéristiques.
2. Construction du graphe de voisinage en considérant les patchs autour des points.
3. Projection des données dans l'espace réduit et classification par  $k$ -means.
4. Diffusion des germes classés dans l'espace initial.

À la **première étape**, notre choix de points caractéristiques s'est porté sur les extrema et sur les points SIFT (*Scale Invariant Feature Transform*). Plusieurs descripteurs ont été proposés dans la littérature et le descripteur d'images SIFT est aujourd'hui considéré comme étant le plus performant [Mikolajczyk et Schmid, 2005]. Ce descripteur introduit par Lowe [Lowe, 1999], décrit le voisinage local d'un point par un histogramme 3D de la distribution des orientations du gradient. C'est un détecteur de points d'intérêts de Harris-Laplace, invariant par échelle, qui extrait  $m$  points de l'image. Une fois les points d'intérêt détectés, la région de l'image autour de chaque point est utilisée pour calculer un descripteur. En fonction de la complexité de l'image, entre 100 et 300 points sont détectés. Pour chacun de ces points, un vecteur caractéristique de dimension 128 est calculé à partir des pixels voisins. Il est alors possible d'établir des correspondances entre les points d'intérêts détectés dans les images en utilisant une mesure de similarité entre les descripteurs locaux.

L'information des extrema locaux permet de réduire le nombre de pixels à traiter dans l'image tout en donnant une bonne description de l'image également.

La **deuxième étape** consiste à associer, à chacun des points, un patch sur lequel sera calculé le descripteur caractéristique. Nous obtenons ainsi des descripteurs locaux qui encode l'information utile. Bien que l'information des extrema locaux va nous permettre de réduire l'espace de données à traiter, nous espérons qu'elle permette de distinguer le fond des objets d'intérêt.

La **troisième étape** comporte la décomposition spectrale du noyau de diffusion et la projection des coordonnées de diffusion sur l'espace réduit des deux premiers vecteurs propres engendrés par les deux valeurs propres  $\lambda_1$  et  $\lambda_2$ . Ensuite, le nuage de points dans l'espace réduit fait l'objet d'une classification par  $k$ -means par exemple.

Dans la **quatrième étape**, il s'agit de revenir à l'espace de départ pour regrouper les nœuds dont les représentants dans l'espace réduit ont été regroupés dans une même classe. L'idée de base de notre approche de segmentation est basée sur la diffusion géométrique globale des maxima locaux sur graphes. Ensuite, une fois les germes sont classés, nous utilisons la méthode de segmentation des lignes de partage des eaux (LPE) pour segmenter les objets d'intérêts. Nous pouvons également utiliser le principe de diffusion



de labels pour la segmentation semi-supervisée que nous avons présenté dans 4.2.4.3. Dans ce cas, les labels correspondent aux germes classés. L'algorithme suivant présente les principales étapes de construction de notre approche de segmentation :

---

**Algorithme 6.2** Segmentation par classification-diffusion

---

**ENTRÉES :** l'image  $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^n$  et  $Y$  les points caractéristiques de  $X$

- Construction et décomposition spectrale de noyau de diffusion.
  - Classification des points caractéristiques avec l'opérateur de diffusion géométrique par marches aléatoires sur  $Y : diff: Y \rightarrow C(Y)$  tel que  $diff(y) = c_i \cdot i \in [2, \dots, k]$ , où  $C(Y)$  est l'ensemble des partitions de  $Y$  et  $c_i$  est la classe d'appartenance de chaque point caractéristique de l'ensemble  $Y$ . Dans notre cas, l'ensemble  $Y$  correspond à deux classes :  $C_F$  (le fond) et  $C_2$  (l'objet).
  - Croissance de régions avec la ligne de partage des eaux (LPE) ou diffusion de labels par fonctions  $p$ -harmoniques.
- 

La figure 6.2.1 illustre les différentes étapes de l'algorithme 6.2. Des résultats de segmentation obtenus par diffusion de labels à la figure 6.2.

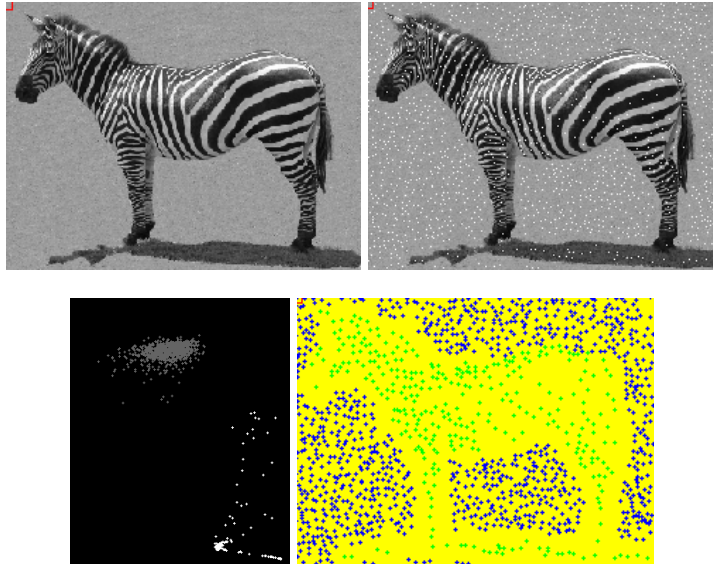


FIGURE 6.1 – Transformation du problème de l'espace initial à l'espace spectral. De gauche à droite et de haut en bas, l'image d'origine, les extrema, la projection dans l'espace réduit et le résultat de la classification par  $k$ -means.

Les figures 6.3, 6.4 et 6.5 montrent le regroupement spectral des patches similaires en utilisant la distance SIFT sur des images texturées.

### 6.2.2 Caractérisation des éléments saillants

Cette méthode de détection de saillance repose sur la caractérisation des interactions entre pixels dans un voisinage local. Elle est basée sur les marches aléatoires locales sur un graphe de grande dimension. Les arrêtes sont définies en termes de mesures de similarité entre les propriétés locales des pixels ou des motifs considérés. Le principe de cette méthode repose sur l'exploration du graphe par analyse spectrale, ce qui permet de capturer les interactions locales et non locales entre pixels ou motifs des structures importantes telles que la texture.

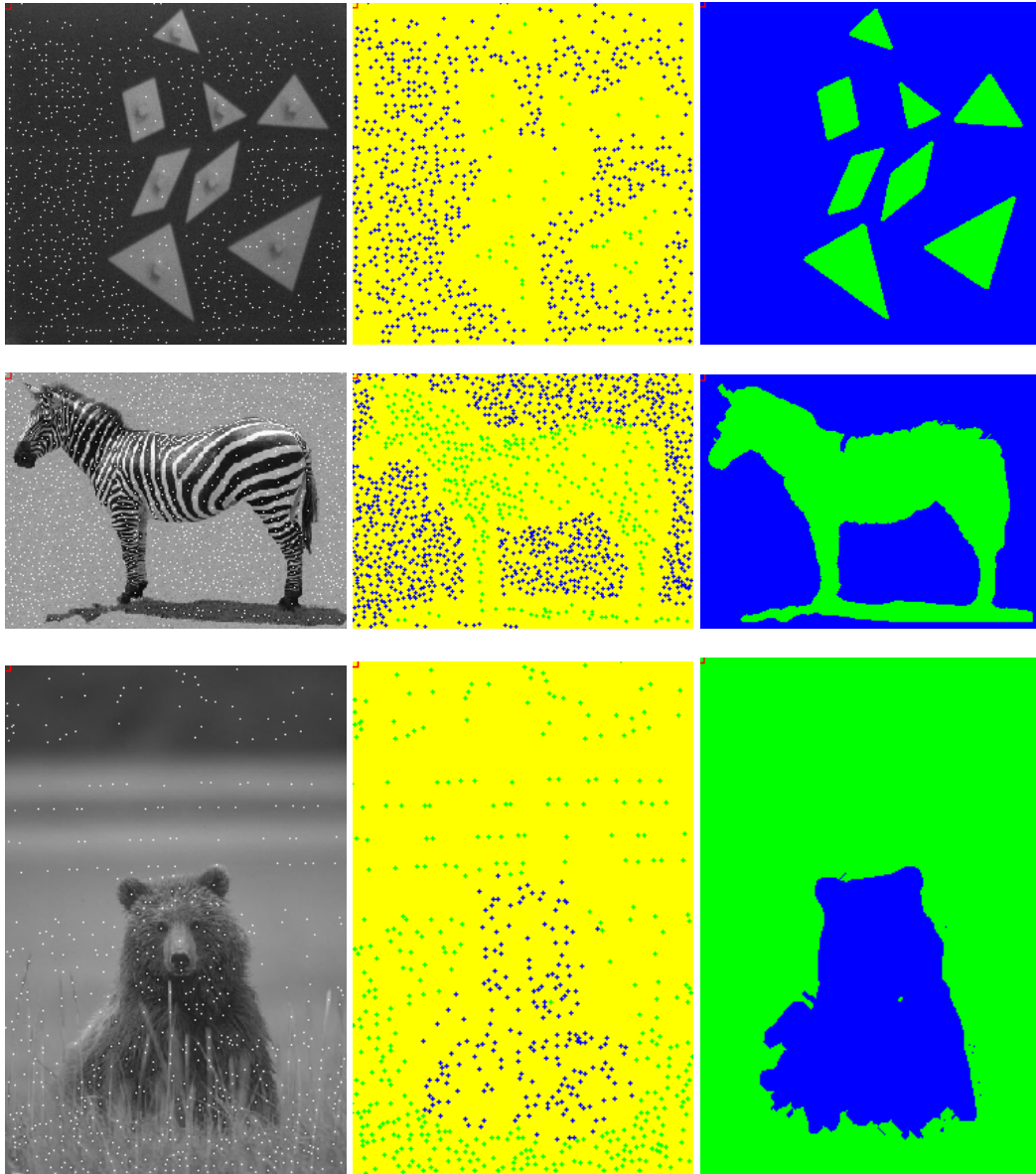


FIGURE 6.2 – Résultats de segmentation. De gauche à droite, les germes, le résultat de la classification par  $k$ -means et la segmentation obtenue par LPE.

Le descripteur, formé de la somme des valeurs propres, fournit l'information sur la dépendance linéaire entre les lignes et les colonnes de la matrice locale des descripteurs (niveaux de gris/couleur) de chaque pixel (voir algorithme 6.3).

*Remarque 6.* Le descripteur  $\Gamma(x_i)$  est égal à la trace de la matrice de transition représentant le graphe :  $\sum_i \lambda_i$ . Ainsi, le descripteur peut être calculé sans passer par la décomposition spectrale de la matrice de transition  $P$ . Ceci offre le grand avantage de se défaire des difficultés calculatoires que peut présenter la décomposition spectrale.

La figure 6.6 montre des résultats de l'utilisation de ce descripteur dans un processus de caractérisation et de segmentation d'objets visuels.

Cette approche est facilement extensible aux séquences d'images. Grâce à la représentation sur graphes, les résultats doivent être aussi intéressants que sur les images.

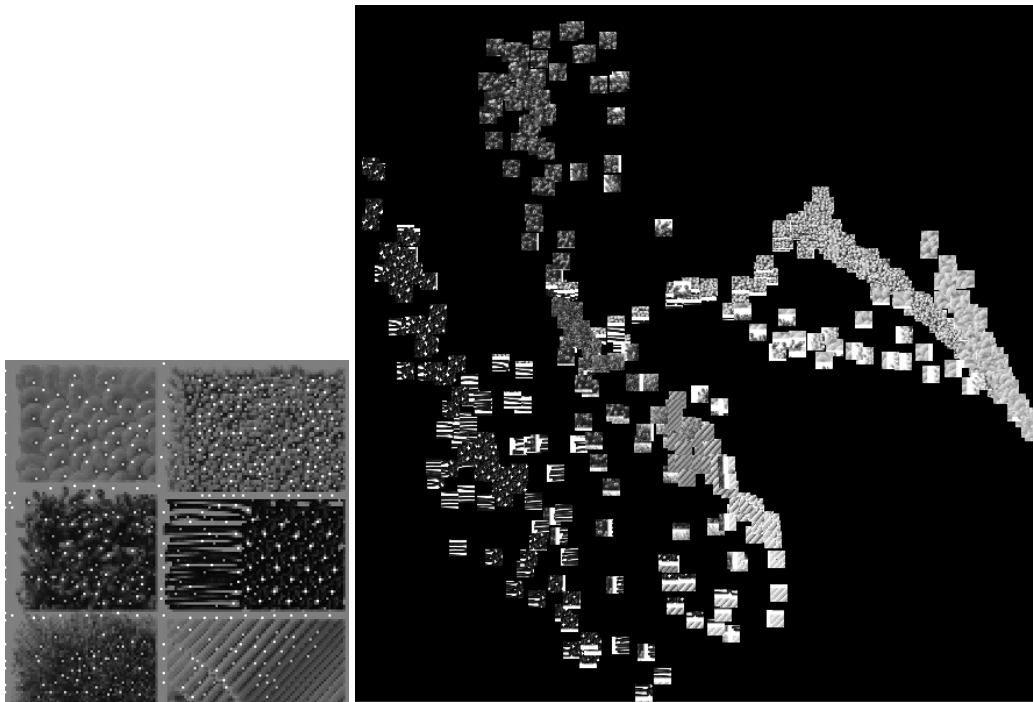


FIGURE 6.3 – Classification des patches  $15 \times 15$  autour des points SIFT sur l'image de textures, *rugs*. De gauche à droite, les points SIFT et la projection dans l'espace réduit.



FIGURE 6.4 – Classification des patches  $20 \times 20$  autour des points SIFT sur l'image de *Lena*. De gauche à droite, les points SIFT et la projection dans l'espace réduit.

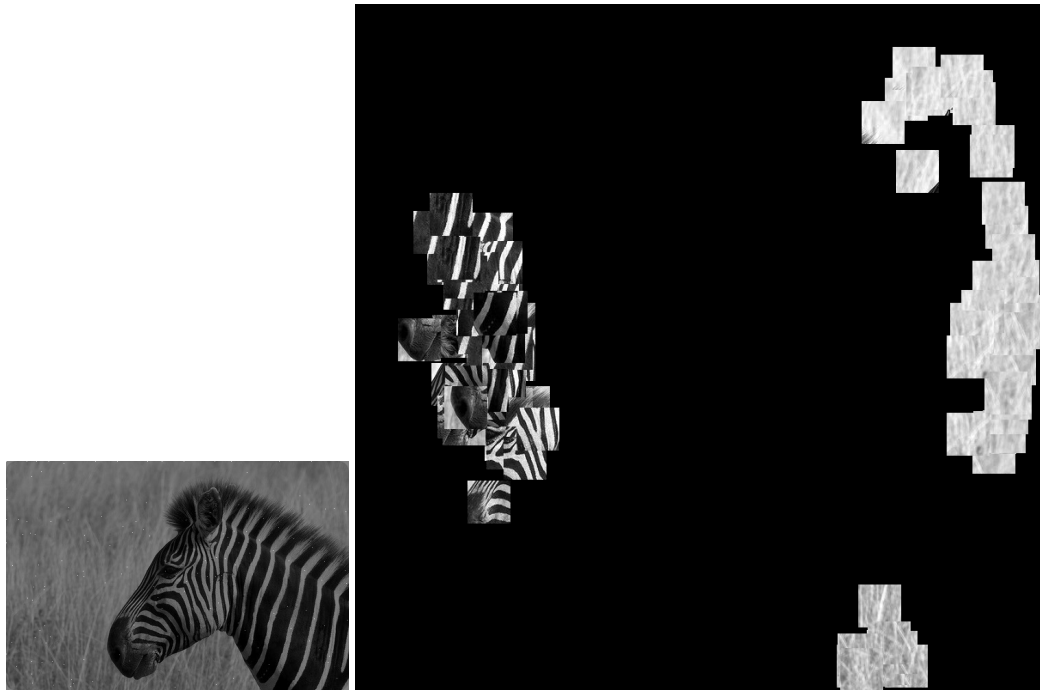


FIGURE 6.5 – Classification des patches  $50 \times 50$  sur l'image *Zebra* autour des points SIFT sur l'espace réduit.

---

**Algorithme 6.3** Caractérisation visuelle par analyse spectrale

---

ENTRÉES : l'image  $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^n$

**pour tout**  $x_i$ , **faire**

Construction et décomposition de la matrice de transition de Markov de chaque fenêtre locale de  $x_i$ . Récupération des fonctions propres de la matrice de transition.

$1 = \lambda_0 \geq \lambda_1 \geq \dots \geq 0$  et  $\{\psi_j\}_{j=i,w}$  tel que  $P\psi_j = \lambda_j\psi_j$

Remplacer  $x_i$  par la somme des valeurs singulière :  $\Gamma(x_i) = \sum_{h=r}^w \lambda_h$ .

**fin pour**

---



(a) Image initiale



(b) Résultat

FIGURE 6.6 – Résultats de caractérisation visuelle.

### 6.2.3 Classification de lettres manuscrites

Après avoir rappelé les propriétés spectrales des marches aléatoires sur graphes, nous présentons dans cette section une application de classification de données hétérogènes liées à l'écriture. Cette application s'inscrit dans un projet réel sur l'apprentissage d'écriture.

#### 6.2.3.1 Contexte et objectifs de l'application

Le projet repose sur des productions écrites de chaque lettre de l'alphabet par des enfants scolarisés du CP au CM2. L'objectif principal est de concevoir un système d'aide à la reconnaissance des lettres et de mettre en évidence certains gestes par rapport au modèle du maître. À terme, le projet est destiné à aider les enfants à l'apprentissage de l'écriture, en analysant et en corrigeant le tracé d'une lettre. Cette analyse passe par l'extraction de primitives de bases d'une lettre et leur catégorisation. Cette étude s'effectue en collaboration entre les psychologues de l'Université de Caen de l'équipe PALM et notre équipe du GREYC.

À ce jour, nous avons récolté les productions de 300 enfants à l'aide d'une tablette graphique. Chaque production comporte une trace de la dynamique d'écriture, un rendu image ainsi que d'autres données liées aux *conditions* d'écriture (voir figure 6.7).

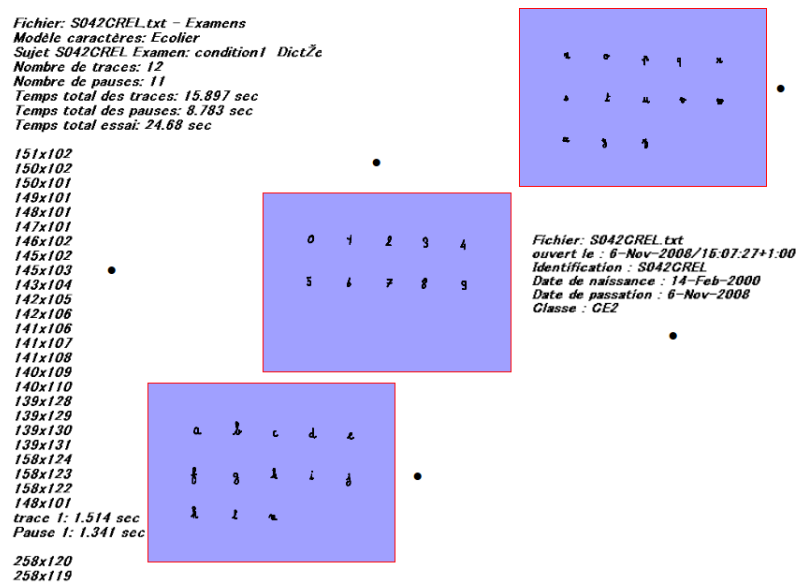


FIGURE 6.7 – Exemple des données hétérogènes d'écriture.

La figure 6.8 présente la chaîne de traitements pour la classification des lettres.

#### 6.2.3.2 La reconnaissance des formes

La reconnaissance des formes consiste à associer à une donnée une signification symbolique en lui attribuant une étiquette. La difficulté consiste à reconnaître un objet malgré les difficultés telles que le changement de point de vue, des conditions d'observation.

L'étape de reconnaissance ne peut s'opérer directement sur les données brutes. Étant donné le très grand nombre d'informations, il est nécessaire d'effectuer une sélection de ces informations et de ne garder que les plus pertinentes pour réduire la dimension du problème.

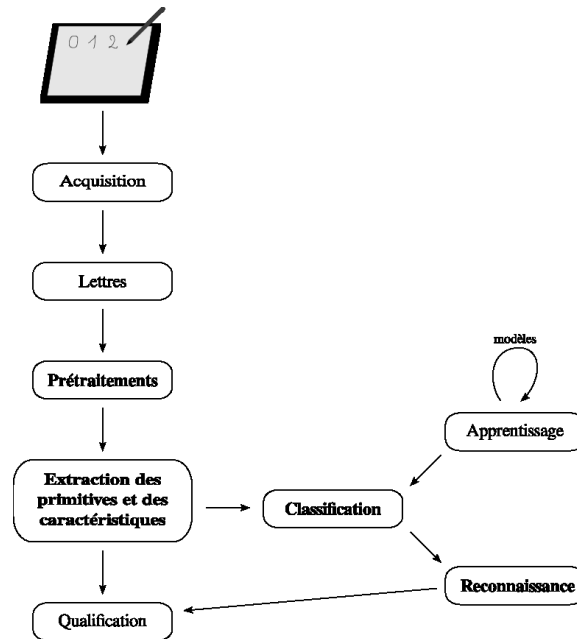


FIGURE 6.8 – Chaîne de traitements pour la reconnaissance des lettres.

Les caractéristiques à extraire sont nombreuses et variées. Il peut s'agir de caractéristiques *globales* (surface, périmètre, largeur, hauteur, élongation, circularité, moments statistiques) ou *locales* (coins, sommets, angles, segments). L'important est de retenir les caractéristiques les plus informatives et les plus pertinentes, c'est-à-dire celles qui garantissent une faible variabilité intra-classe et une grande variabilité inter-classes.

Les processus mis en jeu pour la reconnaissance des formes suivent généralement les étapes suivantes :

1. PRÉTRAITEMENTS
  - réduction du bruit,
  - binarisation,
  - localisation,
  - normalisation, ...
2. EXTRACTION DE CARACTÉRISTIQUES
  - intensité des pixels, textures,
  - concavités, occlusions, droites,
  - aire, périmètre, circularité,
  - histogramme,
  - compacité,
  - moments invariants,
  - contours, squelette, ...
3. CLASSIFICATION
  - apprentissage,
  - décision, ...

**Les prétraitements** Afin d'améliorer l'efficacité de l'algorithme de reconnaissance d'écriture, il est nécessaire d'appliquer des prétraitements particuliers sur les caractères. Il s'agit principalement de traiter les caractères tracés afin qu'ils correspondent au mieux aux modèles de caractères mais aussi de réduire le bruit induit par le scripteur. En effet, les principaux problèmes de la reconnaissance des caractères manuscrits sont la présence de bruit



et surtout la variabilité inter-scripteur : la taille, l'inclinaison, la structure des traits, ...

Une étape de régularisation permet de diminuer l'effet d'escalier des traces dû à l'échantillonnage du système de capture de l'écriture. Le nombre de points capturés dépend de la vitesse à laquelle on exécute un geste et de la fréquence d'échantillonnage du système de capture. La figure 6.9 illustre la simplification du caractère à la suite des prétraitements.

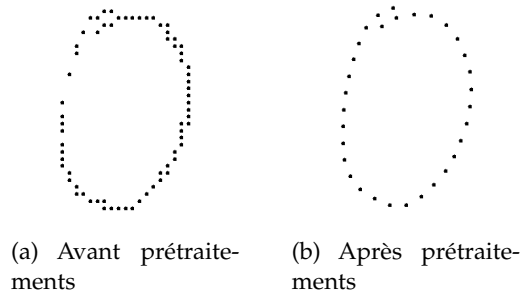


FIGURE 6.9 – Un exemple de caractère avant et après prétraitements.

**Analyse des primitives** Les primitives permettent d'analyser les caractéristiques d'un caractère. On distingue deux types de primitives : les primitives *globales* et les primitives *classiques*. De même, on peut ensuite distinguer les primitives *topologiques*, *structurelles* ou *statistiques*. Les primitives topologiques consistent à mesurer dans une image brute le nombre de trous, les concavités, les pentes, l'orientation, etc. Les primitives structurelles s'appliquent au squelette de la forme ou à son contour, il s'agit alors de déterminer des boucles au lieu de trous, etc.

Il existe plusieurs méthodes dans la littérature pour décrire le sens et la direction d'un tracé. Nous utilisons le *codage de Freeman* (voir figure 6.2.3.2) qui permet de mettre en évidence les composantes morphocinétiques des tracés de caractères.

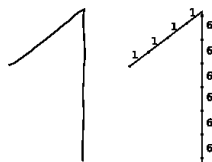


FIGURE 6.10 – Exemple de chaîne de Freeman pour le chiffre 1.

Notons que le codage de Freeman classique est invariant aux translations mais pas aux rotations. Pour y remédier, on peut utiliser le *codage de Freeman relatif* qui code le changement de direction plutôt que la direction elle-même. On peut ainsi calculer la longueur d'un chemin (8-connextité) :

$$L = \text{nombre de codes pairs} + \sqrt{2} \text{ nombre de codes impairs} \quad (6.10)$$

On peut également inverser un chemin, ce qui est utile pour repérer les cas où l'on ne trace pas un même caractère dans le même sens. Il suffit d'inverser tous les descripteurs de code, puis d'inverser la séquence. L'inverse d'un descripteur de code  $c$  (en  $n$ -connextité) est :

$$\bar{c} = \frac{n}{2} + c \cdot \text{mod}(n) \quad (6.11)$$

La réduction d'un chemin calcule sa longueur minimale pour relier les deux extrémités du tracé. On pourrait aussi tester si le tracé se recoupe à un endroit, ce qui mettrait en évidence les points de jonction. La détection des boucles dans un caractère peut servir pour préciser l'appartenance à des classes génériques. En effet, en fonction de la présence ou non d'une boucle, de sa position (supérieure, médiane, inférieure), du nombre de boucles, on peut classer les caractères en plusieurs catégories.

Les points dominants sont les points de forte courbure, les maxima locaux de courbure, c'est-à-dire les points symbolisant un changement significatif de direction dans la trajectoire.

**Visualisation et organisation** La classification des lettres se base sur le choix de la distance utilisée pour comparer les différents éléments. Nous présentons les résultats de visualisation d'un ensemble d'écrits d'un même caractère en utilisant les cartes de diffusion avec une similarité basée sur les points SIFT. La mesure de similarité que nous avons retenu est basée sur les points SIFT pour leurs propriétés d'invariances, notamment par changement d'échelle :

$$\sum_{k=0}^{127} \left\| \sqrt{(S_k^i - S_k^j)} \right\|$$

Par conséquent, la variabilité de l'écriture au sein d'une même catégorie est mise en évidence (voir figure 6.11).

La reconnaissance est l'une des dernières étapes intervenant dans le processus de classification, après l'acquisition des données et l'extraction d'attributs caractéristiques. Elle permet d'obtenir une catégorisation des caractères selon des caractéristiques précises en mesurant la similarité avec les classes référentes. Concrètement, il s'agit de déterminer, à partir d'un tracé inconnu, la classe d'appartenance du caractère parmi les chiffres (10 classes), les lettres (26 classes) elles-mêmes subdivisées en lettres minuscules/majuscules et aussi cursives/d'imprimerie, soit un total de 114 classes.

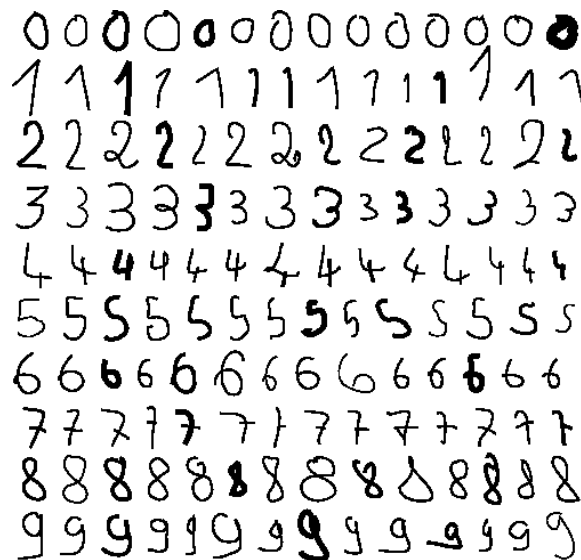


FIGURE 6.11 – Variabilité inter-scripteurs de l'écriture pour dix classes.

**Conclusion** L'avancement actuel des travaux n'a pas atteint l'étape de la classification sur la base d'écrits à notre disposition. Cependant, nous avons effectué les travaux préliminaires nécessaires à la classification. L'étape restante est l'application d'un classifieur et

l'analyse des résultats obtenus. Par la suite, nous nous intéressons à l'aspect dynamique de l'écriture en considérant les données temporelles capturées durant l'écriture.

### 6.2.3.3 Analyse des données temporelles

L'analyse des données temporelles liées à l'écriture nécessite le recours à des méthodes de traitement de séries temporelles. Pour cela, nous utilisons une distance appropriée aux données temporelles.

**Distance par déformation temporelle (*Dynamic Time Warping*)** L'algorithme de *dynamic time warping* 6.4, que l'on peut traduire par « déformation temporelle dynamique », permet de comparer la similarité entre deux séquences qui varient dans le temps. Il a été initialement utilisé pour la reconnaissance automatique de la parole et a fourni de très bons résultats. Il permet de comparer des séquences ayant un nombre différent de valeurs et de résoudre les problèmes de distorsion sur l'axe de temps et de variation d'échelle de temps. La particularité de cet algorithme est sa capacité à gérer les dilatations et/ou les compressions temporelles qui peuvent exister entre les séries comparées.

En effet, au lieu d'associer chaque point d'une série à celui d'une autre série qui intervient au même instant, la DTW permet d'associer chaque point d'une série à un ou plusieurs points consécutifs de l'autre série. Ces derniers peuvent être décalés dans le temps, afin de rechercher le meilleur alignement des deux séries à comparer.

L'inconvénient de cet algorithme, dans sa version de base, est qu'il nécessite un temps de calcul important qui augmente avec le nombre d'éléments dans les séries comparées. Des techniques d'optimisation de cet algorithme sont proposées afin de réduire le temps de calcul sans détériorer ses performances [Salvador et Chan, 2007]. Cette distance est une semi-métrique car elle ne respecte pas l'inégalité triangulaire. Le principe de la distance par déformation temporelle consiste à mettre en correspondance les sous-séquences qui « se ressemblent » même si elles ne correspondent pas au même intervalle de temps (voir figure 6.2.3.3).

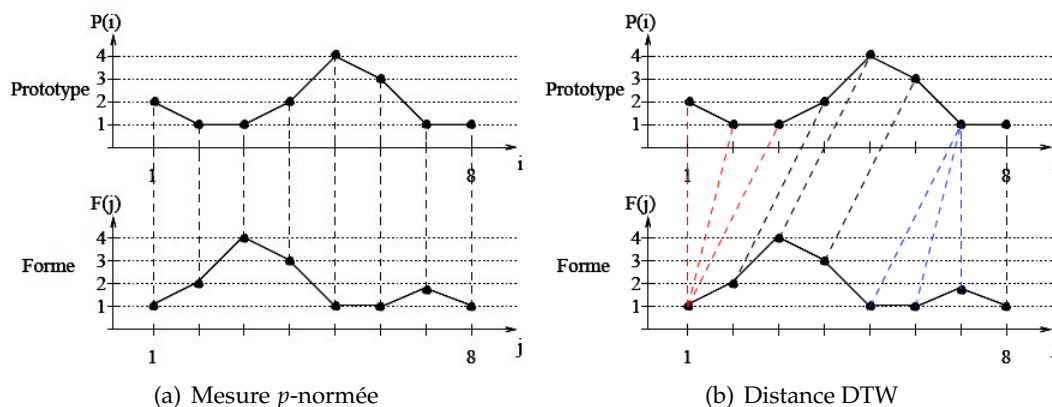


FIGURE 6.12 – Illustration de l'intérêt de la DTW (6.12(b)) par rapport à la mesure euclidienne (6.12(a)) pour la comparaison des séries temporelles.

Cet algorithme permet de trouver un alignement optimal entre deux séquences données sous certaines contraintes comme la monotonie, la continuité, etc. Ce principe permet de suivre la trajectoire d'un tracé en temps-réel et de la comparer avec des modèles. Il

permet non seulement de vérifier si le sens du tracé est conforme à un modèle, mais aussi d'estimer la distance de la trajectoire du tracé avec les trajectoires des modèles.

$$DTW(courbe_1, courbe_2) = \min_w \frac{1}{N} \sum_{k=1}^{k=N} w_k \quad (6.12)$$

---

**Algorithme 6.4** Algorithme DTW
 

---

**ENTRÉES :**  $courbe_1[1..l_1]$ ,  $courbe_2[1..l_2]$

**SORTIES :** la distance dtw entre  $courbe_1$  et  $courbe_2$

DTW = tableau[0.. $l_1$ ][0.. $l_2$ ]

$i, j$ , cout = 0

**pour**  $i$  de 1 à  $l_2$  **faire**

DTW[0][ $i$ ] =  $\infty$

**fin pour**

**pour**  $i$  de 1 à  $l_1$  **faire**

DTW[ $i$ ][0] =  $\infty$

**fin pour**

**pour**  $i$  de 1 à  $l_1$  **faire**

**pour**  $j$  de 1 à  $l_2$  **faire**

cout = distance( $courbe_1[i]$ ,  $courbe_2[j]$ )

DTW[ $i$ ][ $j$ ] = cout + min(DTW[ $i-1$ ][ $j$ ], DTW[ $i$ ][ $j-1$ ], DTW[ $i-1$ ][ $j-1$ ])

**fin pour**

**fin pour**

**return** DTW[ $l_1$ ][ $l_2$ ]

---

**Application à la reconnaissance des caractères** Pour évaluer l'intérêt de la distance DTW, nous avons développé une première application pour la reconnaissance des dix chiffres. Le principal problème de la reconnaissance d'écriture est de faire face à la grande variabilité inter-scripteurs de l'écriture due au style propre de chaque écriture (taille, vitesse, inclinaison, forme).

On utilise la méthode du perceptron<sup>1</sup> pour la classification. Le perceptron peut être défini comme le type de réseau de neurones le plus simple. Soit un vecteur  $w \in \mathbb{R}^n$ , la fonction  $f_w$  de  $\mathbb{R}^n$  dans l'ensemble 0, 1 des booléens est définie de la manière suivante :

$$f_w(x) = \begin{cases} 1 & \text{si } \sum_i w_i x_i > 0 \\ 0 & \text{sinon} \end{cases} \quad (6.13)$$

La fonction  $f_w$  est appelée un perceptron.  $f_w$  définit la séparation des éléments de  $\mathbb{R}^n$  situés des deux côtés de l'hyperplan défini par  $w$ . On cherche un vecteur  $w$  tel que pour tout élément  $(x, y)$  de  $D$ , on a  $f_w(x) = y$  :

1:  $w \leftarrow (0, \dots, 0)$

2: **tantque** il existe un élément  $(x, y)$  de  $D$  tel que  $f_w \neq y$  **faire**

3:  $w_i \leftarrow w_i + (y - f_w(x))x_i$  pour tout  $i$

4: **fin tantque**

L'algorithme ci-dessus se termine sur la solution  $w$  si elle existe.

Soit une image en noir et blanc représentant un chiffre manuscrit. Prenons par exemple l'image 6.14(a) de taille 100x100 pixels qui représente le chiffre 8. On détermine la boîte

---

1. Référence pour la description du perceptron : <http://www.enseignement.polytechnique.fr/profs/informatique/Jean-Christophe.Filliatre/09-10/INF431/perceptron/>

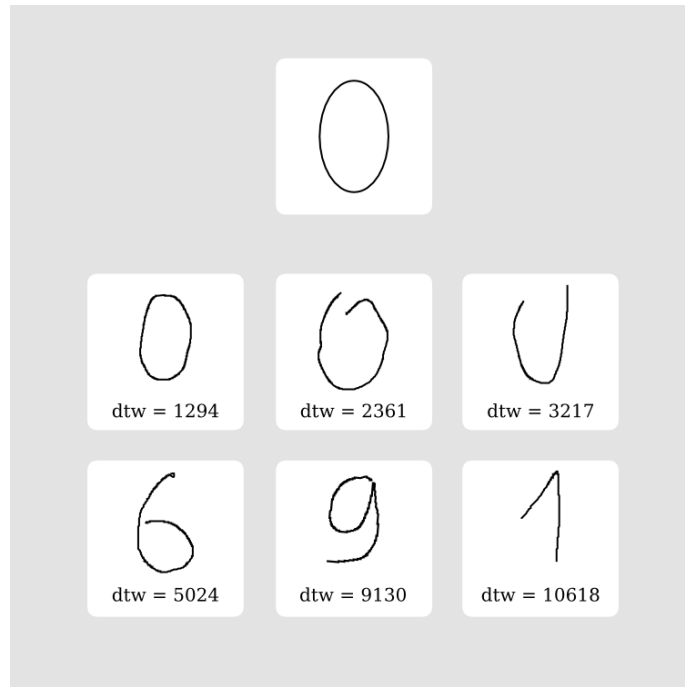


FIGURE 6.13 – Distances DTW pour des exemples de la lettre 0.

englobante (*bounding box*) contenant les pixels de l'image, comme sur la figure 6.14(b). Puis on la subdivise en  $w$  colonnes et  $h$  lignes,  $w$  et  $h$  étant fixés. Par exemple, avec  $w = 3$  et  $h = 5$ , on obtient le découpage 6.14(c). On calcule ensuite la proportion de pixels noirs pour chacune des cases obtenues. On obtient  $(w * h)$  réels compris entre 0 et 1, soit un vecteur de  $\mathbb{R}^n$  où  $n = w * h + 1$ . Dans notre exemple, on a  $n = 16$  et on obtient le vecteur suivant :  $V = (0.010, 0.095, 0.099, 0.010, 0.005, 0.094, 0.012, 0.023, 0.005, 0.096, 0.000, 0.012, 0.019, 0.010, 0.018)$  en considérant le sens de haut en bas et de gauche à droite.

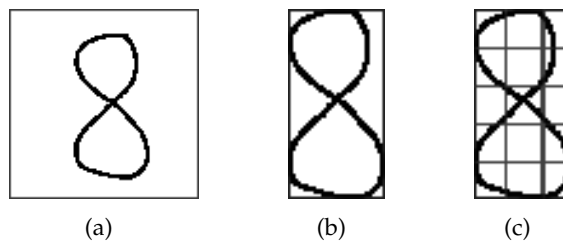


FIGURE 6.14 – Traitements d'une image du chiffre 8.

À partir d'images référentes représentant plusieurs échantillons de chacun des dix caractères 0 à 9, on construit dix perceptrons  $p_0, \dots, p_9$  reconnaissant chacun un caractère. Pour chaque caractère  $i$ , le perceptron  $p_i$  est établi en considérant toutes les images représentant  $i$  comme devant être acceptées et toutes les autres images comme devant être refusées. Ensuite pour reconnaître un caractère, on le présente à chacun des dix perceptrons. Si l'un d'eux et un seul l'accepte, alors on a identifié correctement le caractère. Sinon on considère que la reconnaissance est incertaine.

Pour pouvoir tester la reconnaissance et mesurer les performances, une applet<sup>2</sup> Java (voir capture à la figure 6.15) implémentant ce système a été réalisée. Des modèles de

2. L'applet Java est l'œuvre de E. Pillu.

lettres (les chiffres de 0 à 9) sont présents par défaut, mais lorsque l'on crée et teste soi-même ses modèles, la reconnaissance est bien meilleure. Naturellement, le système peut aussi accueillir d'autres caractères que les chiffres. Il suffit d'ajouter les modèles correspondants.

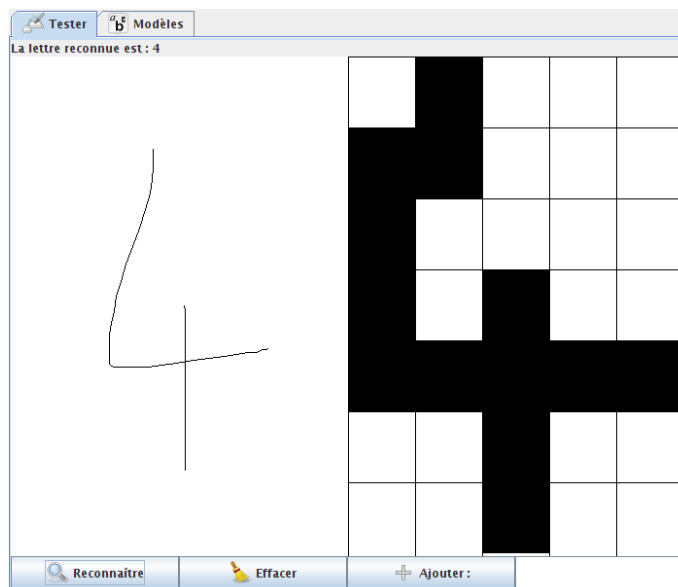


FIGURE 6.15 – Interface de l'applet Java pour la reconnaissance de caractères.

Nous obtenons comme résultats préliminaires, un taux de réussite d'environ 75%, sans compter que les modèles ne contiennent qu'un seul échantillon de chaque classe et que chaque modèle est tracé à la main. Ce système présente l'avantage d'être sensible au sens du tracé et de prendre en compte la façon dont est écrite une lettre comme nous pouvons le voir sur la figure (fig :reconnaissance). De plus amples expérimentations sont en perspectives.

## Conclusion

Dans ce chapitre, nous avons présenté un cadre générique de classification et de caractérisation par cartes de diffusion. Dans un premier temps, nous avons utilisé les points caractéristiques tels que les extrema et les points SIFT pour réduire l'information dans les images. Ceci nous a permis de proposer un algorithme rapide de segmentation par classification et diffusion de ces points. Ensuite, nous avons appliqué ce cadre à la caractérisation des éléments saillants. Et enfin, nous avons présenté nos travaux actuels sur la classification des données temporelles dans le cadre d'un projet sur des données réelles recueillies pour l'apprentissage de l'écriture. Ces données nécessitent des distances appropriées. Nous avons fait le choix d'utiliser la distance DTW pour traiter ces données. Les premiers résultats de classification reflètent la qualité des résultats des étapes intermédiaires de choix des distances, de visualisation et de catégorisation. La suite des travaux doit permettre d'évaluer précisément les résultats de la classification.

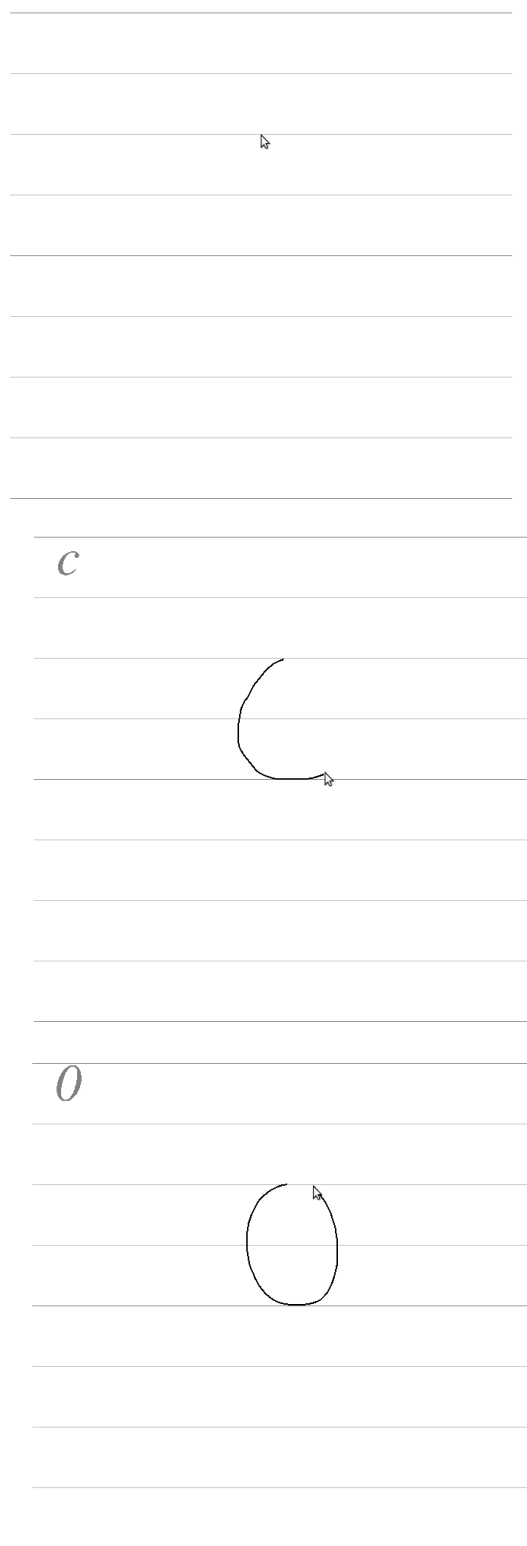


FIGURE 6.16 – Reconnaissance des caractères pendant l'écriture.



---

# CONCLUSION ET PERSPECTIVES

---

Dans le présent travail, nous avons eu l'occasion de nous intéresser à la régularisation spatio-temporelle sur graphes. Tout d'abord, nous avons étendu un formalisme des équations aux différences partielles pour effectuer deux applications de régularisation : le débruitage et la simplification, et l'inpainting et l'interpolation des images et des séquences d'images. Ensuite, nous avons proposé une approche de régularisation spectrale. Dans un premier temps, nous avons établi des méthodes pour la segmentation et la caractérisation d'éléments saillants des images. Nous avons également présenté une application réelle de classification de données sur laquelle nous travaillons encore dans le cadre d'un projet d'apprentissage de l'écriture.

## Débruitage et simplification

Dans la première partie, nous avons abordé la régularisation spatio-temporelle des images et des vidéos et ses applications au débruitage et à la simplification. Nous avons dressé un panorama des techniques de lissage pour la réduction du bruit dans les images et les vidéos. Nous nous sommes particulièrement intéressé aux approches *basées sur la régularisation* continue par EDP. Ensuite, nous avons proposé l'extension d'un cadre de régularisation discrète aux séquences d'images. Nous avons mis en place des algorithmes de régularisation sur graphes par  $p$ -Laplacien, avec  $p \in ]0, +\infty[$ , pour le débruitage et la simplification des séquences d'images. Afin d'augmenter la rapidité de la régularisation, nous avons fourni une optimisation de l'algorithme non local que nous avons proposé. En outre, nous avons introduit un algorithme nouveau de  $p$ -régularisation, avec  $p = \infty$ . En plus de son intérêt théorique, cet algorithme est plus rapide que ses concurrents pour  $1 \leq p < \infty$  tout en assurant une qualité similaire du résultat.

Nous souhaitons dans les travaux à venir explorer d'autres pistes d'amélioration des performances et de la qualité des résultats.

Pour améliorer la qualité obtenue, une solution envisageable est l'utilisation d'une approche multi-résolution. Nous espérons ainsi rendre la comparaison entre patches encore plus efficace. Une autre idée que nous souhaitons évaluer consiste à considérer les points SIFT dans la formule des distances entre patches. Par ailleurs, pour le traitement des couleurs, nous traitons actuellement chaque composante RGB à part. Il serait intéressant de corrélérer le traitement des trois composantes pour ne pas obtenir des résultats incorrects visuellement à cause d'une valeur anormale sur l'une des couleurs.

En termes de performance, une piste pour accélérer davantage la régularisation consiste à combiner les traitements de  $p = 2$  et  $p = \infty$ . Nous pensons qu'il serait avantageux de commencer par une régularisation avec  $p = \infty$  et de poursuivre avec  $p = 2$  vers la fin pour plus de précision.

En outre, nous souhaitons utiliser le cadre de régularisation spatio-temporelle proposé pour détecter les événements rares des vidéos. En effet, dans ce cadre, nous nous basons

sur l'analyse de l'auto-similarité. Par conséquent, un événement rare, qui, par définition, aura une faible auto-similarité dans la vidéo, est facilement détectable.

## Inpainting et interpolation

Dans la deuxième partie, nous avons traité le problème de la restauration des données manquantes. Tout d'abord, nous avons rappelé les grandes familles des approches proposées dans la littérature pour résoudre ce problème. Ensuite, nous avons présenté deux approches d'inpainting et d'interpolation.

Premièrement, nous avons proposé l'extension des méthodes d'interpolation continues au domaine discret. Pour réaliser l'inpainting, nous avons développé un algorithme combinant l'interpolation et la régularisation itérative dans le but d'améliorer la qualité de la reconstruction des régions manquantes. Les expérimentations réalisées sur des images et des séquences d'images mettent en évidence la performance de notre approche.

Deuxièmement, nous nous sommes intéressés aux fonctions  $p$ -harmoniques qui constituent un cadre général d'interpolation permettant l'accomplissement de nombreuses applications dont l'inpainting, l'agrandissement et la segmentation semi-supervisée sur graphes. Nous avons montré que ce cadre unifie les approches géométriques et les approches basées sur la synthèse de texture. Ainsi de nombreux travaux de la littérature peuvent être retrouvés en considérant certaines valeurs des paramètres. Ensuite, nous avons introduit le cas nouveau de l'interpolation par fonctions  $\infty$ -harmoniques. Les résultats obtenus à l'aide de cet algorithme fournissent des résultats de qualité visuelle similaire à ce que l'on obtient avec des valeurs plus classiques comme  $p = 2$  et  $p = 1$  et les dépasse en termes de vitesse de calcul et de convergence. Nous avons eu l'occasion d'éprouver nos algorithmes sur diverses applications comme l'inpainting, l'agrandissement et la segmentation semi-supervisée.

De même que pour le débruitage, certaines améliorations sont envisagées pour les travaux futurs. La multi-résolution sur graphes hiérarchiques et la corrélation des couleurs pourraient augmenter la robustesse des algorithmes actuels. De nombreuses applications intéressantes en découlent de manière directe. Notre approche est utilisable pour des applications telles que la colorisation qui consiste à diffuser des couleurs à la place des labels. Le parallèle avec la segmentation semi-supervisée est immédiat. L'incrustation d'objets se base sur le même principe de l'inpainting. En particulier, notre algorithme utilisant l'interpolation et la régularisation devrait être parfaitement adapté à ce genre d'applications. Nous avons appliqué notre algorithme à l'agrandissement des données. Cette application a des retombées dans le domaine de la compression des données. Nous pouvons exploiter notre approche à des fins de compactage des images et des séquences.

## Régularisation spectrale

Enfin, dans la dernière partie, nous avons abordé la régularisation spectrale. Dans un grand nombre d'applications en traitement d'images, nous sommes souvent confrontés à des données massives de grandes dimensions. Afin de faciliter leurs visualisations, leurs organisations et les rendre plus facilement exploitables, il est indispensable de réduire leurs dimensions. Nous avons brièvement rappelé les méthodes classiques d'analyse spectrale qui se sont avérées des outils puissants.

Ensuite, nous avons proposé une approche de régularisation spectrale sur graphes basée sur les cartes de diffusion. Premièrement, nous avons utilisé cette approche pour formuler un algorithme rapide de segmentation par classification et diffusion des points caractéristiques. Nous avons montré également une application de détection des éléments

saillants des images par marches aléatoires sur graphes. Et enfin, nous avons présenté un projet d'apprentissage d'écriture auquel nous contribuons. Dans le cadre de ce projet, nous nous sommes intéressé à la visualisation et la catégorisation de données hétérogènes par régularisation dans l'espace spectral réduit. La finalité de ce travail est la classification des caractères écrits que nous avons à disposition. Les données temporelles recueillies pendant l'écriture ont fait l'objet d'un travail supplémentaire exploitant la dynamique d'écriture. La distance DTW, adaptée aux données temporelles, a été proposée. Les premiers résultats de classification sont encourageants. De plus, les résultats des étapes intermédiaires de choix des distances, de visualisation et de catégorisation sont importants pour le reste des étapes menant à la classification.

La suite des travaux doit permettre d'évaluer précisément les résultats de la classification. Une des difficultés des méthodes de régularisation est le coût de la décomposition spectrale qui peut être très élevé pour une grande matrice. Pour augmenter la vitesse de calcul, notre intérêt porte sur deux approches : l'approche de Nyström et le calcul itératif de la valeur propre et des vecteurs propres dominants par la méthode de la puissance itérée.

Par ailleurs, l'approche de caractérisation des éléments saillants est facilement extensible en 3D et peut être appliquée aux séquences d'images. Grâce à la représentation sur graphes, les résultats doivent être aussi intéressants que sur les images. Nous développerons cet aspect dans les travaux à venir.



---

# BIBLIOGRAPHIE

---

- [Amer et Dubois, 2005] AMER, A. et DUBOIS, E. (2005). Fast and reliable structure-oriented video noise estimation. *CirSysVideo*, 15(1):113–118.
- [Arias et al., 2009] ARIAS, P., CASELLES, V. et SAPIRO, G. (2009). A variational framework for non-local image inpainting. In *EMMCVPR09*, pages 345–358.
- [Aronsson, 1967] ARONSSON, G. (1967). Extension of functions satisfying lipschitz conditions. *Arkiv för Matematik*, 6(6):551–561.
- [Aubert et Kornprobst, 2006] AUBERT, G. et KORNPBST, P. (2006). *Mathematical Problems in Image Processing : Partial Differential Equations and the Calculus of Variations (second edition)*, volume 147 de *Applied Mathematical Sciences*. Springer-Verlag.
- [Balasubramanian et Schwartz, 2002] BALASUBRAMANIAN, M. et SCHWARTZ, E. L. (2002). The isomap algorithm and topological stability. *Science*, 295(5552):7.
- [Ballester et al., 2001] BALLESTER, C., BERTALMIO, M., MEMBER, A., SAPIRO, G. et VERDERA, J. (2001). Filling-in by joint interpolation of vector fields and gray levels. *IEEE Trans. Image Processing*, 10:1200–1211.
- [Barash et Comaniciu, 2004] BARASH, D. et COMANICIU, D. (2004). A common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean shift. *Image and Video Computing*, 22:73–81.
- [Barron et al., 2008] BARRON, E. N., EVANS, L. et JENSEN, R. (2008). The infinity laplacian, aronsson’s equation and their generalizations. *Trans. Amer. Math. Soc.*, 360:77–101.
- [Belkin et Niyogi, 2001] BELKIN, M. et NIYOGI, P. (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, pages 585–591. MIT Press.
- [Belkin et Niyogi, 2003] BELKIN, M. et NIYOGI, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396.
- [Bertalmio et al., 2001] BERTALMIO, M., BERTOZZI, A. et SAPIRO, G. (2001). Navier-stokes, fluid dynamics, and image and video inpainting. In *CVPR01*, pages 355–362.
- [Bertalmio et al., 2000] BERTALMIO, M., SAPIRO, G., CASELLES, V. et BALLESTER, C. (2000). Image inpainting. In *SIGGraph-2000*, pages 417–424.
- [Bertero et al., 1988] BERTERO, M., POGGIO, T. A. et TORRE, V. (1988). Ill-posed problems in early vision. In *Proceedings of the IEEE*, pages 869–889.
- [Bertozzi et al., 2007a] BERTOZZI, A., GLU, S. E. et GILLETTE, A. (2007a). 1 analysis of a two-scale cahn-hilliard model for image inpainting.
- [Bertozzi et al., 2007b] BERTOZZI, A. L., ESEDOGLU, S. et GILLETTE, A. (2007b). Inpainting of binary images using the cahn-hilliard equation. *IEEE Trans Image Process*, 16(1): 285–91.
- [Boulanger et al., 2007] BOULANGER, J., KERVRANN, C. et BOUTHEMY, P. (2007). Space-time adaptation for patch-based image sequence restoration. *IEEE Trans. PAMI*, 29: 1096–1102.

- [Brailean *et al.*, 1995] BRAILEAN, J. C., KLEIHORST, R. P., EFSTRATIADIS, S., KATSAGGELLOS, A. K. et LAGENDIJK, R. L. (1995). Noise reduction filters for dynamic image sequences : a review. *Proceedings of the IEEE*, 83(9):1272–1292.
- [Buades *et al.*, 2005a] BUADES, A., COLL, B. et MOREL, J. (2005a). Denoising image sequences does not require motion estimation. In *Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference on*, pages 70–74.
- [Buades *et al.*, 2005b] BUADES, A., COLL, B. et MOREL, J. M. (2005b). A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2):490–530.
- [Buades *et al.*, 2008] BUADES, A., COLL, B. et MOREL, J.-M. (2008). Nonlocal image and movie denoising. *Int. J. Comput. Vision*, 76(2):123–139.
- [Burges, 2005] BURGES, C. J. C. (2005). Geometric methods for feature extraction and dimensional reduction. In *In L. Rokach and O. Maimon (Eds.), Data*. Kluwer Academic Publishers.
- [Canny, 1986] CANNY, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698.
- [Caselles *et al.*, 1998] CASELLES, V., MOREL, J.-M. et SBERT, C. (1998). An axiomatic approach to image interpolation. *IEEE Transactions on Image Processing*, 7(3):376–386.
- [Chambolle et Lions, 1997] CHAMBOLLE, A. et LIONS, P. L. (1997). Image recovery via total variation minimization and related problems. *Numer. Math.*, 76:167–188.
- [Chan *et al.*, 2002a] CHAN, T., KANG, S. et SHEN, J. (2002a). Euler’s elastica and curvature based inpainting. *Journal of Applied Mathematics*, 63(2):564–592.
- [Chan *et al.*, 2002b] CHAN, T., KANG, S. et SHEN, J. (2002b). Euler’s elastica and curvature based inpaintings. *SIAM J. Appl. Math*, pages 564–592.
- [Chan et Shen, 2001] CHAN, T. et SHEN, J. (2001). Mathematical models for local nontexture inpaintings. *Journal of Applied Mathematics*, 62(3):1019–1043.
- [Chan et Shen, 2005a] CHAN, T. et SHEN, J. (2005a). *Image Processing And Analysis : Variational, PDE, Wavelet, and Stochastic Methods*. SIAM.
- [Chan et Shen, 2005b] CHAN, T. F. et SHEN, J. (2005b). Variational image inpainting. *Comm. Pure Applied Math*, 58:579–619.
- [Chang *et al.*, 2004] CHANG, H., YEUNG, D.-Y. et XIONG, Y. (2004). Super-resolution through neighbor embedding. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I-275 – I-282 Vol.1.
- [Choi et Choi, 2007] CHOI, H. et CHOI, S. (2007). Robust kernel isomap. *Pattern Recognition*, 40:2007.
- [Chu *et al.*, 1998] CHU, C., GLAD, I., GODTLIEBSEN, F. et MARRON, J. (1998). Edge-preserving smoothers for image processing. *Journal of the American Statistical Association*, 93:526–541.
- [Coifman *et al.*, 2005] COIFMAN, R. R., LAFON, S., LEE, A. B., MAGGIONI, M., WARNER, F. et ZUCKER, S. (2005). Geometric diffusions as a tool for harmonic analysis and structure definition of data : Diffusion maps. In *Proceedings of the National Academy of Sciences*, pages 7426–7431.
- [Cong *et al.*, 2004] CONG, G., ESSER, M., PARVIN, B. et BEBIS, G. (2004). Shape metamorphism using p-laplacian equation. In *ICPR*, volume 4, pages 15–18.

- [Crandall, 2008] CRANDALL, M. (2008). A visit with the infinity laplace equation. *Calculus of Variations and Nonlinear Partial Differential Equations*, pages 75–122.
- [Criminisi et al., 2003] CRIMINISI, A., PÉREZ, P. et TOYAMA, K. (2003). Object removal by exemplar-based inpainting. In *IEEE Computer Vision and Pattern Recognition*, pages 721–728.
- [Criminisi et al., 2004] CRIMINISI, A., PÉREZ, P. et TOYAMA, K. (2004). Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13:1200–1212.
- [Dabov et al., 2007] DABOV, K., FOI, A. et EGIAZARIAN, K. (2007). Video denoising by sparse 3d transform-domain collaborative filtering.
- [de Haan, 2003] de HAAN, G. (2003). Video processing for multimedia systems.
- [Didas et Weickert, 2007] DIDAS, S. et WEICKERT, J. (2007). Integrodifferential equations for continuous multiscale wavelet shrinkage. (*submitted to*) *Inverse Problems and Imaging*, 1(1):47–62.
- [Dijkstra, 1959] DIJKSTRA, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271–271.
- [Do et al., 2006] DO, V., LEBRUN, G., MALAPERT, L., SMET, C. et TSCHUMPERLÉ, D. (2006). Inpainting d’images couleurs par lissage anisotrope et synthèse de textures. In *Reconnaissance des Formes et Intelligence Artificielle*.
- [Dobrosotskaya et Bertozzi, 2008] DOBROSOTSKAYA, J. A. et BERTOZZI, A. L. (2008). A wavelet-laplace variational technique for image deconvolution and inpainting. *IEEE Transactions on Image Processing*, 17(5):657–663.
- [Duraiswami et Raykar, 2005] DURAISWAMI, R. et RAYKAR, V. C. (2005). The manifolds of spatial hearing. In *In Proceedings of International Conference on Acoustics, Speech and Signal Processing*, pages 285–288.
- [Efros et Leung, 1999] EFROS, A. et LEUNG, T. (1999). Texture synthesis by non-parametric sampling. In *IEEE International Conference on Computer Vision*, pages 1033–1038, Corfu, Greece.
- [Efros et Freeman, 2001] EFROS, A. A. et FREEMAN, W. T. (2001). Image quilting for texture synthesis and transfer. In *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346, New York, NY, USA. ACM.
- [Elad, 2002] ELAD, M. (2002). On the origin of the bilateral filter and ways to improve it. *Image Processing, IEEE Transactions on*, 11(10):1141–1151.
- [Elion et Vese, 2007] ELION, C. et VESE, L. A. (2007). An image decomposition model using the total variation and the infinity laplacian. In *Proceedings SPIE*, volume 6498, pages 64980W–1–64980W–10.
- [Elmoataz et al., 2011] ELMOATAZ, A., GHONIEM, M., LÉZORAY, O. et TA, V. (2011). Non-local infinity laplacian on graphs with applications to image and data processing. *IEEE transactions on Pattern Analysis and Machine Intelligence*. Submitted.
- [Elmoataz et al., 2008] ELMOATAZ, A., LEZORAY, O. et BOUGLEUX, S. (2008). Nonlocal discrete regularization on weighted graphs : a framework for image and manifold processing. *IEEE Transactions on Image Processing*, 17(7):1047–1060.
- [Elmoataz et al., 2008] ELMOATAZ, A., LEZORAY, O., BOUGLEUX, S. et TA, V. T. (2008). Unifying local and nonlocal processing with partial difference operators on weighted graphs. In *International Workshop on Local and Non-Local Approximation in Image Processing*, pages 11–26, Switzerland.



- [Esedoglu et Shen, 2002] ESEDOGLU, S. et SHEN, J. (2002). Digital inpainting based on the mumford-shah-euler image model. *European J. Appl. Math*, 13:353–370.
- [Floyd, 1962] FLOYD, R. W. (1962). Algorithm 97 : Shortest path. *Commun. ACM*, 5(6):345.
- [Fukunaga, 1990] FUKUNAGA, K. (1990). *Introduction to Statistical Pattern Recognition*. Academic Press, San Diego, California.
- [Geman et Geman, 1984] GEMAN, S. et GEMAN, D. (1984). Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741.
- [Ghoniem et al., 2008a] GHONIEM, M., CHAHIR, Y. et ELMOATAZ, A. (2008a). Video denoising and simplification via discrete regularization on graphs. *In ACIVS*, pages 380–389.
- [Ghoniem et al., 2008b] GHONIEM, M., CHAHIR, Y. et ELMOATAZ, A. (2008b). Video denoising via discrete regularization on graphs. *In ICPR08*, pages 1–4.
- [Ghoniem et al., 2009] GHONIEM, M., CHAHIR, Y. et ELMOATAZ, A. (2009). Geometric and texture inpainting based on discrete regularization on graphs. *In ICIP*, pages 1349–1352.
- [Ghoniem et al., 2010] GHONIEM, M., CHAHIR, Y. et ELMOATAZ, A. (2010). Nonlocal video denoising, simplification and inpainting using discrete regularization on graphs. *Signal Processing*, 90(8):2445–2455. Special Section on Processing and Analysis of High-Dimensional Masses of Image and Signal Data.
- [Gilboa et Osher, 2007] GILBOA, G. et OSHER, S. (2007). Nonlocal linear image regularization and supervised segmentation. *Multiscale Modeling & Simulation*, 6:595–630.
- [Gilboa et Osher, 2009] GILBOA, G. et OSHER, S. (2009). Nonlocal operators with applications to image processing. *Multiscale Modeling & Simulation*, 7(3):1005–1028.
- [Goldberg et al., 2008] GOLDBERG, Y., ZAKAI, A., KUSHNIR, D. et RITOV, Y. (2008). Manifold learning : The price of normalization. *J. Mach. Learn. Res.*, 9:1909–1939.
- [Grossauer et Scherzer, 2003] GROSSAUER, H. et SCHERZER, O. (2003). Using the complex ginzburg-landau equation for digital inpainting in 2d and 3d. *In ScaleSpace03*, pages 225–236.
- [Guillot et Guyader, 2009] GUILLOT, L. et GUYADER, C. L. (2009). Extrapolation of vector fields using the infinity laplacian and with applications to image segmentation. *In SSVM*, pages 87–99.
- [He et al., 2005] HE, X., CAI, D., YAN, S. et ZHANG, H.-J. (2005). Neighborhood preserving embedding. *Computer Vision, IEEE International Conference on*, 2:1208–1213.
- [Holopainen et Soardi, 1997] HOLOPAINEN, I. et SOARDI, P. (1997).  $p$ -harmonic functions on graphs and manifolds. *manuscripta mathematica*, 94:95–110.
- [Hotelling, 1933] HOTELLING, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24.
- [Jenkins et Mataric, 2002] JENKINS, O. C. et MATARIC, M. J. (2002). Deriving action and behavior primitives from human motion. *In In International Conference on Intelligent Robots and Systems*, pages 2551–2556.
- [Jolliffe, 2002] JOLLIFFE, I. T. (2002). *Principal Component Analysis*. Springer, second édition.
- [Juutinen et Kawohl, 2006] JUUTINEN, P. et KAWOHL, B. (2006). On the evolution governed by the infinity laplacian. *Mathematische Annalen*, 335(4):819–851.

- [Kindermann *et al.*, 2005] KINDERMANN, S., OSHER, S. et W., J. P. (2005). Deblurring and denoising of images by nonlocal functionals. *Multiscale Modeling & Simulation*, pages 1091–1115.
- [Kokiofoulou et Saad, 2005] KOKIOPOULOU, E. et SAAD, Y. (2005). Orthogonal neighborhood preserving projections. *In IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 26–30.
- [Lafon *et al.*, 2006] LAFON, S., KELLER, Y. et COIFMAN, R. R. (2006). Data fusion and multicue data matching by diffusion maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1784–1797.
- [Lafon et Lee, 2006] LAFON, S. et LEE, A. B. (2006). Diffusion maps and coarse-graining : A unified framework for dimensionality reduction, graph partitioning and data set parameterization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28: 1393–1403.
- [Lee et Verleysen, 2005] LEE, J. A. et VERLEYSSEN, M. (2005). Nonlinear dimensionality reduction of data manifolds with essential loops. *Neurocomput.*, 67:29–53.
- [Lee et Verleysen, 2007] LEE, J. A. et VERLEYSSEN, M. (2007). *Nonlinear Dimensionality Reduction*. Springer Publishing Company, Incorporated.
- [Lee, 1983] LEE, J.-S. (1983). Digital image smoothing and the sigma filter. *Computer Vision, Graphics, and Image Processing*, 24(2):255–269.
- [Lim *et al.*, 2003] LIM, I. S., CIECHOMSKI, P. D. H., SARNI, S. et THALMANN, D. (2003). Planar arrangement of high-dimensional biomedical data sets by isomap coordinates. *In In Proceedings of the 16 th IEEE Symposium on Computer-Based Medical Systems*, pages 50–55.
- [Lowe, 1999] LOWE, D. G. (1999). Object recognition from local scale-invariant features. *In Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157 vol.2, Los Alamitos, CA, USA. IEEE Computer Society.
- [Masnou, 2002] MASNOU, S. (2002). Disocclusion : a variational approach using level lines. *IP*, pages 68–76.
- [Masnou et Morel, 1998] MASNOU, S. et MOREL, J. (1998). Level lines based disocclusion. *In Proceedings of International Conference on Image Processing, ICIP*, pages 259–263.
- [Mikolajczyk et Schmid, 2005] MIKOLAJCZYK, K. et SCHMID, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630.
- [Mrázek *et al.*, 2004] MRÁZEK, P., WEICKERT, J. et BRUHN, A. (2004). On robust estimation and smoothing with spatial and tonal kernels. *In Proc. Dagstuhl Seminar : Geometric Properties from Incomplete Data*, pages 3–88722. Springer.
- [Mumford, 1994] MUMFORD, D. (1994). Bayesian rationale for energy functionals. *In In Geometry-driven diffusion in Computer Vision*, pages 141–153. Kluwer Academic.
- [Mumford et Shah, 1989] MUMFORD, D. et SHAH, J. (1989). Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, pages 577–685.
- [Nadler *et al.*, 2005] NADLER, B., LAFON, S., COIFMAN, R. R. et KEVREKIDIS, I. G. (2005). Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *ArXiv Mathematics e-prints*.

- [Ng *et al.*, 2001] NG, A. Y., JORDAN, M. I. et WEISS, Y. (2001). On spectral clustering : Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press.
- [Niskanen et Silvén, 2003] NISKANEN, M. et SILVÉN, O. (2003). Comparison of dimensionality reduction methods for wood surface inspection. In *Proceedings of the 6<sup>th</sup> International Conference on Quality Control by Artificial Vision*, pages 178–188.
- [Pearson, 1901] PEARSON, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572.
- [Perona et Malik, 1990] PERONA, P. et MALIK, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:629–639.
- [Pizarro *et al.*, 2010] PIZARRO, L., MRÁZEK, P., DIDAS, S., GREWENIG, S. et WEICKERT, J. (2010). Generalised nonlocal image smoothing. *Int. J. Comput. Vision*, 90(1):62–87.
- [Protter et Elad, 2007] PROTTER, M. et ELAD, M. (2007). Sparse and redundant representations and motion-estimation-free algorithm for video denoising. In *Wavelets XII. Proceedings of the SPIE, Volume 6701*, volume 6701, page 43.
- [Protter et Elad, 2009] PROTTER, M. et ELAD, M. (2009). Image sequence denoising via sparse and redundant representations. *Trans. Img. Proc.*, 18(1):27–35.
- [Protter *et al.*, 2009] PROTTER, M., ELAD, M., TAKEDA, H. et MILANFAR, P. (2009). Generalizing the nonlocal-means to super-resolution reconstruction. *IEEE Transactions on Image Processing*, pages 36–51.
- [Rajpoot *et al.*, 2007] RAJPOOT, N., ARIF, M. et BHALERAO, A. (2007). Unsupervised learning of shape manifolds. In *Proceedings of the British Machine Vision Conference*.
- [Raytchev *et al.*, 2004] RAYTCHEV, B., YODA, I. et SAKAUE, K. (2004). Head pose estimation by nonlinear manifold learning. In *ICPR '04 : Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 4*, pages 462–466, Washington, DC, USA. IEEE Computer Society.
- [Roweis et Saul, 2000] ROWEIS, S. T. et SAUL, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326.
- [Rudin *et al.*, 1992] RUDIN, L. I., OSHER, S. et FATEMI, E. (1992). Nonlinear total variation based noise removal algorithms. *Phys. D*, pages 259–268.
- [Saint-Marc *et al.*, 1991] SAINT-MARC, P., CHEN, J.-S. et MEDIONI, G. (1991). Adaptive smoothing : a general tool for early vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 13(6):514–529.
- [Salvador et Chan, 2007] SALVADOR, S. et CHAN, P. (2007). Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.*, 11:561–580.
- [Saul *et al.*, 2003] SAUL, L. K., ROWEIS, S. T. et SINGER, Y. (2003). Think globally, fit locally : Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155.
- [Saul *et al.*, 2006] SAUL, L. K., WEINBERGER, K. Q., HAM, J. H., SHA, F. et LEE, D. D. (2006). Spectral methods for dimensionality reduction. *Semisupervised Learning. MIT Press : Cambridge, MA*.
- [Saxena *et al.*, 2004] SAXENA, A., GUPTA, A. et MUKERJEE, A. (2004). Non-linear dimensionality reduction by locally linear isomaps. lecture. In *Neural Information Processing 2004*, pages 1038–1043.

- [Scherzer et Weickert, 1998] SCHERZER, O. et WEICKERT, J. (1998). Relations between regularization and diffusion filtering. *Journal of Mathematical Imaging and Vision*, 12:43–63.
- [Shi et Malik, 2000] SHI, J. et MALIK, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.
- [Sinop et Grady, 2007] SINOP, A. et GRADY, L. (2007). A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In *ICCV*, pages 1–8.
- [Smith et Brady, 1997] SMITH, S. M. et BRADY, J. M. (1997). SUSAN - a new approach to low level image processing. *Int. Journal of Computer Vision*, 23(1):45–78.
- [Spearman, 1904] SPEARMAN, C. (1904). "general intelligence," objectively determined and measured. *American Journal of Psychology*, 15:201–293.
- [Steidl et al., 2004] STEIDL, G., WEICKERT, J., BROX, T., MRÁZEK, P. et WELK, M. (2004). On the equivalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and sides. *SIAM J. Numer. Anal.*, 42:686–713.
- [Sun et al., 2006] SUN, J., BOYD, S., XIAO, L. et DIACONIS, P. (2006). The fastest mixing markov process on a graph and a connection to a maximum variance unfolding problem. *SIAM Review*, 48:2006.
- [Ta et al., 2008a] TA, V., ELMOATAZ, A. et LEZORAY, O. (2008a). Partial difference equations over graphs : Morphological processing of arbitrary discrete data. In *European Conference on Computer Vision*, volume 5304 de LNCS, pages 668–680.
- [Ta et al., 2009] TA, V., ELMOATAZ, A. et LEZORAY, O. (2009). Nonlocal graph morphology. In *ISMM Abstract Book*, pages 5–9.
- [Ta et al., 2011] TA, V., ELMOATAZ, A. et LÉZORAY, O. (2011). Nonlocal pdes-based morphology on weighted graphs for image and data processing. *IEEE transactions on Image Processing*.
- [Ta et al., 2008b] TA, V.-T., ELMOATAZ, A. et LÉZORAY, O. (2008b). Partial difference equations over graphs : Morphological processing of arbitrary discrete data. In *ECCV '08 : Proceedings of the 10th European Conference on Computer Vision*, pages 668–680, Berlin, Heidelberg. Springer-Verlag.
- [Tenenbaum et al., 2000] TENENBAUM, J. B., SILVA, V. et LANGFORD, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500): 2319–2323.
- [Tikhonov et Arsenin, 1977] TIKHONOV, A. N. et ARSENIN, V. Y. (1977). *Solutions of Ill-Posed Problems*. V. H. Winston & Sons, Washington, D.C. : John Wiley & Sons, New York,.
- [Tomasi et Manduchi, 1998] TOMASI, C. et MANDUCHI, R. (1998). Bilateral filtering for gray and color images. In *Proceedings of the Sixth International Conference on Computer Vision, ICCV '98*, pages 839–, Washington, DC, USA. IEEE Computer Society.
- [Torgerson, 1952] TORGERSON, W. (1952). Multidimensional scaling : I. theory and method. *Psychometrika*, 17:401–419. 10.1007/BF02288916.
- [Torre et Poggio, 1986] TORRE, V. et POGGIO, T. (1986). On edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:147–163.
- [van den Boomgaard et van de Weijer, 2002] van den BOOMGAARD, R. et van de WEIJER, J. (2002). On the equivalence of local-mode finding, robust estimation and mean-shift analysis as used in early vision tasks. In *Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 3 - Volume 3, ICPR '02*, pages 30927–, Washington, DC, USA. IEEE Computer Society.

- [van der Maaten *et al.*, 2009] van der MAATEN, L. J. P., POSTMA, E. O. et van den HERIK, H. J. (2009). Dimensionality reduction : A comparative review. Rapport technique, Tilburg University.
- [Venna, 2007] VENNA, J. (2007). *Dimensionality reduction for visual exploration of similarity structures*. Thèse de doctorat, Helsinki University of Technology, Espoo, Finland.
- [Weickert, 1997] WEICKERT, J. (1997). A review of nonlinear diffusion filtering. In ter HAAR ROMENY, B., FLORACK, L., KOENDERINK, J. et VIERGEVER, M., éditeurs : *Scale-Space Theory in Computer Vision*, volume 1252 de *Lecture Notes in Computer Science*, pages 1–28. Springer Berlin / Heidelberg.
- [Weickert, 1998] WEICKERT, J. (1998). *Anisotropic Diffusion In Image Processing*. Teubner-Verlag, Stuttgart ; Germany.
- [Weinberger *et al.*, 2005] WEINBERGER, K. Q., PACKER, B. D. et SAUL, L. K. (2005). Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization. In *In Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pages 381–388.
- [Winkler, 2006] WINKLER, G. (2006). *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods : A Mathematical Introduction (Stochastic Modelling and Applied Probability)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- [Winkler *et al.*, 1999] WINKLER, G., AURICH, V., HAHN, K., MARTIN, A. et RODENACKER, K. (1999). Noise reduction in images : Some recent edge-preserving methods.
- [Wong et Orchard, 2008] WONG, A. et ORCHARD, J. (2008). A nonlocal-means approach to exemplar-based inpainting. In *ICIP*, pages 2600–2603.
- [Yaroslavsky, 1985] YAROSLAVSKY, L. (1985). Digital picture processing-an introduction, springer. BERLIN, 85.
- [Yuille et Poggio, 1986] YUILLE, A. et POGGIO, T. (1986). Scaling theorems for zero-crossings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:15–25.
- [Zhou et Schölkopf, 2004] ZHOU, D. et SCHÖLKOPF, B. (2004). A regularization framework for learning from graph data. In *ICML Workshop on Statistical Relational Learning*, pages 132–137.
- [Zhou et Schölkopf, 2005] ZHOU, D. et SCHÖLKOPF, B. (2005). Regularization on discrete spaces. In *Pattern Recognition*, pages 361–368. Springer.
- [Zhu *et al.*, 1998] ZHU, S. C., WU, Y. et MUMFORD, D. (1998). Filters, random fields and maximum entropy (frame) – towards a unified theory for texture modeling. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 27(2):1–20.
- [Zhu *et al.*, 1997] ZHU, S. C., WU, Y. N. et MUMFORD, D. (1997). Minimax entropy principle and its application to texture modeling. *Neural Computation*, 9:1627–1660.
- [Zlokolic *et al.*, 2006] ZLOKOLICA, V., PIZURICA, A. et PHILIPS, W. (2006). Wavelet-domain video denoising based on reliability measures. *CirSysVideo*, 16(8):993–1007.