

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ MOULOUD MAMMÈRI, TIZI-OUZOU



FACULTE DE GENIE ÉLECTRIQUE ET D'INFORMATIQUE
DÉPARTEMENT AUTOMATIQUE

MEMOIRE DE MAGISTER

en Automatique

Option **Traitement d'Images et
Reconnaissance de Formes**

Présenté par

ARKOUB Nouara

Ingénieur UMMTO

Reconstruction de données manquantes dans une image.

Mémoire soutenu le : / /2012

devant le jury d'examen composé de :

DIAF Moussa

Professeur à l'U.M.M.T.O.

Président

HAMMOUCHE Kamal

M.C.A. à l'U.M.M.T.O.

Rapporteur

MAIDI Ahmed

M.C.A. à l'U.M.M.T.O.

Examineur

LAHDIR Mourad

M.C.A. à l'U.M.M.T.O.

Examineur

Sujet de thèse de Magister proposé à

Melle Arkoub Nouara

Thème : Reconstruction de données manquantes dans une image.

Encadrant : Mr Hammouche Kamal, Maître de Conférences à l'UMMTO

Cadre et objectif du travail

La reconstruction de données manquantes dans une image, également connue sous le terme d' " inpainting ", joue un rôle très important en vision par ordinateur et touche plusieurs domaines d'applications comme par exemple la restauration de manière numérique des images dégradées par des artefacts ayant détruit de manière complète certaines parties des images (rayures sur des films anciens ou taches sur des photographies), pour supprimer un logo, un texte ou une image incrustée sur une autre image. De manière générique, ce terme désigne le fait de déterminer le niveau de gris ou la couleur des pixels considérés comme manquants dans une image, c'est à dire, dont on ne connaît pas les valeurs à priori. Le problème d'inpainting peut être formulé mathématiquement sous forme d'une interpolation spatiale des pixels connus de l'image. L'objectif dans ce mémoire est d'étudier puis implémenter quelques méthodes d'inpainting.

Plan du travail

- 1- Effectuer une recherche bibliographique sur les méthodes d'inpainting
- 2- Etablir un état de l'art sur les méthodes d'inpainting
- 3- Etudier puis implémenter quelques méthodes d'inpainting.
- 4- Tests et résultats

Bibliographie

[1] M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester, Image Inpainting, Tech. Report, ECE-University of Minnesota, 1999.

[2] C.A.Z. Barcelos and M.A. Batista. Image restoration using digital inpainting and noise removal. Image and Vision Computing, vol. 25, pp. 61–69, 2007.

Tizi-Ouzou le 16/11/2009
K. Hammouche

Avant propos

Le travail que nous présentons dans ce mémoire a été effectué au sein de laboratoire « Robotique et vision » du département automatique, Faculté de Génie Electrique et Informatique, Université Mouloud Mammeri, Tizi-Ouzou.

J'exprime mes remerciements et ma gratitude à mon directeur de mémoire, Monsieur **HAMMOUCHE KAMAL** maître de conférences classe A à l'Université Mouloud Mammeri, pour l'aide et le soutien et les conseils qu'il m'a apporté tout au long de ce travail.

Nos vifs remerciements vont aussi à Monsieur **DIAF MOUSSA** professeur à l'Université Mouloud Mammeri, pour avoir fait l'honneur de présider le jury de ce mémoire et l'intérêt qu'il a apporté à ce travail.

Nous tenons à remercier Monsieur **MAIDI AHMED**, maître de conférences classe A à l'Université Mouloud Mammeri, pour avoir bien voulu faire partie du jury de ce mémoire.

Monsieur **LAHDIR MOURAD**, maître de conférences classe A à l'Université Mouloud Mammeri à bien voulu participer au jury de ce mémoire, qu'il trouve, ici, l'expression de notre profonde gratitude pour l'intérêt qu'il a apporté à notre travail.

Nous tenons aussi à remercier vivement tout les membres de laboratoire Robotique et vision pour l'aide qu'ils nous ont apportée.

À l'âme de mon père que Dieu le garde dans son paradis.

Sommaire

Introduction générale	1
Chapitre 1 : Les outils mathématiques utilisés dans l'inpainting.	
1.1. Introduction.....	3
1.2. Interpolation.....	3
1.2.1 Interpolation polynomiale.....	4
1.2.1.1 Polynôme de Lagrange	4
1.2.1.2 Polynôme de Newton.....	5
1.2.1.3 Polynôme d'interpolation de Hermite.....	6
1.2.1.4 Polynôme d'interpolation de Neville-Aitken.....	6
1.2.1.5 Interpolation polynomiale par morceaux- splines cubiques.....	7
1.2.1.6 B-splines.....	10
1.2.2 Interpolation linéaire	11
1.2.3 Interpolation par le plus proche voisin	11
1.2.2 Interpolation de données en 2D.....	11
1.2.2.1 Interpolation bilinéaire	12
1.2.4.2 Interpolation par le plus proche voisin.....	13
1.2.4.3 Spline cubique en deux dimensions.....	13
1.3. Généralités sur les équations aux dérivées partielles.....	13
1.3.1. Définitions.....	13
1.3.2 Conditions aux limites, condition initiales.....	15
1.3.3 Classification des EDPs linéaires du second ordre.....	15
1.3.3.1 Les équations elliptiques.....	16
1.3.3.2 Les équations paraboliques	16
1.3.3.3 Les équations hyperboliques.....	17
1.3.4 La formulation variationnelle ou énergétique.....	17
1.3.4.1 La formulation variationnelle.....	17
1.3.4.2 La formulation énergétique.....	18
1.3.4.3 Equation d'Euler Lagrange.....	19
1.3.5 Méthodes de résolution des EDP.....	20
1.3.5.1. Méthodes analytiques.....	20

1.3.5.2 Avantages et inconvénients des méthodes analytiques.....	20
1.3.6 Méthodes numériques.....	21
1.3.6.1 Méthode des différences finies.....	21
1.3.6.2 Méthode des éléments finis.....	24
1.3.6.3 Méthode des volumes finis.....	27
1.3.7 Notions de consistance, convergence et stabilité des méthodes numériques.....	30
1.3.7 Les EDP en traitement d'image.....	30
1.3.7.1 Détection de contours	30
1.3.7.2 Segmentation.....	31
1.3.7.3 Restauration.....	32
1.3.7.1.1 Diffusion isotropique.....	32
1.3.7.1.2 Diffusion anisotropique.....	33
1.3.7.1.3 Restauration par minimisation de la variation totale (TV).....	33
1.4 Conclusion.....	34

Chapitre 2 : Etat de l'art sur les méthodes d'inpainting.

2.1. Introduction.....	35
2.2. Définition d'inpainting.....	35
2.3. Principe d'inpainting.....	35
2.4 Sources de détérioration des données et applications.....	36
2.5 Propriétés de l'inpainting.....	38
2.6 Méthodes d'inpainting.....	39
2.7 Méthodes basées sur l'interpolation.....	39
2.8 Méthodes basées sur les EDPs.....	40
2.8.1 Approche de Bertalmio, Sapiro, Casselles et Ballester.....	41
2.8.2 Approche basée sur la variation totale.....	42
2.8.3 Approche de Chan et Shen	43
2.8.4 Approche de Telea.....	44
2.9 Méthodes basées sur la recherche de similarités.....	46
2.10 Méthodes hybrides	49
2.11 Conclusion.....	50

Chapitre 3 : Méthode d'inpainting basée sur l'interpolation.

3.1 Introduction.....	51
3.2 Principe de l'inpainting par interpolation.....	51
3.3 Inpainting par des méthodes d'interpolation monodimensionnelle.....	52
3.3.1 Courbes de remplissage d'espace	52
3.4 Testes et résultats.....	56
3.4.1 Influence de la taille du domaine D à reconstruire.....	62
3.4.2 Influence de la taille de la fenêtre de voisinage.....	67
3.4.3 Application sur des images couleur.....	72
3.5 Conclusion.....	76

Chapitre 4 : Méthode d'inpainting à la base de la diffusion isotropique.

4.1. Introduction.....	77
4.2 Principe de l'inpainting par diffusion	77
4.3 Inpainting par diffusion isotropique.....	77
4.4 Inpainting par diffusion anisotropique.....	78
4.5 Tests et résultats	80
4.3.1 Images tests.....	80
4.3.2 Influence de la taille du domaine D à reconstruire.....	82
4.3.3 Influence du coefficient dt	82
4.3.4 Influence du nombre d'itérations maximal (T_{max}).....	84
4.3.5 Influence du paramètre k dans le cas de la diffusion anisotropique.....	85
3.4.6 Application sur des images couleur.....	86
4.4. Conclusion.....	87

Conclusion générale.....	88
---------------------------------	-----------

Références

Annexes

Introduction générale

Aujourd'hui le concept d'inpainting attire de plus en plus l'attention des chercheurs notamment dans le domaine du traitement d'image. De manière générique, ce terme désigne le fait de déterminer, de la manière la plus automatique possible, le niveau de gris ou la couleur de pixels considérés comme manquants dans une image, c'est-à-dire, dont on ne connaît pas les valeurs à priori.

La reconstruction de données manquantes est intéressante à plus d'un titre, de par les nombreuses applications concrètes qu'il peut traiter. Initialement, l'inpainting a été utilisé par des dessinateurs pour restaurer des anciens tableaux ou des gravures anciens dégradés par l'usure du temps ou par une mauvaise manipulation. Par la suite, l'inpainting a été utilisé pour restaurer de manière numérique des images dégradées par des artefacts ayant détruit de manière complète certaines parties des images. Ces artefacts peuvent être causés par des rayures sur des films anciens ou taches sur des photographies, lors de l'acquisition ou la transmission d'une image. Dans une autre application, on cherche à cacher certains effets spéciaux cinématographiques, des éléments indésirables sur une photo. L'inpainting peut être également appliqué pour zoomer (agrandir) une image ou pour reconstruire, un visage, une empreinte digitale dont certaines parties sont altérées ou cachées.

Ce sujet a réellement fait l'objet d'une attention particulière. Plusieurs méthodes ont été proposées pour l'inpainting d'images. Certaines se basent sur des méthodes d'interpolation des fonctions (Newton, Lagrange, les splines,...), d'autres mettent l'accent sur la reconstruction de la géométrie globale de l'image en utilisant des équations aux dérivées partielles. Comme, il existe des méthodes qui combinent ces dernières et la synthèse de texture.

Notre but dans ce mémoire est d'appliquer quelques méthodes d'interpolation ainsi que le principe de la diffusion pour résoudre le problème d'inpainting d'images.

Le mémoire est organisé en quatre chapitres. Le premier chapitre sera consacré aux outils mathématiques utilisés dans le domaine du traitement d'image et plus particulièrement dans celui d'inpainting. Il s'agit précisément de l'interpolation des fonctions ainsi que les équations différentielles partielles (EDP). Dans le deuxième chapitre, nous allons présenter la problématique d'inpainting, ses applications, ainsi que les différentes méthodes existantes dans ce domaine. Le troisième chapitre est consacré à l'application des méthodes d'interpolation monodimensionnelles et bidimensionnelles pour reconstruire des images

détériorées. Quant au quatrième chapitre, il sera consacré à l'application d'une méthode basée sur EDP formulée à partir du processus de diffusion isotope ou anisotropique. Une conclusion générale synthétise le travail et trace quelques perspectives.

1.1 Introduction

Les méthodes de traitement d'images sont en général indissociables des techniques mathématiques. Dans ce chapitre, nous allons présenter brièvement quelques outils mathématiques utilisés dans le domaine du traitement d'image et plus particulièrement dans celui d'inpainting. Il s'agit précisément de l'interpolation des fonctions ainsi que les équations différentielles partielles (EDP).

1.2 Interpolation

Le problème d'interpolation consiste à chercher une fonction $F(x)$ à partir d'un ensemble de couple de points $(x_i, y_i) \quad i = 0, 1, \dots, n$, appelés points d'appuis ou points d'interpolation, tels que $x_i \in [a, b]$ et $y_i = f(x_i)$. Géométriquement cela signifie qu'il faut trouver une courbe d'équation $y = F(x)$ qui passe par tous les points d'interpolation et qui vérifie la condition $F(x_i) = f(x_i) = y_i, \forall i$.

Le problème peut être complexe car la fonction $F(x)$ peut prendre plusieurs formes. Comme il peut avoir une solution, plusieurs solutions ou aucune solution. Cependant, on peut simplifier le problème si on utilise une forme générale pour $F(x)$, la plus usuelle s'écrit de la manière suivante :

$$F(x) = a_0 g_0(x) + a_1 g_1(x) + \dots + a_m g_m(x) = \sum_{i=1}^m a_i g_i(x). \tag{1.1}$$

où a_0, a_1, \dots, a_m sont des coefficients à déterminer et $g_i(x)$ une fonction élémentaire simple et connue à priori telle que :

$$g_i(x) = x^i, \quad g_i(x) = e^{\alpha_i x} \quad \text{ou} \quad g_i(x) = \sin ix, \quad \text{ect...}$$

Dans le premier cas la fonction $F(x)$ correspond à un polynôme. On s'intéressera par suite à l'interpolation polynomiale.

Une autre manière de définir la fonction d'interpolation $F(x)$ est d'utiliser les fonctions radiales de base $\Phi(x)$ telle que [2] :

$$F(x) = \sum_{i=0}^m a_i \Phi(|x - x_i|) \tag{1.2}$$

$f(x) = f(|x|)$ est une fonction radialement symétrique qui peut prendre plusieurs formes comme :

$$\text{Gaussien : } f(r) = e^{-x^2/a^2},$$

$$\text{Quadratique inverse: } f(r) = \frac{1}{1 + a^2 x^2},$$

$$\text{Multiquadratique : } f(r) = \sqrt{1 + a^2 x^2},$$

$$\text{Thin- plate spline : } f(r) = x^2 \ln x.$$

Pour déterminer les coefficients a_i , la solution la plus évidente consiste à résoudre le système d'équation linéaire de dimension $(n \times m)$ qui sont :

$$F(x_j) = y_j \quad \forall i = 0, 1, \mathbf{K}, m \quad (1.3)$$

avec

$$F(x_j) = \sum_{i=0}^m a_i g(x_j) \quad (1.4)$$

$$\text{Soit : } F(x_j) = \sum_{i=0}^m a_i \Phi(|x_j - x_i|). \quad (1.5)$$

1.2.1 Interpolation polynomiale

Etant données $n+1$ point distinct $x_0, x_1, \mathbf{L}, x_n$ et $n+1$ valeurs correspondantes $y_0, y_1, \mathbf{L}, y_n$. Le problème consiste à trouver un polynôme $P_m(x)$ d'ordre m , appelé polynôme d'interpolation, ou polynôme interpolant :

$$P_m(x) = a_0 + a_1 x + a_2 x^2 + \mathbf{L} + a_m x^m. \quad (1.6)$$

$$\text{tel que } P_m(x_i) = y_i \quad i = 0, \mathbf{K}, n. \quad (1.7)$$

Ce problème revient à déterminer les coefficients $a_0, a_1, \mathbf{K}, a_m$. Il admet une solution unique si $n=m$. Le problème de l'interpolation est concerné par ce cas de figure. Dans le cas où $n \neq m$, le problème est sur ou sous déterminé et sa résolution peut être résolu par la méthode des moindres carrés [1]. Il s'agit de l'approximation des fonctions qui ne sera pas abordée dans ce chapitre.

1.2.1.1 Polynôme de Lagrange

Le polynôme d'interpolation $P_n(x)$ est calculé sous la forme d'une combinaison linéaire des polynômes caractéristiques de Lagrange de degré i , définis par :

$$l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{(x - x_j)}{(x_i - x_j)} \quad i = 0, \mathbf{K}, n. \quad (1.8)$$

tels que
$$\begin{cases} l_i(x_j) = 1 & \text{si } i = j \\ l_i(x_j) = 0 & \text{si } i \neq j \end{cases} \quad (1.9)$$

Le polynôme d'interpolation de Lagrange est donné par la formule suivante :

$$P_n(x) = \sum_{i=0}^n y_i l_i(x) = \sum_{i=0}^n f_i(x_i) l_i(x) = \sum_{i=0}^n y_i l_i(x). \quad (1.10)$$

Notons que le polynôme de Lagrange introduit le phénomène de Runge qui est décrit comme étant l'apparition des oscillations importantes aux bords de l'intervalle lorsque le degré de polynôme augmente, autrement dit le polynôme ne converge pas toujours vers la fonction interpolée en tous points (la divergence s'observe aux bords de l'intervalle). Afin d'éviter ce problème, on peut choisir des points d'interpolation correspondants aux racines des polynômes orthogonaux de Tchebychev [2].

1.2.1.2 Polynôme de Newton

Le polynôme d'interpolation de Newton propose une forme alternative plus commode d'un point de vue pratique que le polynôme de Lagrange, il est décrit par la formule suivante :

$$P(x) = y_0 + [x_0 \ x_1](x - x_0) + [x_0 \ x_1 \ x_2](x - x_0)(x - x_1) + [x_0 \ x_1 \ x_2 \ x_3](x - x_0)(x - x_1)(x - x_2) + \dots + [x_0 \ x_1 \ \mathbf{L} \ x_n](x - x_0)(x - x_1)\mathbf{L}(x - x_{n-1}). \quad (1.11)$$

où les expressions entre crochets correspondent aux différences divisées telles que :

- $[x_i \ x_{i+1}] = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = \frac{p_{i+1,0} - p_{i,0}}{x_{i+1} - x_i} = p_{i1}$. (Ordre un)
- $[x_i \ x_{i+1} \ x_{i+2}] = \frac{[x_{i+1} \ x_{i+2}] - [x_i \ x_{i+1}]}{x_{i+2} - x_i} = \frac{p_{i+1,1} - p_{i,1}}{x_{i+2} - x_i} = p_{i2}$. (Ordre deux)
- $[x_i \ x_{i+1} \ x_{i+n}] = \frac{[x_{i+1} \ x_{i+n}] - [x_i \ \mathbf{L} \ x_{i+n-1}]}{x_{i+n} - x_i} = \frac{p_{i+1,n-1} - p_{i,n-1}}{x_{i+n} - x_i} = p_{in}$. (Ordre n)

D'une manière générale, la différence divisée d'ordre k au point x_i peut s'écrire:

$$p_{ik} = \frac{p_{i+1,k-1} - p_{i,k-1}}{x_{i+k} - x_i} \quad \text{avec} \quad \begin{cases} k = 1, 2, \mathbf{L} \ n \\ i = 0, 1, \mathbf{L} \ , n - k \\ p_{i0} = y_i \quad \forall i = 0 : n \end{cases} \quad (1.12)$$

Le polynôme de Newton prend alors la forme suivante :

$$P(x) = y_0 + \sum_{k=1}^n p_{0k} \prod_{j=0}^{k-1} (x - x_j). \quad (1.13)$$

1.2.1.3 Polynôme d'interpolation de Hermite

Plutôt que de faire coïncider la fonction f et le polynôme P_n aux points x_i , on peut chercher à faire aussi coïncider les dérivées de f et de P_n en ces points.

Soient $(x_i, f^{(k)}(x_i))$, avec $i = 1, \mathbf{L}, n$ et $k = 1, \mathbf{L}, m_i$. $f^{(k)}(x_i) = y_i^{(k)}$ correspond à la k -ième dérivée de f à au point x_i .

Posons $N = 1 + m_0 + m_1 + \mathbf{L} + m_i = \sum_{i=1}^n (1 + m_i)$. Si f admet des dérivées d'ordre a_i aux points x_i , il existe un unique polynôme $H_{N-1}(x)$ appelé polynôme d'interpolation d'Hermite, tel que :

$$H_{N-1}(x) = \sum_{i=0}^n \sum_{k=0}^{m_i} y_i^{(k)} L_{ik}(x) \quad (1.14)$$

Ce polynôme vérifie la condition suivante

$$H_{N-1}^{(k)}(x_i) = f^{(k)}(x_i) = y_i^{(k)}. \quad i = 0, \mathbf{L}, n. \quad k = 0, \mathbf{L}, m_i. \quad (1.15)$$

$H_{N-1}(x_i)$ est le polynôme d'interpolation de Hermite de f aux points x_i aux ordres m_i .

Les fonctions L_{ik} sont appelées polynômes caractéristiques d'Hermite et sont définies par les relations

$$\frac{d^p}{dx^p}(L_{ik})(x_j) = \begin{cases} 1 & \text{si } i = j \text{ et } k = p \\ 0 & \text{sin on.} \end{cases} \quad (1.16)$$

En définissant les polynômes

$$l_{ij}(x) = \frac{(x - x_i)^j}{j!} \prod_{\substack{k=0 \\ k \neq i}}^n \left(\frac{x - x_k}{x_i - x_k} \right)^{m_k+1}; \quad i = 0, \mathbf{L}, n \text{ et } j = 0, \mathbf{L}, m_i. \quad (1.17)$$

et en posant $L_{im_i}(x) = l_{im_i}(x)$ pour $i = 0, \mathbf{L}, n$, on obtient les relations de récurrence suivante pour les polynômes L_{ij} :

$$L_{ij}(x) = l_{ij}(x) - \sum_{k=j+1}^{m_i} l_{ij}^{(k)}(x_i) L_{ik}(x), \quad j = m_i - 1, m_i - 2, \mathbf{L}, 0. \quad (1.18)$$

1.2.1.4 Polynôme d'interpolation de Neville-Aitken

Dans la méthode de Neville, le polynôme d'interpolation est construit à l'aide d'une formule de récurrence. Le polynôme de Neville de degré n est calculé à partir de polynôme de degré inférieur suivant le schéma de récurrence illustré sur le tableau (1.1) suivant :

	$P_{i,i}$	$P_{i,i+1}$	$P_{i,i+2}$	$P_{i,i+3}$	$P_{i,i+4}$	
$x_0 :$	$y_0 = P_{0,0}$					
$x_1 :$	$y_1 = P_{1,1}$	$P_{0,1}$				
$x_2 :$	$y_2 = P_{2,2}$	$P_{1,2}$	$P_{0,2}$			
$x_3 :$	$y_3 = P_{3,3}$	$P_{2,3}$	$P_{1,3}$	$P_{0,3}$		
$x_4 :$	$y_4 = P_{4,4}$	$P_{3,4}$	$P_{2,4}$	$P_{1,4}$	$P_{0,4}$	(1.19)

Tableau 1.1 : Illustration du schéma récurrent dans le polynôme d'interpolation de Neville

Dans ce tableau, $p_{i,i}(x)$ représente la valeur de l'unique polynôme de degré 0 passant par (x_i, y_i) donc $p_{i,i}(x) = y_i$. $p_{i,i+1}(x)$ est la valeur de l'unique polynôme de degré 1 passant par les points (x_i, y_i) et (x_{i+1}, y_{i+1}) . De même, $p_{i,i+n}(x)$ est la valeur de l'unique polynôme de degré n passant par les $n + 1$ points.

D'une manière générale, un polynôme $p_{i,j}(x)$ peut être déterminé à partir de l'équation itérative suivante :

$$p_{i,j}(x) = \frac{(x - x_i)p_{i+1,j}(x) - (x - x_j)p_{i,j-1}(x)}{(x_j - x_i)} \tag{1.20}$$

avec $i = 0, \mathbf{K}, n - 1, j = 1, \mathbf{K}, n$ et $j > i$.

Finalement, le polynôme d'interpolation de Neville est donné par :

$$p_n(x) = p_{0,n}(x). \tag{1.21}$$

Remarque : Il existe d'autres polynômes (Polynômes de Stirling, Gauss, ect...) qui permettent d'interpoler des points d'appui équidistants.

1.2.1.5 Interpolation polynomiale par morceaux- splines cubiques

En introduisant les fonctions splines dans les années 40, Schoenberg apporta plus de souplesse dans l'approximation polynomiale [3]. Il permet de diminuer le degré du polynôme approchant une fonction à interpoler en considérant des fonctions polynomiales par morceaux appelées splines. Nous envisageons ici le cas des splines cubiques.

Soit (x_0, \mathbf{K}, x_n) , $n+1$ nœuds distincts de l'intervalle $[a, b]$ avec $a = x_0 < x_1 < \mathbf{K} < x_n = b$.

On note $f_i = f(x_i) = y_i$. Sur chaque intervalle élémentaire $[x_i, x_{i+1}]$, on cherche un polynôme

p_i vérifiant les conditions d'interpolation $p_i(x_i) = f_i = y_i$

La forme générale de l'interpolation par des splines cubiques s'écrit :

$$S(x) = \begin{cases} p_0(x) & \text{si } x_0 \leq x < x_1 \\ p_1(x) & \text{si } x_1 \leq x < x_2 \\ \mathbf{M} & \mathbf{M} \\ p_{n-1}(x) & \text{si } x_{n-1} \leq x < x_n \end{cases} \quad (1.22)$$

Où $p_i(x)$ est un polynôme de degré inférieur ou égale à 3 tel que:

$$p_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i. \quad i = 0, \mathbf{K}, n-1. \quad (1.23)$$

Les dérivées premières et secondes de ces n équations s'écrivent de la manière suivant, pour $i = 0, \mathbf{K}, n-1$:

$$p_i'(x) = 3a_i(x - x_i)^2 + 2b_i(x - x_i) + c_i. \quad (1.24)$$

$$p_i''(x) = 6a_i(x - x_i) + 2b_i \quad (1.25)$$

La fonction interpolatrice $S(x)$ est constituée de morceaux de polynôme de degré 3 qui se raccordent, ainsi que leurs dérivées premières et secondes, aux points connus. $S(x)$ respecte alors les propriétés suivantes :

- $S(x)$ doit passer par tous les points connus : $S(x_i) = y_i$ (1.26)

- $S(x)$ doit être continue sur l'intervalle $[x_0, x_n]$: $p_i(x_i) = p_{i-1}(x_i)$ (1.27)

- $S'(x)$ doit être continue sur l'intervalle $[x_0, x_n]$: $p_i'(x_i) = p_{i-1}'(x_i)$ (1.28)

- $S''(x)$ doit être continue sur l'intervalle $[x_0, x_n]$: $p_i''(x_i) = p_{i-1}''(x_i)$ (1.29)

Le problème consiste alors à déterminer les coefficients (a_i, b_i, c_i, d_i) du polynôme $p_i(x)$.

$x_i = [x_i, x_{i+1}]$ donc $S(x_i) = p_i(x_i)$.

D'après les équations (1.23) et (1.26), on a pour $i = 0, \mathbf{K}, n-1$:

$$y_i = d_i \quad (1.30)$$

Sachant que $y_i = d_i = p_i(x_i)$ et en combinant les équations (1.23) et (1.27), on obtient pour

$i = 0, \mathbf{K}, n-1$:

$$d_i = a_{i-1}h^3 + b_{i-1}h^2 + c_{i-1}h + d_{i-1} \quad \text{avec } h = (x_i - x_{i-1}) \quad (1.31)$$

D'après les équations (1.24) où $p'_i(x_i) = c_i$ et (1.28), on a :

$$c_i = 3a_{i-1}h^2 + 2b_{i-1}h + c_{i-1} \quad (1.32)$$

De même, d'après les équations (1.25) où $p''_i(x_i) = 2b_i$ et (1.29), on a :

$$2b_{i+1} = 6a_i h + 2b_i \quad (1.33)$$

Les expressions précédentes peuvent être simplifiées en posant $M_i = p''_i(x_i)$:

$$d_i = y_i \text{ et } b_i = \frac{M_i}{2}$$

Les coefficients a_i et c_i sont calculés à partir des équations (1.33) et (1.31) respectivement, ce qui donne après simplification :

$$\begin{aligned} a_i &= \frac{M_{i+1} - M_i}{6h} \\ b_i &= \frac{M_i}{2} \\ c_i &= \frac{y_{i+1} - y_i}{h} - \left(\frac{M_{i+1} + 2M_i}{6} \right) h \\ d_i &= y_i \end{aligned} \quad (1.34)$$

En remplaçant ces coefficients dans l'équation (1.32), on déduit une équation plus générale

$$M_i + 4M_{i+1} + M_{i+2} = 6 \left(\frac{y_i - 2y_{i+1} + y_{i+2}}{h^2} \right) \quad (1.35)$$

Cette équation peut s'écrire sous une forme matricielle comme suit :

$$\begin{bmatrix} 1 & 4 & 1 & 0 & \cdot & \cdot & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & \cdot & \cdot & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & \cdot & \cdot & 0 & 0 & 0 & 0 \\ \cdot & \cdot \\ \cdot & \cdot \\ 0 & 0 & 0 & 0 & \cdot & \cdot & 4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdot & \cdot & 1 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdot & \cdot & 0 & 1 & 4 & 1 \end{bmatrix} \begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ M_3 \\ \cdot \\ \cdot \\ M_{n-4} \\ M_{n-3} \\ M_{n-2} \\ M_{n-1} \end{bmatrix} = \frac{6}{h^2} \begin{bmatrix} y_0 - 2y_1 + y_2 \\ y_1 - 2y_2 + y_3 \\ y_2 - 2y_3 + y_4 \\ \cdot \\ \cdot \\ y_{n-5} - 2y_{n-4} + y_{n-3} \\ y_{n-4} - 2y_{n-3} + y_{n-2} \\ y_{n-3} - 2y_{n-2} + y_{n-1} \end{bmatrix} \quad (1.36)$$

Ce système comporte (n-1) lignes et (n+1) colonnes, il est par conséquent sous dimensionné.

Pour générer une unique solution, l'ajout de deux conditions aux limites est nécessaire.

On suppose que les dérivées secondes sont nulles aux extrémités en x_0 et x_n ($M_o = M_n = 0$). Une matrice de taille $(n-1) \times (n-1)$ est ainsi obtenue.

$$\begin{bmatrix} 4 & 1 & 0 & \cdot & \cdot & 0 & 0 & 0 \\ 1 & 4 & 1 & \cdot & \cdot & 0 & 0 & 0 \\ 0 & 1 & 4 & \cdot & \cdot & 0 & 0 & 0 \\ \cdot & \cdot \\ \cdot & \cdot \\ 0 & 0 & 0 & \cdot & \cdot & 4 & 1 & 0 \\ 0 & 0 & 0 & \cdot & \cdot & 1 & 4 & 1 \\ 0 & 0 & 0 & \cdot & \cdot & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ \cdot \\ \cdot \\ M_{n-4} \\ M_{n-3} \\ M_{n-2} \end{bmatrix} = \frac{6}{h^2} \begin{bmatrix} y_0 - 2y_1 + y_2 \\ y_1 - 2y_2 + y_3 \\ y_2 - 2y_3 + y_4 \\ \cdot \\ \cdot \\ y_{n-5} - 2y_{n-4} + y_{n-3} \\ y_{n-4} - 2y_{n-3} + y_{n-2} \\ y_{n-3} - 2y_{n-2} + y_{n-1} \end{bmatrix} \quad (1.37)$$

La résolution de ce système permet de déterminer les valeurs des M_i pour en déduire ensuite les valeurs des coefficients (a_i, b_i, c_i, d_i) recherchés.

1.2.1.6 B-splines

Soit (x_0, \mathbf{K}, x_n) , $n+1$ nœuds distincts de l'intervalle $[a, b]$ avec $a = x_0 < x_1 < \mathbf{K} < x_n = b$. On note $f_i = f(x_i) = y_i$. Sur chaque intervalle élémentaire $[x_i, x_{i+1}]$, on cherche un polynôme S_i vérifiant les conditions d'interpolation $S(x_i) = f_i = y_i$. On définit la B-splines de degré k relatif aux nœuds distincts $x_i, \mathbf{K}, x_{i+k+1}$, les courbes définies pour $x \in \mathfrak{R}$ et $0 \leq i \leq m - k - 1$ par

$$\begin{cases} B_{i,0}(x) = 1 & \text{si } x \in [x_i, x_{i+1}] \\ 0 & \text{sin on} \end{cases} \quad (1.38)$$

Si $x_i < x_{i+k}$ et $x_{i+1} < x_{i+k+1}$, on a :

$$B_{i,k}(x) = \frac{x - x_i}{x_{i+k} - x_i} B_{i,k-1}(x) + \frac{x_{i+k+1} - x}{x_{i+k+1} - x_{i+1}} B_{i+1,k-i}(x). \quad (1.39)$$

Soit $(p_0, p_1, \mathbf{K}, p_{n-1})$ n point de \mathfrak{R} . On appelle fonction spline (ou courbe spline) de degré k associe aux points $(p_0, p_1, \mathbf{K}, p_{n-1})$ la courbe définie par l'expression suivante :

$$S_k(x) = \sum_{i=0}^{n-1} p_i B_{i,k}(x) \quad (1.40)$$

Les réels p_i sont les coefficients de la fonction spline. Ils sont calculés en fonction des coefficients $S(x_i) = S_i = y_i$ aux points $x = x_i$. On doit donc résoudre un système d'équations

linéaire $S_i = y_i = \sum_{i=0}^{n-1} p_i B_{i,k}(x)$.

1.2.2 Interpolation linéaire

L'interpolation linéaire est une méthode d'interpolation la plus simple pour les fonctions à seul variable, elle découle de l'interpolation polynomiale d'ordre 1 de Newton telle que :

$$P(x) = y_k + (x - x_k) \left(\frac{y_{k+1} - y_k}{x_{k+1} - x_k} \right) \quad (1.30)$$

Où k est l'indice du point d'appui ayant une valeur x_k la plus proche de x .

Ce polynôme peut mettre sous la forme d'une fonction linéaire suivante :

$$F(x) = ax + b. \quad (1.31)$$

Où a , et b sont à déterminer à partir des coordonnées de deux points (x_{k+1}, y_{k+1}) et (x_k, y_k)

tels que:

$$a = \frac{y_{k+1} - y_k}{x_{k+1} - x_k} \text{ et } b = \frac{x_{k+1}y_k - x_k y_{k+1}}{x_{k+1} - x_k} \quad (1.32)$$

1.2.3 Interpolation par le plus proche voisin

C'est une méthode simple qui consiste à affecter au point x la valeur de la fonction de son plus proche voisin parmi les points d'appui. Elle consiste à chercher les deux points d'appui x_k et x_{k+1} qui englobe la variable x puis d'appliquer la règle suivante :

$$\text{Si } x > \frac{x_k + x_{k+1}}{2} \text{ alors } F(x) = y_{k+1} \text{ sinon } F(x) = y_k \quad (1.33)$$

1.2.4 Interpolation de données en 2D

Les méthodes d'interpolation de fonctions monovariées peuvent être étendues au cas bidimensionnel. Nous désignons par Ω une région bornée de R^2 et par (x, y) les coordonnées d'un point de Ω . Dans une situation particulièrement simple, où le domaine d'interpolation est le produit tensoriel de deux intervalles monodimensionnels $\Omega = [a, b] \times [c, d]$, on peut introduire $(n+1)$ points x_i et $(m+1)$ points y_i dans chaque intervalle tel que :

$$a \leq x_0 < x_1 < \mathbf{K} < x_n \leq b \quad \text{et} \quad c \leq y_0 < y_1 < \mathbf{K} < y_m \leq d.$$

Ces points forment un réseau cartésien de nœuds. On peut construire facilement un polynôme d'interpolation de Lagrange sur un tel réseau.

On considère alors deux polynômes caractéristiques de Lagrange unidimensionnels en x et y .

$$l_i(x) = \prod_{\substack{k=0 \\ k \neq i}}^n \frac{(x - x_k)}{(x_i - x_k)} \quad \text{et} \quad l_j(y) = \prod_{\substack{m=0 \\ m \neq j}}^n \frac{(y - y_m)}{(y_j - y_m)} \quad (1.34)$$

Où $l_i, i = 0, \mathbf{K}, n$ et $l_j, j = 0, \mathbf{K}, m$

Soit $f(x, y)$ une fonction continue sur Ω , on définit le polynôme d'interpolation de Lagrange de $f(x, y)$ sur Ω , de degré n en x et de degré m en y par $P_{n,m}(x, y)$ tel que :

$$P_{n,m}(x, y) = \sum_{i=0}^n \sum_{j=0}^m f(x_i, y_j) l_i(x) l_j(y). \quad (1.35)$$

$f(x_i, y_j)$ étant la valeur de la fonction au point (x_i, y_j) du réseau.

1.2.4.1 Interpolation bilinéaire

L'interpolation bilinéaire est une méthode d'interpolation pour les fonctions à deux variables, sur une grille régulière, très utilisée en imagerie numérique.

L'interpolation bilinéaire pour le point (x, y) tel que $x_{i-1} \leq x \leq x_i$ et $y_{j-1} \leq y \leq y_j$ peut se mettre sous la forme suivante :

$$F(x, y) = ax + by + cxy + d. \quad (1.36)$$

où $F(x, y)$ est la valeur interpolée au point de coordonnées (x, y) , et $a, b, c, et d$ sont des constantes à déterminer à partir des 4 points voisins $(x_{i-1}, y_{j-1}), (x_i, y_{j-1}), (x_{i-1}, y_j), (x_i, y_j)$, tel que :

$$F(x, y) = \frac{y_j - y}{y_j - y_{j-1}} f(x, y_{j-1}) + \frac{y - y_{j-1}}{y_j - y_{j-1}} f(x, y_j)$$

$$\text{avec } f(x, y_{j-1}) = \frac{x_i - x}{x_i - x_{i-1}} f(x_{i-1}, y_{j-1}) + \frac{x - x_{i-1}}{x_i - x_{i-1}} f(x_i, y_{j-1}).$$

$$\text{et } f(x, y_j) = \frac{x_i - x}{x_i - x_{i-1}} f(x_{i-1}, y_j) + \frac{x - x_{i-1}}{x_i - x_{i-1}} f(x_i, y_j).$$

Soit alors :

$$F(x, y) = \frac{1}{(x_i - x_{i-1})(y_j - y_{j-1})} [(x_i - x)(y_j - y)f(x_{i-1}, y_{j-1}) + (x - x_{i-1})(y_j - y)f(x_i, y_{j-1}) + (x_i - x)(y - y_{j-1})f(x_{i-1}, y_j) + (x - x_{i-1})(y - y_{j-1})f(x_i, y_j)]. \quad (1.37)$$

1.2.4.2 Interpolation par le plus proche voisin

L'interpolation par le plus proche voisin au point (x, y) tel que $x_{i-1} \leq x \leq x_i$ et $y_{j-1} \leq y \leq y_j$ consiste à déterminer la fonction $z = F(x, y)$ telle que $F(x, y) = f(x_k, y_k)$ où (x_k, y_k) est le point d'appui le plus proche du point (x, y) .

1.2.4.3 Spline cubique en deux dimensions

Soit $z = f(x, y)$ une fonction bidimensionnelle à interpoler. On se donne une grille rectangulaire (ih, jl) de points de pas $h > 0$ en x et $l > 0$ en y avec $(i, j) \in \mathbf{Z}$. On se donne aussi les valeurs z_{ij} aux points de la grille (ih, jl) , telles que $z_{ij} = f(ih, jl)$. Le problème est comment interpoler la fonction f entre les points de la grille. La solution est d'effectuer un produit cartésien des fonctions spline $B_{i,k}(x)$ vue dans la partie (1.2.1.6) et poser pour $i = 0, \mathbf{K}, n$ et $j = 0, \mathbf{K}, m$:

$$F(x, y) = \sum_{i,j} p_{i,j} B_{i,n}(x) B_{j,m}(y) \quad (1.38)$$

Pour calculer les coefficients $p_{i,j}$ de la fonction $f(x, y)$, qui permet d'interpoler entre les valeurs nodales z_{ij} , on écrit l'équation (1.38) au point (x_i, y_j) , en tenant compte de $z_{ij} = f(ih, jl)$:

$$z_{ij} = F(x, y) = \sum_{i,j} p_{i,j} B_{i,n}(x) B_{j,m}(y). \quad (1.39)$$

La relation (1.39) définit un système d'équations linéaires à résoudre pour calculer les coefficients $p_{i,j}$.

1.3. Généralités sur les équations aux dérivées partielles

Utilisées depuis de nombreuses années en physique du solide, les Equations aux Dérivées Partielles (EDP) n'ont pris que récemment de l'importance dans le domaine du traitement des images, pour des applications de détection des contours, de segmentation de restauration, ainsi que dans le domaine d'inpainting.

1.3.1. Définitions

Soit $u = u(x_1, x_2, \mathbf{K}, x_d)$ une fonction de plusieurs variables indépendantes en nombre fini. Une EDP pour la fonction u est une relation entre u (variable dépendante où la solution

recherchée), les variables indépendantes $(x_1, x_2, \mathbf{K}, x_d)$, et un nombre fini de dérivées partielles de u . [7]

$$F(x_1, x_2, \mathbf{K}, x_d, u, \frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2}, \frac{\partial^2 u}{\partial x_1^2}, \frac{\partial^2 u}{\partial x_2^2}, \mathbf{K} \frac{\partial^m u}{\partial x_1^m}) = 0, \quad (1.40)$$

Où m est le degré de l'équation. Le problème d'EDP est posé sur un domaine (ouvert convexe) $\Omega \subset R^d$ où $(x \in \Omega)$. On note par $\partial\Omega$ la frontière de Ω sur laquelle on se donne des conditions aux bords et, si le temps intervient, des conditions initiales.

On dit que u est solution de l'équation aux dérivées partielles dans $\Omega \subset R^d$ si après substitution de u et de ses dérivées partielles, F s'annule pour tout $(x_1, x_2, \mathbf{K}, x_d) \in \Omega$.

Les équations aux dérivées partielles sont classées en fonction de plusieurs critères. Des théories de résolution sont aussi données en fonction du type de l'EDP.

- **L'ordre de l'EDP.** L'ordre de l'EDP est l'ordre de plus haute dérivée partielle présente dans l'équation.

$$\frac{\partial u}{\partial t} = \frac{\partial u}{\partial x}, \quad \text{première ordre,} \quad (1.41)$$

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad \text{second ordre,} \quad (1.42)$$

- **Nombre de variable.** Le nombre de variables est le nombre de variables indépendantes.

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad \text{deux variables } t \text{ et } x. \quad (1.43)$$

- **Linéarité.** Les EDP sont linéaires ou non-linéaires. Pour qu'elle soit linéaire, il faut que la variable u ne soit pas multipliée ou élevée à une puissance par elle-même ou par une de ses dérivées. A titre d'exemple, l'EDP suivante est linéaire

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \sin x \quad (1.44)$$

Par contre l'équation ci-dessous est non linéaire

$$\frac{\partial u}{\partial t} + u \frac{\partial^2 u}{\partial x^2} = 0, \quad (1.45)$$

- **Homogénéité.** Une EDP est dite homogène quand elle ne contient que des termes faisant intervenir u et ses dérivées partielles. L'équation linéaire à deux variables suivante est homogène

$$a \frac{\partial^2 u}{\partial x_1^2} + b \frac{\partial^2 u}{\partial x_1 \partial x_2} + c \frac{\partial^2 u}{\partial x_2^2} + d \frac{\partial u}{\partial x_1} + e \frac{\partial u}{\partial x_2} + fu = 0, \quad (1.46)$$

a, b, c, d, e, f peuvent être des constantes ou des fonctions des variables x_1 et x_2 .

- **Type de coefficients.** Si les coefficients a, b, c, d, e, f dans l'équation précédente sont constants, alors l'équation est dite à coefficients constants.

1.3.2 Conditions aux limites, condition initiales

La résolution des EDPs n'a de sens que si on impose un certain nombre de conditions aux limites que la solution doit respecter. Ces conditions peuvent être des conditions initiales (le temps par exemple) pour certaines variables et des conditions aux limites sur une région pour d'autres.

La limite d'une région représente la frontière $\partial\Omega$ dans le cas de deux variables, la surface dans le cas de trois variables. En général, on impose sur la frontière (en tout point), des conditions portant soit sur la fonction u (problème de Dirichlet), soit sur le gradient de u (problème de Neumann), soit sur la fonction u ainsi que son gradient (problème de Cauchy).

Les trois types de conditions aux limites sont :

- **Condition de Dirichlet** : on impose la valeur de u sur le bord $\partial\Omega$.
- **Condition de Neumann** : on impose la valeur de la dérivée normale de u sur le bord $\partial\Omega$.
- **Condition de Cauchy** : On impose ces deux conditions sur $\partial\Omega$.

Quand les valeurs imposées sont nulles, on dit que les conditions aux limites sont homogènes.

1.3.3 Classification des EDPs linéaires du second ordre

Les EDPs du second ordre représentent une classe importante des EDPs car elles permettent de modéliser plusieurs phénomènes naturels. La forme générale d'une EDP de second ordre est :

$$a \frac{\partial^2 u}{\partial x_1^2} + b \frac{\partial^2 u}{\partial x_1 \partial x_2} + c \frac{\partial^2 u}{\partial x_2^2} + d \frac{\partial u}{\partial x_1} + e \frac{\partial u}{\partial x_2} + fu = g, \quad (1.48)$$

Où a, b, c, d, e, f, g peuvent être des constantes ou des fonctions des variables x_1 et x_2 .

Dans cette partie, on s'intéresse aux EDP linéaires à coefficients constants. Selon le signe du déterminant $b^2 - 4ac$, l'EDP est qualifiée de :

- Elliptique si est seulement si $b^2 - 4ac < 0$,

- Parabolique si est seulement si $b^2 - 4ac = 0$,
- hyperbolique si est seulement si $b^2 - 4ac > 0$.

1.3.3.1 Les équations elliptiques

Les équations elliptiques régissent les problèmes stationnaires, d'équilibre, généralement définis sur un domaine spatial borné Ω de frontières $\partial\Omega$, sur laquelle l'inconnue est soumise à des conditions aux limites, le plus souvent de type Dirichlet ou Neumann. Ce type d'équation décrit les régimes permanents. Dans ce cas, on parle de problème d'équilibre ou de valeurs aux limites (PVL). Les équations de Laplace, Poisson ou de Helmholtz sont toutes elliptiques.

L'équation de Laplace (ou de Poisson si $f \equiv 0$) soumise à des conditions aux limites, par exemple de Dirichlet est:

$$\begin{cases} -\Delta u = f, & \text{dans } \Omega, \\ u = u_0, & \text{sur } \partial\Omega. \end{cases} \quad (1.49)$$

Cette équation modélise le déplacement d'une membrane soumise à une force extérieure f , avec un déplacement u_0 imposé sur la frontière $\partial\Omega$.

1.3.3.2 Les équations paraboliques

Les équations paraboliques régissent les problèmes d'évolution ou instationnaire dans lesquels intervient le mécanisme de diffusion ou de dissipation. Ces problèmes sont généralement définis sur un domaine spatial borné Ω de frontières $\partial\Omega$ sur laquelle l'inconnue est soumise à des conditions aux limites du même type qu'en elliptique, ainsi qu'à des conditions initiales. Il s'agit d'un problème de valeurs initiales (PVI). L'équation de la chaleur, de Schrödinger et de Navier-Stokes en sont des exemples d'EDPs paraboliques. L'équation de chaleur, soumise à des conditions aux limites, par exemple de Dirichlet, ainsi qu'à des conditions initiales est définie par:

$$\begin{cases} \frac{\partial u}{\partial t} - \Delta u = f, & \text{dans } \Omega, \\ u = g, & \text{sur } \partial\Omega, \\ u(x, t = 0) = u_0(x), & \text{dans } \Omega. \end{cases} \quad (1.50)$$

L'équation avec $f \equiv 0$ est appelée équation de chaleur ou de diffusion homogène.

1.3.3.3 Les équations hyperboliques

Les équations hyperboliques modélisent la propagation d'ondes sans dissipation (elles décrivent les systèmes oscillants et les mouvements d'onde). Le problème est dit à valeurs propres (PVP). L'exemple le plus classique d'EDP hyperbolique est l'équation d'ondes, celle-ci est décrite comme suit :

$$\begin{cases} \frac{\partial u}{\partial t} - \Delta u = f, & \text{dans } \Omega \times (0, T), \quad T > 0, \\ u = g, & \text{sur } \partial\Omega \times (0, T), \\ u(x, t = 0) = u_0(x), \\ \frac{\partial u}{\partial t}(x, t = 0) = v_0 & x \in \Omega. \end{cases} \quad (1.51)$$

C'est l'équation des ondes qui modélise des phénomènes d'évolution : corde ou membrane vibrante, ondes électromagnétiques, ondes sismiques, . . .

1.3.4 Formulation variationnelle ou énergétique

Souvent, il est plus avantageux de présenter les EDPs par des opérateurs différentiels classiques. On considère par exemple une EDP avec le problème de Dirichlet homogène suivant :

$$\begin{cases} -Lu = f & \text{sur } \Omega; \\ u = u_0 & \text{sur } \partial\Omega \end{cases} \quad (1.52)$$

L'opérateur différentiel L peut être le Laplacien Δ , l'Hamiltonien ∇ , ect. f est une fonction donnée.

L'EDP peut être alors reformuler sous certaines conditions comme un problème variationnel ou comme un problème de minimisation d'énergie. La méthode des éléments finis se fonde sur une formulation variationnelle pour déterminer des solutions numériques approchées du problème d'origine.

1.3.4.1 Formulation variationnelle

Ecrire un problème EDP sous forme variationnelle FV consiste à trouver :

$$u \in V \quad \text{tel que } a(u, v) = L(v) \quad \forall v \in V. \quad (1.53)$$

Où V est un espace de Hilbert, appelé aussi espaces des fonctions tests et a l'opérateur différentiel dérivé de L , v est une fonction test arbitraire de classe C^1 .

Pour trouver la formulation variationnelle de (1.52), on multiplie tout d'abord par une fonction test v , ensuite on intègre sur le domaine Ω (intégration par partie) en appliquant la formule de Green.

Si u est solution de la formulation variationnelle, alors u est solution de l'EDP associée.

1.3.4.2 Formulation énergétique

Dans le cas où la forme a est symétrique, on vérifie que toute solution u de (1.53) est aussi un minimiseur de la fonctionnelle

$$J(v) = \frac{1}{2} a(v, v) - L(v). \tag{1.54}$$

Pour $v \in V$, la formulation énergie revient à chercher $u \in V$ tel que

$$J(u) = \inf_{v \in V} J(v) \tag{1.55}$$

Une fonction u est solution de la formulation variationnelle si et seulement si u est solution de la formulation énergétique.

▼ Formules de Green

Définition : Un ouvert Ω de R^d ($d > 1$) est dit régulier de classe C^1 si son bord $\partial\Omega$ est une hypersurface (de dimension $d - 1$) régulière et si Ω est situé d'un seul côté de sa frontière.

Théorème1 : si u et v sont des fonctions de $C^1(\bar{\Omega})$, on a pour tout indice $i = 1, \dots, n$;

$$\int_{\Omega} \frac{\partial u}{\partial x_i} v dx = - \int_{\Omega} u \frac{\partial v}{\partial x_i} dx + \int_{\partial\Omega} u v n_i dS$$

Théorème2 : si $u \in C^2(\bar{\Omega})$ et $v \in C^1(\bar{\Omega})$, alors

$$\int_{\Omega} \Delta u v dx = - \int_{\Omega} \nabla u \cdot \nabla v dx + \int_{\partial\Omega} \frac{\partial u}{\partial n} v dS$$

Où $\frac{\partial u}{\partial n} = \nabla u \cdot n = \sum_{i=1}^n \frac{\partial u}{\partial x_i} n_i$ est la dérivé normale de u sur $\partial\Omega$, dS désigne la mesure surfacique sur $\partial\Omega$. Et on note par $C^k(\bar{\Omega})$ la restriction à $\bar{\Omega}$ de fonctions de $C^k(R^n)$.

Pour illustrer cette méthode considérons l'équation différentielle suivante avec les conditions de Dirichlet :

$$\begin{cases} -\Delta u = f, & \text{dans } \Omega \\ u = 0 & \text{sur } \partial\Omega. \end{cases} \quad (1.56)$$

En multipliant l'équation (1.42) par une fonction v arbitraire de classe C^1 , et en intégrant sur le domaine, on obtient

$$-\int_{\Omega} \Delta u v = \int_{\Omega} f v,$$

Et en appliquant la formule de green

$$\int_{\Omega} \Delta u \Delta v - \int_{\partial\Omega} \frac{\partial u}{\partial n} v = \int_{\Omega} f v,$$

Comme le terme de bord est nul: $\int_{\partial\Omega} \frac{\partial u}{\partial n} v = 0$, on obtient ainsi

$$\int_{\Omega} \Delta u \Delta v = \int_{\Omega} f v, \quad \forall v \in V$$

Alors u est solution de (18) si seulement si u est solution de la formulation énergétique

$$J(v) = \frac{1}{2} \int_{\Omega} |\nabla v|^2 dx - \int_{\Omega} f v dx, \quad \forall v \in V$$

1.3.4.3 Equation d'Euler Lagrange

Dans les méthodes basées sur une minimisation d'une certaine énergie, l'EDP apparaît quand on dérive une formule de l'équation d'Euler-Lagrange où on repère le minimum globale ou locale. On considère les fonctionnelles du type :

$$J(u) = \int_{\Omega} F(x, u, \nabla u, \Delta u) dx = \int_{\Omega} F(x, u, u_x, u_{xx}) dx \quad (1.57)$$

On cherche u qui minimise $J(u)$, revient à résoudre $\inf_u J(u)$ en appliquant l'équation d'Euler Lagrange suivante :

$$\frac{\partial F}{\partial u} - \frac{d}{dx} \left(\frac{\partial F}{\partial u_x} \right) + \frac{d^2}{dx^2} \left(\frac{\partial F}{\partial u_{xx}} \right) = 0. \quad (1.58)$$

Par exemple, soit à minimiser la fonctionnelle suivante :

$$J(u) = \int_{\Omega} (2xu + u_x^2 - u^2) dx \quad (1.59)$$

L'équation d'Euler Lagrange sera :

$$2x - 2u - 2u_{xx} = 0. \quad (1.60)$$

1.3.5 Méthodes de résolution des EDP

La littérature fait apparaître plusieurs techniques pour résoudre les EDPs. Celles-ci se regroupent principalement en 2 familles : Méthodes analytiques et Méthodes numériques. Notons qu'en traitement d'image, la résolution d'EDP s'effectue généralement par des méthodes numériques.

1.3.5.1. Méthodes analytiques

De nombreuses méthodes peuvent être utilisées pour résoudre une EDP, parmi lesquelles on peut citer la méthode de la solution générale et la méthode de séparation des variables. Ces méthodes consistent à transposer une EDP en équations aux dérivées ordinaires (EDO). La première méthode consiste à déterminer d'abord la solution générale puis la particulariser en utilisant les conditions aux limites. La seconde méthode consiste, par contre, à rechercher des solutions particulières puis les combiner pour retrouver la solution du problème. La méthode de séparation des variables, encore appelée méthode Fourier, est la plus utilisée à cause de sa simplicité. Elle est résumée comme suit :

- 1- Exprimer la solution comme le produit de fonctions inconnues d'une seule variable indépendante $u(x_1, x_2, \mathbf{K}, x_d) = U_1(x_1)U_2(x_2)..U_d(x_d)$ puis substituer cette solution dans l'EDP pour aboutir à des EDOs à une seule variable.
- 2- Résoudre chaque EDO séparément puis déterminer la solution finale en combinant les solutions séparées.

1.3.5.2 Avantages et inconvénients des méthodes analytiques

Il est toujours souhaitable d'avoir recours aux méthodes analytiques pour avoir la solution analytique (exacte) du problème. Cependant ces méthodes sont souvent difficiles à réaliser à cause des inconvénients suivants :

- Utilisation d'un grand nombre de termes pour obtenir la solution d'après le principe de superposition.
- La solution est toujours difficile à trouver. La méthode est spécifique à chaque type de problème.
- La difficulté s'accroît davantage si les conditions aux limites et l'EDP sont non homogènes.

1.3.6 Méthodes numériques

En vue du passage d'un problème exact (continu) au problème approché (discret), on dispose de plusieurs méthodes, les éléments finis, les volumes finis et les différences finies. Ces méthodes permettent de changer une EDP en un système d'équations linéaires qui peuvent être résolus sur le plan informatique par des techniques itératives. Dans ces méthodes, chaque dérivée est approchée par une expression discrétisée.

- **Différences finies :** La méthode des différences finies consiste à remplacer les dérivées apparaissant dans le problème continu par des différences divisées ou combinaisons de valeurs ponctuelles de la fonction en un nombre fini de points discrets ou nœuds du maillage.
- **Éléments finis :** La méthode des éléments finis consiste à approcher, dans un sous-espace de dimension finie, un problème écrit sous forme variationnelle (comme minimisation de l'énergie, en général) dans un espace de dimension infinie. La solution approchée est dans ce cas une fonction déterminée par un nombre fini de paramètres comme, par exemple, ses valeurs en certains points (les nœuds du maillage).
- **Volumes finis :** La méthode des volumes finis intègre, sur des volumes élémentaires de forme simple, les équations écrites sous forme de loi de conservation. Elle fournit ainsi de manière naturelle des approximations discrètes conservatives, elle donc particulièrement bien adaptée aux équations de la mécanique des fluides. Sa mise en œuvre est simple avec des volumes élémentaires rectangles.

En traitement d'image, la méthode la plus couramment employée pour discrétiser les EDPs est celle des différences finies, cela est dû à la structure des images qui sont formées par un ensemble de pixels uniformément distribués.

1.3.6.1 Méthode des différences finies

Cette méthode consiste à approximer les dérivées partielles d'une équation au moyen du développement en série de Taylor de la variable considérée au voisinage d'un point.

1.3.6.1.1 Formule de Taylor

Soit $u(x, y, t)$ une fonction de l'espace et du temps. Par définition de la dérivée, on a :

$$\frac{\partial u}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{u(x + \Delta x, y, t) - u(x, y, t)}{\Delta x} \quad (1.61)$$

Si Δx est petit, un développement de Taylor de $u(x + \Delta x, y, t)$ au voisinage de x donne :

$$u(x + \Delta x, y, t) = u(x, y, t) + \Delta x \frac{\partial u}{\partial x}(x, y, t) + \frac{\Delta x^2}{2} \frac{\partial^2 u}{\partial x^2}(x, y, t) + \frac{\Delta x^3}{6} \frac{\partial^3 u}{\partial x^3}(x, y, t) + \mathbf{L}$$

En tronquant la série au premier ordre en Δx , on obtient :

$$\frac{u(x + \Delta x, y, t) - u(x, y, t)}{\Delta x} = \frac{\partial u}{\partial x}(x, y, t) + \mathbf{q}(\Delta x)$$

$$\frac{\partial u}{\partial x}(x, y, t) \approx \frac{u(x + \Delta x, y, t) - u(x, y, t)}{\Delta x} \tag{1.62}$$

Ceci est appelé le schéma avant.

De la même manière, nous pouvons aussi donner le schéma arrière qui est de la forme :

$$\frac{\partial u}{\partial x}(x, y, t) \approx \frac{u(x, y, t) - u(x - \Delta x, y, t)}{\Delta x} \tag{1.63}$$

La somme de ces deux schémas nous donne le schéma centré suivant

$$\frac{\partial u}{\partial x}(x, y, t) \approx \frac{u(x + \Delta x, y, t) - u(x - \Delta x, y, t)}{2\Delta x} \tag{1.64}$$

1.3.6.1.2 Schéma numérique de la dérivée première

Considérons l'évolution d'une grandeur $u(x, t)$ en fonction de l'espace et du temps. Le domaine de définition de u est décomposé en N nœuds x_i répartis régulièrement avec un pas d'espace Δx , tel que $x_{i+1} = x_i + \Delta x$. De même, le temps est décomposé en intervalle élémentaire de pas constant Δt . On notera u_i^n la valeur discrète de la grandeur $u(x, t)$ au nœud x_i et au temps $n\Delta t$.

Le schéma aux différences finies d'ordre 1 présenté précédemment s'écrit, en notation indicielle

- $\left(\frac{\partial u}{\partial x}\right)_i \approx \frac{u_{i+1} - u_i}{\Delta x}$, Différence finie avant ou progressive
- $\left(\frac{\partial u}{\partial x}\right)_i \approx \frac{u_i - u_{i-1}}{\Delta x}$, Différence finie arrière ou rétrograde
- $\left(\frac{\partial u}{\partial x}\right)_i \approx \frac{u_{i+1} - u_{i-1}}{2\Delta x}$, Différence finie centrée

1.3.6.1.3 Schéma numérique de la dérivée deuxième

Le principe est identique et repose sur les développements de Taylor au voisinage de x_i . Par exemple, pour construire un schéma d'approximation de la dérivée seconde de u , on écrit

$$u_{i+1} = u_i + \Delta x \left(\frac{\partial u}{\partial x}\right)_i + \frac{\Delta x^2}{2} \left(\frac{\partial^2 u}{\partial x^2}\right)_i + \frac{\Delta x^3}{6} \left(\frac{\partial^3 u}{\partial x^3}\right)_i + q(\Delta x^4) \tag{1.65}$$

$$u_{i+1} = u_i - \Delta x \left(\frac{\partial u}{\partial x}\right)_i + \frac{\Delta x^2}{2} \left(\frac{\partial^2 u}{\partial x^2}\right)_i - \frac{\Delta x^3}{6} \left(\frac{\partial^3 u}{\partial x^3}\right)_i + q(\Delta x^4) \tag{1.66}$$

En faisant la somme de ces deux égalités, on aboutit à $u_{i+1} + u_{i-1} - 2u_i = \Delta x^2 \left(\frac{\partial^2 u}{\partial x^2}\right)_i + q(\Delta x^4)$.

Ce qui permet d'obtenir le schéma d'ordre deux pour approximer la dérivée seconde de u

- $\left(\frac{\partial^2 u}{\partial x^2}\right)_i \approx \frac{u_{i+1} + u_{i-1} - 2u_i}{\Delta x^2}$ Différence finie centrée,
- $\left(\frac{\partial^2 u}{\partial x^2}\right)_i \approx \frac{u_{i+2} - u_{i+1} + u_{i-1}}{\Delta x^2}$ Différence finie avant ou progressive,
- $\left(\frac{\partial^2 u}{\partial x^2}\right)_i \approx \frac{u_i - 2u_{i-1} + u_{i-2}}{\Delta x^2}$ Différence finie avant ou rétrograde.

La résolution d'une EDP par les différences finies suit les étapes suivantes [8] :

- Construire le maillage ou une grille du domaine Ω . Notons que le maillage est un ensemble de points du domaine de définition sur lequel on va appliquer la méthode des différences finies.
- Transformer l'équation aux dérivées partielles et l'exprimer sous forme de schéma numérique de différences finies.
- Ecrire l'équation de différences finie aux points du maillage.
- Obtenir un système d'équations algébriques discrètes $Au = b$, tel que b est le vecteur connu donné par les conditions aux limites non homogènes, A est la matrice de coefficients et u est le vecteur solution recherché en tout point du maillage.
- Trouver la solution u en résolvant le système d'équations $Au = b$.

Pour illustrer cette méthode, considérons l'équation différentielle suivante avec les conditions de Dirichlet :

$$\begin{cases} \frac{\partial^2 u}{\partial x^2} = f(x) \quad , x \in]0,1[\\ u(0) = a \quad \text{et} \quad u(1) = b. \end{cases} \quad (1.67)$$

Où f est une fonction continue.

Le maillage est construit en introduisant $N + 1$ nœuds x_i avec $i = 0, 1, \dots, N$, régulièrement espacés avec un pas Δx . La quantité u_i désignera la valeur de la fonction $u(x)$ au nœud x_i .

L'équation à résoudre s'écrit, sous forme discrète en chaque nœud x_i :

$$-\left(\frac{\partial^2 u}{\partial x^2}\right)_i = f(x_i) = f_i. \quad (1.68)$$

L'approximation de la dérivée seconde de u au moyen d'un schéma centré à l'ordre deux donne:

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_i = \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2}. \quad (1.69)$$

L'équation discrétisée est ainsi :

$$\frac{2u_i - u_{i+1} - u_{i-1}}{\Delta x^2} = f_i \quad \text{pour } i = 1, \dots, N-1. \quad (1.70)$$

Il est très pratique d'utiliser une formulation matricielle en faisant apparaître le vecteur des inconnues discrètes :

$$\begin{bmatrix} 2 & -1 & 0 & \mathbf{L} & 0 \\ -1 & 2 & -1 & \mathbf{L} & 0 \\ \mathbf{M} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{M} \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \mathbf{M} \\ u_{N-2} \\ u_{N-1} \end{bmatrix} = \begin{bmatrix} f_1 + \frac{a}{\Delta x^2} \\ f_2 \\ \mathbf{M} \\ f_{N-2} \\ f_{N-1} + \frac{b}{\Delta x^2} \end{bmatrix}. \quad (1.71)$$

1.3.6.2 Méthode des éléments finis

La méthode des éléments finis est une technique de discrétisation pour le problème mathématique posé sous la forme d'un problème de minimisation ou sous une forme variationnelle. Soit le problème mis sous la forme de l'équation (1.53), sa discrétisation consiste à remplacer V par un sous espace de dimension finie V_h , tel que

$$a(u_h, j_h) = l(j_h), \quad \forall j_h \in V_h. \quad (1.72)$$

Pour illustrer cette méthode considérons l'équation différentielle suivante avec les conditions de Dirichlet :

$$\begin{cases} -u'' = f(x) & , x \in]0,1[\\ u(0) = u(1) = 0. \end{cases} \quad (1.73)$$

L'écriture variationnelle de cette équation est sous la forme suivante

$$-\int_0^1 u''(x)v(x) dx = \int_0^1 f(x)v(x) dx \quad (1.74)$$

En intégrant par partie, on aura

$$\int_0^1 u'(x)v'(x) dx = \int_0^1 f(x)v(x) dx, \quad \forall v \in V. \quad (1.75)$$

On cherche alors à écrire un problème approché dans un espace vectoriel de dimension finie.

Soit \tilde{V} un sous espace vectoriel de V de dimension N finie. Soient j_1, j_2, \dots, j_N , N fonctions linéairement indépendantes de V . Ces fonctions constituent une base de sous espace \tilde{V} .

Ainsi, toute fonction \tilde{u} de \tilde{V} peut se décomposer selon l'expression suivante :

$$\tilde{u}(x) = \sum_{j=1}^N u_j j_j(x). \quad (1.76)$$

Résoudre le problème différentiel de départ revient à chercher une solution $\tilde{u} \in \tilde{V}$, telle que :

$$\int_0^1 \tilde{u}'(x)\tilde{v}'(x) dx = \int_0^1 f(x)\tilde{v}(x) dx, \quad \forall \tilde{v} \in \tilde{V}. \quad (1.77)$$

C'est-à-dire chercher N réels u_1, u_2, \dots, u_N vérifiant :

$$\sum_{j=1}^N u_j \int_0^1 j_j'(x)\tilde{v}'(x) dx = \int_0^1 f(x)\tilde{v}(x) dx \quad (1.78)$$

Ou encore

$$\sum_{j=1}^N u_j \int_0^1 j_j'(x)j_i'(x) dx = \int_0^1 f(x)j_i(x) dx, \quad \forall j_i \in \tilde{V}. \quad (1.79)$$

Soient A la matrice de $N \times N$ d'élément courant a_{ij} et B le vecteur à N composantes b_i

définies respectivement par $a_{ij} = \int_0^1 j_j'(x)j_i'(x) dx$ et $b_i = \int_0^1 f(x)j_i(x) dx$.

Par définition, la matrice A est symétrique. Notons U le vecteur des inconnues u_1, u_2, \dots, u_N .

Le problème différentiel se ramène à la résolution du système linéaire $A \cdot U = B$.

L'intervalle $]0,1[$ est discrétisé en N points de coordonnées x_i . Les fonctions $j_i(x)$ sont choisies comme fonction polynomiale d'ordre 1 définie par :

$$j_i(x) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}} & \text{si } x_{i-1} < x < x_i \\ \frac{x - x_{i+1}}{x_i - x_{i+1}} & \text{si } x_i < x < x_{i+1} \\ 0 & \text{sin on.} \end{cases} \quad (1.80)$$

Ces fonctions sont appelées les éléments finis de degré 1 (il est aussi possible de choisir pour éléments finis des fonctions de degré 2 ou plus). Le calcul de la matrice A fait intervenir les dérivées $j'_i(x)$ simples à calculer :

$$j'_i(x) = \begin{cases} \frac{1}{x_i - x_{i-1}} & \text{si } x_{i-1} < x < x_i \\ \frac{1}{x_i - x_{i+1}} & \text{si } x_i < x < x_{i+1} \\ 0 & \text{sin on.} \end{cases} \quad (1.81)$$

Les éléments de la matrice sont calculés comme suit :

$$a_{ii} = \int_0^1 j'_i(x) j'_i(x) dx = \frac{1}{x_i - x_{i-1}} + \frac{1}{x_{i+1} - x_i}. \quad (1.82)$$

$$a_{i,i+1} = \int_0^1 j'_{i+1}(x) j'_i(x) dx = \frac{-1}{x_{i+1} - x_i}. \quad (1.83)$$

$$a_{i-1,i} = \int_0^1 j'_i(x) j'_{i-1}(x) dx = \frac{-1}{x_i - x_{i-1}}. \quad (1.84)$$

Les composantes du vecteur B peuvent être calculées par une méthode des trapèzes, soit :

$$b_i = \int_0^1 f(x) j_i(x) dx = f_i \left(\frac{x_{i+1} - x_{i-1}}{2} \right). \quad (1.85)$$

Le système linéaire à résoudre s'écrit donc sous forme indicielle

$$\frac{u_i - u_{i-1}}{x_i - x_{i-1}} - \frac{u_{i+1} - u_i}{x_{i+1} - x_i} = \frac{x_{i+1} - x_{i-1}}{2} f_i, \text{ pour } i = 1, \mathbf{K}, N. \quad (1.86)$$

Dans le cas où les N points de l'intervalle $]0,1[$ sont régulièrement espacés avec un pas h . La discrétisation en éléments finis devient :

$$\frac{2u_i - u_{i+1} - u_{i-1}}{h^2} = f_i, \text{ pour } i = 1, \mathbf{K}, N. \quad (1.87)$$

Soit sous forme matricielle :

$$\frac{1}{h^2} \begin{bmatrix} 2 & -1 & 0 & \mathbf{L} & 0 \\ -1 & 2 & -1 & \mathbf{L} & 0 \\ \mathbf{M} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{M} \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \mathbf{M} \\ u_{N-1} \\ u_N \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \mathbf{M} \\ f_{N-1} \\ f_N \end{bmatrix}. \quad (1.88)$$

La méthode des Eléments Finis consiste donc à :

- Choisir N points entre 0 et 1 et choisir les fonctions j_i .
- Construire la matrice A .
- Déterminer le vecteur B (avec une méthode d'intégration).
- Résoudre le système linéaire $A \cdot U = B$ où B désigne le vecteur des inconnues.

1.3.6.3 Méthode des volumes finis

La méthode des volumes finis consiste à intégrer sur des volumes élémentaires les équations écrites sous forme intégrale. C'est une méthode particulièrement bien adaptée à la discrétisation spatiale des lois de conservation, contrairement aux éléments finis. Elle est très utilisée en mécanique des fluides.

Considérons une loi de conservation d'une grandeur physique w dans une maille de volume Ω , faisant intervenir un flux $F(w)$ et un terme source $S(w)$. L'expression de cette loi de conservation est donnée sous forme intégrale:

$$\frac{\partial}{\partial t} \int_{\Omega} w d\Omega + \int_{\Omega} \text{div} F(w) d\Omega = \int_{\Omega} S(w) d\Omega. \quad (1.89)$$

La méthode des volumes finis consiste à :

- Décomposer la géométrie en mailles élémentaires (élaborer un maillage).
- Initialiser la grandeur w sur le domaine de calcul.
- Lancer le processus d'intégration temporelle jusqu'à convergence avec :
 1. Calcul du bilan de flux par maille par un schéma numérique.
 2. Calcul du terme source.
 3. Calcul de l'incrément temporel par une méthode numérique d'intégration.
 4. Application des conditions aux limites.

Pour illustrer cette méthode, considérons l'équation différentielle précédente :

$$\begin{cases} \frac{\partial^2 u}{\partial x^2} = f(x) & , x \in]0,1[\\ u(0) = a \quad \text{et} \quad u(1) = b. \end{cases} \quad (1.90)$$

Où f est une fonction continue.

L'intervalle $]0,1[$ est discrétisé en N mailles de centre x_i et de taille $h_i = \frac{x_{i+1}}{2} - \frac{x_{i-1}}{2}$. La fonction $u(x)$ est supposée constante dans chaque maille et égale à une valeur approchée de la moyenne sur la maille considérée. On notera u_i cette valeur dans i -ème maille :

$$\forall x \in \left[\frac{x_{i-1}}{2}, \frac{x_{i+1}}{2} \right], u(x) = u_i.$$

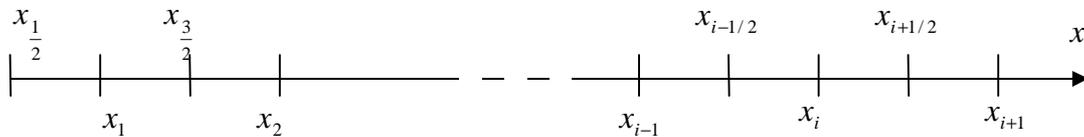


Fig. 1.1 : Maillage 1D

La discrétisation spatiale par les volumes finis consiste à intégrer maille par maille l'équation différentielle du problème, soit pour la i -ème maille :

$$\int_{x_{i-1}/2}^{x_{i+1}/2} -\frac{\partial^2 u}{\partial x^2}(x) dx = \int_{x_{i-1}/2}^{x_{i+1}/2} f(x) dx. \quad (1.91)$$

Ce qui donne par intégration :

$$\frac{\partial u}{\partial x}(x_{i-1}/2) - \frac{\partial u}{\partial x}(x_{i+1}/2) = h_i \tilde{f}_i \quad \text{pour } i = 1, \mathbf{L}, N. \quad (1.92)$$

Où \tilde{f}_i désigne la valeur moyenne de f sur la i -ème maille : $\tilde{f}_i = \frac{1}{h_i} \int_{x_{i-1}/2}^{x_{i+1}/2} f(x) dx$.

Il reste maintenant à exprimer $\frac{\partial u}{\partial x}(x_{i-1}/2)$ en fonction des inconnus u_i . L'approximation la plus naturelle est de prendre la valeur moyenne $\frac{\partial u}{\partial x}$ sur le segment $[x_{i-1}, x_i]$, soit :

$$\frac{\partial u}{\partial x}(x_{i-1}/2) = \frac{1}{h_{i-1} + h_i} \int_{x_{i-1}}^{x_i} \frac{\partial u}{\partial x} dx = \frac{u(x_i) - u(x_{i-1})}{h_{i-1/2}} = \frac{u_i - u_{i-1}}{h_{i-1/2}}. \quad (1.93)$$

Avec $h_{i-1/2} = \frac{h_{i-1} + h_i}{2}$.

Cette dernière expression n'est pas valable au bord gauche, pour $i = 1$ (en $x_{1/2}$), car elle fait intervenir le point x_0 qui n'est pas défini. Il se pose alors le problème du traitement des bords qui exige une formulation particulière. Une possibilité est de définir une maille fictive à gauche de l'intervalle $[x_0, x_1]$ et d'affecter une valeur moyenne de la fonction u dans cette maille. Une autre possibilité est de considérer la valeur moyenne de $\frac{\partial u}{\partial x}(x_{1/2})$ non plus sur le segment $[x_0, x_1]$ qui n'est pas défini mais sur le segment $[x_{1/2}, x_1]$, c'est ce que nous choisissons dans cet exemple. Ainsi on écrit :

$$\frac{\partial u}{\partial x}(x_{1/2}) = \frac{2}{h_1} \int_{x_{1/2}}^{x_1} \frac{\partial u}{\partial x} dx = \frac{2(u_1 - u(0))}{h_1} = \frac{2(u_1 - a)}{h_1}. \quad (1.94)$$

De même pour le terme $\frac{\partial u}{\partial x}(x_{i+1/2})$, on écrit que $\frac{\partial u}{\partial x}(x_{i+1/2}) = \frac{u_{i+1} - u_i}{h_{i+1/2}}$. Le même problème survient au bord droit, pour $i = N$, en $x_{N+1/2} = 1$. On considère la valeur moyenne de $\frac{\partial u}{\partial x}(x_{N+1/2})$ non plus sur le segment $[x_N, x_{N+1}]$ qui n'est pas défini mais sur le segment $[x_N, x_{N+1/2}]$, soit :

$$\frac{\partial u}{\partial x}(x_{N+1/2}) = \frac{2}{h_N} \int_{x_N}^{x_{N+1/2}} \frac{\partial u}{\partial x} dx = \frac{2(u(1) - u_N)}{h_N} = \frac{2(b - u_N)}{h_N}. \quad (1.95)$$

La discrétisation en Volumes Finis donne finalement :

$$\begin{aligned} \frac{u_i - u_{i-1}}{h_{i-1/2}} - \frac{u_{i+1} - u_i}{h_{i+1/2}} &= h_i \tilde{f}_i, \quad \text{pour } i = 2, \mathbf{L}, N-1. \\ \frac{2(u_1 - a)}{h_1} - \frac{u_2 - u_1}{h_{3/2}} &= h_1 \tilde{f}_1, \\ \frac{u_N - u_{N-1}}{h_{N-1/2}} - \frac{2(b - u_N)}{h_N} &= h_N \tilde{f}_N. \end{aligned} \quad (1.96)$$

Dans le cas particulier d'un maillage régulier de pas h .

$$\begin{aligned} \frac{2u_i - u_{i-1} - u_{i+1}}{h^2} &= \tilde{f}_i \quad \text{pour } i = 2, \mathbf{L}, N-1. \\ \frac{3u_1 - u_2}{h^2} &= \tilde{f}_1 + \frac{2a}{h^2} \\ \frac{3u_N - u_{N-1}}{h^2} &= \tilde{f}_N + \frac{2b}{h^2}. \end{aligned} \quad (1.97)$$

Sous forme matricielle, ceci s'exprime :

$$\begin{bmatrix} 3 & -1 & 0 & \mathbf{L} & 0 \\ -1 & 2 & -1 & \mathbf{L} & 0 \\ \mathbf{M} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{M} \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 3 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \\ u_N \end{bmatrix} = \begin{bmatrix} \tilde{f}_1 + \frac{2a}{h^2} \\ \tilde{f}_2 \\ \vdots \\ \tilde{f}_{N-1} \\ \tilde{f}_N + \frac{2b}{h^2} \end{bmatrix}. \quad (1.98)$$

La résolution de ce système d'équations linéaires fournit la solution de l'EDP.

1.3.7 Notions de consistance, convergence et stabilité des méthodes numériques

Un certain nombre de notions est nécessaire lors de la résolution d'équations aux dérivées partielles (EDP) au moyen de leurs équivalents discrétisés. Les trois principales sont la convergence, la stabilité et la consistance. Ces trois propriétés permettent de relier la solution exacte des équations continues à la solution exacte des équations discrétisées et à la solution numérique obtenue.

- **La stabilité :** C'est la propriété qui assure que la différence entre la solution numérique obtenue et la solution exacte des équations discrétisées est bornée.
- **La consistance :** C'est la propriété qui assure que la solution exacte des équations discrétisées tend vers la solution exacte des équations continues lorsque le pas de discrétisation Δx et Δt tendent vers zéro.
- **La convergence :** C'est la propriété qui assure que la solution numérique tend vers la (ou une) solution exacte des équations continues.

1.3.7 Les EDP en traitement d'image

De nombreuses méthodes basées sur des EDPs sont apparues ces dernières années pour résoudre le problème de la restauration, de la segmentation, de la détection, d'inpainting, etc.... des images. Dans cette section, nous n'aborderons que quelques points importants sur les méthodes d'EDP appliquées en traitement d'images.

1.3.7.1 Détection de contours

Un des objectifs important en traitement d'image est de pouvoir déterminer les contours d'objets. Pour cela, un moyen efficace consiste à utiliser la méthode des contours actifs. Celle-ci consiste à faire évoluer une courbe notée C (un snake) autour ou à l'extérieur de l'objet à détecter [10]. Ce snake se déforme progressivement jusqu'à épouser la forme de

l'objet. Cette évolution est décrite par une EDP obtenue en générale en minimisant la fonctionnelle suivante :

$$E(C) = \int_0^1 (a|v'(s)|^2 + b|v''(s)|^2) ds - I \int_0^1 |\nabla u(v(s))| ds \quad (1.99)$$

Notons que $v(s) = (x(s), y(s))$ constitue le point courant du contour C (s représente l'abscisse curviligne $s \in [0,1]$).

Les deux premiers termes constituent l'énergie interne de la courbe C , le troisième terme constitue l'énergie externe et dépend du gradient de l'image u , a, b et I sont des constantes réelles. Le but est donc de trouver la courbe C et les constantes a, b et I telle que $E(C)$ soit minimale. Ceci peut être résolu en résolvant l'équation d'Euler Lagrange suivante :

$$\frac{\partial u}{\partial s} = \left(a \frac{\partial^2 v}{\partial s^2} \right) - \left(b \frac{\partial^2 v}{\partial s^2} \right) - \nabla u(v(s)). \quad (1.100)$$

1.3.7.2 Segmentation

La segmentation joue un rôle prépondérant en analyse d'images. Elle permet notamment de décomposer une image en régions homogènes afin de pouvoir appréhender facilement son contenu. Les travaux traduisant cette problématique sous forme d'EDP sont nombreux. Parmi ces derniers, on trouve le modèle de Chan et Vese qui consiste à minimiser la fonctionnelle suivante :

$$E(C, c_1, c_2) = \int_{\Omega_{int}} (u_0 - c_1)^2 dx + \int_{\Omega_{ext}} (u_0 - c_2)^2 dx + v \int ds \quad (1.101)$$

où C est la courbe qui va permettre de détecter l'objet, Ω_{int} et Ω_{ext} représentent l'intérieur et l'extérieur de la courbe C . c_1 et c_2 sont défini comme étant la moyen intérieur et extérieur de C . Afin de résoudre ce problème variationnel avec contrainte, une EDP peut être appliquée. La solution vérifie nécessairement l'équation d'Euler-Lagrange

$$\left(I_1 (u_0 - c_1)^2 - I_2 (u_0 - c_2)^2 - v \cdot \nabla \left(\frac{\nabla u}{|\nabla u|} \right) \right) = 0 \quad (1.102)$$

$$\frac{\partial u}{\partial t} = \left(v \nabla \cdot \frac{\nabla u}{|\nabla u|} - I_1 (u_0 - c_1)^2 + I_2 (u_0 - c_2)^2 \right) \quad (1.103)$$

I_1, I_2 et v sont des constantes.

1.3.7.3 Restauration

La restauration est une technique du traitement d'images qui s'applique sur une image dégradée. Elle a pour but de corriger les distorsions introduites lors des étapes d'acquisition ou de transmission des images. Plusieurs techniques de correction basées sur des EDPs de diffusion ont été proposées.

1.3.7.1.1 Diffusion isotropique

L'une des premières idées a été d'établir une analogie entre la restauration d'une image et un phénomène physique qui est celui de la diffusion de chaleur. De même façon que, dans un matériau, la chaleur se diffuse d'un point à un autre, de proche en proche, et tend ainsi à se répartir uniformément au fur et à mesure que le temps s'écoule. On peut imaginer diffuser de proche en proche les niveaux de gris dans une image. Ainsi, les irrégularités des niveaux de gris diminueront et l'on retrouve un niveau de gris uniforme. L'équation de chaleur est une EDP linéaire parabolique qui peut s'écrire sous la forme suivante :

$$\begin{cases} \frac{\partial u}{\partial t} = \Delta u, & t \geq 0, \text{ et } x \in R^2 \\ u(x,0) = u_0. \end{cases} \quad (1.104)$$

La résolution de cette équation mène à la convolution linéaire (lissage) de l'image initiale avec un noyau gaussienne $G_t(x)$ [11].

$$u(t, x) = (G_{\sqrt{2t}} * u_0)(x) \quad (1.105)$$

Avec

$$G_t(x) = \frac{1}{4pt} \exp\left(-\frac{\|x\|^2}{4t^2}\right). \quad (1.106)$$

Cette équation de diffusion (1.104) peut être réécrite sous la forme divergence suivante :

$$\begin{cases} \frac{\partial u}{\partial t} = \operatorname{div}(\nabla u), & t \geq 0, \text{ et } x \in R^2 \\ u(x,0) = u_0. \end{cases} \quad (1.107)$$

Ce schéma itératif correspond à un filtrage passe bas, où le lissage est le même dans toute les directions, on le dit isotropique. Il ne prend pas en compte la structure de l'image et engendre la disparition des contours et rend l'image de plus en plus floue. Afin de préserver les structures géométriques de l'image des équations non linéaires (diffusion anisotropique) ont été proposées.

1.3.7.1.2 Diffusion anisotropique

La diffusion anisotropique a été introduite par Perona et Malik [12]. L'idée première est de rendre la diffusion dépendante du contenu de l'image et en particulier des contours. Elle consiste à effectuer une diffusion conditionnelle, c.-à-d. forte diffusion dans les zones à faible gradient et faible diffusion dans les zones à fort gradient. L'équation s'écrit alors avec une fonction décroissante qui varie en fonction de la valeur de gradient. L'équation de Perona Malik s'écrit :

$$\begin{cases} \frac{\partial u}{\partial t} = \text{div}(g(|\nabla u|)\nabla u), & t \geq 0, \text{ et } x \in R^2 \\ u(x,0) = u_0. \end{cases} \tag{1.108}$$

avec g une fonction décroissante qui satisfait les conditions suivantes

$$g(0) = 1 \text{ et } \lim_{k \rightarrow +\infty} g(k) = 0$$

Celles proposées par Perona et Malik sont :

$$g(|\nabla u|) = \exp(-(|\nabla u|/k)^2), \tag{1.109}$$

$$g(|\nabla u|) = \frac{1}{1 + (|\nabla u|/k)^2}. \tag{1.110}$$

k étant un paramètre à fixer.

1.3.7.1.3 Restauration par minimisation de la variation totale

La variation totale (TV) d'une image mesure l'amplitude totale des oscillations d'une fonction [13]. Etant donnée une image $u(x_1, x_2)$, définie sur une région Ω bornée, ouverte et convexe, sa variation totale (TV) est définie comme suit :

$$TV(u) = \int_{\Omega} |\nabla u| \, dx = \int_{\Omega} \sqrt{u_{x_1}^2 + u_{x_2}^2} \, dx_1 dx_2. \tag{1.111}$$

L'un des plus populaires modèles de la variation totale a été proposé par Rudin, Osher et Fatemi pour la restauration d'images [13]. Ce modèle a prouvé son efficacité à supprimer les oscillations tout en préservant les discontinuités qui peuvent se voir comme les contours d'une image. Il est défini par :

$$TV(u) = \int_{\Omega} |\nabla u| \, dx, \text{ tel que } \int_{\Omega} (u_0 - u)^2 \, dx = S^2. \tag{1.112}$$

Où u_0 est l'image observée et dégradée par un bruit gaussien de moyenne zéro et de variance S^2 . Afin de résoudre ce problème variationnel avec contrainte, une EDP peut être appliquée. La solution de (1.112) vérifie nécessairement l'équation d'Euler-Lagrange

$$\nabla \cdot \frac{\nabla u}{|\nabla u|} - I(u - u_0) = \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) - I(u - u_0) = 0. \quad (1.113)$$

et la condition de bord de Neumann homogène. I est un multiplicateur de Lagrange qui est choisi tel que la contrainte puisse être satisfaite.

1.4. Conclusion

Nous avons présenté dans ce chapitre quelques notions sur l'interpolation des fonctions mono et bi-variables ainsi qu'un bref rappel sur les équations différentielles partielles. Nous avons également donné quelques exemples sur l'utilisation des EDPs en traitement d'images. Ces notions d'interpolation et d'EDPs vont nous servir à la compréhension et à la résolution du problème de la restauration automatique de données manquantes ou inpainting. L'inpainting se révèle comme une nouvelle tendance dans le domaine du traitement d'images. Il s'agit de reconstituer l'intégralité d'une ou de plusieurs zones perdues d'une image, en utilisant les données valides présentes au voisinage de ces zones. Un état de l'art sur l'inpainting fera l'objet du prochain chapitre.

2.1 Introduction

L'inpainting d'image correspond aux techniques de restauration d'image qui permettent de remplir des parties manquantes dans une image. Plusieurs méthodes ont été proposées ces dix dernières années et qu'on peut classer en deux grande catégories. La première catégorie se base sur des méthodes d'interpolation classiques. La second met l'accent sur la reconstruction de la géométrie globale de l'image et regroupent des modèles exprimés sous forme de EDP.

2.2 Définition de l'inpainting

L'inpainting est un terme utilisé par des dessinateurs pour restaurer des anciens tableaux ou des gravures anciens dégradés par l'usure du temps ou par une mauvaise manipulation. Il est le synonyme artistique qui désigne l'interpolation d'image, c'est aussi un terme qui désigne la reconstruction de données manquantes dans une image. Cette reconstruction a pour but de déterminer le niveau de gris ou la couleur des pixels dans les régions d'une image où l'information est manquante en fonction de l'information disponible (niveau de gris ou couleur des pixels connus) d'une manière la plus automatique possible.

2.3 Principe de l'inpainting

Le problème d'inpainting d'image peut être formulé de la manière suivante. Etant donnée une image u_0 définie sur un domaine complet Ω . Soit un sous-ensemble $D \subseteq \Omega$ tel que l'information de l'image sur D est manquante ou détériorée (voir fig.2.1). Le but d'inpainting est de reconstruire l'image idéale u à partir de l'image u_0 . Cela revient à déterminer l'information manquante dans le domaine D à partir de l'information valide autour de D (ou bien sur la frontière ∂D). Autrement dit, l'inpainting consiste à trouver une fonction u_D définie dans D tel que u_D est une bonne approximation de u_0 dans le domaine $D(u_0 /_D)$.

Ce problème tel qu'il est posé s'apparente à celui de l'interpolation des fonctions (chapitre 1). Par conséquent, l'inpainting peut être formulé mathématiquement sous forme d'interpolation ou extrapolation spatiale des valeurs de pixels inconnus de l'image.

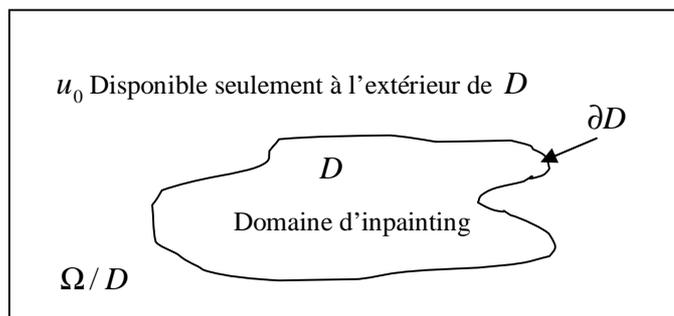


Fig. 2.1 : Principe de l'inpainting

2.4 Sources de détérioration des données et applications

Il existe plusieurs facteurs qui peuvent conduire à la détérioration ou au manque d'informations dans une image. Ce problème touche plusieurs applications. Ainsi, l'inpainting peut être utilisée pour restaurer de manière numérique des images dégradées par des artefacts ayant détruit de manière complète certaines parties des images. Ces artefacts peuvent être causés par des rayures sur des films anciens ou taches sur des photographies (**Fig.2.2**). Une application dérivée consiste en la *suppression cohérente* d'objets réels dans des images, en supposant que ces objets sont considérés comme des artefacts que l'on veut corriger (**Fig.2.3**). Ce type d'application cherche à cacher certains effets spéciaux cinématographiques, des éléments indésirables sur une photo, etc. La figure (2.4) montre aussi comment l'inpainting peut être utilisée pour effacer un texte dans une image. Lors de l'acquisition ou de la transmission d'une image, certaines parties de l'image peuvent disparaître ou altérées (**Fig.2.5**), il est alors utile de procéder à leurs reconstruction par des méthodes d'inpainting.

L'inpainting peut être également appliqué pour zoomer (agrandir) une image. Dans cette application, de nouveaux pixels sont incorporés entre les pixels de l'image originale par sous échantillonnage. Ces nouveaux pixels sont considérés comme des artefacts et leurs niveaux de gris ou couleurs peuvent être alors déterminés par des techniques d'inpainting (**Fig. 2.6**). Une application intéressante consiste à colorer une image en niveaux de gris en spécifiant la couleur de certaines régions de l'image (**Fig. 2.7**). L'inpainting peut être également utilisé pour reconstruire, un visage ou une empreinte digitale dont certaines parties sont manquantes [18]. Il peut également servir pour protéger des informations confidentielles des personnes présentes dans une image ou vidéo [33]



Fig. 2.2 : Suppression d'une tache sur une photographie.



Fig. 2.3 : Suppression des objets dans une image



Fig.2.4 : Suppression d'un texte dans une image



Fig.2.5 : Restauration et amélioration d'une image



Fig. 2.6 : Zoomer une image.

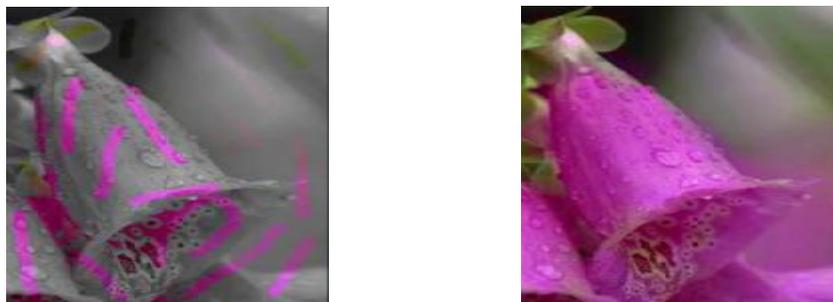


Fig. 2.7 : Coloriage d'une image

2.5 Propriétés de l'inpainting

L'opération d'inpainting doit être :

- Locale : l'information utilisée pour interpoler un domaine D doit être disponible dans le voisinage de ce domaine.
- Fonctionnelle : l'information doit être prélevée de l'image seulement (information bas niveau). Aucune information de haut niveau ne doit être incorporée.
- Automatique : L'intervention humaine doit être réduite le plus possible. Plus cette intervention est réduite, plus un algorithme d'inpainting devient puissant.
- Stable : Insensible à des dégradations mineurs comme le bruit ou le flou qui peuvent apparaître dans les zones où l'information est disponible.
- Générique : L'inpainting doit pouvoir traiter tout type d'image avec une précision satisfaisante.

Notons qu'un algorithme ou méthode d'inpainting est dit d'ordre k si :

$$\|u_D - u_{0/D}\|_{\infty} = O(d^{k+1}) \quad (2.1)$$

Il est linéaire si $k=1$:

$$\|u_D - u_{0/D}\|_{\infty} = O(d^2) \quad (2.2)$$

Avec d est le diamètre de domaine d'inpainting D tend vers 0 .

Deux facteurs influent généralement sur l'inpainting :

- Complexité du domaine à interpoler : ce domaine peut être de forme ou de taille quelconque. Il dépend en générale de l'application envisagée. Ce domaine est connu a priori. Il est fourni par l'utilisateur sous forme d'un masque ou obtenu d'une manière automatique ou semi-automatique.

- Complexité de l'image : l'image peut être composée de plusieurs régions qui peuvent être de texture différentes ou contenir des coins, des jonctions, etc. L'inpainting idéale doit tenir compte de ces caractéristiques géométriques ou de l'inhomogénéité des paramètres de texture.

2.6 Méthodes d'inpainting

Il existe une multitude d'approches d'inpainting qu'on peut classer en cinq catégories :

- Méthodes basées sur l'interpolation.
- Méthodes basées sur les EDP.
- Méthodes basées sur la recherche des similarités.
- Méthodes hybrides.

2.7 Méthodes basées sur l'interpolation

Certaines méthodes d'interpolation comme celle basée sur les fonctions radiales de base ont été utilisées pour résoudre le problème d'inpainting [34]. Cette méthode consiste à chercher le niveau de gris d'un pixel situé dans le domaine D à partir des niveaux de gris des pixels situés dans la région Ω/D .

Soient $(X_i)(i=1,\dots,n)$ l'ensemble des pixels de la région Ω/D tels que leurs niveaux de gris $f(X_i) = h_i$ sont connus. $X_i = (x_i, y_i)^T$ représentent les coordonnées des pixels dont le niveau de gris ou couleur est connu. Le niveau de gris d'un pixel de coordonnées $X = (x, y)^T$ appartenant au domaine à reconstruire D peut être déterminé par :

$$\tilde{f}(X) = \sum_{i=1}^n I_i \Phi(\|X - X_i\|) + P(X) \quad (2.3)$$

Où I_i sont des coefficients à déterminer, $\Phi(\|X - X_i\|)$ est une fonction radiale, $\|X - X_i\|$ est la distance, généralement euclidienne, entre les deux pixels de coordonnées X et X_i . $P(X)$ est un polynôme de degré m dépendant de la fonction radiale Φ . $P(x) = a_0x + a_1y + a_2x^2 + a_3$.

La fonction radiale peut avoir plusieurs formes comme par exemple :

$$\text{Gaussien : } f(x) = e^{-ax^2}, a \in R$$

$$\text{Quadratique inverse: } f(x) = \frac{1}{1 + a^2x^2},$$

Multiquadratique : $f(x) = \sqrt{1 + a^2 x^2}$,

Spline : $f(x) = x^2 \ln x$.

Biharmonic : $f(x) = |x|^2$,

Triharmonic : $f(x) = |x|^3$.

Pour déterminer $f(x)$ à partir de l'équation (2.3), il faut déterminer auparavant les coefficients I_i et les $(m + 1)$ paramètres du polynôme $P(X)$. Pour cela, on utilisera l'hypothèse des points d'appui suivante :

$$\tilde{f}(x_j) = f(x_j) = h_j \quad \forall j = 1, \dots, n. \tag{2.4}$$

Soit $P(X) = a_0 x + a_1 y + a_2$, alors

$$\tilde{f}(x_j) = \sum_{i=1}^n I_i \Phi(\|x_j - x_i\|) + P(x_j) \quad j = 1, \dots, n \tag{2.5}$$

Ces équations forment le système d'équations linéaires suivant :

$$\begin{bmatrix} A_{11} & A_{12} & & A_{1n} & x_1 & y_1 & 1 \\ A_{21} & A_{22} & & A_{2n} & x_2 & y_2 & 1 \\ \mathbf{M} & \mathbf{M} & \mathbf{M} & \mathbf{M} & \mathbf{M} & \mathbf{M} & \mathbf{M} \\ A_{n1} & A_{n2} & \mathbf{K} & A_{nn} & x_n & y_n & 0 \\ x_1 & x_2 & \mathbf{K} & x_n & 0 & 0 & 0 \\ y_1 & y_2 & \mathbf{K} & y_n & 0 & 0 & 0 \\ 1 & 1 & \mathbf{K} & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ \\ I_n \\ a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \\ \\ h_n \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.6}$$

Où $A_{ij} = \Phi(\|x_j - x_i\|)$.

Ce système peut se mettre sous la forme matricielle suivante :

$$\begin{bmatrix} A & \mathbf{M} & P \\ \mathbf{K} & \mathbf{M} & \mathbf{K} \\ P^T & \mathbf{M} & 0 \end{bmatrix} \begin{bmatrix} I \\ \mathbf{M} \\ a \end{bmatrix} = \begin{bmatrix} h \\ \mathbf{K} \\ 0 \end{bmatrix} \tag{2.7}$$

La résolution de ce système par l'une des méthodes de résolution des systèmes d'équations linéaires nous fournira les valeurs des coefficients I_i et les coefficients a_i du polynôme $P(X)$.

2.8 Méthodes basées sur les EDPs

Cette approche regroupe un grand nombre de méthodes qui sont récentes que celles basées sur l'interpolation polynomiale.

2.8.1 Approche de Bertalmio, Sapiro, Casselles et Ballester

Dans [17], Bertalmio et al, ont été les premiers à présenter un algorithme pour l'inpainting numérique « digital inpainting ». Ils se sont inspirés des artistes lorsqu'ils restaurent les tableaux de peinture en repeignant les parties dégradées. C'est donc une méthode de reconstruction de l'image à partir des informations manquantes situées sur le bord de la zone à reconstruire. Ils définissent ainsi l'inpainting comme une propagation de l'information L présenté sur le contour ∂D du domaine D suivant la direction N des isophotes pour préserver au mieux les lignes de contours. Notons qu'un isophote correspond aux lignes de même niveau de gris.

L'algorithme proposé consiste à propager l'information L de la frontière à l'intérieur de D . Pour cela, on définit une famille d'images $u(i, j, n)$ où n représente le paramètre de temps tel que $u(i, j, 0) = u_0(i, j)$ et $\lim_{n \rightarrow +\infty} u(i, j, n) = u_R(i, j)$. $u_R(i, j)$ étant l'image résultante ou l'image inpaintée. L'évolution de l'algorithme est régit par l'équation suivante :

$$u^{n+1}(i, j) = u^n(i, j) + \Delta t u_t^n(i, j) \quad \forall (i, j) \in D \quad (2.8)$$

$u_t^n(i, j)$ dénote le changement d'information à l'intérieur du domaine D et Δt représente le taux de changement ou le taux d'amélioration. Pour propager l'information de l'extérieur de D vers l'intérieur, $u_t^n(i, j)$ est défini par :

$$u_t^n(i, j) = \vec{d} L^n(i, j) \cdot \vec{N}^n(i, j). \quad (2.9)$$

$\vec{N}(i, j)$ représente la direction de propagation, $\vec{L}^n(i, j)$ est l'information à propager du contour ∂D vers le domaine D , elle est donnée par le Laplacien de l'image $u^n(i, j)$

$$\vec{L}^n(i, j) = \Delta u^n(i, j) = u_{xx}^n(i, j) + u_{yy}^n(i, j). \quad (2.10)$$

$\vec{d} L^n$ est la variation du Laplacien, autrement dit, la variation de la régularité de l'image. Elle est définie comme suit :

$$\vec{d} L^n(i, j) = (L^n(i+1, j) - L^n(i-1, j), L^n(i, j+1) - L^n(i, j-1)). \quad (2.11)$$

$$\vec{d} L(i, j) = \nabla L^n(i, j) \quad (2.12)$$

La direction \vec{N} de propagation est estimée par la direction des isophotes qui est la normale à la direction de gradient de l'image :

$$\vec{N}^n(i, j) = (\nabla u(i, j))^\perp \quad (2.13)$$

$$\vec{N}^n(i, j) = \frac{(C_y^n(i, j), C_x^n(i, j))}{\sqrt{C_x^n(i, j)^2 + C_y^n(i, j)^2 + e}} \quad (2.14)$$

C_x^n et C_y^n sont respectivement les composantes en x et en y du gradient de l'image u^n , et e est une valeur fixe et négligeable qui permet de ne pas avoir la division par zéro dans le cas où la norme est nulle.

Autrement dit, elle représente la direction normale aux contours comme le montre la figure (Fig. 2.8) où la zone en blanc représente la région à inpainter.

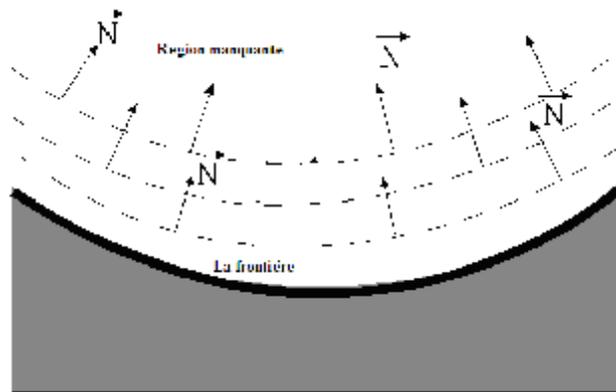


Fig. 2.8 : Choix de la direction de propagation.

2.8.2 Approche basée sur la variation totale

Chan et Shen ont proposé une approche pour l'inpainting basée sur la restauration d'image par minimisation de variation totale (TV) dans le domaine à restaurer [18, 19, 20]. Elle est connue sous le nom de TV inpainting. Cette approche variationnelle consiste à minimiser l'énergie suivante :

$$J(u) = \int_{\Omega/D} |\nabla u| dx + \frac{I}{2} \int_D (u - u_0)^2 dx. \quad (2.16)$$

Le premier terme de cette énergie correspond à la variation totale de l'image dans le domaine D/Ω telle que définie dans le premier chapitre alors que le deuxième terme définit l'adéquation entre l'image recherchée u et l'image initiale u_0 à l'intérieur du domaine D .

I est une constante multiplicative, appelée multiplicateur de Lagrange.

La minimisation du critère J est équivalente à la résolution de l'EDP correspondant à l'équation d'Euler Lagrange suivante :

$$\frac{\partial u}{\partial t} = \nabla \cdot \left(\frac{1}{|\nabla u|} \cdot \nabla u \right) + Ie(u_0 - u). \quad (2.17)$$

Où $Ie = I \cdot c_{\Omega/D}$ est un multiplicateur de Lagrange étendu, $c_{\Omega/D}$ est la fonction caractéristique (ou bien le masque) du domaine d'inpainting.

$$c_{\Omega/D} = \begin{cases} 1 & \text{si } x \in \Omega/D, \\ 0 & \text{si } x \in D. \end{cases} \quad (2.18)$$

L'équation (2.17) est valable sur l'intégralité du domaine Ω . Si on considère juste son premier terme (valable dans le domaine D), cette équation peut être simplifiée à l'EDP de diffusion anisotrope suivante:

$$\frac{\partial u}{\partial t} = \nabla \cdot \left(\frac{1}{|\nabla u|} \cdot \nabla u \right). \quad (2.19)$$

Cette équation montre qu'un flux de chaleur purement anisotrope orienté le long de courbure de l'image est appliqué à l'intérieur du domaine d'inpainting D .

Parmi les avantages principaux de l'équation TV inpainting comparée à tous les autres modèles définis par une EDP, et qu'elle est la seule équation du second ordre, son schéma numérique est simple et converge rapidement. Cette approche est employée pour reconstruire des petites régions, car elle ne peut pas prolonger les isophotes lorsque la zone à reconstruire est grande.

2.8.3 Méthode de Chan et Shen

En 2001, Chan et Shen [21] ont proposé un nouveau modèle d'inpainting basé sur la diffusion conduite par la courbure (CCD inpainting) en s'inspirant du modèle d'inpainting à variation totale. L'inconvénient majeur du modèle TV d'inpainting est qu'il ne suit pas le principe de connectivité pour les objets simples qui sont en grande partie déconnectés par le domaine d'inpainting (Fig. 2.9). Ceci a poussé les auteurs à ajouter l'information de la courbure des isophotes dans l'équation de diffusion. Dans ce nouveau modèle, le terme de diffusivité $D = 1/|\nabla u|$ est remplacé par un autre terme dépendant de la courbure k . Ce terme, également nommé force de diffusion, est défini par $D' = g(k)/|\nabla u|$.

Le modèle d'inpainting par CCD s'exprime alors par l'EDP suivante :

$$\frac{\partial u}{\partial t} = \nabla \cdot \left(\frac{g(k)}{|\nabla u|} \cdot \nabla u \right) + Ie(u_0 - u). \quad (2.20)$$

Avec $k = \nabla \cdot \left[\frac{\nabla u}{|\nabla u|} \right]$ et $g(k)$ est une fonction qui permet de pénaliser les grandes courbures et encourager les petites. $g(k)$ doit satisfaire la condition suivante :

$$g(k) = \begin{cases} 0, & \text{si } k = 0 \\ +\infty, & \text{si } k = +\infty \end{cases} \quad (2.21)$$

Ce choix permet une forte diffusion lorsque k est élevé, ce qui a pour effet de relier même les objets en grande partie interrompus.

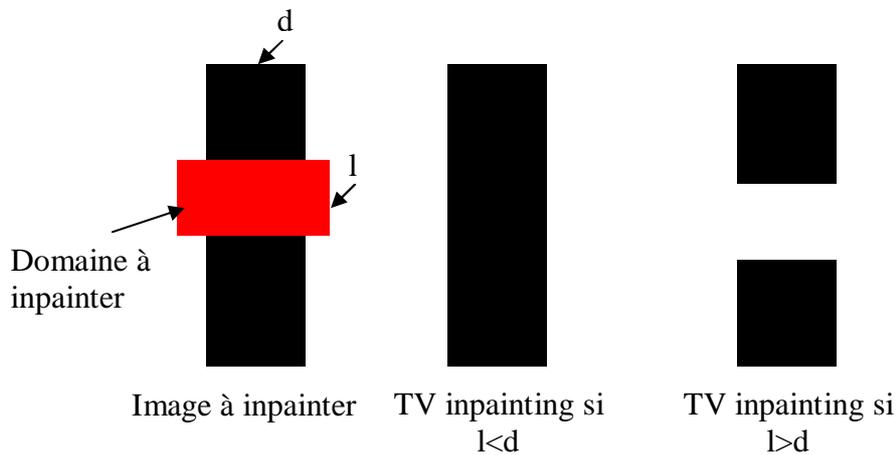


Fig. 2.9 : Limite d'une reconstruction par la variation totale.

2.8.4 Approche de Telea

L'algorithme, proposé par Telea en 2004 [24], consiste à propager l'image lissée le long du gradient d'image. L'image lissée est estimée par la moyenne pondérée de tous les pixels voisins connus du pixel à inpainter. Afin de propager cette information, l'auteur utilise la méthode Fast Marching (FMM) initialement utilisée pour résoudre rapidement le problème de la segmentation par la méthode des level sets [32].

Pour mieux expliquer cet algorithme, on considère l'exemple de la figure (2.10), dans laquelle, on doit chercher l'information au point p situé à la frontière ∂D de la région à inpainter D . On définit un petit voisinage $B_e(p)$ autour de p de la taille e des pixels de l'image dont les niveaux de gris sont connus: $B_e(p) = \{q, q \in D\}$. Pour e assez petit, on considère une première approximation d'ordre un de l'image $u_q(p)$ au point p en fonction des valeurs d'image $u(q)$ et du gradient $\nabla u(q)$ au point q .

$$u_q(p) = u(q) + \nabla u(q)(p - q). \quad (2.22)$$

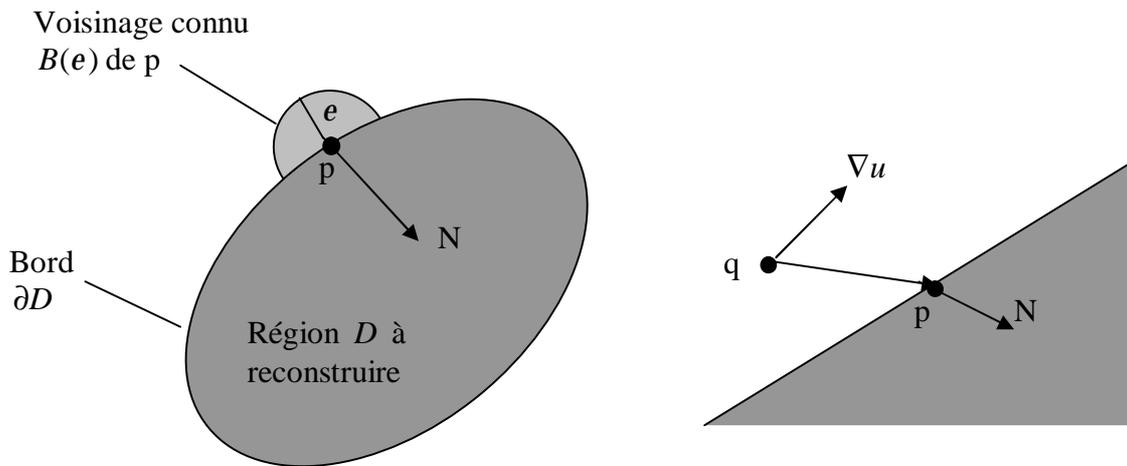


Fig. 2.10 : Principe de l'inpainting par la méthode de Telea

Ainsi, le niveau de gris au point p est remplacé par une fonction de tous les points q de $B_e(p)$ en additionnant les estimations de tous les points q , pondérés par une fonction de poids $w(p, q)$.

$$u(p) = \frac{\sum_{q \in B_e(p)} w(p, q) [u(q) + \nabla u(q)(p - q)]}{\sum_{q \in B_e(p)} w(p, q)}. \quad (2.23)$$

Cette opération est répétée pour chaque point p appartenant à ∂D jusqu'à ce que le domaine D soit complètement inpainté. Ceci permet de déplacer à chaque itération la frontière ∂D dans D . A chaque itération le point p de ∂D à interpoler est choisi selon sa distance à la frontière initiale ∂D_i (position initiale de la frontière de D) en utilisant l'algorithme FMM. Ce dernier consiste à résoudre l'équation d'Eikonale [32] :

$$\|\nabla T\| = 1 \text{ dans } D \quad T = 0 \text{ dans } \partial D. \quad (2.24)$$

La solution T de l'équation (2.24) est la carte de distance des pixels de D à la frontière ∂D . L'algorithme FMM garantit que les pixels de ∂D sont traités par ordre croissant de leur distance à la frontière T , c.à.d. inpainter d'abord le pixel le plus proche au secteur connu de l'image.

La fonction de pondération $w(p, q)$ se calcule comme suit :

$$w(p, q) = \text{dir}(p, q) \cdot \text{dst}(p, q) \cdot \text{lev}(p, q). \quad (2.25)$$

Où $dir = \frac{p-q}{\|p-q\|} N(p)$:est la composante directionnelle, $dest = \frac{d_0^2}{\|p-q\|^2}$ est la composante de distance, $lev(p,q) = \frac{T_0}{1+|T(p)-T(q)|}$ est la composante distance des ensembles de niveau (p,q) . T_0 et d_0 sont des constantes fixées à 1.

Cet algorithme est considérablement rapide et simple à implémenter, cependant il introduit une tache floue quand la région à inpainter qui contient plus de 10 pixels [24].

2.9.1 Méthodes basées sur la recherche de similarités

Proposée en 2004 par Criminisi et al [26], cette méthode, également appelée ‘Région filing and Object removal by Exemplar-based Image Inpainting’, consiste à rechercher de blocs similaires dans l’image initiale, puis les copier localement dans la zone à inpainter en allant de l’extérieur du domaine D vers l’intérieur de D avec une priorité de remplissage dépendante de la structure de l’image.

Le succès de cette méthode dépend de l’ordre du choix des pixels se trouvant sur le bord de la région détériorée. L’algorithme consiste à associer à chaque pixel p du bord de la région détériorée un patch Ψ_p centré sur ce pixel puis à déterminer sa priorité à partir de deux termes, un terme dit de confiance et un terme de données (data term). Le patch correspond à un carré de taille fixe et centré sur un pixel. La figure (2.11) illustre ce principe.

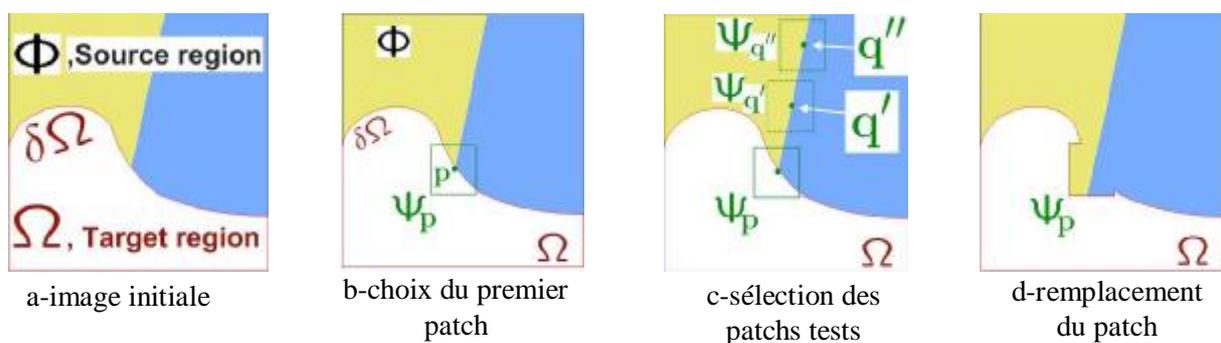


Fig. 2.11 : Étapes de remplissage du patch, ces schémas sont tirés de l’article [11]

Pour une meilleure compréhension de la méthode, nous allons le décrire point par point son déroulement.

Soient u l'image source, $\Omega = D$ la partie manquante ou la région à remplir et $\Phi = u - \Omega$ l'image entière sans la partie manquante ou région source. Soit $\partial\Omega = \partial D$ la frontière entre Φ et Ω .

Chaque pixel $p \in \partial\Omega$, sera caractérisé par :

- Son niveau de gris ou sa couleur en RGB.
- une valeur de confiance.
- une valeur temporaire de priorité.

Etape1 : Définir une priorité :

Soit Ψ_p le patch à remplir centré au point $p \in \partial\Omega$ (voir Fig. 2.11), on définit sa priorité $P(p)$ comme étant un produit de deux termes :

$$P(p) = C(p) * D(p) \tag{2.26}$$

$C(p)$ définit le terme de confiance. Il correspond au nombre de pixels non détériorés se trouvant dans le patch ou bien au nombre de pixels précédemment corrigés.

$$C(p) = \frac{\sum_{\tilde{p} \in (\Psi_p \cap \Phi)} C(\tilde{p})}{|\Psi_p|} \tag{2.27}$$

$|\Psi_p|$ est l'aire de Ψ_p , Initialisation $C(p) = \begin{cases} 1 & \forall p \in \Phi \\ 0 & \forall p \in \Omega \end{cases}$

$D(p)$ correspond à donnée inertielle ou bien au terme de donnée de p ,

$$D(p) = \frac{|\nabla u_p^\perp \cdot n_p|}{a} \tag{2.28}$$

a est un facteur de normalisation égal à 255 pour une image en niveaux de gris, n_p est le vecteur unitaire orthogonal à la surface $\partial\Omega$, et ∇u_p^\perp correspond à la valeur maximum du gradient image dans la fenêtre $\Psi_p \cap u$.

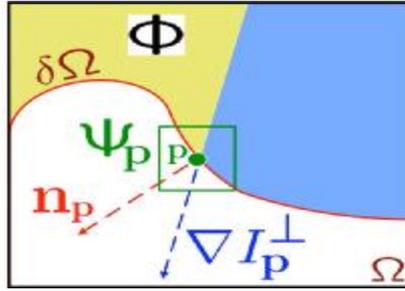


Fig. 2.12 : Schéma du calcul de valeur inertielle, cette figure est tirée de l'article [11]

La priorité produit un équilibre sensible entre le terme de confiance et le terme de données. Le terme de données tend à pousser des isophotes rapidement vers l'intérieur de la zone à inpainter, alors que le terme de confiance tend à supprimer avec précision cette sorte d'irruption dans la région à traiter. A chaque étape le pixel \tilde{p} ayant une priorité élevée est alors choisi pour être reconstruit.

Etape2 : Propagation de l'information de texture et de structure

Elle consiste à propager la texture d'image en la prélevant directement de la région de source. Pour cela, on cherche le meilleur patch Ψ_q proche de Ψ_p

$$\Psi_{\tilde{q}} = \arg \min_{\Psi_q \in \Phi} d(\Psi_{\tilde{p}}, \Psi_q) \quad (2.28)$$

La distance d entre deux patches est le plus souvent la somme des différences au carré.

Une fois Ψ_q trouvé il ne reste plus qu'à remplir les pixels $p \in \Psi_p \cap \Omega$ avec ceux de Ψ_q .

Etape3 : Mise à jour des valeurs de confiance

Après la propagation de l'information dans la zone à interpoler, une mise à jour des coefficients de confiance des points inpaintés est effectuée :

$$C(p) = C(\hat{p}), \quad \forall p \in \Psi_{\tilde{q}} \cap \Omega. \quad (2.29)$$

Les étapes 1 à 3 sont réitérées jusqu'à ce que toute la zone Ω soit remplie.

Cette méthode est rapide, elle a donné de bons résultats pour les exemples présentés, elle permet de reconstruire les textures, mais elle a comme inconvénient de ne pas construire la géométrie globale de la zone à inpainter (reconstruction difficile) [26].

2.10 Méthodes hybrides

Dans [27] Tschumperlé et al, ont proposé un algorithme qui combine la diffusion anisotrope et la synthèse de textures. Cet algorithme permet à la fois de bénéficier de la reconstruction d'une géométrie globale mais aussi de reconstruire les détails (synthétiser la texture à l'intérieur de ces régions). Cet algorithme fonctionne en deux phases : Une phase de reconstruction qui utilise une EDP de lissage anisotrope par EDP multivaluée avec une contrainte de courbure qui diffuse les couleurs (intensités) des pixels connus de l'extérieur vers l'intérieur des régions à inpainter et une phase de synthèse des textures à l'intérieur des régions des données manquantes. Cette phase se base sur les techniques de synthèse de texture utilisant des correspondances de bloc d'image. Des blocs similaires dans l'image sont déterminés, puis copiés localement dans la zone à inpainter de l'extérieur vers l'intérieur.

Phase1 : Reconstruction:

Soit $u : \Omega \rightarrow R^3$ une image couleur, $v : \Omega \rightarrow R^2$ un champ de vecteurs à deux dimensions. Le principe de cette méthode consiste à lisser localement u suivant v en utilisant une diffusion anisotrope avec préservation des courbures selon l'EDP suivante :

$$\frac{\partial u_i}{\partial t} = \text{trace}(v v^T H_i) + \nabla u_i^T J_v v \quad (2.30)$$

v Correspond à l'estimation de la direction de l'isophote en chaque point. Il est calculé comme le vecteur propre correspondant à la valeur propre la plus faible du tenseur de

$$\text{structure } G_s = \left(\sum_{k=0}^3 \nabla u_k \nabla u_k^T \right) * g_s, \text{ tel que } g_s = \frac{1}{2ps^2} \exp\left(-\left(\frac{x^2 + y^2}{2s^2}\right)\right).$$

J_v représente la matrice jacobéenne de v (matrice des dérivées premières).

Au fur et à mesure des itérations de l'EDP (2.30), les pixels voisins aux régions manquante sont diffusé à l'intérieur de ces régions jusqu'à les remplir complètement. Ceci est fait en chaque point dans la direction des isophotes u , qui est elle-même ré-évaluée itération par itération. Notons que l'équation (2.30) ne modifie l'image u que dans les régions à inpainter.

Le terme de contrainte $(\nabla u_i^T J_v v)$ dans (2.30) permet la préservation des courbures.

Phase2 : synthèse des textures

Dans cette phase, un algorithme de synthèse de texture est appliqué. Cet algorithme cherche des blocs similaires dans l'image, puis copie locale dans la zone à inpainter de l'extérieur vers l'intérieur.

2.9 Conclusion

Nous avons exposé dans ce chapitre quelques notions élémentaires sur l'inpainting et nous avons effectué un tour d'horizon sur les différentes méthodes d'inpainting. On en déduit que l'inpainting est une opération qui peut être utilisée dans plusieurs applications et que les moyens de sa réalisation sont très diversifiés. Principalement, deux approches se distinguent : celle qui est basée sur l'utilisation des méthodes mathématiques d'interpolation des fonctions et celle qui est basée sur des équations différentielles partielles. Dans la première approche, on fait généralement appel à l'interpolation polynomiale comme les splines ou les RBF. En ce qui concerne la deuxième approche, la plus part des méthodes développées cherchent à mimer la restauration manuelle des tableaux et sont fondées sur la préservation ou la continuation des lignes isophotes à l'intérieur de la zone à reconstruire.

3.1 Introduction

On présentera dans ce paragraphe l'application des méthodes d'interpolation des fonctions pour résoudre le problème de l'inpainting.

Comme méthodes d'interpolation, on s'est intéressé à l'interpolation par les méthodes du plus proche voisin linéaires et splines. Ces méthodes seront utilisées selon deux stratégies : la première considère l'image comme une fonction monovariante tandis que dans la deuxième, l'image est considérée comme une fonction à deux variables. On utilisera ainsi l'interpolation monodimensionnelle dans le premier cas et l'interpolation à 2D dans le deuxième cas.

3.2 Principe de l'inpainting par interpolation

On suppose que le domaine D ou l'information est manquante contenant les pixels à interpoler est connu a priori. Le problème de l'inpainting est de déterminer les niveaux de gris des pixels du domaine D en fonction des niveaux de gris connus des autres pixels de l'image. Pour effectuer de l'inpainting à base des méthodes d'interpolation, nous proposons la démarche suivante :

On parcourt l'image pixel par pixel et dès qu'un pixel appartenant au domaine D est rencontré, on consulte alors son voisinage défini par une fenêtre de voisinage carrée de taille $(2w+1)^2$ centrée sur ce pixel afin de retrouver l'information valide autour de D ainsi qu'à sa frontière ∂D . Tous les pixels voisins qui n'appartiennent pas au domaine D et dont le niveau de gris est connu, sont alors considérés comme des points d'appuis ou d'interpolation. On applique alors une des méthodes d'interpolation (PPV, linéaire, splines) au pixel considéré en utilisant les points d'appuis afin de déterminer son information manquante, c'est-à-dire son niveau de gris. Ce pixel ne fera alors plus partie du domaine D . On continue le balayage à la recherche des pixels de domaine D et à leurs interpolations jusqu'à ce que ce domaine D devient vide. Les étapes de cette approche proposée pour l'inpainting par l'interpolation sont résumées par l'algorithme suivant :

Pour chaque pixel du domaine D

1. On consulte son voisinage défini par une fenêtre de taille $(2w+1)^2$.
2. Tous les pixels de son voisinage qui n'appartiennent pas au domaine D sont considérés comme des points d'interpolation.
3. Appliquer une méthode d'interpolation au pixel courant à partir des points d'appuis.

Notons qu'en pratique le domaine D est représentés par des point ayant la valeur 0 d'un masque ou image binaire de même dimension que l'image originale ou à inpainter. Les autres points du masque ont la valeur 1.

La figure (3.1) illustre le principe de cette approche.

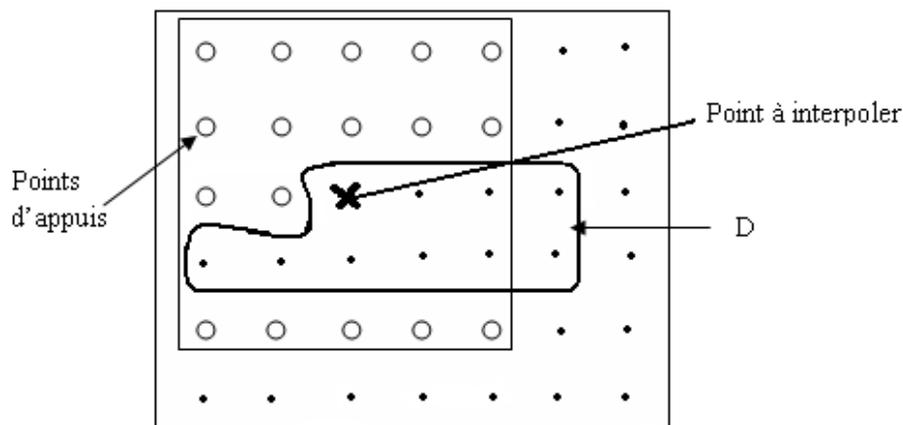


Fig.3.1 : Principe de l'inpainting par interpolation

Nous avons choisi trois méthodes d'interpolations PPV, linéaire et splines. Ces trois méthodes peuvent être appliquées dans le cas monovarié ou en deux dimensions. Dans le deuxième cas, les points d'appuis sont représentés par le triplet $(x, y, u(x, y))$ ou (x, y) et $u(x, y)$ représentent les coordonnées du pixel correspondant et $u(x, y)$ son niveau de gris, et dans le premier cas, les points d'appuis sont représentés par le doublet $(s, u(s))$ ou s représente l'abscisse curviligne (scalaire) et $u(s)$ son niveau de gris. Cette représentation est obtenue en transformant l'image 2D en un vecteur.

3.3 Inpainting par des méthodes d'interpolation monodimensionnelles

Il est possible d'utiliser les méthodes d'interpolation des fonctions monovariées dans l'étape trois de l'algorithme. Pour cela, il est indispensable de transformer la fonction image à deux variables spatiales (x, y) en une seule variable en utilisant les courbes de remplissage.

3.3.1 Courbes de remplissage de l'espace

La transformation d'une image bidimensionnelle en une représentation vectorielle (monodimensionnelle) se fait généralement en concaténant les lignes ou les colonnes (Raster ou vidéo). Cependant, d'autres manières d'obtenir cette transformation est d'utiliser la notion

de courbes de remplissage d'espace SFC (Space Filling Curves) introduites par Giuseppe Peano en 1890.

Une courbe qui remplit l'espace est une courbe qui parcourt un espace multidimensionnel en passant, de point en point, et d'une manière unique par tous les points de cet espace. Les SFC sont souvent utilisées pour définir le sens d'un parcours d'une image. Soit $u(x, y)$ la fonction image représentant le niveau de gris au point de coordonnées $\mathbf{r}_v = (x, y)^T$. On définit une courbe de parcours $s = C(v)$ qui peut être considérée comme l'abscisse curviligne le long de la courbe. La figure (3.2) montre sur un exemple comment l'image peut être représentée par une fonction monodimensionnelle.

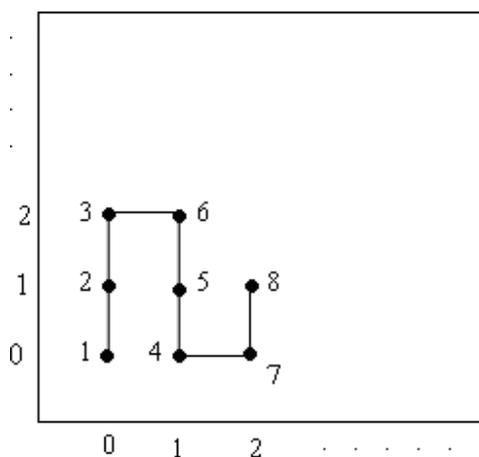


Fig.3.2 : Représentation d'une image par une fonction monodimensionnelle.

$$\mathbf{r}_v \begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightarrow s = C(\mathbf{r}_v) = 1, \mathbf{r}_v \begin{bmatrix} 0 \\ 1 \end{bmatrix} \rightarrow s = C(\mathbf{r}_v) = 2, \dots, \mathbf{r}_v \begin{bmatrix} 2 \\ 1 \end{bmatrix} \rightarrow s = C(\mathbf{r}_v) = 8.$$

Il existe plusieurs courbes de remplissage

- **Courbe de Zigzag scan**

Cette courbe est utilisée en compression d'images JPEG, elle respecte l'ordre canonique, mais des points voisins dans l'espace des composantes peuvent correspondre à des points très éloignés sur le parcours de la courbe.

- **Courbe de Regazzoni**

La structure en couches de cette courbe traduit la volonté de construire un ordre proche de celui que donnerait une distance. Cette courbe est utilisée beaucoup plus en filtrage.

- **Balayage Vidéo (lexicographique)**

On constate la très forte dissymétrie de cette courbe qui traduit le déséquilibre et la non préservation de la notion de voisinage.

- **Courbe entrelacement de bits**

Cette courbe se construit de manière récursive, ce qui lui confère une structure auto similaire. Notons que cette courbe est également utilisée dans l'espace 2D de l'image, notamment pour la construction quadtree lors de la décomposition d'une image en structure hiérarchique.

- **Courbe de Hilbert et de Peano**

Les courbes de Hilbert (souvent attribuées à Peano) et de Peano sont des SFC abondamment étudiées et souvent utilisées en traitement d'image pour permettre un parcours efficace du plan image.

Ces courbes présentent toutes les deux une propriété essentielle: le passage d'un point de la courbe à son voisin (sur la courbe) ne s'accompagne jamais d'un déplacement important dans l'espace des composantes, d'où la préservation de la topologie et d'information spatiale.

La figure ci-dessous regroupe les formes géométriques des différentes courbes de remplissage d'espace.

Les différentes représentations scalaires (une seule variable) s'accompagnent inévitablement de distorsions topologiques, c'est-à-dire la notion de voisinage est modifiée. En effet deux points proches dans l'espace initial peuvent se retrouver éloignés dans leur numérotation le long de la courbe. Cela risque d'introduire du bruit. Pour évaluer quantitativement la distorsion introduite par chaque type de balayage, un critère basé sur la notion de voisinage a été proposé [35]. Ce critère est défini comme suit: étant donné les 8 voisinages d'un vecteur, on calcule le pourcentage de ces voisins qui se retrouvent encore parmi les 8 voisins du pixel après codage sur la courbe, comme le montre la figure (3.4).

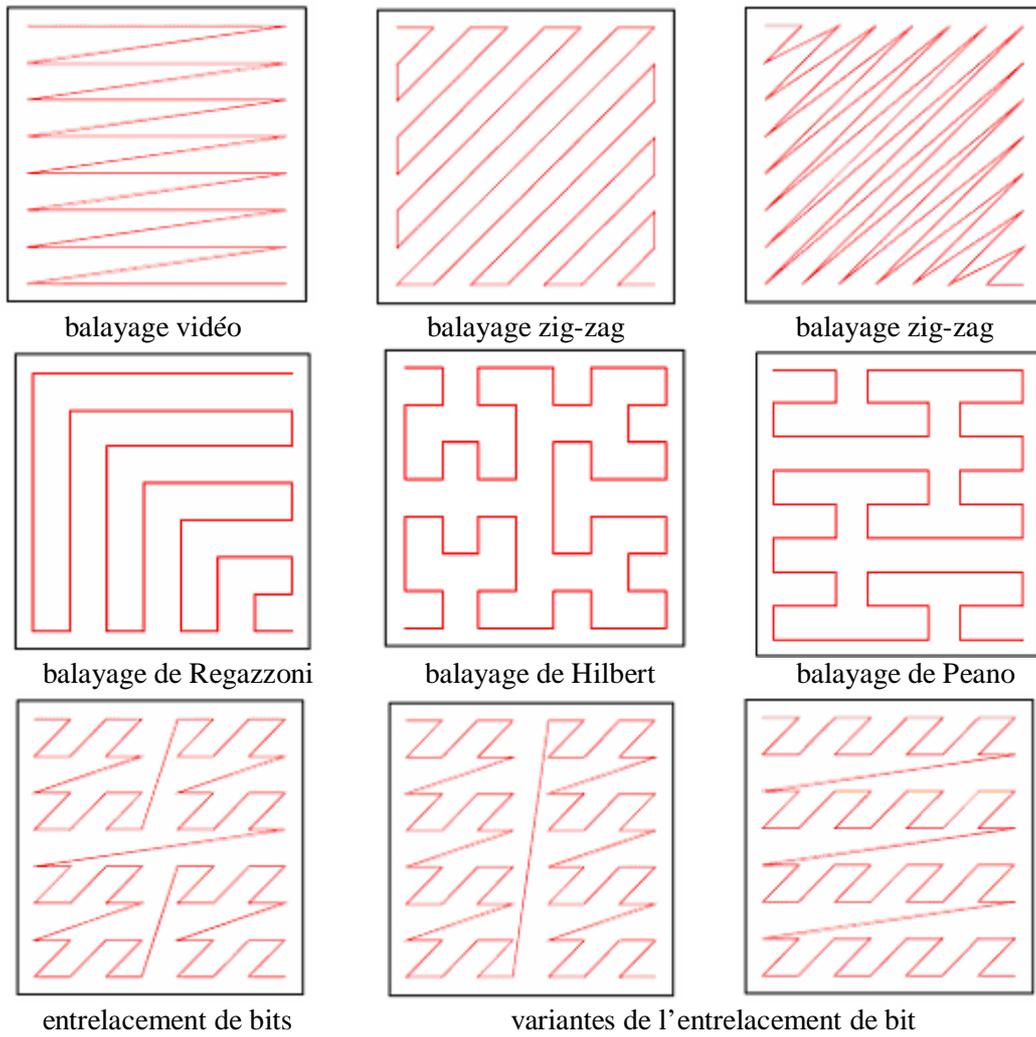


Fig.3.3 : Représentation géométrique des SFC

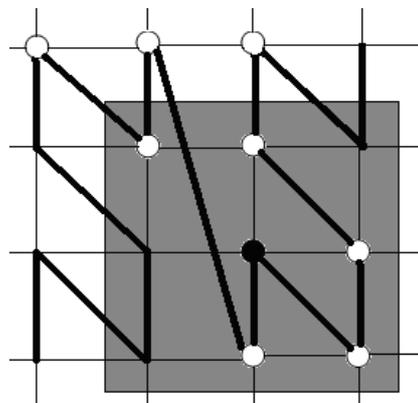


Fig.3.4 : Critère de distorsion.

Le résultat final de cette étude est regroupé dans le tableau suivant:

Courbe	Taux de préservation du voisinage
Hilbert	58%
Peano	57%
Entrelacement de bits	47%
Zigzag scan	27%
Regazzoni	26%
Lexicographique (vidéo)	25%

Tableau 3.1: Taux de préservation topologique

Ce tableau montre clairement que la courbe de Hilbert est celle qui préserve le plus le voisinage d'un pixel. Nous avons ainsi choisi ce type de balayage pour transformer l'image 2D en une fonction monovariante. Le balayage Vidéo est également utilisé pour des raisons de comparaison.

3.4 Test et Résultats

Nous allons présenter dans cette section les résultats de l'inpainting obtenues par les différentes méthodes d'interpolation. Pour évaluer les performances de chaque méthode d'inpainting, nous avons déterminé l'erreur globale moyenne entre les niveaux de gris reconstruits et les niveaux de gris réels [36].

$$E = \frac{1}{N} \sum_{(x_1, x_2)} |u(x, y) - u_r(x, y)|^k$$

$u_r(x, y)$ représente le niveau de gris reconstruit et N représente le nombre de pixels. Pour $k=1$, l'erreur est dite linéaire et pour $k=2$, on parle de l'erreur quadratique. C'est cette deuxième que nous avons utilisée.

Les cinq images tests (charrette, Lena, perroquet, ballon dans le ciel, bébé) de la figure (3.3.), connues dans le domaine de l'inpainting pour tester les algorithmes d'inpainting, ont été choisies pour mener notre évaluation.

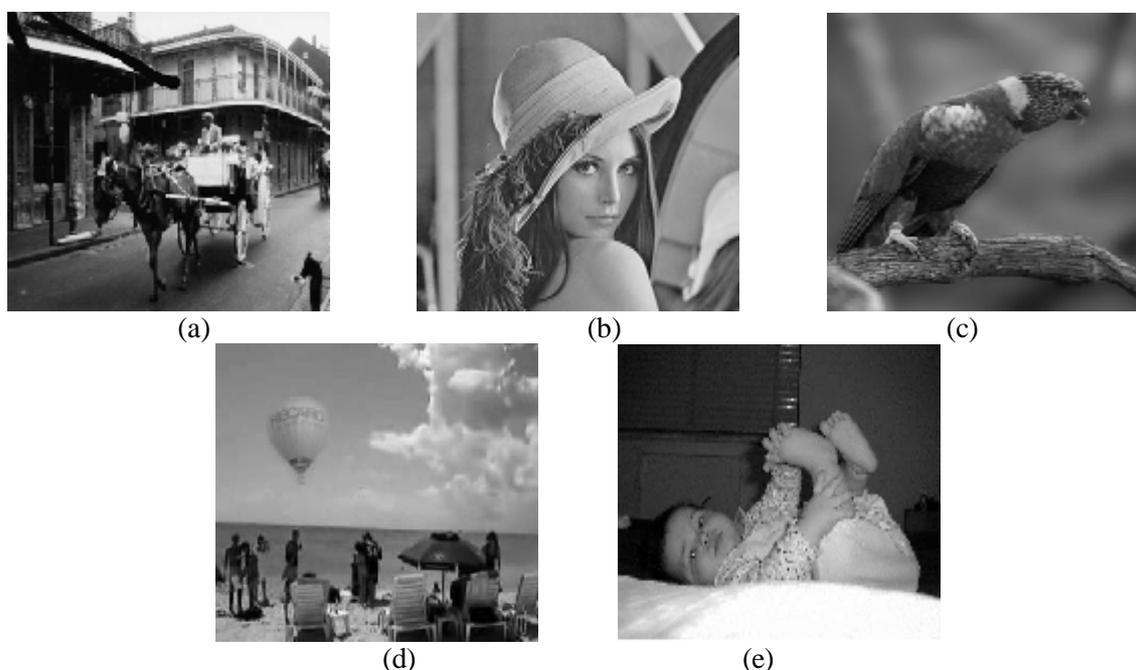


Fig. 3.5: Images tests. (a) charrette, (b) Lena, (c) perroquet, (d) ballon dans le ciel et (e) bébé.

Ces images seront superposées avec des masques binaires définissant les zones à reconstruire (domaine D) comme le montre la figure (3.6.). Ces masques sont définis manuellement de manière à obtenir une image binaire M telle que $M(x, y) = 1$ si le pixel (x, y) se situe dans la zone à reconstruire et $M(x, y) = 0$ sinon. Ces masques ont des formes et des tailles différentes. Les images résultantes sont représentées sur la figure (3.7).

Pour ces images, l'application de l'inpainting a pour objet d'éliminer l'écriture dans l'image charrette, d'effacer le dessin de la rose dans l'image Lena, de supprimer le grillage dans l'image perroquet, d'ôter le ballon du ciel et de supprimer les ratures dans l'image bébé.

Les résultats de l'inpainting par les trois méthodes d'interpolation monodimensionnelle sont représentés sur la figure (3.8) pour le balayage vidéo et la figure (3.9) pour le balayage de Hilbert. La figure (3.10) regroupe les résultats de l'inpainting obtenues par les méthodes d'interpolation bidimensionnelle. La taille $(2w+1)^2$ de la fenêtre de voisinage utilisée est indiquée sous chaque image reconstruite.

Le tableau (3.2) regroupe les différentes valeurs de l'erreur d'inpainting obtenues.

Since 1966, when french explorers landed at the great bend of the Mississippi River and celebrated the first Mardi Gras in North America, New Orleans has brewed a fascinating melange of cultures. It was French, then Spanish, then French again, then sold the United States. Through all these years and even into the 1900s, others arrived from every where: Acadians (Cajuns), Africans, indige-

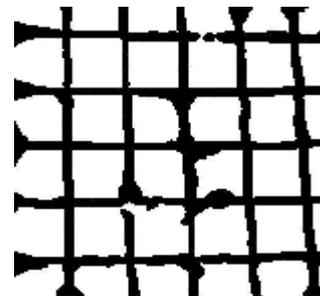


Fig.3.6 : Images binaire des domaines D à reconstruire.

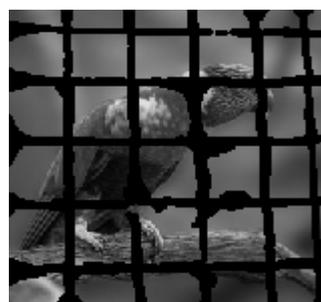


Fig. 3.7: Images à inpainter.

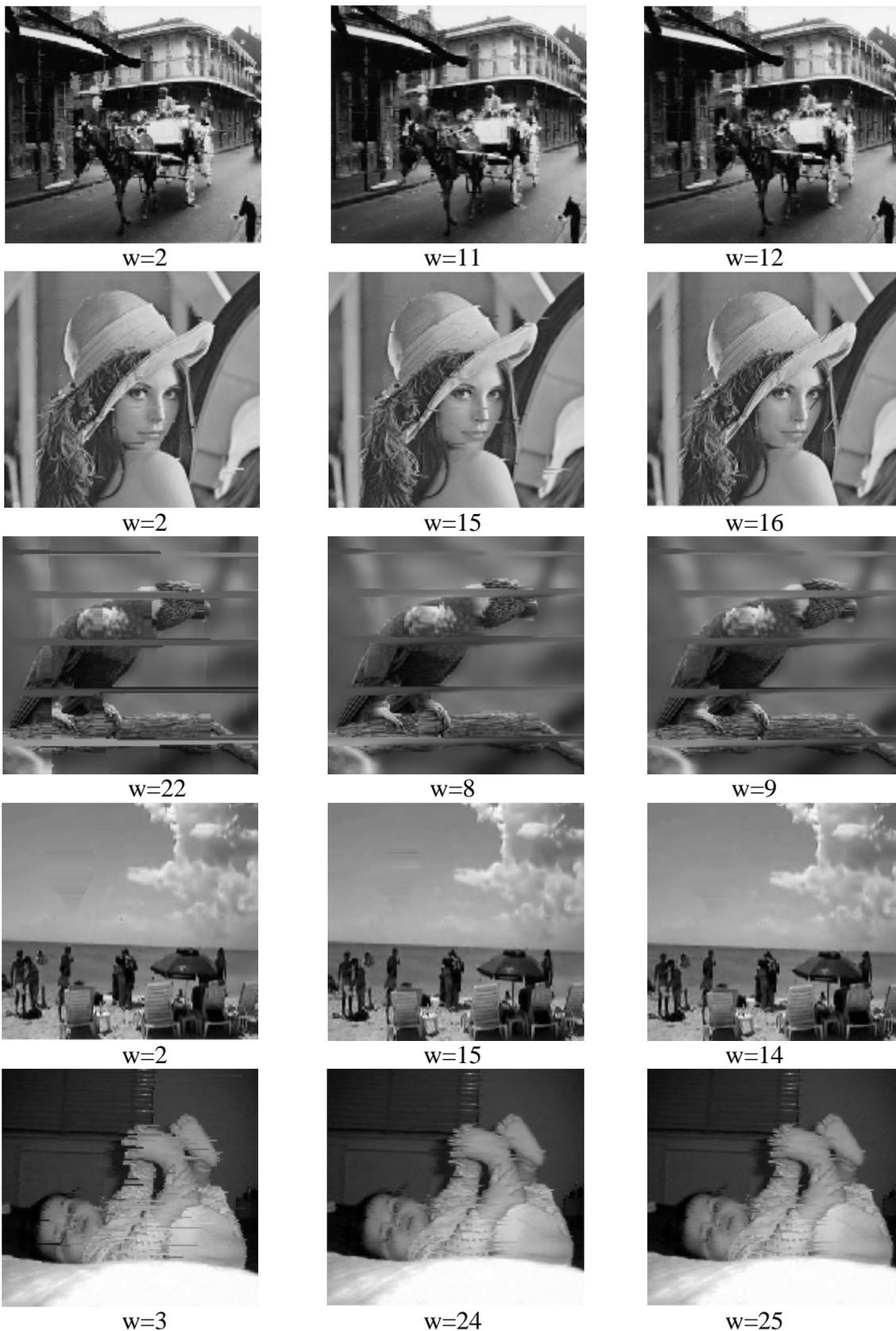


Fig. 3.8: Résultats d'inpainting selon le balayage vidéo par la méthode d'interpolation monodimensionnelle PPV (1^{ère} colonne), Linéaire (2^{ème} colonne) et Spline (3^{ème} colonne).

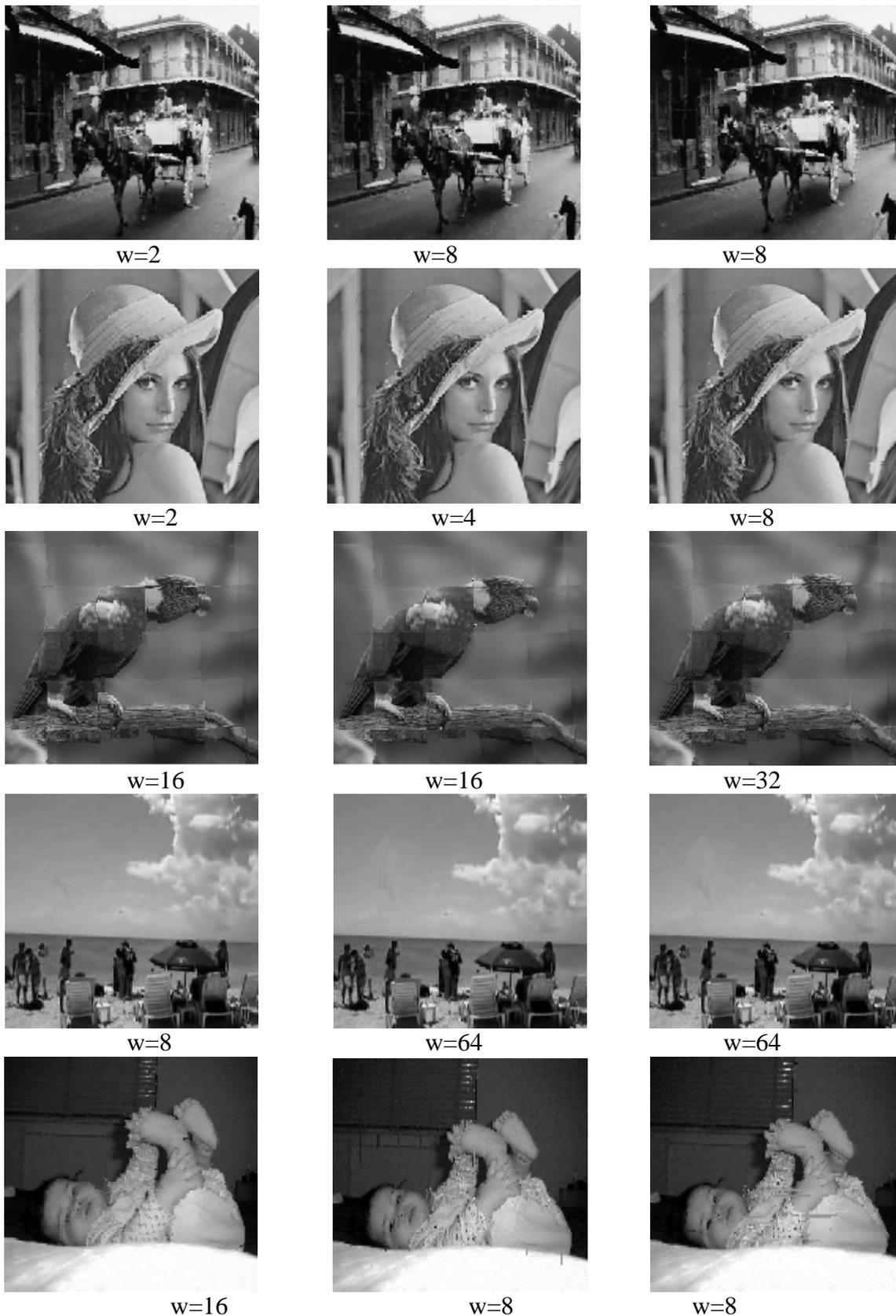


Fig. 3.9: Résultats d'inpainting selon le balayage Hilbert par la méthode d'interpolation monodimensionnelle PPV (1^{ère} colonne), Linéaire (2^{ème} colonne) et Spline (3^{ème} colonne).

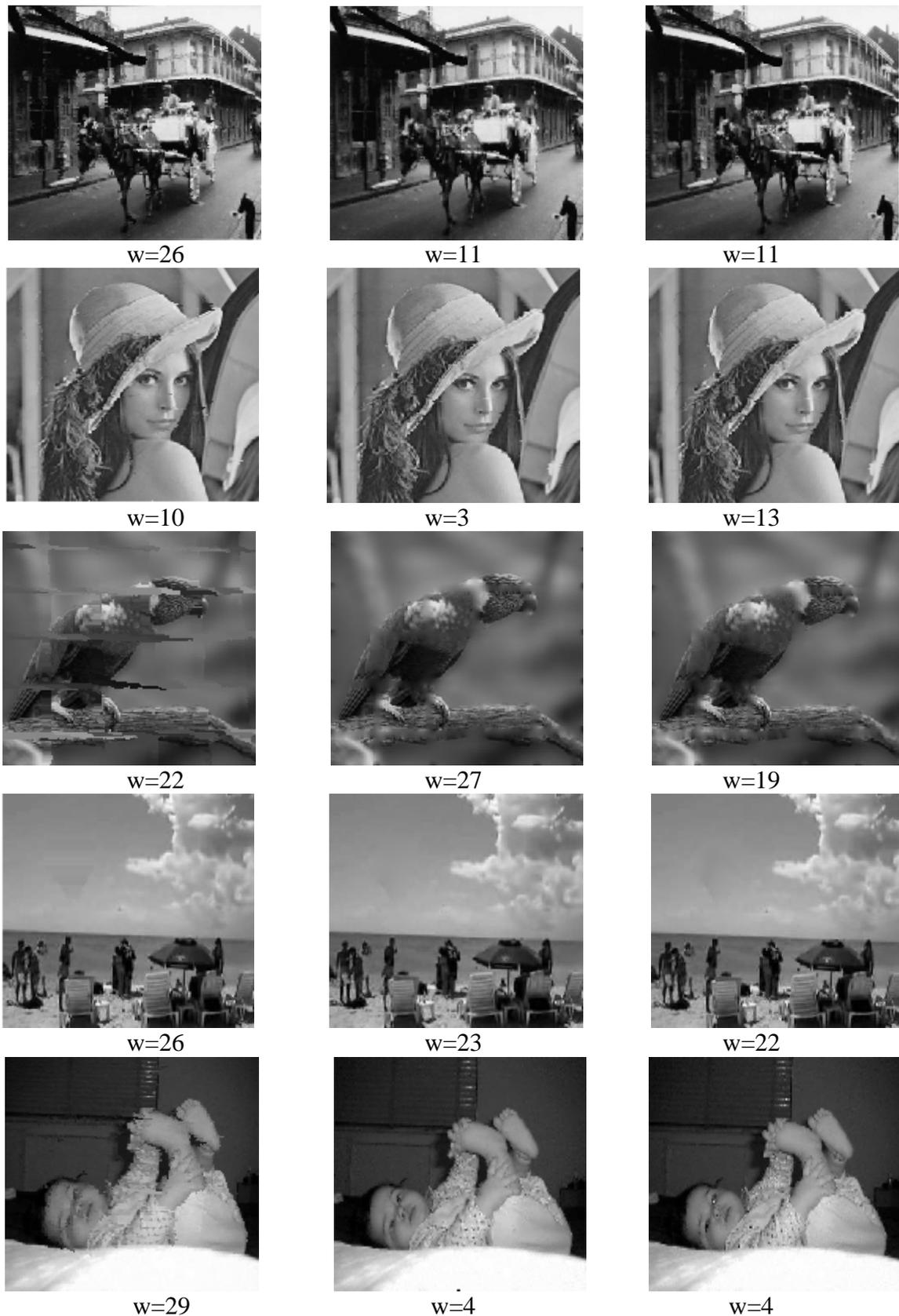


Fig. 3.10: Résultats d'inpainting par la méthode d'interpolation bidimensionnelle PPV (1^è colonne), Linéaire (2^{ème} colonne) et Spline (3^{ème} colonne).

	Méthodes	Charrette	Lena	Perroquet	Ballon	Bébé
Interpolation monodimensionnelles avec le balayage vidéo	PPV	129.8377	63.3885	224.7595	1.4846	81.2877
	Linéaire	81.7679	32.7080	127.2725	1.0108	41.6307
	Spline	78.8166	33.0109	137.2343	1.0755	40.4819
Interpolation monodimensionnelles avec le balayage Hilbert	PPV	139.6676	22.9367	125.0730	0.9340	36.3245
	Linéaire	124.0730	27.2891	115.3614	1.4203	31.4083
	Spline	124.3010	28.3120	121.1172	1.5317	31.7443
Interpolation bidimensionnelles	PPV	167.8124	30.0255	167.6481	1.4098	49.6655
	Linéaire	58.6811	7.8466	56.8209	1.2773	15.4292
	Spline	53.4110	6.5039	56.4184	1.0317	14.3138

Tableau 3.2 : Erreurs d'inpainting obtenues par les trois méthodes d'interpolation

Interprétation des résultats

On remarque que les méthodes d'interpolation bidimensionnelle ont donné de meilleurs résultats comparativement aux méthodes d'interpolation monodimensionnelle et que le balayage de Hilbert est plus performant que le balayage vidéo. En consultant les erreurs d'inpainting (tableau 3.2) obtenus par les trois méthodes d'interpolation, on constate que la méthode d'interpolation par spline cubique a donnée de meilleurs résultats dans le cas bidimensionnel et la méthode d'interpolation linéaire dans le cas monodimensionnel avec le balayage de Hilbert.

3.4.3 Influence de la taille du domaine D à reconstruire

La taille du domaine à reconstruire joue un rôle important. Pour évaluer les méthodes d'inpainting par rapport à ce facteur, nous avons généré sur l'image Lena plusieurs domaines D de forme rectangulaire avec des largeurs différentes (Fig.3.11). Quatre largeurs différentes ayant respectivement 2, 10, 15, 20 pixels sont utilisées. Les figures (3.12) à (3.14) montrent les images reconstruites par chaque méthode d'interpolation. Les erreurs calculées entre l'image originale et les images reconstruites par chaque méthode d'interpolation sont regroupées dans le tableau (3.3).

	Méthodes	Taille1	Taille2	Taille3	Taille4
Interpolation monodimensionnelles avec le balayage vidéo	PPV	25.3546	73.7241	121.0449	158.3540
	Linéaire	13.4676	42.6538	62.5314	79.6761
	Spline	15.1999	43.4596	64.5634	82.4134
Interpolation monodimensionnelles avec le balayage Hilbert	PPV	4.7583	32.9489	49.0989	73.9238
	Linéaire	4.8446	28.8506	55.6626	84.5723
	Spline	5.2609	31.6985	60.1017	89.9625
Interpolation bidimensionnelles	PPV	6.5934	39.6389	59.4405	81.8478
	Linéaire	1.7415	13.2646	22.3176	35.0675
	Spline	1.6369	13.5660	26.8683	45.3257

Tableau 3.3 : Erreurs d'inpainting obtenues par les trois méthodes d'interpolation



Fig. 3.11: Image test Lena avec des domaines D à reconstruire de tailles différentes.



Fig. 3.12: Résultats d'inpainting des images de la figure (3.11) selon le balayage vidéo par la méthode d'interpolation monodimensionnelle PPV (1^{ère} colonne), Linéaire (2^{ème} colonne) et Spline (3^{ème} colonne).



Fig. 3.13: Résultats d'inpainting des images de la figure (3.11) selon le balayage Hilbert par la méthode d'interpolation monodimensionnelle PPV (1^{ère} colonne), Linéaire (2^{ème} colonne) et Spline (3^{ème} colonne).



Fig. 3.14: Résultats d'inpainting des images de la figure (3.11) par la méthode d'interpolation bidimensionnelle PPV (1^{ère} colonne), Linéaire (2^{ème} colonne) et Spline (3^{ème} colonne).

Interprétation

Il apparaît clairement que lorsque la taille du domaine à reconstruire augmente, l'erreur d'inpainting obtenue par chaque méthode augmente. L'interpolation bidimensionnelle combinée avec les méthodes d'interpolations linéaire et par les splines donne les meilleurs résultats quelque soit la taille du domaine. On constate aussi quelques imperfections dans la reconstruction telle que les taches floues surtout lorsque le domaine d'inpainting augmente.

3.4.4 Influence de la taille de la fenêtre de voisinage

Les méthodes d'inpainting basées sur l'interpolation sont conditionnées par la taille $(2w+1)^2$ de la fenêtre de voisinage. Le tableau (3.4) regroupe quelques valeurs de l'erreur d'inpainting en fonction du paramètre w . Ces résultats sont obtenus par les méthodes d'interpolation sur les images de la figure (3.11).

	Méthodes	w	Taille1	Taille2	Taille3	Taille4
Interpolation mono-Dimensionnelles avec le balayage vidéo	PPV	4	25.3546	73.7241	121.0449	158.3540
		16	25.3546	73.7241	121.0449	158.3540
		32	25.3546	73.7241	121.0449	158.3540
	Linéaire	4	15.2502	71.6301	137.6297	187.5547
		16	18.0851	54.1292	84.9687	111.1174
		32	16.7002	43.4238	62.5340	78.9520
	Spline	4	16.9105	93.5905	196.0470	271.4137
		16	18.3355	55.2164	87.3110	115.2810
		32	16.4937	44.2211	64.5663	82.4149
Interpolation mono-dimensionnelles avec le balayage Hilbert	PPV	4	5.9521	51.3289	71.5422	107.9794
		16	5.9521	32.9489	69.4999	100.7058
		32	5.9521	32.9489	69.4999	103.9518
	Linéaire	4	4.8446	291.6490	431.1841	589.1249
		16	5.3509	28.8506	55.6626	114.6007
		32	5.3509	29.0180	55.6906	84.5723
	Spline	4	5.2609	304.5164	468.8844	635.1399
		16	5.7703	31.7936	60.2366	122.8988
		32	5.7703	31.6985	60.1017	89.9625
Interpolation bidimensionnelles	PPV	4	24.8460	77.0444	100.7480	146.0283
		16	8.3452	71.2298	124.7845	122.7249
		32	16.8033	66.8095	92.5001	123.5707
	Linéaire	4	1.8970	356.9399	461.0447	633.8079
		16	1.8262	13.3381	22.9657	35.1438
		32	1.8715	13.4734	22.7885	35.1897
	Spline	4	1.8029	358.8062	462.0847	634.8479
		16	1.6852	13.8250	27.6744	45.3743
		32	1.7258	13.8806	27.2225	46.5943

Tableau 3.4 : Erreurs d'inpainting obtenues par les trois méthodes d'interpolation en fonction de la taille de la fenêtre de voisinage w .

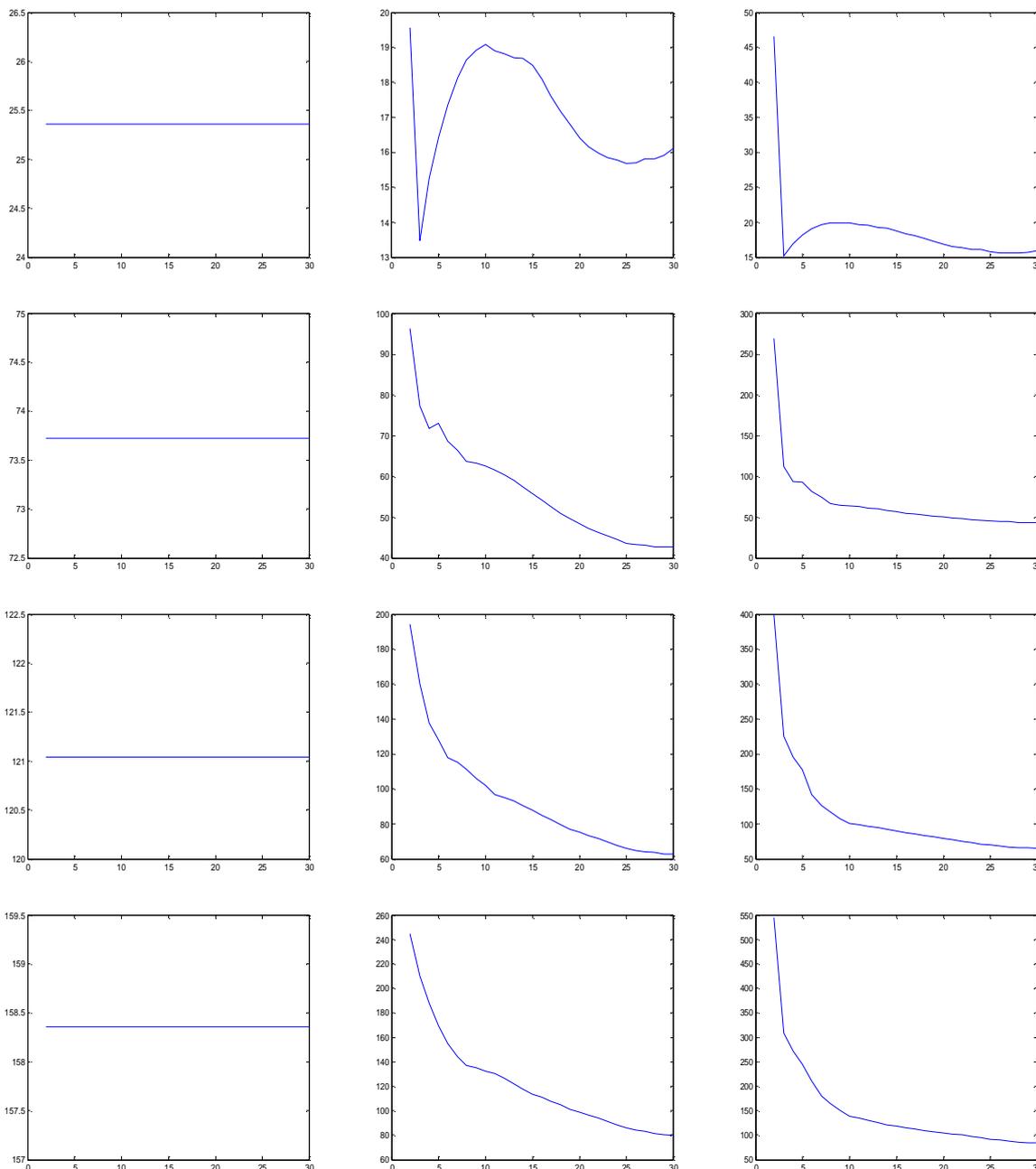


Fig. 3.15: Les graphes des erreurs en fonction de w en fonction de la taille du domaine D et selon le balayage vidéo par la méthode d'interpolation monodimensionnelle PPV(1ère colonne), Linéaire (2^{ème} colonne) et Spline (3^{ème} colonne).

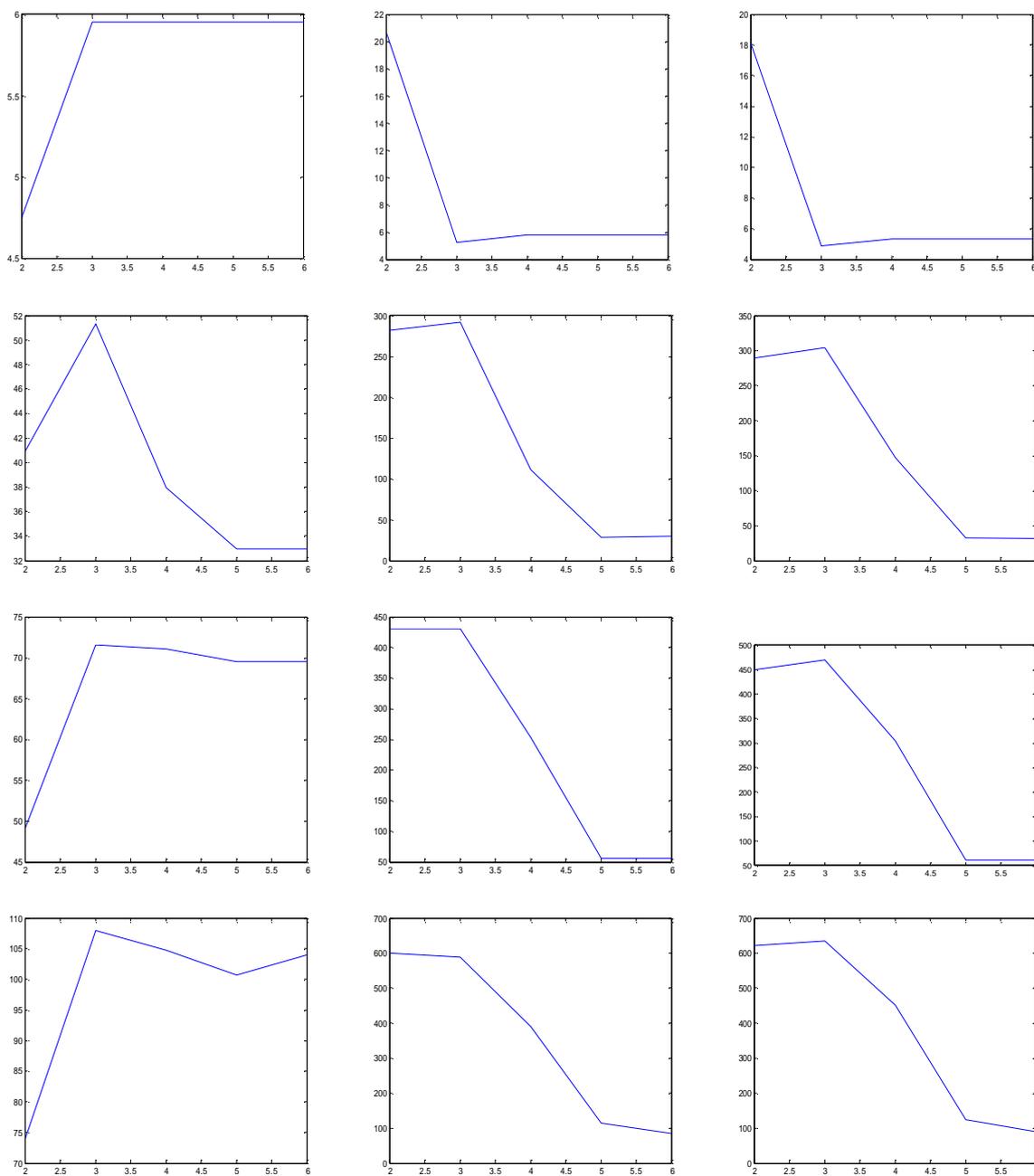


Fig. 3.16: Les graphes des erreurs en fonction de w en fonction de la taille du domaine D et selon le balayage Hilbert par la méthode d'interpolation monodimensionnelle PPV (1ère colonne), Linéaire (2^{ème} colonne) et Spline (3^{ème} colonne).

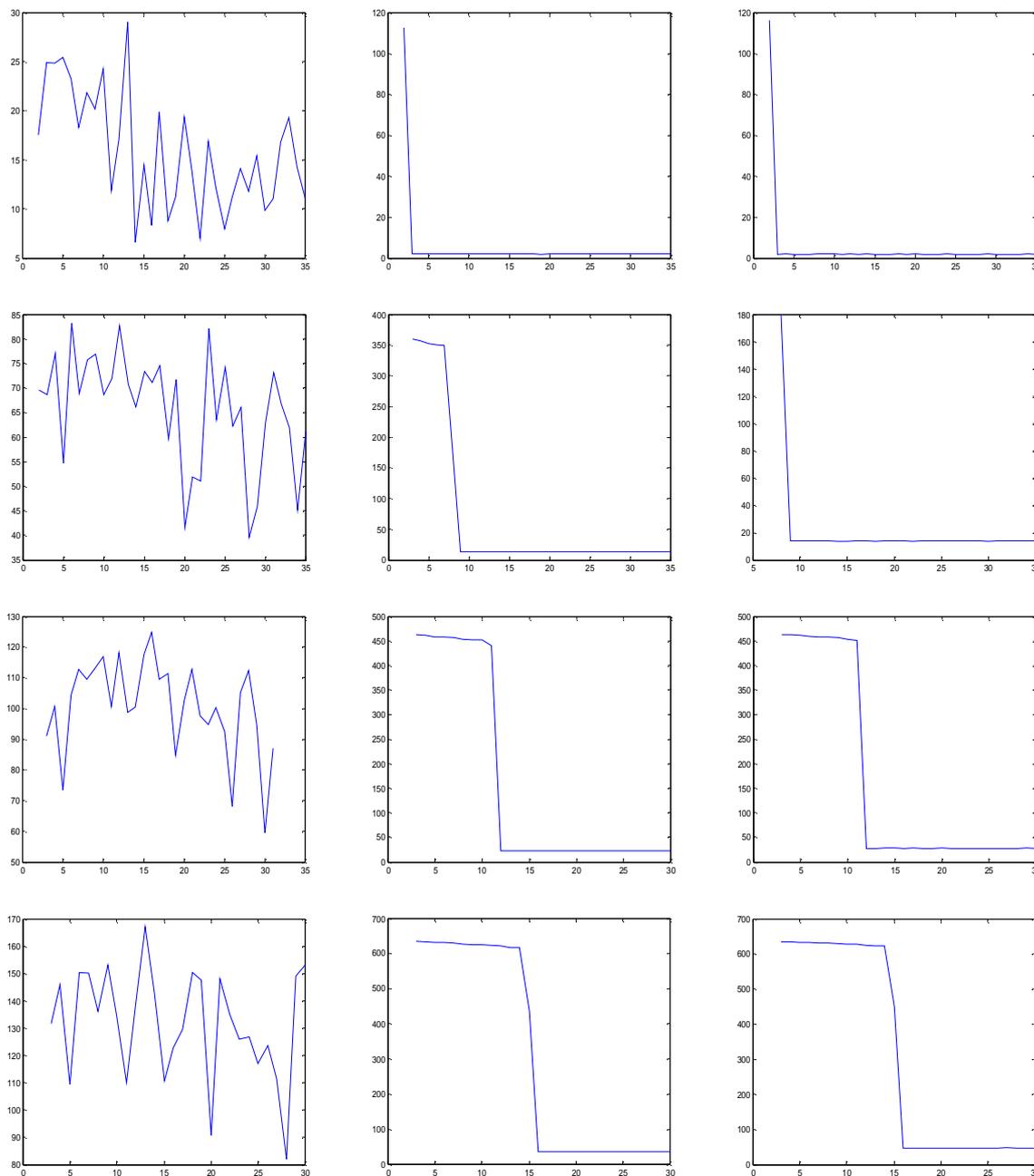


Fig. 3.17: Les graphes des erreurs en fonction de w en fonction de la taille du domaine D et selon la méthode d'interpolation bidimensionnelle PPV (1ère colonne), Linéaire (2^{ème} colonne) et Spline (3^{ème} colonne).

Nous avons également évalué le temps de calcul de toutes les méthodes d'interpolation implémentées. Le tableau (3.5) regroupe le temps consommé par chaque méthode en fonction du paramètre w .

	Méthodes	w	Taille1	Taille2	Taille3	Taille4
Interpolation mono-Dimensionnelles avec le balayage vidéo	PPV	4	0.50	0.90	1.11	1.37
		16	3.32	8.40	11.19	13.74
		32	26.53	74.33	98.92	123.48
	Linéaire	4	0.45	0.87	1.08	1.33
		16	3.56	8.48	11.21	13.70
		32	26.32	71.38	99.59	122.29
	Spline	4	0.85	1.66	2.15	3.16
		16	4.81	9.92	13.35	16.30
		32	28.77	75.08	103.07	137.95
Interpolation mono-dimensionnelles avec le balayage Hilbert	PPV	4	0.47	0.86	1.10	1.33
		16	3.41	8.76	10.66	13.10
		32	25.01	69.57	92.70	120.35
	Linéaire	4	0.47	0.83	1.26	1.32
		16	3.79	7.88	10.53	12.99
		32	24.93	67.47	93.17	117.31
	Spline	4	0.78	1.57	2.10	2.94
		16	3.67	9.26	12.60	15.59
		32	26.84	71.35	96.67	124.22
Interpolation bidimensionnelles	PPV	4	2.43	4.52	6.08	8.54
		16	21.49	57.02	77.99	94.43
		32	104.98	273.92	384.36	490.41
	Linéaire	4	2.07	4.70	6.31	7.91
		16	20.69	54.75	74.12	89.45
		32	102.79	261.72	369.45	477.07
	Spline	4	2.10	6.04	8.33	10.63
		16	22.47	59.55	78.46	95.43
		32	101.84	272.01	383.27	493.97

Tableau 3.5 : Temps de calcul (en seconde) obtenus par les trois méthodes d'interpolation en fonction de la fenêtre de voisinage w et la taille du domaine D à reconstruire.

Interprétation

On peut remarquer à travers les résultats obtenus que le choix de w est crucial. La méthode d'interpolation PPV donne des résultats différents selon la dimension mono ou bidimensionnelle, la taille le type de balayage utilisé. L'erreur d'inpainting reste constante quelque soit w dans le cas du balayage vidéo, varie peut ou d'une manière erratique dans les autres cas. Par contre pour les méthodes d'interpolation linéaire et par les splines cubiques la variation de l'erreur d'inpainting en fonction de w est plus régulière, elle est grande pour w petit puis décroît pour finir à des valeurs plus faibles lorsque w devient grand. La décroissance est lente dans le cas monodimensionnelle et rapide dans le cas bidimensionnel. Pour ce dernier cas, le choix de w devient plus simple à faire.

Par contre, sans surprise, le temps de calcul augmente avec la taille du domaine à interpoler et la taille w de la fenêtre de voisinage. L'interpolation monodimensionnelle apparaît naturellement plus rapide que l'interpolation bidimensionnelle.

3.4.4 Application sur des images couleur

Les tests présentés précédemment ont été effectués sur des images en niveaux de gris. Or il est toujours possible d'appliquer les méthodes d'inpainting sur des images couleurs. Pour cela, il suffit d'appliquer la même procédure sur chaque composante RVB de l'image. La figure (3.18) montre quelques images couleurs à inpainter. Les figures (3.19) à (3.21) montrent les résultats obtenus par les différentes méthodes d'interpolation.

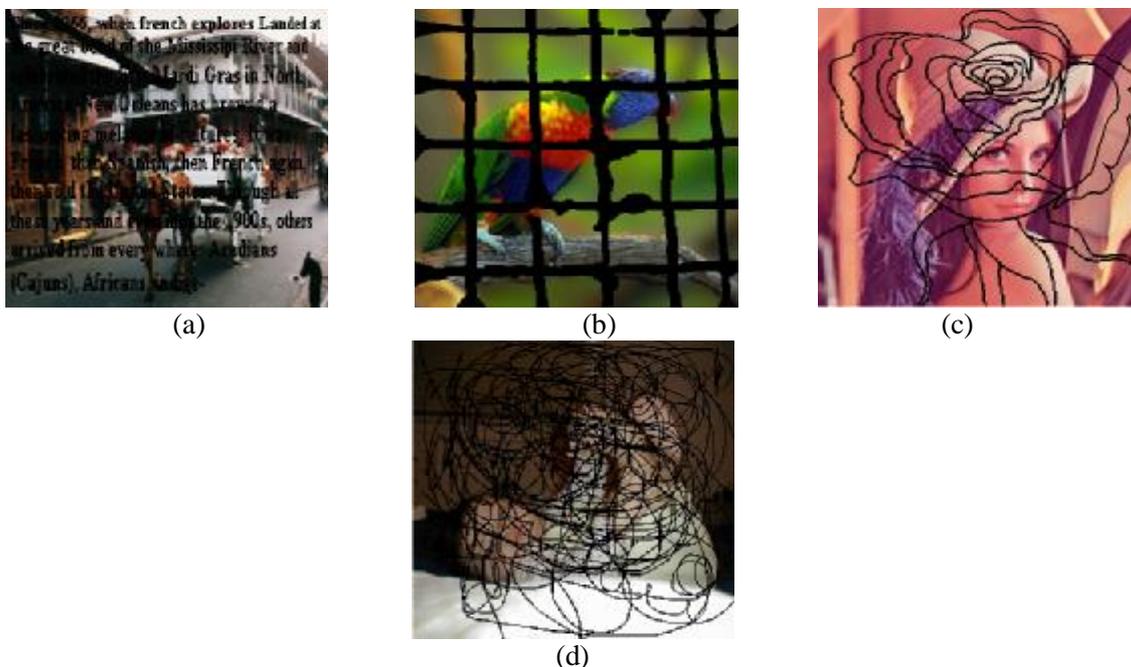


Fig. 3.18: Image couleur à inpainter.

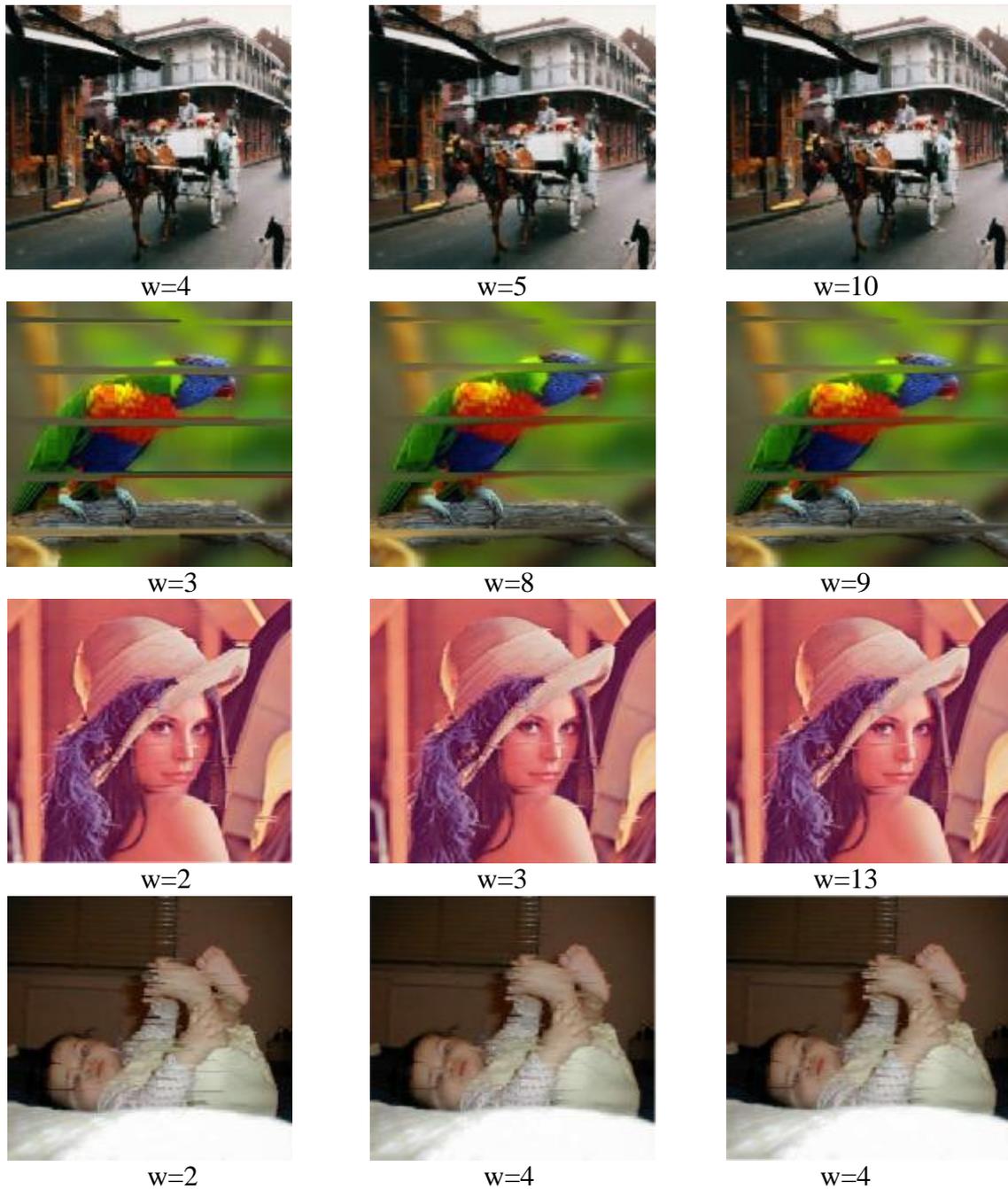


Fig. 3.19: Résultats d'inpainting des images couleur par la méthode d'interpolation monodimensionnelle PPV (1^{ère} colonne), Linéaire (2^{ème} colonne) et Spline (3^{ème} colonne) et selon le balayage vidéo

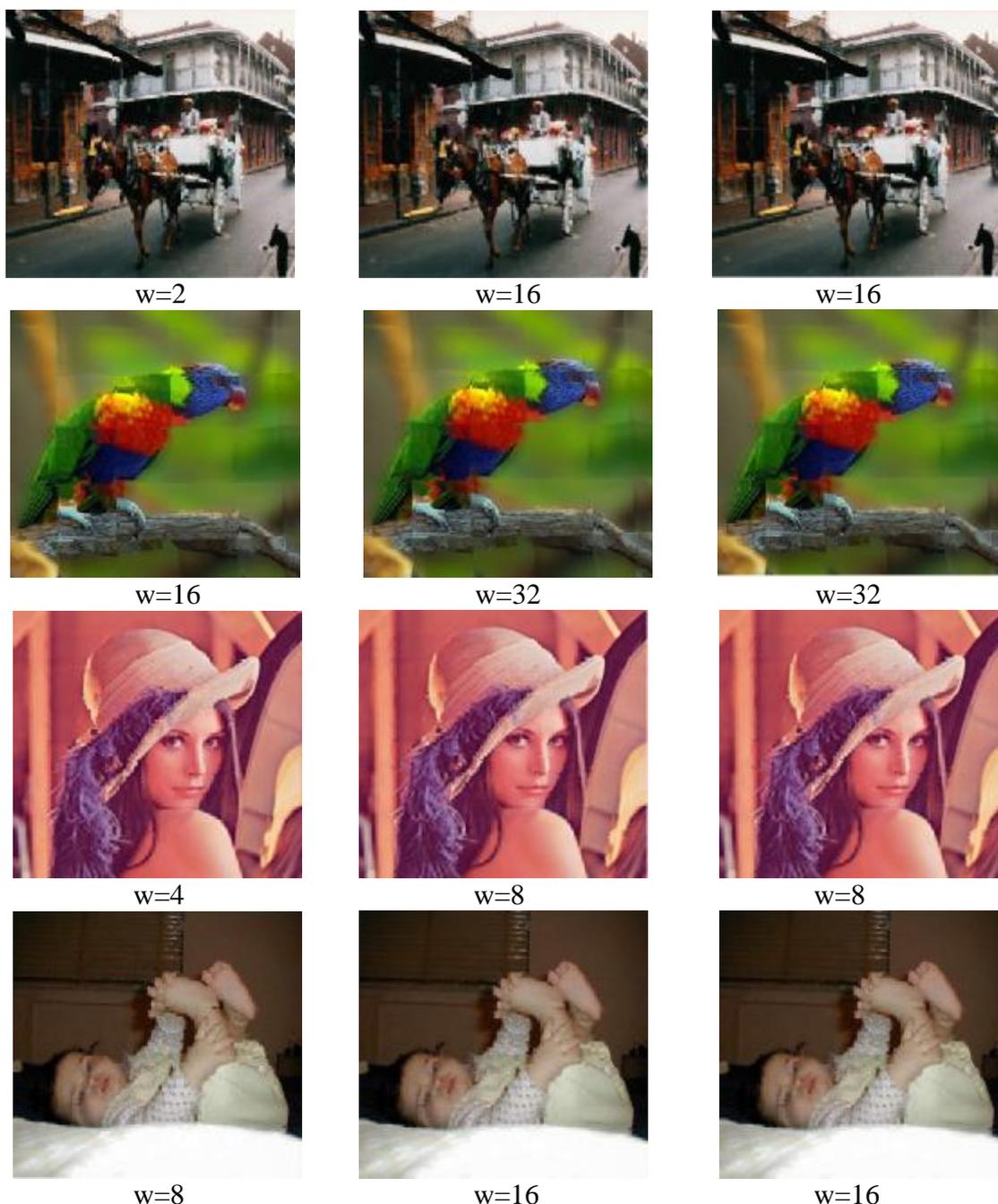


Fig. 3.20: Résultats d'inpainting des images couleur par la méthode d'interpolation monodimensionnelle PPV (1^{ère} colonne), Linéaire (2^{ème} colonne) et Spline (3^{ème} colonne) et selon le balayage de Hilbert

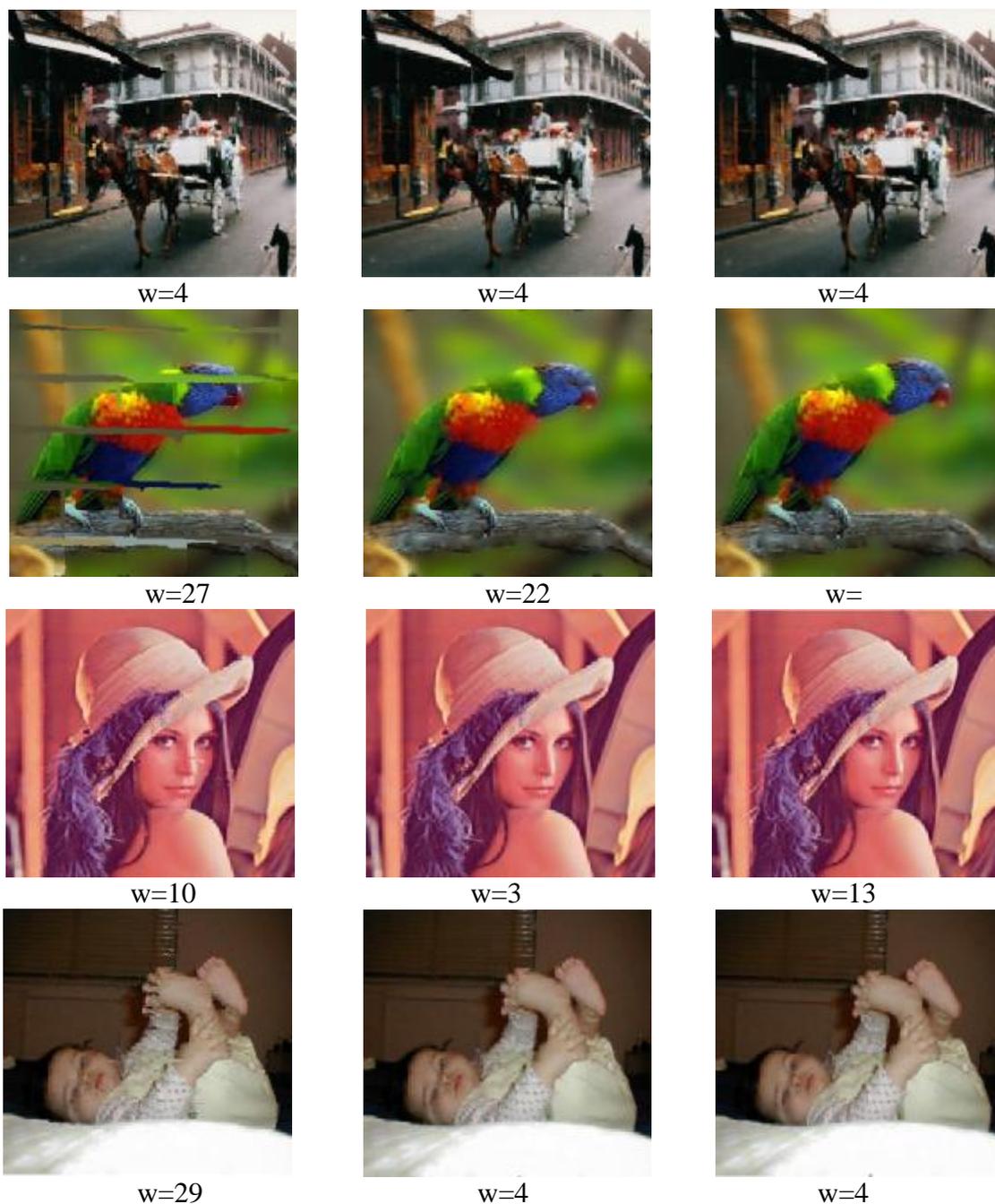


Fig. 3.21: Résultats d'inpainting des images couleur par la méthode d'interpolation bidimensionnelle PPV (1^{ère} colonne), Linéaire (2^{ème} colonne) et Spline (3^{ème} colonne).

	Méthodes	charrette	Perroquet	Lena	Bébé
Interpolation monodimensionnelles avec le balayage vidéo	PPV	408.9970	981.5382	175.3179	251.4064
	Linéaire	256.9697	545.3400	94.4598	129.7790
	Spline	247.6887	584.4232	89.2198	127.2953
Interpolation monodimensionnelles avec le balayage Hilbert	PPV	434.9979	439.2668	66.1354	112.7425
	Linéaire	387.2452	411.3059	77.6091	97.7754
	Spline	387.8880	432.5634	80.7063	98.7404
Interpolation bidimensionnelles	PPV	526.8661	791.5713	86.7887	160.2028
	Linéaire	183.1069	185.8091	24.2411	48.1322
	Spline	166.0417	185.2037	20.6806	44.4683

Tableau 3.6 : Erreurs d'inpainting obtenues par les trois méthodes d'interpolation sur les images couleur

Interprétation des résultats

Les résultats d'inpainting obtenues par les trois méthodes d'interpolation mono et bidimensionnelle avec les deux types de balayage pour les méthodes monodimensionnelles sont mieux mis en évidence sur les images couleur. Ces résultats confirment ceux obtenus sur les images en niveaux de gris à savoir que l'interpolation bidimensionnelle est plus performante que l'interpolation monodimensionnelle et que l'interpolation linéaire et par les splines cubiques est meilleur que l'interpolation par PPV.

3.5 Conclusion

Dans ce chapitre, nous avons montré comment appliquer les méthodes d'interpolation mono et bidimensionnelle pour reconstruire des parties manquantes dans une image. Trois méthodes d'interpolation (linéaire, PPV, spline) dans le cas mono et bidimensionnel ont été testés. L'utilisation des méthodes d'interpolation monodimensionnelle nécessite la transformation de l'image 2D en une fonction monovariante. Cette transformation est basée sur le balayage de l'image selon un parcours bien déterminé. Deux types de balayage (vidéo et Hilbert) ont été alors comparés. Plusieurs tests faisant intervenir différentes images en niveaux de gris ou en couleur avec différents domaines à interpoler ont montré d'une manière objective la supériorité des méthodes d'interpolation bidimensionnelle et que le balayage de Hilbert donne de meilleurs résultats que le balayage vidéo. Les tests ont également montré que l'interpolation linéaire et par les splines cubiques donne de meilleurs résultats que l'interpolation par PPV.

4.1. Introduction

Nous allons présenter dans ce chapitre une méthode d'inpainting basée sur les équations de diffusion. Cette méthode consiste à adapter l'équation de diffusion de la chaleur afin de diffuser l'information depuis les zones où cette information est disponible vers les zones où l'information est manquante. La diffusion de cette information s'effectuera d'une manière isotrope ou anisotrope.

4.2. Principe de l'inpainting par diffusion

Rappelons que le but de l'inpainting consiste à restituer l'information dans un domaine D de l'image $u(x, y)$ en diffusant l'information existante sur la frontière de ∂D vers le domaine D . Dans le cas général, la solution à ce problème peut être obtenue en résolvant une EDP de diffusion dérivée de l'équation de la chaleur avec une condition initiale et les conditions aux limites de Neumann:

$$\begin{cases} \frac{\partial u}{\partial t}(x, y, t) = \text{div}(f(\nabla u)) & \text{dans } D \\ u(x, y, t)|_{\partial D} = u_0(x, y, t)|_{\partial D} \\ u(x, y, 0) = 0 \end{cases} \quad (4.1)$$

f est une fonction dépendant du gradient de u . Autrement dit, la démarche adoptée pour effectuer l'inpainting à base de l'EDP de diffusion consiste à parcourir le domaine à reconstituer pixel par pixel et pour chaque pixel détérioré dans l'image u_0 , on lui applique l'équation de diffusion qui sera explicité ci-dessous.

4.3. Inpainting par diffusion isotropique

Dans le cas d'une diffusion isotropique, la fonction f est une fonction identité et l'équation (4.1) s'écrira :

$$\begin{cases} \frac{\partial u}{\partial t}(x, y, t) = \text{div}(\nabla u) & \text{dans } D \\ u(x, y, t)|_{\partial D} = u_0(x, y, t)|_{\partial D} \\ u(x, y, 0) = 0 \end{cases} \quad (4.2)$$

La résolution de cette équation d'une manière numérique nécessite le passage de l'équation de la chaleur du domaine continu au domaine discret. Pour cela, nous avons utilisé la méthode basée sur les différences finies. On peut alors écrire :

$$\begin{aligned} \frac{\partial u}{\partial t} &= \text{div}(\nabla u) = \Delta u \\ \frac{u^{t+1} - u^t}{dt} &= \Delta u^t \\ u^{t+1} - u^t &= (\Delta u^t) dt \\ u^{t+1} &= u^t + dt(\Delta u^t) \end{aligned} \quad (4.3)$$

Le terme (Δu^t) représente le Laplacien de l'image à l'itération t , il est donné par l'équation :

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \quad (4.4)$$

En utilisant les différences finies, les deux termes du Laplacien sont approximés par :

$$\left(\frac{\partial^2 u}{\partial x^2} \right)_{i,j} \approx \frac{u_{i+1,j} + u_{i-1,j} - 2u_{i,j}}{\Delta x^2} \quad (4.5.a)$$

$$\left(\frac{\partial^2 u}{\partial y^2} \right)_{i,j} \approx \frac{u_{i,j+1} + u_{i,j-1} - 2u_{i,j}}{\Delta y^2} \quad (4.5.b)$$

avec Δx est le pas de discrétisation suivant x et Δy est le pas de discrétisation suivant y .

En remplaçant ces deux termes par leurs expressions dans l'équation (4.4), on obtient l'équation suivante :

$$\Delta u = \frac{u_{i+1,j} + u_{i-1,j} - 2u_{i,j}}{\Delta x^2} + \frac{u_{i,j+1} + u_{i,j-1} - 2u_{i,j}}{\Delta y^2} \quad (4.5)$$

En substituant l'équation (4.5) dans l'équation (4.3), on aura

$$u_{i,j}^{t+1} = u_{i,j}^t + dt \left(\frac{u_{i+1,j}^t + u_{i-1,j}^t - 2u_{i,j}^t}{\Delta x^2} + \frac{u_{i,j+1}^t + u_{i,j-1}^t - 2u_{i,j}^t}{\Delta y^2} \right) \quad (4.6)$$

L'inpainting par diffusion isotropique consiste finalement à réitérer cette équation (4.6) pendant un certain nombre d'itérations maximal (T_{\max}), pour chaque pixel de coordonnées (i, j) du domaine D .

4.4. Inpainting par diffusion anisotropique

Contrairement à la diffusion isotropique, la diffusion anisotropique permet une diffusion différente d'un pixel à un autre. Dans le cas d'une diffusion anisotropique, la fonction f prend la forme $f = g(|\nabla u|)\nabla u$ et l'équation (4.1) devient :

$$\begin{cases} \frac{\partial u}{\partial t}(x, y, t) = \text{div}(g(|\nabla u|)\nabla u) & \text{dans } D \\ u(x, y, t)|_{\partial D} = u_0(x, y, t)|_{\partial D} \\ u(x, y, 0) = 0 \end{cases} \quad (4.7)$$

La fonction g doit satisfaire les conditions suivantes [12] :

$$g(0) = 1 \text{ et } \lim_{k \rightarrow +\infty} g(k) = 0$$

Perona et Malik ont proposé dans [12] les fonctions suivantes :

$$g(|\nabla u|) = \exp(-(|\nabla u|/k)^2) \quad (4.8)$$

$$g(|\nabla u|) = \frac{1}{1 + (|\nabla u|/k)^2}. \quad (4.9)$$

k étant un paramètre à fixer.

Ces fonctions permettent d'effectuer une diffusion conditionnelle qui dépend du gradient de l'image. Cette diffusion est forte dans les zones à faible gradient et faible dans les zones à fort gradient. Elle permettra, contrairement à la diffusion isotropique, de conserver les contours de l'image lors de la diffusion. Dans notre cas, nous avons utilisé la deuxième fonction.

Comme pour l'équation (4.2), la résolution de cette équation (4.7) est effectuée en utilisant les différences finies. L'équation récurrente qui en découle est:

$$u_{i,j}^{t+1} = u_{i,j}^t + dt \left(\frac{Cn_{i,j}^t \nabla n_{i,j}^t + Cs_{i,j}^t \nabla s_{i,j}^t}{\Delta x^2} + \frac{Cw_{i,j}^t \nabla w_{i,j}^t + Ce_{i,j}^t \nabla e_{i,j}^t}{\Delta y^2} \right) \quad (4.10)$$

avec

$$\begin{aligned} Cn_{i,j}^t &= (u_{i-1,j}^t - u_{i,j}^t) & \nabla n_{i,j}^t &= g(Cn_{i-1,j}^t - Cn_{i,j}^t) \\ Cs_{i,j}^t &= (u_{i+1,j}^t - u_{i,j}^t) & \nabla s_{i,j}^t &= g(Cs_{i+1,j}^t - Cs_{i,j}^t) \\ Cw_{i,j}^t &= (u_{i,j-1}^t - u_{i,j}^t) & \nabla w_{i,j}^t &= g(Cw_{i,j-1}^t - Cw_{i,j}^t) \\ Ce_{i,j}^t &= (u_{i,j+1}^t - u_{i,j}^t) & \nabla e_{i,j}^t &= g(Ce_{i,j+1}^t - Ce_{i,j}^t) \end{aligned} \quad (4.11)$$

Finalement, l'inpainting par diffusion anisotropique revient à réitérer le calcul de l'équation (4.10) pendant un certain nombre d'itérations maximal (T_{\max}), pour chaque pixel de coordonnées (i, j) du domaine D .

4.5 Tests et résultats

Nous allons présenter dans ce paragraphe les résultats de l'inpainting obtenues par la diffusion. Les mêmes images tests (en niveaux de gris ou en couleur) et les mêmes critères d'évaluation (erreur d'inpainting et temps de calcul), présentées dans le chapitre précédent, sont utilisées dans ce paragraphe pour évaluer les performances de la méthode d'inpainting par diffusion. La méthode d'inpainting basée sur la diffusion dépend du coefficient dt , du nombre maximal d'itérations T_{\max} et de la taille du domaine D à reconstruire ainsi que du paramètre k pour la diffusion anisotropique.

4.3.1 Images tests

La figure (4.1) montre les résultats de l'inpainting obtenues par la diffusion isotropique et anisotropique. Dans les deux cas, le nombre maximal d'itérations est fixé à 3000 et la valeur du coefficient dt est fixé à 0.3. Par contre, dans le cas de l'anisotropie, la valeur du coefficient k varie selon l'image. Le tableau (4.1) regroupe les différentes valeurs de l'erreur d'inpainting correspondantes.

Diffusion	Charrette	Lena	Perroquet	Ballon	Bébé
Isotopique	58.3553	9.9886	92.0498	0.9837	22.8334
Anisotropique	56.3576	9.6748	73.6828	0.9793	22.5110

Tableau 4.1 : Erreurs d'inpainting obtenues par diffusion sur les images tests.

Interprétation des résultats

Comme on peut le voir sur la figure (4.1) les résultats d'inpainting obtenues par les méthodes de diffusion isotrope et anisotrope sont appréciables. La diffusion anisotrope reste légèrement meilleure que la diffusion isotrope. Ces résultats sont meilleurs que ceux obtenus par l'interpolation monodimensionnelle mais moins bons (sauf pour l'image "ballon") que ceux obtenus par la méthode d'interpolation bidimensionnelle des splines cubiques (voir Chapitre 3).

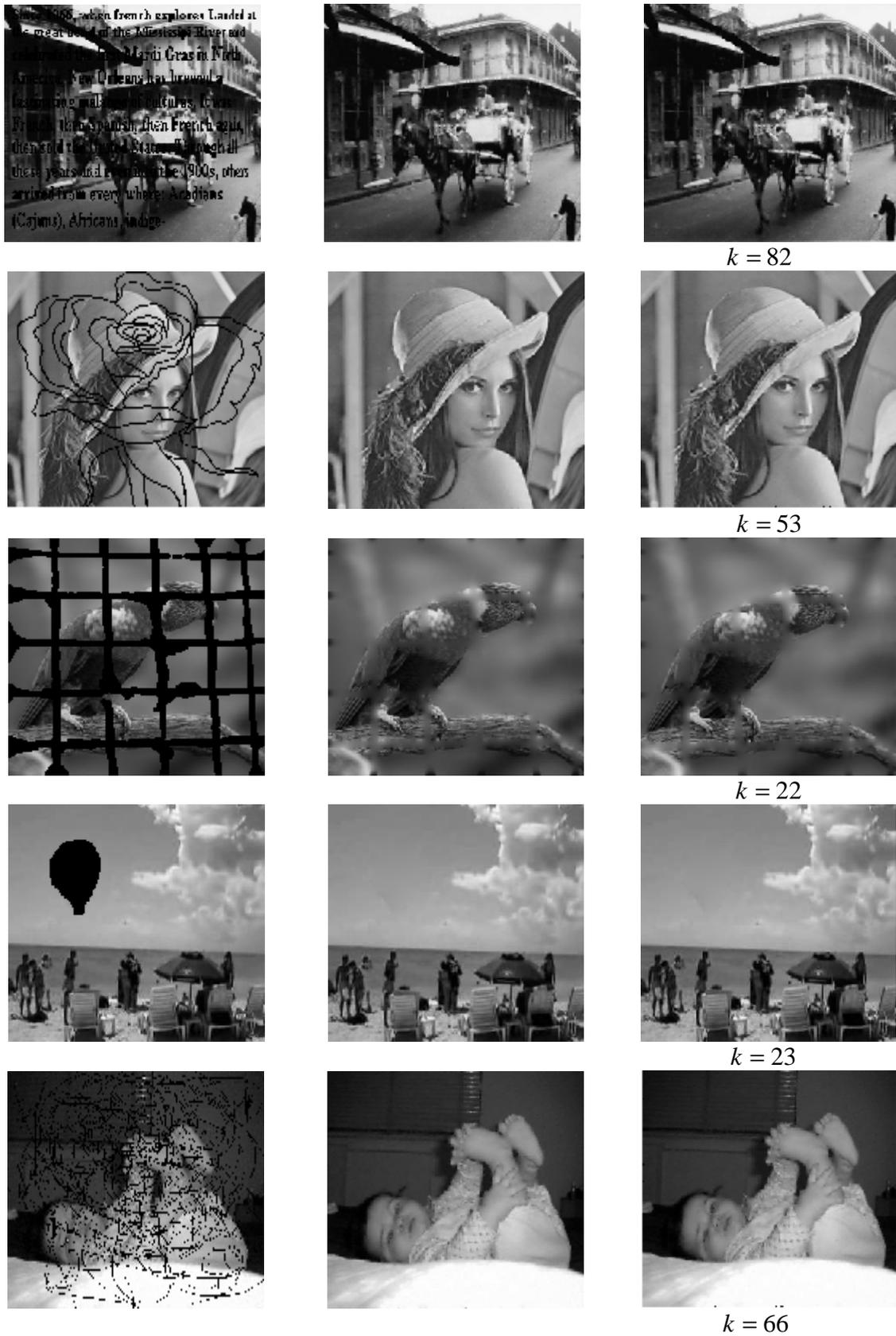


Fig. 4.1: Résultats d'inpainting par la méthode de diffusion. Colonne 1 : Images tests. Colonne 2 : Diffusion isotropique. Colonne 3 : Diffusion anisotropique.

4.3.2 Influence de la taille du domaine D à reconstruire

Comme pour les méthodes d'inpainting basées sur l'interpolation des fonctions, nous avons étudié l'influence de la taille du domaine à reconstruire sur la qualité de l'image reconstruite par la méthode de diffusion. L'évaluation des résultats est également effectuée sur la même image Lena présentant plusieurs domaines D de forme rectangulaire avec des largeurs différentes. La figure (4.2) montre les images reconstruites par la diffusion isotropique et anisotropique. Les erreurs calculées entre l'image originale et les images reconstruites sont regroupées dans le tableau (4.2). Ces résultats sont obtenus avec paramètres suivants : $T_{\max} = 3000$, $dt = 0.3$. La valeur du coefficient k varie d'une image à l'autre.

Diffusion	Taille1	Taille2	Taille3	Taille4
Isotropique	1.9826	14.3839	21.8031	33.3923
Anisotropique	1.9428	14.2262	21.8345	33.3852

Tableau 4.2 : Erreurs d'inpainting obtenues par la diffusion en fonction la taille du domaine D .

Interprétation des résultats

D'après ces résultats, on remarque que lorsqu'on augmente la taille du domaine à reconstruire, les erreurs calculées entre l'image originale et les images reconstruites par chaque méthode augmente aussi. En comparant ces résultats avec ceux obtenus avec l'interpolation bidimensionnelle que l'inpainting par diffusion est moins performante lorsque la taille des domaines à reconstruire est petite mais plus performante lorsque la taille de ces domaines est élevée. La diffusion anisotrope reste identique voir légèrement meilleure que la diffusion isotrope.

4.3.3 Influence du coefficient dt

La résolution d'une manière numérique de l'équation différentielle dans le cas de la diffusion isotropique et anisotropique nécessite de fixer la valeur du coefficient dt . Nous avons ainsi étudié l'influence de ce coefficient sur la qualité de l'image reconstruite. Le tableau (4.3) montre les valeurs de l'erreur d'inpainting en fonction de quelques valeurs du coefficient dt . Ces résultats sont obtenus sur les images de la figure (4.2).



Fig. 4.2: Reconstruction par la diffusion de l'image test Lena sur des domaines D de tailles différentes. Colonne 1 : Images tests. Colonne 2 : Diffusion isotropique. Colonne 3 : Diffusion anisotropique.

Interprétation des résultats

D'après ces résultats, on remarque que lorsqu'on varie la valeur du coefficient dt , les erreurs calculées entre l'image originale et les images reconstruites par chaque méthode changent aussi. Cependant, il est toujours possible de trouver une valeur de dt qui minimise l'erreur d'inpainting.

Diffusion	dt	Taille1	Taille2	Taille3	Taille4
Isotropique	0.0095	1.9826	14.7123	27.0367	67.1108
	0.0150	1.9826	14.3839	21.8031	39.4011
	0.1500	1.9826	14.3950	21.9597	33.3923
	0.3000	1.9826	14.3950	21.9597	33.3923
	0.5000	28.7049	53.9892	72.5499	85.3660
	0.5100	131.0170	427.2621	594.7692	793.1662
Anisotropique	0.0095	1.9434	83.0062	142.7664	73.6888
	0.0150	1.9428	32.8236	62.3809	40.9846
	0.1500	1.9428	14.2262	21.8345	33.3852
	0.3000	1.9428	14.2262	21.8345	33.3852
	0.5000	1.9428	14.2262	21.8345	33.4108
	0.5100	2.1064	14.5935	22.8590	159.5481

Tableau 4.3 : Erreurs d'inpainting obtenues par la diffusion en fonction du coefficient dt.

4.3.4 Influence du nombre d'itérations maximal (Tmax)

L'équation de diffusion doit être réitérée pendant un certain nombre d'itérations noté T_{max} . Or ce nombre doit être fixé au préalable. Pour étudié son influence sur la qualité de la reconstitution, nous avons déterminé l'erreur d'inpainting en fonction de la taille du domaine D à reconstruire sur les images Lena (Tableau 4.4). Le nombre maximal d'itérations est étroitement lié au temps de calcul. Le tableau (4.5) regroupe le temps consommé par chaque méthode de diffusion en fonction du paramètre Tmax et en fonction de la taille du domaine D à reconstruire.

Diffusion	Tmax	Taille1	Taille2	Taille3	Taille4
Isotropique	25	1.9826	18.1404	47.6526	122.1509
	150	1.9826	14.3950	21.9344	33.0418
	300	1.9826	14.3950	21.9597	33.3857
	400	1.9826	14.3950	21.9597	33.3923
	500	1.9826	14.3950	21.9597	33.3923
	1000	1.9826	14.3950	21.9597	33.3923
Anisotropique	25	2.0115	178.0623	259.8463	147.6438
	150	1.9428	14.8267	21.7479	33.0004
	300	1.9428	14.2412	21.8221	33.3820
	400	1.9428	14.2326	21.8302	33.3839
	500	1.9428	14.2262	21.8345	33.3852
	1000	1.9428	14.2262	21.8345	33.3852

Tableau 4.4 : Erreurs d'inpainting obtenues par la diffusion en fonction du nombre maximal d'itérations Tmax.

Diffusion	Tmax	Taille1	Taille2	Taille3	Taille4
Isotropique	25	0.006	0.018	0.025	0.037
	150	0.035	0.105	0.150	0.198
	300	0.072	0.211	0.299	0.390
	400	0.100	0.292	0.397	0.519
	500	0.12	0.39	0.55	0.64
	1000	0.245	0.698	0.975	1.265
Anisotropique	25	0.20	0.25	0.27	0.30
	150	0.25	0.41	0.53	0.63
	300	0.33	0.70	0.92	1.01
	400	0.39	0.85	1.10	1.46
	500	0.45	0.92	1.22	1.85
	1000	0.71	1.68	2.26	3.19

Tableau 4.5 : Temps de calcul en S obtenus par les méthodes de diffusion en fonction du nombre maximal d'itérations T_{\max} et en fonction de la taille du domaine D à reconstruire.

Interprétation des résultats

Le tableaux (4.4) montrent que l'erreur d'inpainting varie en fonction de T_{\max} mais au delà d'une certaine valeur cette erreur reste constante. Ceci peut s'expliquer qu'après un certain nombre d'itération l'information à diffuser reste toujours la même. Le tableau et (4.5) montre par contre que le temps de calcul augmente lorsque le nombre d'itérations augmente même si le taux de croissance reste faible.

En comparant ces temps de calcul avec ceux consommés par les méthodes d'interpolation, on constate que l'inpainting est beaucoup plus rapide.

4.3.5 Influence du paramètre k dans le cas de la diffusion anisotropique

L'équation de diffusion anisotropique nécessite le choix du paramètre k . Le tableau (4.6) donne quelques valeurs de l'erreur d'inpainting en fonction du paramètre k et en fonction de la taille du domaine D à reconstruire.

Diffusion	k	Taille1	Taille2	Taille3	Taille4
Anisotropique	34	2.1125	14.2262	21.8345	33.7196
	66	1.9428	14.3464	21.8946	33.4492
	115	1.9688	14.3873	21.9170	33.3852
	150	1.9740	14.3903	21.9407	33.4013
	200	1.9781	14.3898	21.9476	33.4012

Tableau 4.6 : Erreurs d'inpainting obtenus par les méthodes de diffusion anisotropique en fonction du paramètre k et en fonction de la taille du domaine D à reconstruire.

Interprétation des résultats

Le tableau (4.6) montre les résultats d'inpainting ne varient pas d'une manière significative en fonction de paramètre k . Ce qui nous permet un choix plus aisé pour ce paramètre.

3.4.6 Application sur des images couleur

La figure (4.3) montre les résultats d'inpainting obtenus sur quelques images couleur par la diffusion. Les erreurs d'inpainting correspondantes sont regroupées dans le tableau (4.6).

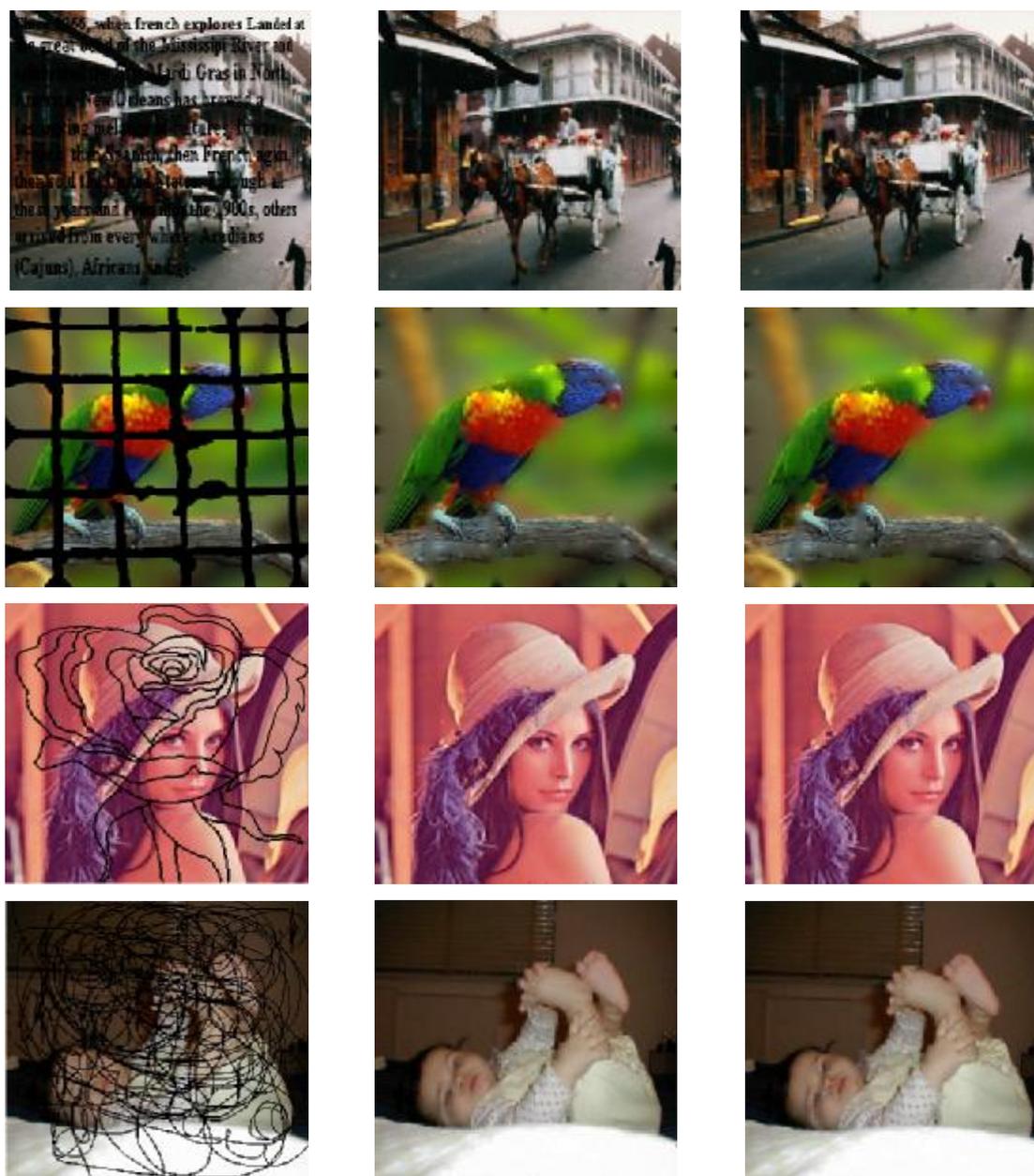


Fig. 4.3: Résultats d'inpainting des images couleur (1^{ère} colonne), par la diffusion isotropique (2^{ème} colonne) et anisotropique (3^{ème} colonne)

Diffusion	Charrette	Perroquet	Lena	Bébé
Isotropique	182.1586	276.6157	31.2063	70.5644
Anisotropique	176.2985	252.7351	30.2907	69.6626

Tableau 4.7 : Erreurs d'inpainting obtenues par la diffusion sur les images couleur

Interprétation des résultats

Il est clair que les résultats d'inpainting sont mieux mis en évidence sur les images couleur. Ces résultats confirment ceux obtenus sur les images en niveaux de gris à savoir la diffusion anisotropique permet une meilleure reconstruction que la diffusion isotropique, que l'interpolation bidimensionnelle est plus performante que l'interpolation par diffusion.

4.4 Conclusion

Nous avons appliqué dans ce chapitre une méthode d'inpainting par diffusion isotropique ou anisotropique de l'information contenu dans la frontière du domaine à reconstruire vers ce domaine. Le processus de diffusion est décrit par une EDP résolue par la technique des différences finies. Plusieurs résultats obtenus sur des images en niveaux de gris ou en couleur ont montré l'efficacité de cette approche à reconstruire des zones de dimensions et de tailles différentes de l'image est manquante. Cette approche s'est avérée plus efficace par rapport à l'approche par interpolation monodimensionnelle mais moins performante que celle basée sur l'interpolation bidimensionnelle. L'utilisation de la diffusion pour l'inpainting s'est avérée plus rapide par rapport à l'interpolation. Nous avons également noté que la diffusion anisotropique permet une meilleure reconstruction que la diffusion isotropique.

Conclusion générale

Nous avons abordé dans ce mémoire le problème de l'inpainting qui est formulé comme un problème de reconstruction de données dans une zone de l'image ou l'information est supposée manquante. Ce problème peut être formulé mathématiquement sous forme d'interpolation spatiale des valeurs de pixels inconnus de l'image.

Nous avons appliqué deux différentes approches pour résoudre le problème de l'inpainting. La première est basée sur l'interpolation des fonctions, la seconde sur le processus de diffusion. Trois techniques d'interpolation: plus proche voisin, linéaire et splines cubiques ont été utilisées et comparées. Ces méthodes sont utilisées selon deux stratégies : la première considère l'image comme une fonction monovariée tandis que dans la deuxième, l'image est considérée comme une fonction à deux variables. L'interpolation est alors monodimensionnelle dans le premier cas et bidimensionnelle dans le deuxième cas. Deux types de balayage Vidéo et Hilbert ont été utilisés afin de transformer une image bidimensionnelle en une fonction monodimensionnelle. Dans la deuxième approche, nous avons adapté le processus de diffusion isotropique et anisotropique afin de diffuser l'information disponible dans une zone de l'image vers la zone où l'information est supposée manquante ou détériorée. Le processus de diffusion est décrit par une EDP que nous avons résolue par la méthode des différences finies.

Plusieurs tests faisant intervenir différentes images en niveaux de gris ou en couleur avec différents domaines à interpoler ont montré d'une manière objective la supériorité des méthodes d'interpolation bidimensionnelle linéaire ou par les splines cubiques sur les méthodes d'interpolation monodimensionnelle. L'inpainting par diffusion s'est avérée plus efficace par rapport à l'approche par interpolation monodimensionnelle mais moins performante que celle basée sur l'interpolation bidimensionnelle. Cependant, l'interpolation bidimensionnelle demeure lente et que la diffusion présente un faible temps de calcul.

Ce travail peut être amélioré en diffusant non pas l'information niveau de gris ou couleur mais plutôt l'information texture. Dans ce cas l'introduction de méthodes de synthèse de la texture est nécessaire.

- [1] A. Quarteroni, R. Sacco et F. Saleri. Méthodes Numériques pour le Calcul Scientifique. Springer-Verlag, 2000.
- [2] F. Jędrzejewski. Introduction aux méthodes numériques. 2^{ème} édition, Springer-Verlag, 2005.
- [3] M. Brishoul. Reconstruction de données. Application à la dosimétrie des radiotéléphones. Thèse de doctorat, pp. 63-74, 2001.
- [4] V. Guinot, B. Cappelaere. Méthodes numériques appliquées. Cour de l'université de Polytech'Montpellier. 2005-2006.
- [5] I. Danaila, P. July, S. M. Kaber, M. Postel. Introduction au calcul scientifique. Dunod, 2005.
- [6] G. Koepfler. Équations aux dérivées partielles. Cour de l'université de René Descartes. France, 2001.
- [7] J.F. Scheid. Méthodes numériques pour la dynamique des fluides. Notes de cour de l'université de H. Poincaré, France, 2007- 2008.
- [8] T. A. Miloud. Méthodes Numériques. OPU, 2003.
- [9] E. Goncalvès. Méthodes, analyse et calculs numériques. Cour de l'institut polytechnique de Grenoble, France, 2005.
- [10] A. Ogier. Méthodes de restauration en imagerie médicale tri-dimensionnelle. Thèse de doctorat, université de Rennes 1, France , 2005.
- [11] R. Dérêche, O. Faugeras, Les EDP en Traitement des Images et Vision par Ordinateur, Traitement du Signal, rapport de recherche, INRIA, 1996.
- [12] P. Perona, J. Malik. Scale-space and edge detection using anisotropic diffusion. IEEE Trans Anal. Machine Intell., pp. 629-639, 1990.
- [13] C.A.Z. Barcelos and M.A. Batista. Image restoration using digital inpainting and noise removal. Image and Vision Computing, vol. 25, pp. 61–69, 2007.
- [14] J. Weickert. Anisotropic Diffusion in Image Processing. Teubner Stuttgart, Germany, 1998.
- [15] J. F. Aujol. Traitement d'image par approche variationnelle et équations aux dérivées partielles. Cour du Semestre d'enseignement UNESCO sur le traitement des images numériques. Tunisie, 2005.
- [16] S. Masnou and J. Morel. Level lines based disocclusion. In IEEE International Conference on Image Processing, Chicago, vol. 3, pp. 259-263, 1998.
- [17] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. Computer Graphics (SIGGRAPH 2000), pp. 417-424, 2000.

- [18] TF. Chan and J. Shen. Image Processing and Analysis. Society of Industrial and Applied Mathematics, 2005.
- [19] TF. Chan and J. Shen. Mathematical models for local non texture inpainting. SIAM Journal of Applied Mathematics, pp. 1019-1043, 2002.
- [20] TF. Chan and J. Shen. Morphologically Invariant PDE inpainting, UCLA CAM Report, 2001.
- [21] T. Chan and J. Shen. (2000, Mar.) Mathematical models for local deterministic inpainting. [Online]. Available: <http://www.math.ucla.edu>
- [22] TF. Chan and J. Shen. Non-texture inpainting by curvature-driven diffusions (CCD). Journal of Visual Communication and Image Representation, pp. 436-449, 2001.
- [23] T. F. Chan, S.H. Kang, and J. Shen. Euler's elastica and curvature based inpainting. SIAM J. Appl. Math., pp. 564-592, 2002.
- [24] A.telea. An image inpainting technique based on the fast marching method. Journal of graphics tools, vol. 9, No. 1, pp. 25-36, 2004.
- [25] M. M. Oliveira , B. bowen ,R. Mckenna , and Y. S chang. Fast digital image Inpainting. In proc. Of the visualization, Imaging, and an image processing IASTED conference, pp. 261-266, 2001.
- [26] A. Criminisi, P. Pérez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. IEEE Trans. Image Processing, pp. 1200-1212, 2004.
- [27] J. F. Aujol, S. Masnou, and S. Ladjal. Exemplar-based inpainting from a variational point of view. CMLA Preprint, 2008.
- [28] V. Do, G. Lebrun, L. Malapert, C. Smet, D. Tschumperlé. Inpainting d'Images Couleurs par Lissage Anisotrope et Synthèse de Textures. Color Image Inpainting by Anisotropic Smoothing and Texture Synthesis. France, 2006.
- [29] D. Tschumperlé and R. Deriche. Diffusion PDE's on vector valued images, local approach and geometric viewpoint. IEEE Signal Processing Magazine - Special issue on Mathematical Methods in Imaging, 2002.
- [30] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. IEEE Transactions on Image Processing, pp. 882-889, 2003.
- [31] C.A.Z. Barcelos and M.A. Batista. Image restoration using digital inpainting and noise removal. Image and Vision Computing, vol. 25, pp. 61-69, 2007.
- [32] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. PNAS, pp. 1591-1595, February 1996.

- [33] V. Venkatesh Mahalingam. Digital Inpainting Algorithm and Evaluation. Thèse de doctorat, université de Kentucky, 2010.
- [34] W. Wang and X. Qin. An Image Inpainting Algorithm based on CSRBF interpolation, vol. 12, No. 6, China, 2006.
- [35] I. Bloch, Y. Gousseau, H. Maître, D. Matignon, B. Pesquet-Popescu, F. Schmitt, M. Sigelle, F. Tupin. Le traitement d'images, Tom2, Polycopié de cours, ANIM Département TSI - Telecom-Paris, version 5.0, 2005.
- [36] K. Uhler and V. Skala. Radial basis function use for the restoration of damaged images. In Computer vision and graphics. Kluwer Academic Publishers, Vol.32, pp.839-844, 2006.

Quelques operateurs usuels

- **Gradient** : le gradient est un vecteur qui correspond à la direction et l'intensité de la plus grande variation d'un scalaire u , il est noté comme suite :

$$\text{grad } u = \nabla u = \left(\frac{\partial u}{\partial x_1}, \mathbf{L}, \frac{\partial u}{\partial x_n} \right)^T.$$

La norme du gradient est donnée par la relation suivante :

$$\|\nabla u\| = \sqrt{\left(\frac{\partial u}{\partial x_1} \right)^2 + \left(\frac{\partial u}{\partial x_2} \right)^2 + \mathbf{L} + \left(\frac{\partial u}{\partial x_n} \right)^2}.$$

- **Laplacien** : le Laplacien correspond à la divergence du gradient, soit à la régularité d'un signal. Il est défini par la formule suivante :

$$\Delta u = \frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} + \mathbf{L}$$

$$\Delta u = \nabla \cdot \nabla u$$

- **Divergence** : Notons aussi la divergence d'une fonction vectorielle $u = (u_1(x), u_2(x), \mathbf{L})$ où $x \in \Omega$

$$\text{div}(u) = \frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} + \mathbf{L}$$

$$\text{div}(\nabla u) = \Delta u.$$

- **Matrice Hessienne** : Pour une fonction u dont les dérivées partielles secondes existent, il existe une matrice Hessienne Hu que l'on calcule grâce à la formule suivante :

$$Hu = \begin{pmatrix} \frac{\partial^2 u}{\partial x_1^2} & \mathbf{K} & \frac{\partial^2 u}{\partial x_1 \partial x_n} \\ \mathbf{M} & \frac{\partial^2 u}{\partial x_i^2} & \mathbf{M} \\ \frac{\partial^2 u}{\partial x_n \partial x_1} & \mathbf{K} & \frac{\partial^2 u}{\partial x_n^2} \end{pmatrix}$$

- **Trace d'une matrice** : la trace d'une matrice est la somme des éléments de sa diagonale, pour une matrice 2×2 :

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \text{trace}(M) = a + d.$$

- **Tenseur** : un tenseur T est une matrice $n \times n$ qui est symétrique et demi-positive (ses valeurs propres λ_i sont positives ou nulles, u_i représente ses vecteurs propre). Un tenseur peut s'écrire :

$$T = \sum_{i=1}^n \lambda_i u_i u_i^T.$$