

**République Algérienne Démocratique et Populaire**

**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université Mouloud MAMMERY, Tizi-Ouzou**



**Faculté de Génie Electrique et d'Informatique  
Département d'Automatique**

**MEMOIRE DE FIN D'ETUDES**

En vue de l'obtention du diplôme

***MASTER ACADEMIQUE EN AUTOMATIQUE  
OPTION : COMMANDE DES SYSTEMES***

# *Thème*

*La restauration des parties manquantes dans l'image en utilisant les Equations aux Dérivées Partielles (EDPs)*

Proposé et dirigé par :  
Melle F. DORBANE

Présenté par :  
Melle MAHMOUDI Faiza  
Melle YAHIAOUI Lilia

Soutenu le : 03 / 10 /2013

*Promotion 2013*

## **Remerciements**

*Comme disait Marcel Proust : "Soyons reconnaissants aux personnes qui nous donnent du bonheur ; elles sont les charmants jardiniers par qui nos âmes fleurissent".*

*Tout d'abord Nous remercions Dieu le tout puissant pour nous avoir donné le courage, la patience, la santé et la volonte d'élaborer ce travail.*

*Nous tenons à exprimer Nos vifs remerciements, et une profonde reconnaissance a notre promotrice Melle F. DORBANE, qui nous a aidé et orienté dans l'élaboration de ce travail.*

*Nous remercions les membres de jury, devant qui nous avons l'honneur d'exposer notre travail, et qui nous font l'honneur de le juger.*

*Nous tenons aussi à remercier tous les enseignants qui ont contribué à notre formation, ainsi qu'à l'équipe pédagogique de notre département ; qu'ils reçoivent l'expression de notre profonde gratitude.*

*Nous plus vifs remerciements vont également à tous ceux qui on participé de prés ou de loin à la réalisation de ce travail mais qui ne sont pas cités ici, nous les remercions tous chaleureusement.*

*Nous ne saurions oublier nos chères familles pour leur soutien moral et matériel, ainsi que tous ceux et celles qui nous on apporter leur soutien et leur amitié tout le long de l'élaboration de ce travail.*

*Merci a tous*

# Dédicaces



## DEDICACES

*Je dédie ce modeste travail :*

*A mes très chers parents que je remercie beaucoup pour leur patience et sacrifice auxquels je ne pourrai rendre assez.*

*A mes très chères grandes mères paternelles et maternelles.*

*A mes très chères sœurs Assia, Thiziri, Thafath, Latifa et Dihia.*

*A tous mes amis(es) qui m'ont beaucoup aidé et soutenu sur tout Souhila, Samira, Sadjia, Amel...*

*A ma binôme et toute sa famille.*

*A tous ceux qui sont proches de mon cœur, que j'aime et dont je n'ai pas cité le nom.*

*Au bonheur des plus chers.*

*Je vous dédie ce modeste travail.*

*Faiza.*







## DEDICACES

*J e dédie ce modeste travail :*

*A mes très chers parents que je remercie beaucoup pour leur patience et sacrifice auxquels je ne pourrai rendre assez.*

*A mes très chères sœurs et leurs époux.*

*A mes très chers frères : **Sofiane** et **Abd enour**.*

*A mes petites princesse **mimane** et **Lina**, et à l'adorable **foufou**.*

*A **MEZIANE ABD ERREZAK** qui ma beaucoup soutenu, aidé et orienté, ainsi que pour toute sa famille.*

*A tous mes amies qui m'ont beaucoup aidé et soutenu sur tout Chana, Katia, Sarah, Samira, Souad, Malika, Kahina.*

*A ma binôme et toute la famille Mahmoudi.*

*A tous ceux qui sont proches de mon cœur, que j'aime et dont je n'ai pas cité le nom.*

*Au bonheur des plus chers.*

*Je vous dédie ce modeste travail.*



**Lilia**



# Sommaire

## SOMMAIRE

<b>Introduction générale .....</b>	<b>01</b>
<b>Chapitre I: les équations aux dérivées partielles</b>	
<b>I.1 Introduction.....</b>	<b>02</b>
<b>I.2 Définition d'une Equation aux Dérivées Partielles.....</b>	<b>02</b>
<b>I.3 Classification des équations aux dérivées partielles .....</b>	<b>03</b>
<b>I.4 Classification des EDPs linéaires du second ordre .....</b>	<b>04</b>
<b>I.4.1 Problèmes elliptiques.....</b>	<b>05</b>
<b>I.4.2 Problèmes parabolique.....</b>	<b>05</b>
<b>I.4.3 Problèmes hyperboliques .....</b>	<b>06</b>
<b>I. 5 Méthodes de résolution des équations aux dérivées partielles.....</b>	<b>06</b>
<b>I.5.1 Méthodes analytiques .....</b>	<b>07</b>
<b>I.5.2 Méthodes numériques .....</b>	<b>07</b>
<b>I.5.2.1 Élément finis .....</b>	<b>07</b>
<b>I.5.2.2 Volume finis .....</b>	<b>07</b>
<b>I.5.2.3 Différences finies .....</b>	<b>08</b>
<b>I.6 Notion de consistance, de convergence et de stabilité des méthodes numériques.....</b>	<b>10</b>
<b>I.7 Les EDPs en traitement d'images .....</b>	<b>11</b>
<b>I.7.1 Détection de contours .....</b>	<b>11</b>
<b>I.7.2 Segmentation .....</b>	<b>12</b>
<b>I.7.3 Filtrage d'image .....</b>	<b>12</b>
<b>I.8 Conclusion .....</b>	<b>13</b>

## Chapitre II Les méthodes d' inpainting basées sur les EDPs

<b>II.1 Introduction</b> .....	<b>12</b>
<b>II.2 Définition de l' inpainting</b> .....	<b>12</b>
<b>II.3 Principe de l' inpainting</b> .....	<b>12</b>
<b>II.4 Les applications d' inpainting</b> .....	<b>13</b>
<b>II.5 Les étapes de l' inpainting</b> .....	<b>15</b>
<b>II.6 Propriétés de l' inpainting</b> .....	<b>16</b>
<b>II.7 Méthodes d' inpainting</b> .....	<b>17</b>
<b>II.7.1 La méthode de Bertalmio, Sapiro, Caselles, Ballester</b> .....	<b>17</b>
<b>II.7.2 Approche de Perona et Malik</b> .....	<b>20</b>
<b>II.7.2.1 La diffusion isotrope</b> .....	<b>21</b>
<b>II.7.2.2 Diffusion anisotrope</b> .....	<b>22</b>
<b>II.7.3 Approche Chan et Shen (méthode de la variation totale</b> .....	<b>23</b>
<b>II.7.4 Méthode de Chan et Shen (CCD inpainting</b> .....	<b>24</b>
<b>II.7.5 Approche de Telea</b> .....	<b>25</b>
<b>II.8 Conclusion</b> .....	<b>26</b>

## Chapitre III Tests et résultats.

<b>III.1 Introduction</b> .....	<b>27</b>
<b>III.2 Les images et leurs masques utilisés</b> .....	<b>27</b>
<b>III.3 Résultats d' application de la méthode de Bertalmio</b> .....	<b>29</b>
<b>III.3.1 Tests sur les images en niveaux de gris</b> .....	<b>29</b>
<b>III.3.2 Interprétation des résultats</b> .....	<b>32</b>
<b>III.3.3 Tests sur les images couleurs</b> .....	<b>32</b>
<b>III.3.4 Interprétation des résultats</b> .....	<b>34</b>
<b>III.4.1 Résultats de la diffusion isotrope sur les images en niveaux de gris</b> .....	<b>34</b>
<b>III.4.2 Interprétation des résultats</b> .....	<b>37</b>

<b>III.4.3 Résultats de la diffusion isotrope sur les images couleurs .....</b>	<b>37</b>
<b>III.4.4 Interprétation des résultats.....</b>	<b>39</b>
<b>III.5.1 Résultats de la diffusion anisotrope sur les images en niveaux de gris.....</b>	<b>40</b>
<b>III.5.2 Interprétation des résultats.....</b>	<b>48</b>
<b>III.5.3 Résultats de la diffusion anisotrope sur les images couleurs .....</b>	<b>48</b>
<b>III.5.4 Interprétation des résultats.....</b>	<b>57</b>
<b>III.6 Calcule d’erreurs quadratiques moyennes des trois méthodes .....</b>	<b>57</b>
<b>III.7 Conclusion .....</b>	<b>58</b>
<b>Conclusion générale .....</b>	<b>56</b>
<b>Quelques opérateurs usuels</b>	
<b>Bibliographie</b>	

# **Introduction générale**

## Introduction Générale

Repeigner les parties dégradées des anciens tableaux de peinture peut apparaître comme une tâche facile pour certains, mais la réalité est toute autre, cette tâche est complexe et difficile à reproduire. Elle ne serait réalisée que par un artiste de restauration d'image qualifié. Cela a attiré l'attention des chercheurs notamment dans le domaine du traitement d'images pour l'application d'algorithmes sophistiqués pour remplacer les pièces perdues ou corrompues des données d'image (principalement de remplir des petites régions ou de supprimer de petits défauts).

Ce type d'algorithmes de reconstruction de données manquantes dans une image est appelé *Inpainting numérique* «digital inpainting». De manière générique, ce terme désigne le fait de déterminer, de la manière la plus automatique possible, la valeur des pixels considérés comme manquants dans une image.

Initialement, l'inpainting a été utilisé par des dessinateurs pour restaurer des anciens tableaux ou des gravures anciennes dégradées par l'usure du temps ou par une mauvaise manipulation. Par la suite, il 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. Récemment un travail intéressant peut être également appliqué pour réparer une vidéo endommagée [13].

Notre but dans ce mémoire est d'appliquer quelques méthodes de résolution du problème de restauration d'image basées sur les équations aux dérivées partielles.

Le mémoire est organisé en trois 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 des équations différentielles partielles (EDP). Dans le deuxième chapitre, nous allons présenter la problématique d'inpainting, ses applications, les différentes méthodes existantes dans ce domaine, ainsi que les méthodes adoptées pour reconstruire les images. Le troisième chapitre est consacré aux tests effectués et aux résultats obtenus lors de l'application des méthodes choisies. Nous finissons par une conclusion générale.

# **Chapitre I :**

## Les équations aux dérivées partielles

## I.1 Introduction

L'application des équations aux dérivées partielles (EDP) en traitement d'images a attiré depuis quelques années l'attention de plusieurs chercheurs en vision par ordinateur. Ceci est dû surtout au formalisme mathématique qui encadre toute approche à base d'EDP et qui permet de donner une bonne interprétation et justification des résultats obtenus par ces méthodes contrairement aux méthodes classiques et heuristiques en traitement d'images. La performance de ces méthodes en traitement d'image dépend en grande partie de la résolution numérique et de la mise en œuvre de ces EDP. Le principe des schémas numérique les plus utilisés se repose sur la discrétisation des dérivées du premier et second ordre [1].

Dans ce chapitre, nous allons présenter les principales généralités sur les équations aux dérivées partielles.

## I.2 Définition d'une Equation aux Dérivées Partielles

Une équation aux dérivées partielles est une équation mathématique contenant, en plus de la variable dépendante ( $u$  ci-dessous) et les variables indépendantes ( $x_1, x_2, \dots, x_d$  ci-dessous), une ou plusieurs dérivées partielles. Cette équation est ainsi de la forme:

$$F\left(x_1, x_2, \dots, x_d, u, \frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2}, \dots, \frac{\partial^2 u}{\partial x_1^2}, \frac{\partial^2 u}{\partial x_2^2}, \dots, \frac{\partial^m u}{\partial x_1^m}\right) = 0. \quad (1.1)$$

Où:

$u$  une fonction inconnue définie sur  $R^d$  à valeur dans  $R$  de variables  $x_1, x_2, \dots, x_d$ , qui est une fonction à déterminer,  $F$  est une fonction donnée.

$m$  est le degré de l'équation. 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  $(x_1, x_2, \dots, x_d) \in \Omega$ .

Une EDP est linéaire si l'équation est linéaire par rapport aux dérivées partielles de la fonction inconnue, par exemple l'équation (1.1) est dite linéaire si  $F$  est une fonction linéaire des quantités  $\frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2}, \dots, \frac{\partial^2 u}{\partial x_1^2}, \frac{\partial^2 u}{\partial x_2^2}, \dots, \frac{\partial^m u}{\partial x_1^m}$ .

Une EDP linéaire du premier ordre(en deux variables réelles indépendantes  $x_1, x_2$ ) s'écrit comme suit :

$$a(x_1, x_2) \frac{\partial u}{\partial x_1} + b(x_1, x_2) \frac{\partial u}{\partial x_2} + c(x_1, x_2)u = f(x_1, x_2) \quad (1.2)$$

De même, la forme générale de l'EDP linéaire du second ordre en deux variables indépendantes  $x_1, x_2$  est :

$$a(x_1, x_2) \frac{\partial^2 u}{\partial x_1^2} + b(x_1, x_2) \frac{\partial^2 u}{\partial x_1 x_2} + c(x_1, x_2) \frac{\partial^2 u}{\partial x_2^2} + d(x_1, x_2) \frac{\partial u}{\partial x_1} + e(x_1, x_2) \frac{\partial u}{\partial x_2} + g(x_1, x_2)u = f(x_1, x_2) . \quad (1.3)$$

Où :  $a, b, c, d, e, f, g$  sont des coefficients à variables  $x_1, x_2$  [2].

### I.3 Classification des équations aux dérivées partielles

Les EDPs sont classés en fonction de cinq critères [3] :

- **L'ordre de l'EDP** : L'ordre de l'EDP est l'ordre de plus grande dérivée présente dans l'équation.

$$\frac{\partial u}{\partial t} = \frac{\partial u}{\partial x} \quad \text{Première ordre,} \quad (1.4)$$

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \quad \text{Second ordre,} \quad (1.5)$$

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

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \quad \text{Deux variables } t \text{ et } x, \quad (1.6)$$

- **Linéarité** : Une EDP peut être linéaire ou non-linéaire. 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

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \sin x , \quad (1.7)$$

Par contre l'équation ci-dessous est non linéaire

$$\frac{\partial u}{\partial t} + u \frac{\partial^2 u}{\partial x^2} = 0, \quad (1.8)$$

- **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 suivantes est homogène.

$$a \frac{\partial^2 u}{\partial x_1^2} + b \frac{\partial^2 u}{\partial x_1 \partial x_2} + c \frac{\partial^2 u}{\partial x_2^2} + d \frac{\partial u}{\partial x_1} + e \frac{\partial u}{\partial x_2} + fu = 0, \quad (1.9)$$

$a, b, c, d, e, f$  Peuvent être des constantes où des fonctions des variables  $x_1$  et  $x_2$ .

- **Type de coefficients** : Si les coefficients  $a, b, c, d, e, f$  dans l'équation (1.9) sont constants donc L'équation est dite à coefficient constants.

#### I.4 Classification des EDPs linéaires du second ordre

La forme générale d'une EDP de second ordre est [4] :

$$a \frac{\partial^2 u}{\partial x_1^2} + b \frac{\partial^2 u}{\partial x_1 \partial x_2} + c \frac{\partial^2 u}{\partial x_2^2} + d \frac{\partial u}{\partial x_1} + e \frac{\partial u}{\partial x_2} + fu = g, \quad (1.10)$$

$a, b, c, d, e, f, g$  Peuvent être des constantes où des fonctions des variables  $x_1$  et  $x_2$ .

Dans notre travail on s'intéresse aux EDPs linéaires du second ordre, pour chaque type de problème : elliptique, parabolique, hyperbolique, on donne un exemple classique ainsi qu'un phénomène physique qu'il modélise.

- Les équations elliptiques décrivent les régimes permanents et satisfont la propriété :

$$b^2 - 4ac < 0$$

- Les équations paraboliques décrivent les procédés de diffusion et satisfont la propriété :

$$b^2 - 4ac = 0$$

- Les équations hyperboliques décrivent les systèmes oscillants et les mouvements d'ondes et satisfont la propriété :

$$b^2 - 4ac > 0$$

$\Delta = b^2 - 4ac$  : est le discriminant de l'équation (1.10)

#### I.4.1 Problèmes elliptiques

Soit  $\Omega$  un domaine borné de  $R^n$  et  $f$  est une fonction dans  $R$ .

Un exemple de problème elliptique est l'équation de poisson .Celle-ci s'écrit

$$\Delta u = f \tag{1.11}$$

Où :

$$\Delta u = \frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} + \dots + \frac{\partial^2 u}{\partial x_n^2} \quad \text{est le laplacien.}$$

Dans le cas particulier  $f = 0$ , l'équation (1.11) est appelée équation de laplace et ses solution sont appelées fonctions harmoniques.

#### I.4.2 Problèmes paraboliques

Un exemple de problème parabolique est l'équation de la chaleur. Elle était initialement conçue pour décrire le phénomène de conduction thermique. Elle définit la diffusion de la température d'un point d'un matériau donné à travers ses voisins. Ramené à l'imagerie numérique, on interprète la température comme étant la valeur des pixels voisins diffusant leurs valeurs au pixel traité. Ceci est régit par l'équation suivante :

$$\frac{\partial u}{\partial t} - \Delta u = f \tag{1.12}$$

Où :

$t$ : est la variable en temps.

$x$ : est la variable en espace.

### I.4.3 Problèmes hyperboliques

Des exemples les plus classiques d'EDP hyperbolique est :

- L'équation de transport :

$$\frac{\partial u}{\partial t}(x, t) + a \frac{\partial u}{\partial x}(x, t) = f(x, t) \quad (1.13)$$

Où  $a \in \mathbb{R}$  est la vitesse de transport.

- L'équation d'ondes :

$$\frac{\partial^2 u}{\partial t^2}(x, t) - \frac{\partial^2 u}{\partial x^2}(x, t) = f(x, t) \quad (1.14)$$

L'équation d'ondes et de la chaleur sont dites d'évolution car elles modélisent en général un phénomène instationnaire, évoluant avec le temps  $t$ . L'équation de poisson est dite stationnaire elles modélisent en général un phénomène à l'équilibre dans l'espace.

Les trois équations sont linéaires, c'est-à-dire qu'elles dépendent linéairement de l'inconnu  $u$ .

Plusieurs méthodes de résolution des EDP ont été proposées, nous sommes basées sur la méthode des différences finie

### I.5 Méthodes de résolution des équations aux dérivées partielles

La résolution des EDPs comporte certaines conditions aux limites qui sont appliquée au bord [5].

- **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 le bord  $\partial\Omega$ .

On dit que les conditions aux limites sont homogènes si les valeurs imposées sont nulles.

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

### **I.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 [6].

### **I.5.2 Méthodes numériques**

En vue du passage d'un problème exact (continu) au problème approché (discret), on dispose de plusieurs méthodes [7], les éléments finis, les volumes finis et les différences finies.

#### **I.5.2.1 Méthode des 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, par exemple, ses valeurs en certains points (les nœuds du maillage).

#### **I.5.2.2 Méthode des 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 est 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'images, la méthode la plus couramment employée pour discrétiser les EDPs est celle des différences finies.

### I.5.2.3 Méthode des 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.

Dans la suite de ce chapitre nous allons présenter la méthode principale utilisée pour la résolution des EDPs qui est la 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.

- **Formule de Taylor**

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

$$\frac{\partial u}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{u(x+\Delta x, y, z, t) - u(x, y, z, t)}{\Delta x}, \quad (1.15)$$

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

$$u(x + \Delta x, y, z, t) = u(x, y, z, t) + \frac{\Delta x}{1!} \frac{\partial u}{\partial x}(x, y, z, t) + \frac{\Delta x^2}{2!} \frac{\partial^2 u}{\partial x^2}(x, y, z, t) + \frac{\Delta x^3}{3!} \frac{\partial^3 u}{\partial x^3}(x, y, z, t) + \dots \quad (1.16)$$

En tronquant la série au premier ordre  $\Delta x$ , on obtient :

$$\frac{u(x+\Delta x, y, z, t) - u(x, y, z, t)}{\Delta x} = \frac{\partial u}{\partial x}(x, y, z, t) + \mathcal{O}(\Delta x), \quad (1.17)$$

Avec  $\mathcal{O}(\Delta x)$  est l'erreur d'approximation.

$$\frac{\partial u}{\partial x}(x, y, z, t) \approx \frac{u(x+\Delta x, y, z, t) - u(x, y, z, t)}{\Delta x}, \quad (1.18)$$

Ceci est appelé le schéma avant.

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

$$\frac{\partial u}{\partial x}(x, y, z, t) \approx \frac{u(x, y, z, t) - u(x - \Delta x, y, z, t)}{\Delta x}, \quad (1.19)$$

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

$$\frac{\partial u}{\partial x}(x, y, z, t) \approx \frac{u(x + \Delta x, y, z, t) - u(x - \Delta x, y, z, t)}{2\Delta x}, \quad (1.20)$$

- **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  $x_i$  partis régulièrement avec un pas d'espace  $\Delta x$ , tel que  $x_{i+1} = x_i + \Delta x$ . De même, le temps est décomposé en intervalle élémentaire de pas constant  $\Delta t$ .

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

$$\left(\frac{\partial u}{\partial x}\right)_i \approx \frac{u_{i+1} - u_i}{\Delta x}, \quad \text{Différence finie avant,} \quad (1.21)$$

$$\left(\frac{\partial u}{\partial x}\right)_i \approx \frac{u_i - u_{i-1}}{\Delta x}, \quad \text{Différence finie arrière,} \quad (1.22)$$

$$\left(\frac{\partial u}{\partial x}\right)_i \approx \frac{u_{i+1} - u_{i-1}}{2\Delta x}, \quad \text{Différence finie centrée,} \quad (1.23)$$

- **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 de  $u$ , on écrit

$$u_{i+1} = u_i + \Delta x \left(\frac{\partial u}{\partial x}\right)_i + \frac{\Delta x^2}{2} \left(\frac{\partial^2 u}{\partial x^2}\right)_i + \frac{\Delta x^3}{6} \left(\frac{\partial^3 u}{\partial x^3}\right)_i + \mathcal{O}(\Delta x^4), \quad (1.24)$$

$$u_{i-1} = u_i - \Delta x \left(\frac{\partial u}{\partial x}\right)_i + \frac{\Delta x^2}{2} \left(\frac{\partial^2 u}{\partial x^2}\right)_i - \frac{\Delta x^3}{6} \left(\frac{\partial^3 u}{\partial x^3}\right)_i + \mathcal{O}(\Delta x^4), \quad (1.25)$$

En faisant la somme de ces deux égalités, on aboutit à

$$u_{i+1} + u_{i-1} - 2u_i = \Delta x^2 \left( \frac{\partial^2 u}{\partial x^2} \right)_i + \mathcal{O}(\Delta x^4), \quad (1.26)$$

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

$$\left( \frac{\partial^2 u}{\partial x^2} \right)_i \approx \frac{u_{i+1} + u_{i-1} - 2u_i}{\Delta x^2}, \quad \text{Différence finie centrée,} \quad (1.27)$$

$$\left( \frac{\partial^2 u}{\partial x^2} \right)_i \approx \frac{u_{i+2} - 2u_{i+1} + u_i}{\Delta x^2}, \quad \text{Différence finie avant,} \quad (1.28)$$

$$\left( \frac{\partial^2 u}{\partial x^2} \right)_i \approx \frac{u_i - 2u_{i-1} + u_{i-2}}{\Delta x^2}, \quad \text{Différence finie arrière,} \quad (1.29)$$

## I.6 Notion de consistance, de convergence et de stabilité des méthodes numériques

Un certain nombre de notions est nécessaire lors de la résolution d'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  $\Delta x$  et  $\Delta t$  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.

## I.7 Les EDPs en traitement d'images

De nombreuses méthodes basées sur des EDPs sont apparues ces dernières années pour résoudre le problème de la segmentation, de la détection de contours, filtrage etc... des images. Dans cette section, nous n'aborderons que quelques points importants sur les méthodes basées sur les EDPs appliquées en traitement d'images [6].

### I.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. 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 :

$$E(C) = \int_0^1 \left( \alpha |v'(s)|^2 + \beta |v''(s)|^2 \right) ds - \lambda \int_0^1 |\nabla u(v(s))| ds, \quad (1.30)$$

Notons  $v(s) = (x(s), y(s))$  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  $u$ ,  $\alpha, \beta$  et  $\lambda$  sont des constantes réelles. Le but est donc de trouver la courbe  $C$  et des constantes  $\alpha, \beta$  et  $\lambda$  telle que  $E(C)$  soit minimale. Ceci peut être résolu en résolvant l'équation d'Euler Lagrange suivante :

$$\frac{\partial u}{\partial s} = \left( \alpha \frac{\partial^2 v}{\partial s^2} \right) - \left( \beta \frac{\partial^2 v}{\partial s^2} \right) - \nabla u(v(s)), \quad (1.31)$$

### I.7.2 Segmentation

La segmentation joue un rôle prépondérant en analyse d'images. Elle permet notamment de décomposer une image 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 :

$$E(C, c_1, c_2) = \int_{\Omega_{int}} (u_0 - c_1)^2 ds + \int_{\Omega_{ext}} (u_0 - c_2)^2 ds + v \int ds, \quad (1.32)$$

Où  $C$  est la courbe qui va permettre de détecter l'objet,  $\Omega_{int}$  et  $\Omega_{ext}$  représentent l'intérieur et l'extérieur de la courbe  $C$ .  $c_1$  et  $c_2$  sont défini comme étant la moyenne intérieure et extérieure de  $C$ . Afin de résoudre ce problème variationnel avec contrainte, une EDP peut être appliquée.

$$\left( \lambda_1 (u_0 - c_1)^2 - \lambda_2 (u_0 - c_2)^2 - v \cdot \nabla \left( \frac{\nabla u}{|\nabla u|} \right) \right) = 0, \quad (1.33)$$

$\lambda_1, \lambda_2$  et  $v$  sont des constantes.

### I.7.3 Filtrage d'image

Le filtrage d'image consiste à réduire les variations brusques d'intensité dans une même région homogène de l'image tout en préservant les transitions entre ces différentes régions. Les bruits sont ainsi réduits, les pixels parasites éliminés, et les contours sont plus définis. Parmi les grandes familles de filtres utilisées dans ce but on trouve les filtres basés sur l'utilisation des EDPs citons par exemple : filtre de choc, le principe de ce filtre est de rehausser le contour dans la direction de gradient de l'image. Une formulation des filtres de choc en vectoriel est donc :

$$\frac{\partial u}{\partial t} = -\text{sing}(u_{nn})u_n \quad (1.34)$$

Avec :

$u_n$  : La dérivée première de  $u$

$u_{nn}$  : La dérivée deuxième de  $u$

Cela revient à rehausser chaque composante dans une direction commune qui est la normale(  $n$ ) [8].

### **I.8 Conclusion**

Nous avons présenté dans ce chapitre quelques notions sur Les équations aux dérivées partielles, et les différentes méthodes utilisés pour la résolution de ce type d'équation nous sommes focalisé sur la méthode des différences finis, Nous avons également donné quelques exemples sur l'utilisation des EDPs en traitement d'image. Ces notions vont nous servir à la compréhension et à la résolution du problème de la restauration automatique de données manquantes dans l'image ou d'inpainting.

## **Chapitre II :**

Les méthodes d'inpainting basées sur  
les EDPs

## II.1 Introduction

L'inpainting d'image correspond aux techniques de restauration d'image qui permettent de remplir des parties manquantes ou détériorée dans une image. Plusieurs méthodes ont été proposées ces dernières années, dans ce chapitre, nous nous focalisons sur les méthodes basées sur les EDPs.

## II.2 Définition de l'inpainting

Le terme Inpainting fait référence aux artistes qui restaurent les tableaux de peinture en repeignant les parties dégradées, numériquement c'est l'opération équivalente pour les images numériques. De manière générique, ce terme désigne le fait de déterminer, de la manière la plus automatique possible, la couleur de pixels considérés comme manquants dans une image, c'est-à-dire, dont on ne connaît pas les valeurs à priori. En d'autres termes c'est une technique qui reconstruit les parties manquantes de l'image à partir des informations situées sur les bords de la zone à reconstruire [9].

## II.3 Principe de l'inpainting

Le problème d'inpainting d'image peut être formulé de la manière suivante. Etant donnée une image originale  $u_0$  définie sur un domaine complet  $\Omega \subset \mathbb{R}^2$ . Soit un sous-ensemble  $D \subset \Omega$  tel que l'information de l'image sur  $D$  est manquante ou détériorée [10] (voir figure II.1)

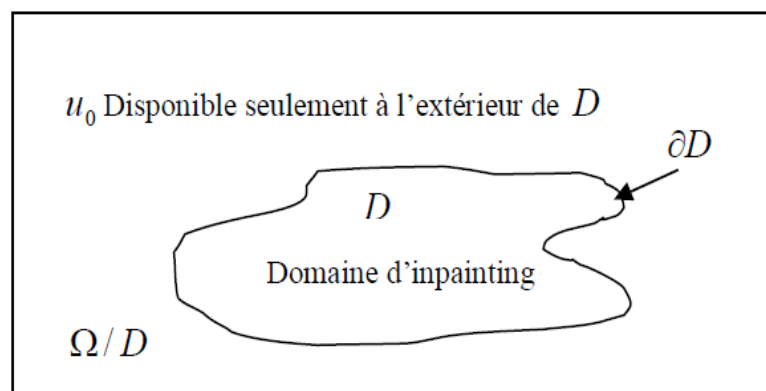


Figure II.1 Principe de l'inpainting

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  $\partial D$ ). Autrement dit, l'inpainting consiste à trouver une fonction  $u_D$  définie dans  $D$  tel que  $u_D$  est une approximation de  $u_0$  dans le domaine  $D$ .

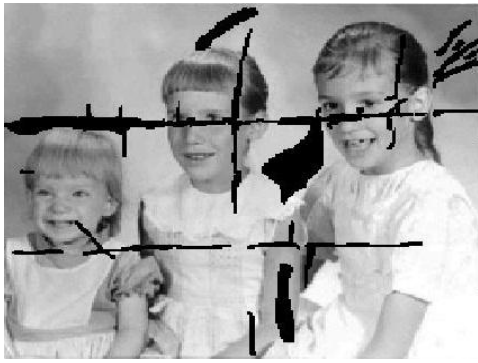
#### II.4 Les applications d'inpainting

Le but d'inpainting est de restaurer des zones manquantes dans une image et cela après numérisation de l'image et en réalisant la reconstruction des données manquantes. Les algorithmes d'inpainting ne se limitent heureusement pas à la restauration, mais nous pouvons obtenir d'autre application utile et impressionnante.

La figure II.2.a montre la possibilité d'effacer un texte dans une image. La figure II.2.b montre aussi comment pouvons nous reconstruire les données perdues lors de l'acquisition ou de la transmission d'image ainsi la restauration d'anciennes photos voir figure II.2.c. L'inpainting est utile si l'on souhaite effacer des parties de l'image des personnes, des objets...comme le montre la figure II.2.d et figure II.2.e. Une autre application intéressante consiste à zoomer l'image (figure II.2.f).



Figure II .2.a : suppression d'un texte dans une image



**Figure II. 2. b :** Restauration et amélioration d'image



**Figure II. 2. c :** suppression d'une tache



**Figure II. 2. d :** suppression d'un objet



Figure II. 2. e : suppression d'une personne



Figure II.2.f : le zoom d'image

## II.5 Les étapes de l'inpainting

L'inpainting nécessite deux étapes :

- Repérer l'information manquante et créer un masque qui délimite de façon précise les zones à reconstruire.

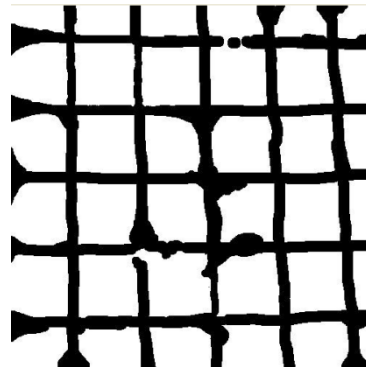
On définit un masque binaire par  $M : \Omega \rightarrow \{0,1\}$  tel que

$$M(x,y) = \begin{cases} 0 & \text{si le pixel } (x,y) \text{ se situe dans la zone à reconstruire } D. \\ 1 & \text{sinon.} \end{cases} \quad (2.1)$$

Comme le montre la figure II.3



(a) Image à restaurer



(b) son masque binaire

**Figure II.3** : création d'un masque binaire

- Reconstruire la partie manquante de l'image en transportant les valeurs des pixels et les informations situées sur les bords  $\partial D$  (On note  $\partial D$  le bord de la zone à reconstruire) à l'intérieur de la zone  $D$  à reconstruire, et fusionner la région retouchée à l'image d'une manière transparente en utilisant des techniques pour diffuser l'information des bords vers l'intérieur de la zone à reconstruire

## II.6 les 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 pratique.

**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.

Deux facteurs influent généralement sur l'inpainting :

**Complexité du domaine à interpoler :** Ce domaine peut être de forme ou de taille quelconque. Il dépend généralement 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.

## II.7 Méthodes d'inpainting

Il existe plusieurs méthodes de résolution du problème d'inpainting, on s'intéresse dans cette partie aux méthodes basées sur les EDPs.

### II.7.1 La méthode de Bertalmio, Sapiro, Caselles et Ballester

Bertalmio et al [9] ont été les premiers à résoudre le problème de la restauration des parties manquantes dans l'image. Cette méthode a pour principe de poursuivre les lignes isophotes existantes aux abords du masque vers l'intérieur de celui-ci. Ainsi, l'algorithme modifie les pixels, de la partie manquante (le domaine  $D$ ), du bord  $\partial D$  vers les pixels de l'intérieur au fur et à mesure des itérations. Notons que le gradient d'un pixel représente la direction de la plus forte variation des valeurs, i.e. la direction orthogonale aux contours. Ainsi nous définissons les isophotes (direction de la plus faible variation) comme étant la rotation de 90 degrés du vecteur gradient (voir figure II.4).

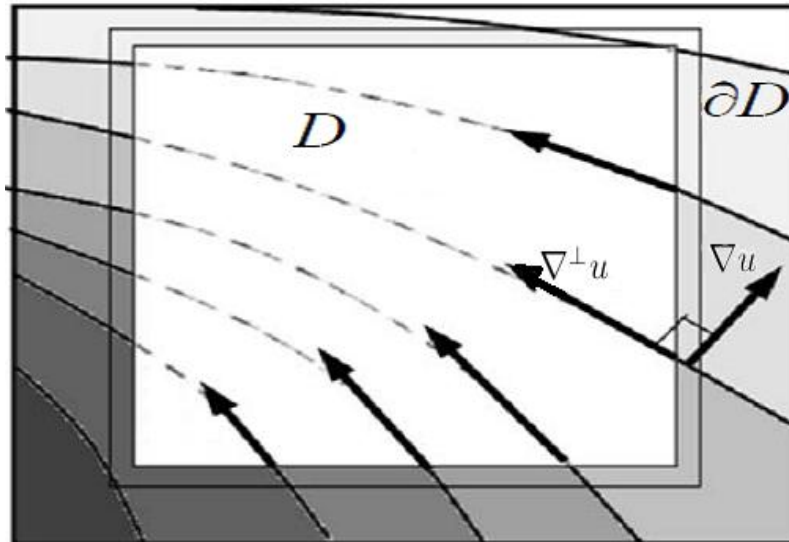


Figure II.4 : la direction des isophotes.

- **Principe de la méthode**

En premier, nous fusionons l'image originale sans détériorations avec un masque binaire qui indique les parties à reconstruire qui sont représentées par les parties noires, nous obtenons l'image à reconstruire  $u(i, j, 0) = u^0(i, j)$  à l'itération 0. A chaque itération  $n$ , nous construisons l'image  $u^{n+1}(i, j)$  connaissant l'image précédente  $u^n(i, j)$ , nous aurons alors quand  $\lim_{n \rightarrow \infty} u^{n+1}(i, j) = u_R(i, j)$  l'image résultante (ou l'image reconstruite). Nous traiterons que les pixels de la partie manquante (le domaine  $D$ ). Cette procédure est donnée par la formule suivante :

$$u^{n+1}(i, j) = u^n(i, j) + \Delta t u_t^n(i, j) \quad \forall (i, j) \in D \quad (2.2)$$

Avec

$\Delta t$  : taux de changement à l'intérieur du domaine  $D$ .

$u_t^n(i, j)$  : est la valeur qu'il faut ajouter à celle du pixel  $(i, j)$  dans le domaine  $D$ , afin que

la ligne isophote sur la frontière  $\partial D$  soit poursuivie.  $u_t^n$  est donnée par la relation suivante :

$$u_t^n(i, j) = B^n(i, j) |\nabla u^n(i, j)| \quad (2.3)$$

Avec

$B^n$  : est la projection de  $\delta \vec{L}^n$  sur la direction isophote représentée par  $\vec{N}(i, j, n)$ . Il est donné par l'équation suivante :

$$B^n(i, j) = \delta \vec{L}^n(i, j) \cdot \frac{\vec{N}(i, j, n)}{|\vec{N}(i, j, n)|} \quad (2.4)$$

Le vecteur  $\vec{N}(i, j, n)$  correspond à la direction isophote au pixel de coordonnées  $(i, j)$  de l'image  $u^n(i, j)$  et de norme unitaire. Il est calculé de la manière suivante :

$$\vec{N}(i, j, n) = \frac{(-c_y^n(i, j), c_x^n(i, j))}{\sqrt{c_x^n(i, j)^2 + c_y^n(i, j)^2 + \varepsilon}} \quad (2.5)$$

$C_y^n$  et  $C_x^n$  sont respectivement les composantes en  $x$  et en  $y$  du gradient de l'image  $u^n(i, j)$ , et  $\varepsilon$  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 II.4 où la zone en blanc représente la région à restaurer.

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

$$\vec{\delta L}(i, j) = (L^n(i+1, j) - L^n(i-1, j), L^n(i, j+1) - L^n(i, j-1)) \quad (2.6)$$

$$\vec{\delta L}(i, j) = \vec{\nabla} L^n(i, j) \quad (2.7)$$

$L_n(i, j)$ : est l'information à propager du contour  $\partial D$  vers le domaine  $D$ , elle est donnée par le Laplacien de l'image  $u^n$ , comme donné dans la relation suivante :

$$L^n(i, j) = \Delta u^n(i, j) = u_{xx}^n(i, j) + u_{yy}^n(i, j) \quad (2.8)$$

$u_{xx}^n(i, j)$  et  $u_{yy}^n(i, j)$  indiquent la deuxième dérivée de  $u$  par rapport à l'axe des  $x$  et des  $y$ .

La norme du gradient  $|\nabla u^n(i, j)|$  introduite dans l'équation (2.3) est calculée d'une manière différente, elle dépend du signe de  $B^n$  comme le montre la relation suivante :

$$|\nabla u^n(i, j)| = \begin{cases} \sqrt{(u_{x_{bm}}^n)^2 + (u_{x_{fM}}^n)^2 + (u_{y_{bm}}^n)^2 + (u_{y_{fM}}^n)^2} & \text{si } B^n > 0 \\ \sqrt{(u_{x_{bM}}^n)^2 + (u_{x_{fM}}^n)^2 + (u_{y_{bM}}^n)^2 + (u_{y_{fM}}^n)^2} & \text{si } B^n < 0 \end{cases} \quad (2.9)$$

Les indices  $x$  et  $y$  indiquent la première dérivée par rapport aux axes  $x$  et  $y$ , par contre les indices  $f$  (forward) et  $b$  (backward) indiquent le type de différence finie à utiliser "différence finie avant" pour  $f$  et "différence finie arrière" pour  $b$  selon l'axe  $x$  ou  $y$ . Les indices  $m$  et  $M$  signifient que le choix respectivement du minimum ou bien du maximum entre la valeur calculée et zéro.

En résumé, l'ordre des opérations à exécuter à partir de l'image détériorée jusqu'à avoir l'image résultante est donné comme suit :

A chaque itérations  $n$ , et pour chaque pixel  $(i,j)$  de la partie à reconstruire nous calculons :

- 1) le vecteur de la normale  $\vec{N}(i, j, n)$  (équation 2.5),
- 2) la variation du Laplacien  $\vec{\delta L}$  (équation 2.6 en utilisant l'équation 2.8),
- 3) le  $B^n(i, j)$  (équation 2.4),
- 4) la valeur ajoutée au pixel à reconstruire  $u_t^n(i, j)$  (équation 2.3),
- 5) la nouvelle valeur du pixel  $(i,j)$   $u^{n+1}(i, j)$  (équation 2.2).

En suivant toutes ces opérations dans l'ordre indiqué, nous obtenons l'image résultante ou restaurée. La restauration est faite pixel par pixel.

### II.7.2 Approche de Perona et Malik

L'idée de Perona et Malik [11] pour résoudre le problème d'inpainting a été d'établir une analogie entre la restauration d'une image et le phénomène physique 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. Nous imaginons diffuser de proche en proche les niveaux de gris dans une image. Ainsi, les irrégularités des niveaux de gris diminueront jusqu'à on retrouve un niveau de gris uniforme. Cette équation de diffusion connue sous le nom d'équation de la chaleur, peut se mettre sous la forme de l'EDP parabolique linéaire suivante :

$$\begin{cases} \frac{\partial u}{\partial t} = \text{div}(f(\nabla u)) & \text{dans } D \\ u(i, j, 0) = u_0 \end{cases} \quad (2.10)$$

Avec  $div$  est un opérateur de divergence

$f$  est une fonction dépendante du gradient de  $u$ . La démarche adoptée pour effectuer l'inpainting en utilisant l'EDP de diffusion consiste à parcourir le domaine et reconstituer pixel par pixel de la partie détériorée dans l'image à reconstruire  $u_0$ ,

### II.7.2.1 La diffusion isotrope

Dans le cas d'une diffusion isotrope la fonction  $f$  dans (2.10) est une constante unité, donc l'équation (2.10) deviendra :

$$\begin{cases} \frac{\partial u}{\partial t} = div(\nabla u) & \text{dans } D \\ u(i, j, 0) = u_0 \end{cases} \quad (2.11)$$

- **Principe de la méthode**

Étant donnée une image originale  $u_0$ , définie sur domaine  $\Omega \subset R^2$ ,  $D \subset \Omega$  et représente les parties manquante de l'image. Pour la reconstruction de ces parties manquantes, il suffit de propager ou diffuser les informations existantes sur la frontière  $\partial D$  à l'intérieure de zones manquantes  $D$ , en utilisant l'équation de chaleur (2.11) qui est équivalente à la relation suivante :

$$\begin{cases} \frac{\partial u}{\partial t} = \Delta u & \text{dans } D \\ u(i, j, 0) = u_0 \end{cases} \quad (2.12)$$

La résolution de cette équation (2.12) d'une manière numérique nécessite le passage de l'équation de la chaleur du domaine continue au domaine discret. Après discrétisation de cette équation nous obtenons l'équation suivante :

$$u^{n+1}(i, j) = u^n(i, j) + dt \Delta u^n(i, j) \quad (2.13)$$

Avec  $\Delta u^n(i, j)$  est le Laplacien au pixel  $(i, j)$ , qui est la valeur ajoutée à chaque itération au pixel à reconstruire jusqu'à convergence.

La méthode de diffusion est décrite de cette manière :

A chaque itérations  $n$ , et pour chaque pixel  $(i, j)$  de la partie à reconstruire nous calculons :

1) le Laplacien  $\Delta u^n(i, j)$ ,

2) la nouvelle valeur du pixel  $(i,j)$ ,  $u^{n+1}(i,j) = u^n(i,j) + dt \Delta u^n(i,j)$  (équation 2.13)

### II.7.2.2 Diffusion anisotrope

Pour pallier aux problèmes de la diffusion isotrope (dans le cas de la restauration de l'image bruitée, apparition du flou et le lissage des contours), Perona et Malik ont proposés de rendre la diffusion dépendante du contenu de l'image et en particulier des contours. Dans le cas de la diffusion anisotrope la fonction  $f$  prend la forme suivante :

$$f = g(|\nabla u|) \tag{2.14}$$

Avec  $g$  est une fonction décroissante positive. En remplaçant la fonction  $f$  dans l'équation (2.11) nous obtenons l'équation qui régit la diffusion anisotrope suivante :

$$\begin{cases} \frac{\partial u}{\partial t} = \text{div} ( g(|\nabla u|)\nabla u ) \text{ dans } D \\ u(x,y,0) = u_0 \end{cases} \tag{2.15}$$

- **Principe de la méthode**

Cette méthode consiste à effectuer une diffusion conditionnelle, c.-à-d forte diffusion dans les zones à faible gradient (zones homogènes) et faible diffusion dans les zones à fort gradient (contours), ce phénomène est contrôlé par la fonction  $g$  qui est une fonction décroissante positive dépendante de la norme du gradient comme le montre l'équation 2.14, cette fonction doit satisfaire les conditions suivantes :

$$\begin{cases} g(0) = 1 \\ \lim_{k \rightarrow \infty} g(k) = 0 \end{cases} \tag{2.16}$$

Les fonctions décroissantes proposées par Perona et Malik sont :

$$g(|\nabla u|) = \exp(-(|\nabla u|/k)^2) \tag{2.17}$$

$$g(|\nabla u|) = \frac{1}{1+(|\nabla u|/k)^2} \tag{2.18}$$

Le paramètre  $k$  est appelé seuil ou barrière de diffusion, il joue un rôle important dans le comportement de la diffusion, les contours correspondant à un gradient supérieur à  $K$  sont préservés. C'est un paramètre à fixer.

Après discrétisation de l'équation 2.15, nous obtenons l'équation suivante :

$$u^{n+1}(i, j) = u^n(i, j) + dt(\nabla(g(|\nabla u|)).\nabla u + g(|\nabla u|).\Delta u) \quad (2.19)$$

La méthode de diffusion anisotrope est décrite de cette manière :

Choisir la valeur de la barrière de diffusion  $k$ .

Choisir la forme de la fonction  $g$  ( $\exp(-(|\nabla u|/k)^2)$  ou  $\frac{1}{1+(|\nabla u|/k)^2}$  )

A chaque itérations  $n$ , et pour chaque pixel  $(i, j)$  de la partie à reconstruire nous calculons :

1) la valeur ajoutée  $\nabla(g(|\nabla u|)).\nabla u + g(|\nabla u|).\Delta u$

2) la nouvelle valeur du pixel  $(i, j)$

$$u^{n+1}(i, j) = u^n(i, j) + dt (\nabla(g(|\nabla u|)).\nabla u + g(|\nabla u|).\Delta u) \quad (\text{équation 2.19}).$$

### II.7.3 Approche Chan et Shen (méthode de 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.

Elle est connue sous le nom de TV inpainting .cette approche variationnelle consiste à minimiser l'énergie suivante [12] :

$$J(u) = \int_{\Omega/D} |\nabla u| dx + \frac{\lambda}{2} \int_D (u - u_0)^2 dx. \quad (2.20)$$

Le premier terme de cette énergie correspond à la variation totale de l'image dans le domaine  $\Omega/D$  alors que le deuxième terme définit l'adéquation entre l'image recherche  $u$  et l'image initiale  $u_0$  à l'intérieur du domaine.

$\lambda$  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 :

$$\frac{\partial u}{\partial t} = \nabla \cdot \left( \frac{1}{|\nabla u|} \cdot \nabla u \right) + \lambda e(u_0 - u). \quad (2.21)$$

L'équation (2.21) est valable sur l'intégralité du domaine  $\Omega$ . Si on considère juste son premier terme (valable dans le domaine D), cette équation peut être simplifiée à l'EDP suivante :

$$\frac{\partial u}{\partial t} = \nabla \cdot \left( \frac{1}{|\nabla u|} \cdot \nabla u \right). \quad (2.22)$$

Cette approche est employée pour reconstruire des petites régions, car elle ne peut pas prolonger les isophotes lorsque la zone à reconstruire est grande.

### II.7.4 Méthode de Chan et Shen (CCD inpainting)

Chan et Shen ont proposé un nouveau model d'inpainting basé sur la diffusion conduite par la courbure (CCD inpainting) en s'inspirant du model d'inpainting à variation totale. Dans ce nouveau modèle, ils ont rajoutés un autre terme dépendant de la courbure  $k$ . Ce terme est  $g(k)$  nommé force de diffusion. Le model d'inpainting par CCD s'exprime alors par l'EDP suivante [12] :

$$\frac{\partial u}{\partial t} = \nabla \cdot \left( \frac{g(k)}{|\nabla u|} \cdot \nabla u \right) + \lambda e(u_0 - u). \quad (2.23)$$

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 :

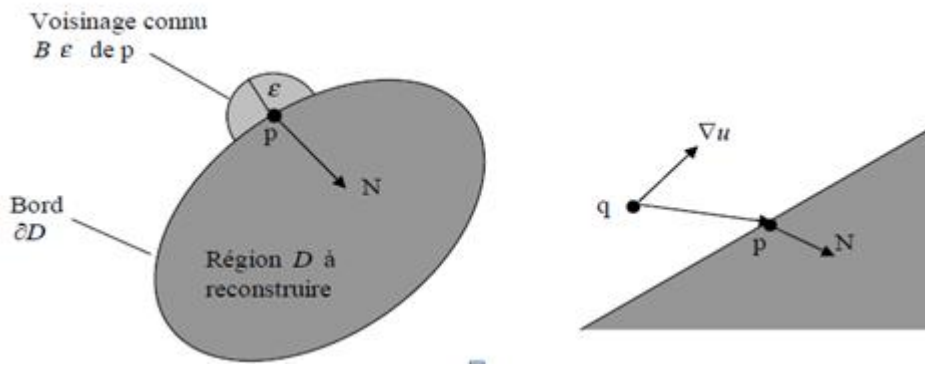
$$g(k) = \begin{cases} 0 & \text{si } k = 0 \\ +\infty & \text{si } k = +\infty \end{cases} \quad (2.24)$$

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

### II.7.5 Approche de Telea

L'algorithme, proposé par Telea en 2004 [6], 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.

Pour mieux expliquer cet algorithme, Nous considérons l'exemple de la figure II.5, dans laquelle, nous cherchons l'information au point  $p$  situé à la frontière  $\partial D$  de la région à inpainter  $D$ .



**Figure II.5** : Principe de l'inpainting par la méthode de Télea

Nous définissons un petit voisinage  $B_\varepsilon(p)$  autour de  $p$  de la taille  $\varepsilon$  des pixels de l'image dont les niveaux de gris sont connus :  $B_\varepsilon(p) = \{q, q \notin D\}$ . Pour  $\varepsilon$  assez petit, nous considérons 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  $q$ .

$$u_q(p) = u(q) + \nabla u(q)(p - q). \quad (2.25)$$

Ainsi, le niveau de gris au point  $p$  est remplacé par une fonction de tous les points  $q$  de  $B_\varepsilon(p)$  en additionnant les estimations de tous les points  $q$ , pondérés par une fonction de poids  $w(p, q)$  (fonction de pondération) comme le montre la relation suivante :

$$u(p) = \frac{\sum_{q \in B_\varepsilon(p)} w(p, q)[u(q) + \nabla u(q)(p - q)]}{\sum_{q \in B_\varepsilon(p)} w(p, q)}. \quad (2.26)$$

Cette opération est répétée pour chaque point  $p$  appartenant à  $\partial D$  jusqu'à ce que le domaine  $D$  soit complètement inpainté.

### II.8 Conclusion

Nous avons présenté dans ce chapitre quelques notions élémentaires sur l'inpainting, et les différentes méthodes d'inpainting basées sur des équations aux dérivées partielles (EDPs). Parmi ces méthodes, nous avons choisies trois à savoir : la méthode de Bertalmio, la méthode de diffusion isotrope et celle de diffusion anisotrope, nous présenterons les résultats de ces méthodes dans le chapitre suivant.

# **Chapitre III :**

## **Tests et Résultats**

### III.1 Introduction

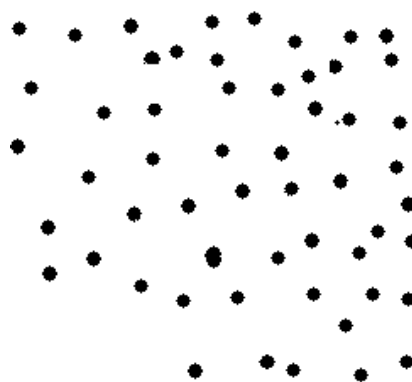
Nous mettrons en pratique l'étude théorique que nous avons abordée dans le chapitre précédent, En effet, nous appliquerons les méthodes de Bertalmio, la diffusion isotrope et anisotrope pour reconstruire nos différentes images. Plusieurs tests seront effectués sur différentes images (niveaux de gris et couleurs) pour évaluer ces méthodes. L'implémentation de notre application se fera avec le langage de programmation MatlabR2009 sur un HP Compaq n7400 (Processeur Intel(R) core (TM) 2CPU [T5500@1.66](#) GHZ et 1528MB de RAM.

### III.2 Les images et leurs masques utilisés

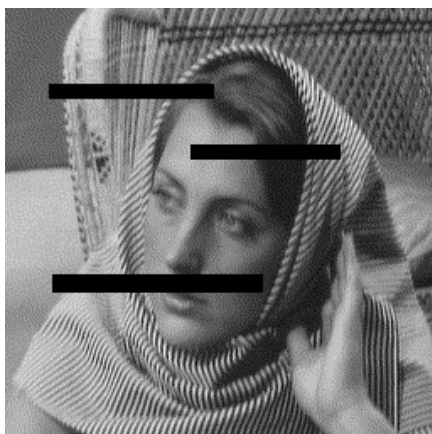
Pour tester les trois méthodes adoptées, nous avons choisies différentes images. Nous avons utilisés les images Cameramen (figure III.2.a) et woman (figure III.2.c) qui sont en niveau de gris et leurs masques binaires sont respectivement : figure III.2.b et figure III.2.d qui définissent les zones à reconstruire.



(a) image cameramen



(b) son masque binaire



(c) image women



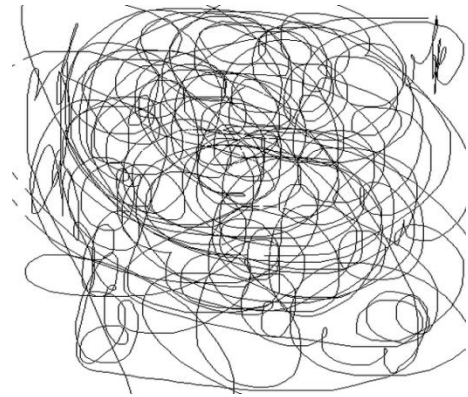
(d) son masque binaire

**Figure III.1** : les images en niveaux de gris à restaurer et leurs masques.

En plus des images en niveaux de gris, nous avons testé ces méthodes sur des images couleurs qui sont : l'image bébé (figure III.3.e), avion (figure III.3.g), perroquet (figure III.3.i) et leurs masques correspondants.



(e) image bébé



(f) son masque binaire



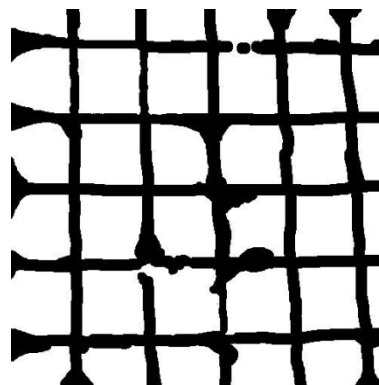
(g) image avion



(h) son masque binaire



(i) Image perroquet



(j) son masque binaire

**Figure III.2** : les images couleurs à restaurer et leurs masques.

Pour réaliser une évaluation objective, nous calculerons un critère donnant la qualité de l'image restaurée. Nous avons choisie d'utiliser le critère de mesure de l'erreur quadratique moyenne(en anglais Mean Square Error MSE) entre l'image originale  $u(i, j)$  et l'image reconstruite  $u_r(i, j)$  donnée par la formule suivante :

$$MSE = \frac{\sum_{i=1}^N \sum_{j=1}^M (u(i, j) - u_r(i, j))^2}{N \times M} \quad (3.1)$$

$u_r(i, j)$  : image reconstruite

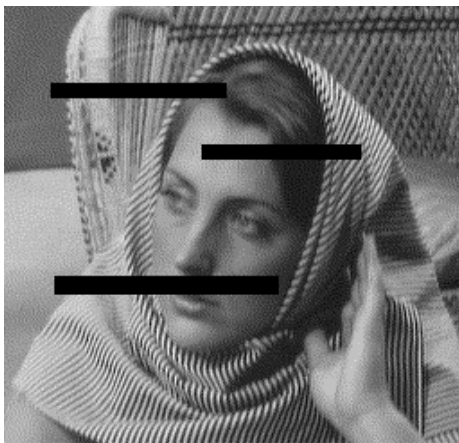
$N$  et  $M$  : sont le nombre de ligne et de colonne.

### III.3 Résultats d'application de la méthode de Bertalmio, Sapiro, Caselles et Ballester

Cette méthode dépend de deux paramètres qui sont  $dt$  et le nombre d'itérations  $N$ , nous fixons l'un et nous faisons varier l'autre.

#### III .3.1 Résultats des images en niveaux de gris.

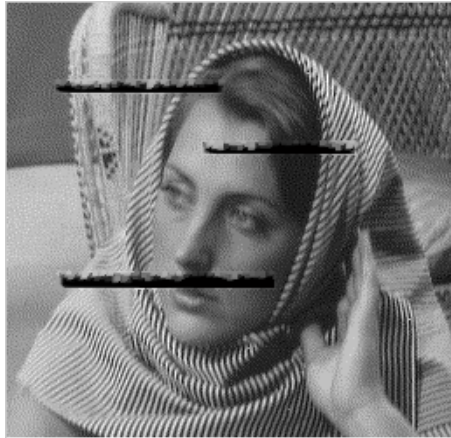
1<sup>er</sup> cas On fixe  $dt=0.001$  et on fait varier  $N$



(a) Image à restaurer



(b) Image à restaurer



(c)

nombre d'itérations  $N=200$



(d)



(e)

Nombre d'itérations  $N=2000$



(f)



(g)

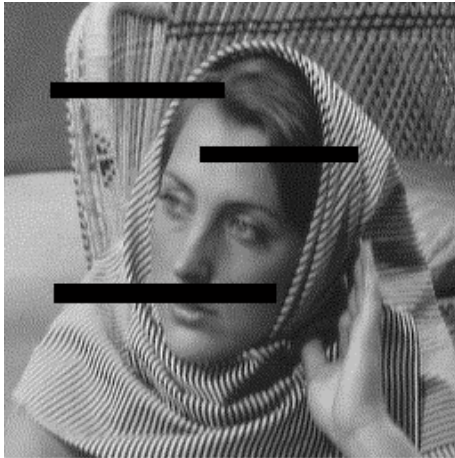
Nombre d'itérations  $N=20000$



(h)

**Figure III.3:** les résultats de la méthode Bertalmio ( $dt=0.001$ , varier  $N$ ).

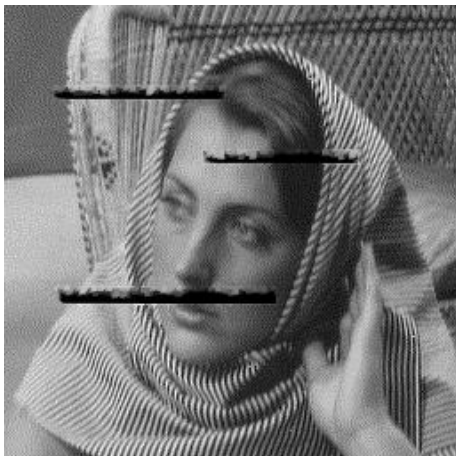
*2<sup>ème</sup> cas On fixe  $N=20000$  et on fait varier  $dt$*



(a) Image à restaurer



(b) image à restaurer



(c)  $dt=0.00001$



(d)



(e)  $dt=0.0001$



(f)



(g)

dt=0.001



(h)

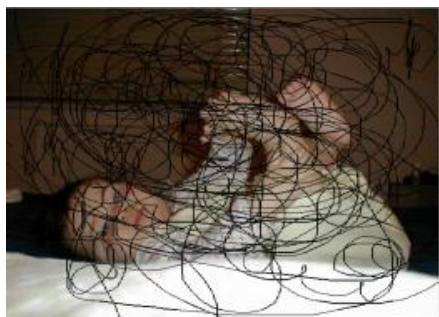
**Figure III.4:** les résultats de la méthode Bertalmio ( $N=20000$ , varier  $dt$ ).

### III.3.2 Interprétation des résultats

Pour l'image women, On remarque les zones perdues sont reconstruites mais la restauration n'est pas parfaite (voir figure III.4.g et III.3.g), après un nombre d'itération  $N=20000$  et  $dt=0.001$ , surtout au niveau du foulard et la chaise parce qu'ils sont texturés. Concernant l'image cameramen, on remarque une bonne restauration sauf qu'elle n'est pas complète dans les deux cas (voir figure III.4.h et figure III.3.h) même si nous augmentons le nombre d'itérations  $N$  ou le  $dt$ . Pour les images en niveau de gris le temps de simulation n'est pas grand (environ 10 mn). Nous concluons que cette méthode ne reconstruit pas la texture et elle ne converge pas rapidement.

### III.3.3 Résultats des images couleurs.

*1<sup>er</sup> cas On fixe  $dt=0.001$  et on fait varier  $N$*



(a) Image à restaurer



(b) Image à restaurer



(c) Image à restaurer



(d)



(e)

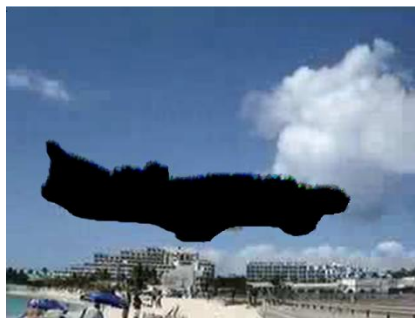


(f)

N=50



(g)



(h)



(i)

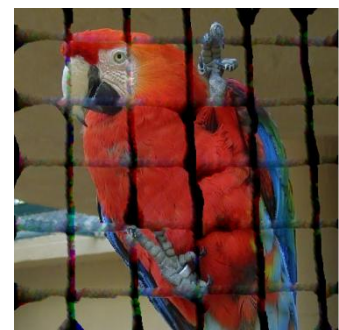
N=500



(h)



(i)



(j)

N=5000



N=10000 (k)

Figure III.5: les résultats de la méthode Bertalmio pour les images couleurs

### III.3.4 Interprétation des résultats

Nous n'avons pas présenté les résultats du deuxième cas (fixer  $N$  et varier  $dt$ ), parcequ'ils sont similaires aux résultats du premier cas.

Pour l'image bébé, nous remarquons que les zones perdues sont reconstruites d'une manière parfaite, dans la figure III.5.h, nous retrouvons aucunes rayures. Par contre pour les images avion et perroquet la restauration n'est pas terminée (voir figure III.5.k et figure III.5.j), nous avons constaté que même si nous augmentons davantage le nombre d'itération  $N$  ou  $dt$  le résultat est toujours le même, nous ne remarquerons aucun changement. Pour cette méthode le temps de simulation est important (environ 3h) pour les images couleurs.

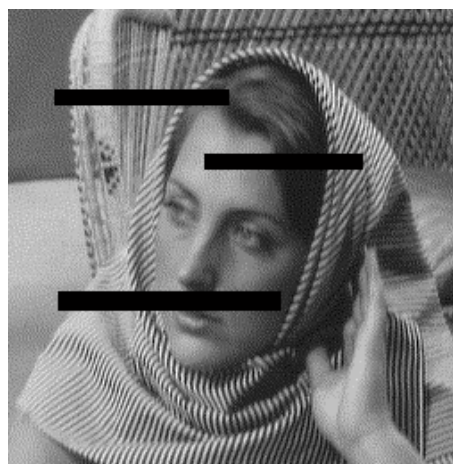
D'après les résultats d'application de la méthode de Bertalmio, nous concluons que cette méthode est appliquée sur des images en niveau de gris et même sur des images couleurs à la différence du temps d'exécution qui est très long pour les images couleurs. Cette méthode est mieux adaptée pour la restauration des zones très finies (le cas de l'image bébé), pour les zones épaisses, la restauration est partielle et au-delà d'un certain nombre d'itération le résultat reste inchangé. Elle n'est pas adaptée pour la reconstruction de la texture.

### III.4.1 Résultats de la diffusion isotrope sur les images en niveaux de gris

*1<sup>er</sup> cas On fixe  $dt=0.3$  et on fait varier  $N$*



(a) Image à restaurer



(b) Image à restaurer



(c)

N=100



(d)



(e)

N=1000



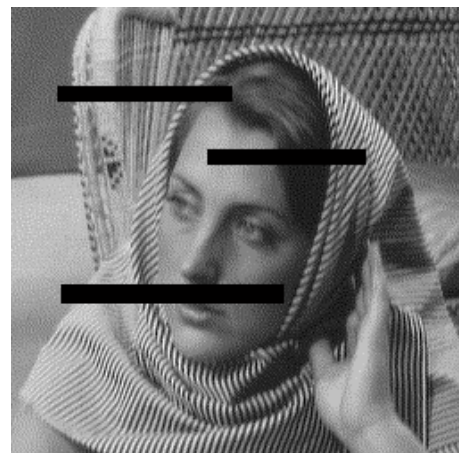
(f)

**Figure III.6:** Résultats d'application de la diffusion isotrope sur des images en niveaux de gris

2<sup>ème</sup> cas On fixe  $N=5000$  et on fait varier  $dt$



(a) Image à restaurer

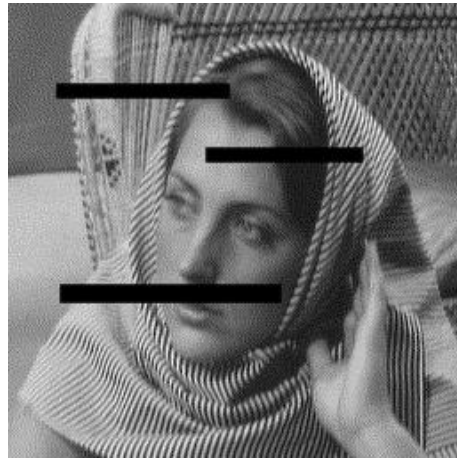


(b) Image à restaurer



(c)

$dt=0.001$

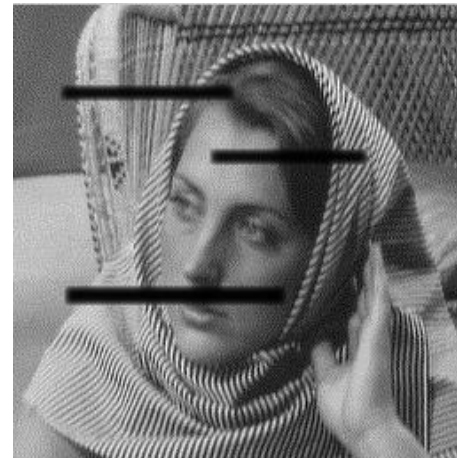


(d)



(e)

$dt=0.01$



(f)



(g)

$dt=0.3$



(h)

**Figure III.7:** Résultats d'application de la diffusion isotrope sur des images en niveaux de gris

**III.4.2 Interprétation des résultats**

Nous constatons que les zones perdues sont reconstruites parfaitement dans les deux images : cameramen (figure III.7.g) et women (figure III.7.h). Dans l’image women de la (figure III.7.h) nous remarquons un certain lissage au niveau de chaise et la chaise car ces zones sont texturées. Pour cette méthode le temps de simulation est très petit (environ 1 mn). Nous concluons que cette méthode est convergente mais elle ne reconstruit pas les zones texturées.

**III.4.3 Résultats de la diffusion isotrope sur des images couleurs**

*1<sup>er</sup> cas on fixe  $dt=0.3$  et on fait varier  $N$*



(a) Image à restaurer



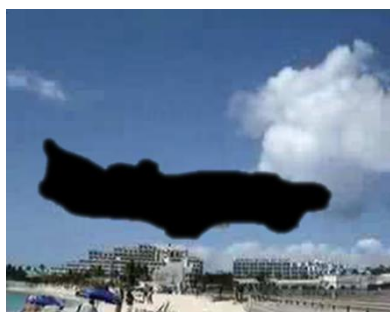
(b) Image à restaurer



(c) Image à restaurer



(d)



(e)



(f)

N=100



(g)



(h)



(i)

N=3000



(j)

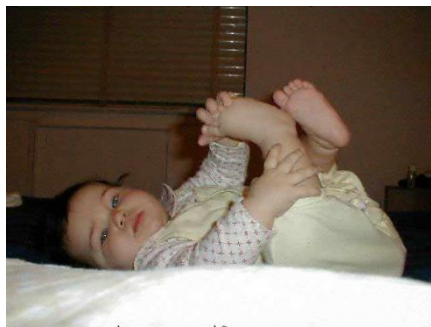


(k)



(l)

N=5000



(m)



(n)

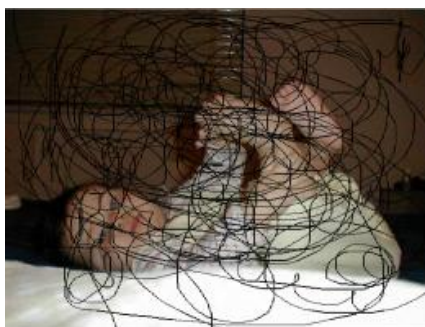


(o)

N=50000

**Figure III.8:** Résultats d'application de la diffusion isotrope sur des images couleurs

2<sup>ème</sup> cas on fixe  $N=50000$  et on fait varier  $dt$



(a) Image à restaurer



(b) Image à restaurer



(c) Image à restaurer



(d)



(e)



(f)

dt=0.001



(g)



(h)



(i)

dt=0.1



(j)



(k)



(l)

dt=0.3

**Figure III.9:** Résultats d'application de la diffusion isotrope sur des images couleurs

### III.4.4 Interprétation des résultats

Nous remarquons à partir de ces résultats que l'image bébé est très bien restaurée après 100 itérations et  $dt=0.3$  (voir figure III.8.d). Nous avons obtenus les mêmes résultats concernant l'image perroquet (voir figure III.1) mais avec un nombre d'itération de 5000 et  $dt = 0.3$ , juste que l'image avion a nécessité beaucoup plus d'itérations que les deux précédentes

$N=50000$ (voir figure III.8.n), nous constatons l'extraction complète de l'avion même si cette partie est très grande et très large. En effet, plus la taille de la partie à reconstruire augmente, la restauration est lente. Après plusieurs tests, nous avons constaté que le taux d'amélioration dt convenable pour cette méthode est de valeur 0.3.

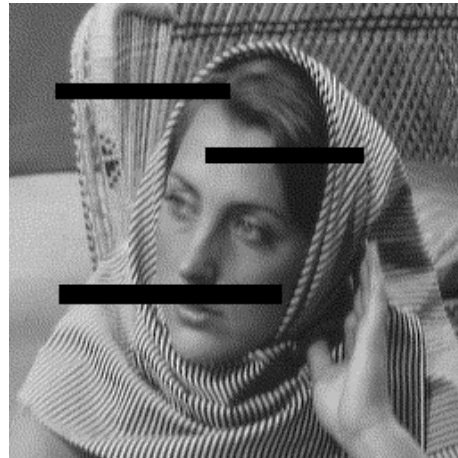
**III.5.1 Résultats de la diffusion anisotrope sur les images en niveau de gris**

Pour cette méthode, les paramètres à contrôler en plus du nombre d'itération  $N$  et le  $dt$  sont : la fonction décroissante  $g$  ( $\exp(-(|\nabla u/k|)^2)$ ) ou  $(\frac{1}{1+(|\nabla u/k|^2)})$  voir les équations 2.17 et 2.18), la barrière de diffusivité  $k$ .

1<sup>er</sup> cas On fixe  $dt=0.3$ ,  $K=22$  et on varie  $N$  avec  $g$  fractionnelle



(a) Image à restaurer



(b) Image à restaurer



(c)

$N=40$



(d)



(e)

N=1000



(f)



(g)

N=5000



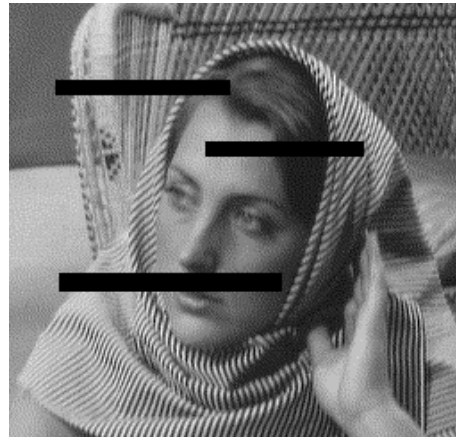
(h)

**Figure III.10:** Résultats d'application de la diffusion anisotrope sur des images en niveau de gris

*2<sup>ème</sup> cas on fixe  $dt=0.3$ ,  $N=5000$  et on fait varier  $k$  avec  $g$  fractionnelle*



(a) Image à restaurer



(b) Image à restaurer



(c)

K=5



(d)



(e)

K=30



(f)



(g)

K=100



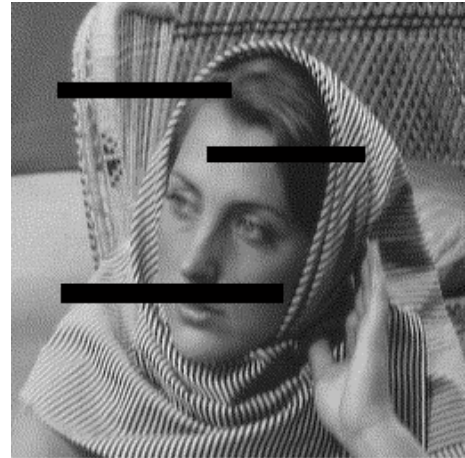
(h)

**Figure III.11:** Résultats d'application de la diffusion anisotrope fractionnelle sur des images en niveau de gris

*3<sup>ème</sup> cas on fixe  $K=100$ ,  $N=5000$  et on fait varier  $dt$  avec  $g$  fractionnelle*



(a) Image à restaurer

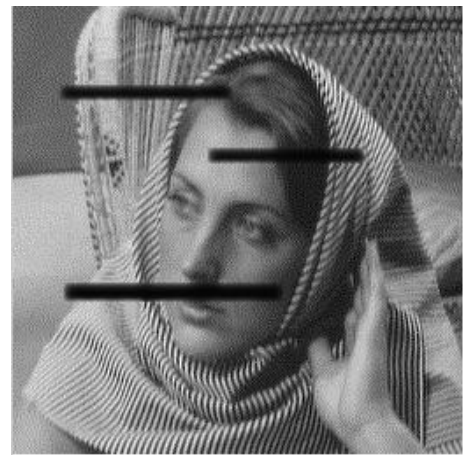


(b) image à restaurer



(c)

$dt=0.001$



(d)



(e)

$dt=0.01$



(f)



(g)



(h)

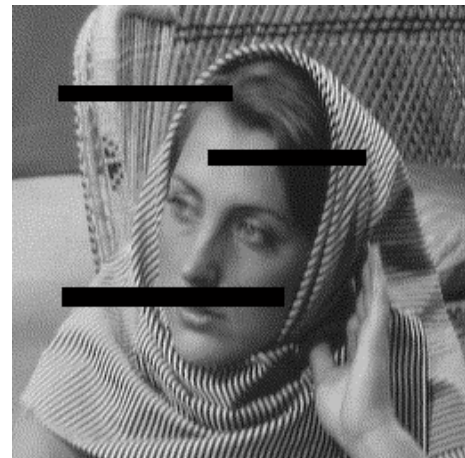
dt=0.1

**Figure III.12:** Résultats d'application de la diffusion anisotrope fractionnelle sur des images en niveau de gris

*1<sup>er</sup> cas on fixe  $dt=0.3$ ,  $K=22$  et on fait varier  $N$  avec  $g$  exponentielle*



(a) Image à restaurer



(b) Image à restaurer



(c)

N=40





(e)

N=1000



(f)



(g)

N=5000



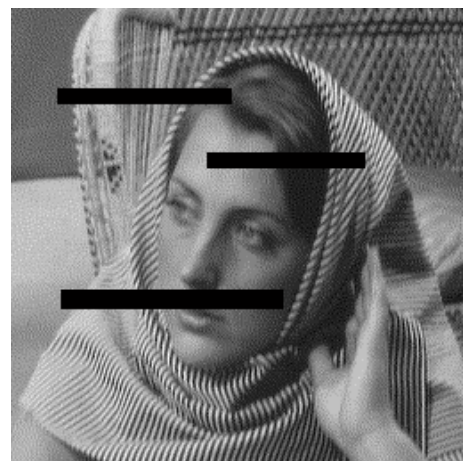
(h)

**Figure III.13:** Résultats d'application de la diffusion anisotrope exponentielle sur des images en niveau de gris

*2<sup>ème</sup> cas on fixe  $dt=0.3$ ,  $N=5000$  et on fait varier  $k$  avec  $g$  exponentielle*



(a) Image à restaurer



(b) image à restaurer



(c)



(d)

K=5



(e)



(f)

K=30



(g)



(h)

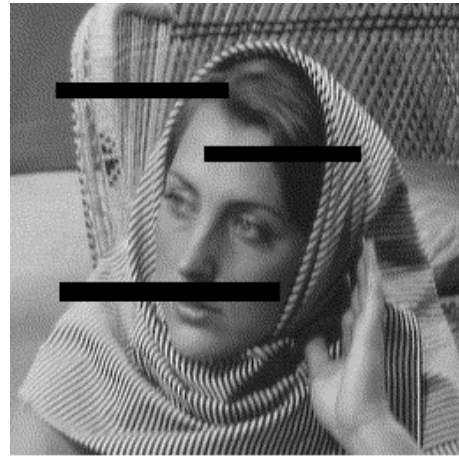
K=100

**Figure III.14:** Résultats d'application de la diffusion anisotrope exponentielle sur des images en niveau de gris

*3<sup>ème</sup> cas on fixe  $K=100$ ,  $N=5000$  et on fait varier  $dt$  avec  $g$  exponentielle*



(a) Image à restaurer

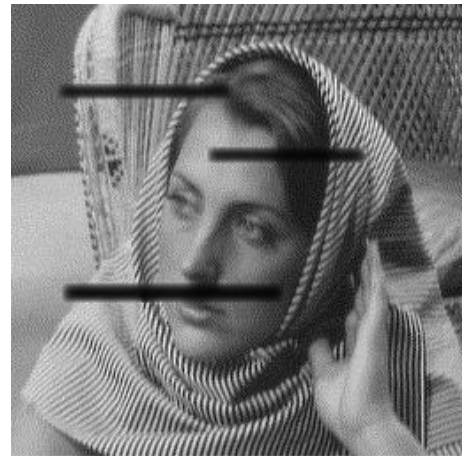


(b) Image à restaurer



(c)

$dt=0.001$



(d)



(e)

$dt=0.01$



(f)



(g)

$dt=0.3$



(h)

**Figure III.15:** Résultats d'application de la diffusion anisotrope exponentielle sur des images en niveau de gris

### III.5.2 Interprétation des résultats

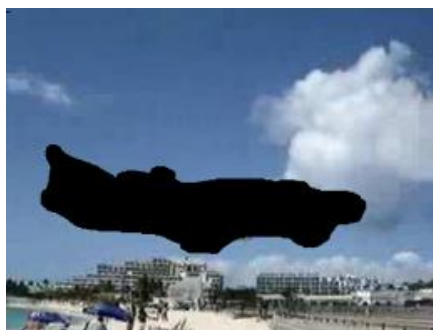
Pour la diffusion anisotrope fractionnelle et exponentielle, la restauration est très bonne dans les deux images : cameramen (figure III.10.g) pour 5000 itération,  $dt=0.3$ ,  $K=22$  dont  $g$  est fractionnelle (figure III.15.g) et avec  $g$  exponentielle, dans l'image women de la (figure III.10.h) nous remarquons que les zones perdues sont reconstruites mais le problème des zones texturées persiste toujours. Pour cette méthode le temps de simulation est très petit (environ 3 mn). Nous concluons que cette méthode est convergente mais elle ne reconstruit pas les zones texturées.

### III.5.3 Résultat de la diffusion anisotrope pour les images couleurs

*1<sup>er</sup> cas on fixe  $dt=0.3$ ,  $K=22$  et on fait varier  $N$ ,  $g$  fractionnelle*



(a) Image à restaurer



(b) Image à restaurer



(c) Image à restaurer



(d)



(e)



(f)

N=1500



(g)



(h)



(i)

N=3000



(j)



(k)



(l)

N=5000

**Figure III.16:** Résultats d'application de la diffusion anisotrope fractionnelle sur des images couleurs

*2<sup>eme</sup> cas on fixe  $dt=0.3$ ,  $N=3000$  et on fait varier  $K$ ,  $g$  fractionnelle*



(a) Image à restaurer



(b) Image à restaurer



(c) Image à restaurer



(d)



(e)



(f)

$K=5$



(g)



(h)



(i)

$K=22$



(j)



(k)



(l)

K=50

**Figure III.17:** Résultats d'application de la diffusion anisotrope fractionnelle sur des images couleurs

*3<sup>ème</sup> cas on fixe  $K=22$ ,  $N=3000$  et on fait varier  $dt$ ,  $g$  fractionnelle*



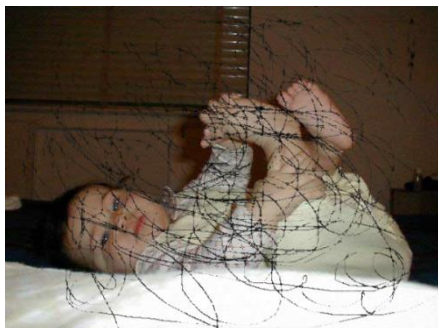
(a) Image à restaurer



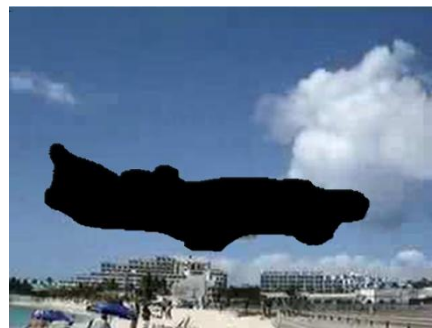
(b) Image à restaurer



(c) Image à restaurer



(d)



(e)



(f)

dt=0.001



(g)



(h)



(i)

$dt=0.3$



(j)



(k)



(l)

$dt=0.5$

**Figure III.18:** Résultats d'application de la diffusion anisotrope fractionnelle sur des images couleurs

*1<sup>er</sup> cas on fixe  $dt=0.3$ ,  $K=22$  et on fait varier  $N$ ,  $g$  exponentielle*



(a) Image à restaurer



(b) Image à restaurer



(c) Image à restaurer



(d)



(e)



(f)

N=1500



(g)



(h)



(i)

N=3000



(j)



(k)



(l)

N=6000

**Figure III.19:** Résultats d'application de la diffusion anisotrope exponentielle sur des images couleurs

2<sup>ème</sup> cas on fixe  $dt=0.3$ ,  $N=3000$  et on fait varier  $K$ ,  $g$  exponentielle



(a) Image à restaurer



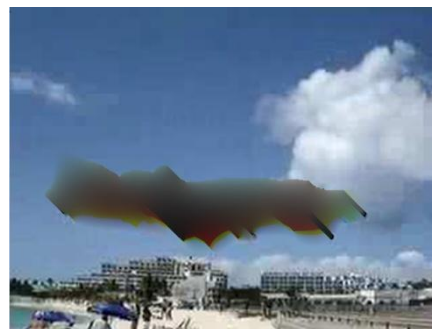
(b) Image à restaurer



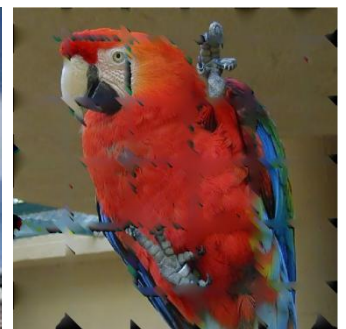
(c) Image à restaurer



(d)



(e)  $K=5$



(f)



(g)

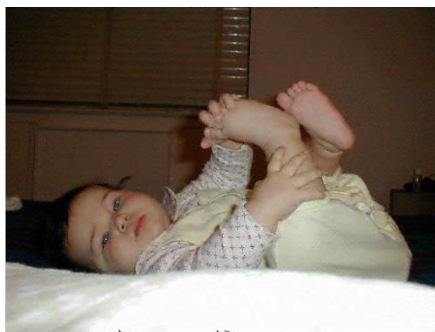


(h)



(i)

K=22



(j)



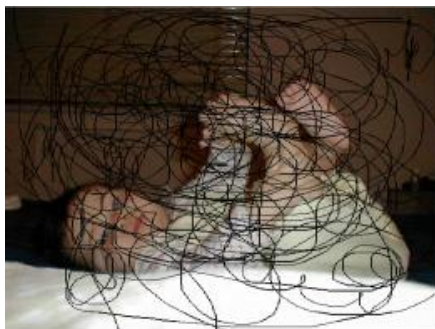
(k) K=50



(l)

**Figure III.20:** Résultats d'application de la diffusion anisotrope exponentielle sur des images couleurs

*3<sup>ème</sup> cas on fixe  $K=22$ ,  $N=3000$  et on fait varier  $dt$ ,  $g$  exponentielle*



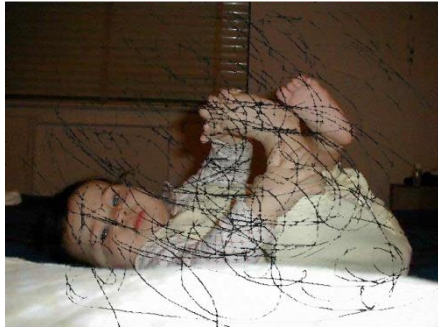
(a) Image à restaurer



(b) Image à restaurer



(c) Image à restaurer



(d)



(e)



(f)

dt=0.001



(g)



(h)



(i)

dt=0.3



(j)



(k)



(l)

dt=0.5

**Figure III.21:** Résultats d'application de la diffusion anisotrope exponentielle sur des images couleurs

### III.5.4 Interprétation des résultats

L'image bébé est très bien restaurée, après 1500 itérations et  $dt=0.3$  (voir figure III.16.d), car le domaine à reconstruire est très fin. Concernant l'image avion même si la partie à reconstruire est très grande et très épaisse la restauration est presque parfaite (voir figure III.18.k) après 3000 itération et  $dt=0.5$ . Pour l'image perroquet nous remarquons un flou (voir figure III.18.l) avec 3000 itération et  $dt=0.5$ . Pour cette méthode le temps de simulation est (environ 10 mn).

### III.6 Calcul d'erreurs quadratiques moyennes des trois méthodes

Pour comparer les trois méthodes : Bertalmio, la diffusion isotrope et la diffusion anisotrope, nous avons calculé le MSE du meilleur résultat (que nous avons jugé visuellement) de chacune de ces méthodes. Comme vous le savez ces trois méthodes nécessitent le contrôle de certains paramètres, la valeur des de ces paramètres sont :

$dt=0.001$  et  $N=20000$  pour la méthode de Bertalmio.

$dt=0.3$  et  $N=50000$  pour la méthode de la diffusion isotrope.

$dt=0.3$ ,  $N=3000$  et  $K=22$  pour la méthode de la diffusion anisotrope.

images \ Méthodes	cameramen	women
Bertalmio	53.6221	10.7959
Diffusion isotrope	7.2691	5.2450
Diffusion anisotrope exponentielle	10.9660	5.9704
Diffusion anisotrope fractionnelle	8.5197	5.4539

**Tableau III.1** : Calcul d'erreurs quadratiques moyennes pour les images aux niveaux de gris

images \ Méthodes	bébé	perroquet	avion
Bertalmio	306.5542	1276.4	1412.0
Diffusion isotrope	15.9666	756.8023	931.0428
Diffusion anisotrope exponentielle	28.0286	779.4958	204.6787
Diffusion anisotrope fractionnelle	16.8035	760.37	213.9051

**Tableau III.2** : Calcul d'erreurs quadratiques moyennes pour les images couleurs.

D'après le tableau III.1, le meilleur résultat est celui de la méthode de la diffusion isotrope pour les deux images aux niveaux de gris cameramen et women. Pour les images couleurs (tableau III.2), le meilleur MSE est celui de la diffusion isotrope pour l'image bébé et perroquet, mais pour l'image avion c'est la diffusion anisotrope exponentielle. C'est à partir du MSE, que nous remarquons une légère différence dans le choix de la fonction de diffusivité  $g$ , d'ailleurs visuellement, nous ne constatons pas cette différence, idem pour le paramètre  $k$ .

### III.7 Conclusion

Dans ce chapitre, nous avons montré comment appliquer les méthodes adoptées (les méthodes basées sur les EDP) pour reconstruire des parties manquantes dans l'image. Nous avons testé trois méthodes à savoir : la méthode de Bertalmio, et la méthode de la diffusion isotrope et la méthode de la diffusion anisotrope.

D'après tous les tests réalisés, nous concluons que la méthode de la diffusion donne de meilleurs résultats par rapport à la méthode de Bertalmio selon deux critères le calcul d'erreur et le temps de simulation, cette méthode est beaucoup plus rapide que celle de Bertalmio. L'inconvénient des deux méthodes est l'incapacité de restaurer les zones texturées.

La différence entre la diffusion isotrope et anisotrope c'est bien le temps de simulation, qui est très petit dans la diffusion isotrope.











# Conclusion générale

### Conclusion générale

Le travail présenté dans ce mémoire s'inscrit dans le cadre de la reconstruction des données manquantes dans une image. Dans cette étude, nous nous sommes focalisés sur les méthodes de restauration d'image. Nous avons implémenté trois méthodes basées sur les EDPs à savoir la méthode de Bertalmio, la méthode de la diffusion isotrope et la méthode de la diffusion anisotrope. Les deux premières méthodes dépendent de deux paramètres qui sont le nombre d'itération  $N$  et le taux de changement  $dt$ , La troisième méthode dépend, en plus de ces deux paramètres, de deux autres qui sont : la barrière de diffusion  $k$  et la fonction de diffusivité  $g$ .

Nous les avons appliqués sur différentes images en niveaux de gris et en couleur avec différentes parties manquantes à restaurer. D'après les résultats obtenus lors de l'application de cette méthode sur différentes images. Nous avons conclu que les deux premiers paramètres ( $N$ ,  $dt$ ) jouent le même rôle qui est l'accélération d'exécution lorsque  $dt$  est grand et pour  $N$  petit, dans le cas des méthodes de diffusion isotrope et anisotrope. En plus, nous avons remarqué que les deux derniers paramètres ( $k$ ,  $g$ ) ont une très légère influence sur la restauration des zones perdues.

Les zones perdues que nous avons utilisé sont de différentes tailles et formes (grandes, petites, épaisses et fines). Après plusieurs tests de restauration de ces zones, nous avons constaté que la méthode de Bertalmio n'est adaptée que pour la restauration des zones fines (grandes ou petites). Contrairement à cette dernière, les méthodes de diffusion reconstruisent même les zones épaisses.

En plus de ces applications, nous avons testé ces trois méthodes sur des images texturées, les parties manquantes ont été restaurées mais la texture n'est pas préservée (apparition du flou dans les zones texturées). Pour remédier à ce problème, nous jugeons qu'une étape de synthèse de texture est primordiale juste après la restauration, afin de reconstruire la texture des zones manquantes.

## Quelques opérateurs usuels

- **Gradient** : Le gradient est un vecteur qui indique la direction et l'intensité de la plus grande variation d'un scalaire  $u$ , il est noté comme suit :

$$\text{grad } u = \nabla u = \left( \frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2} \right) = (u_{x_1}, u_{x_2})$$

La norme du gradient est donnée par la relation suivante :

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

L'orientation du gradient est donnée par la relation suivante :

$$\varphi(\text{grad } u) = \arctan\left(\frac{\partial u}{\partial x_1} / \frac{\partial u}{\partial x_2}\right)$$

- **Laplacien** : le laplacien correspond à la divergence du gradient

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

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

- **Divergence d'un vecteur** : la divergence d'une fonction vectorielle

$$u = (u_1(x), u_2(x) + \dots) \text{ où } x \in \Omega$$

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

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

# Bibliographie

## Bibliographie

[1] : S. Belfkih, A. Dziri, P. Montesinos, R. Beuscart, « *un nouveau schéma Numérique pour la Résolution des Equations aux Dérivées partielles (EDP) en traitement d'images* », Laboratoire CERIM\_ Faculté de Médecine (pôle recherche) 1, place de Verdun 59045 Lille, France.

[2] : S. Chatterji, « *cour d'analyse 3 équations différentielles ordinaires et aux dérivées partielles* », presses polytechniques et universitaires romandes 1998.

[3] : S. Toucherift, K Aouimeur, « *Application des équations aux dérivées partielles en traitement d'image* », Faculté de Génie Electrique et d'Informatique Département d'automatique, UMMTO 2011.

[4] : F. Blanc, C. Ramel, J. Soleilhavoup, « *lissage d'images par équations à dérivées partielles* », projet semestriel, master 1 spécialité imagerie numérique.

[5] : G. Koeper, « *Equation aux Dérivées partielles* », UFR De Mathématiques et Informatique, université René Descartes Paris 5, 2001.

[6] : N. Arkoub, « *Reconstruction de données manquantes dans une image* », Mémoire de magister, Faculté de Génie Electrique et d'Informatique Département Automatique, UMMTO 2012.

[7] : E. Goncalvès, « *Résolution numérique discrétisation des EDP et EDO* », Institut National Polytechnique de Grenoble, 2005.

[8] : R. Deriche, O. Faugeras, « *Les équation aux dérivées partielles en traitement des images et vision par ordinateur* », INRIA Saphi Antipolis, 2004.

[9] : M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, « *Image Inpainting* », IMA Preprint Series, 1999.

[10]: A. Ghazdali, « *Restauration d'image : INPAINTING* », Mémoire du master, Modélisation et calcul scientifique pour l'ingénierie mathématique, Faculté des Sciences et Technique Guéliz, Marrakech 2012.

[11] : P. Perona, J. Malik. « *Scale- espace and edge detection using anisotropic diffusion* », IEEE trans Anal. Machine Intell, pp629-639, 1990.

[12]: Tony. F, Chan and J. Shen, « Morphologically Invariant PDE Inpainting »

[13]: F. Haddad, « *Vidéo Inpainting* », Mastar Professionnel Réseaux, Application Documentaire, Ingénierie et Sécurité (RADIS), 2009.

## Résumé

Repeigner les parties dégradées des anciens tableaux de peinture peut apparaître comme une tâche facile pour certain, mais la réalité est toute autre, cette tâche est complexe et difficile à reproduire. Elle ne serait réalisée que par un artiste de restauration d'image qualifié. Cela a attiré l'attention des chercheurs notamment dans le domaine du traitement d'images pour l'application d'algorithmes sophistiqués pour remplacer les pièces perdues ou corrompues des données d'image (principalement de remplir des petites régions ou de supprimer de petits défauts).

Ce type d'algorithmes de reconstruction de données manquantes dans une image est appelé *Inpainting numérique* «digital inpainting». De manière générique, ce terme désigne le fait de déterminer, de la manière la plus automatique possible, la valeur des pixels considérés comme manquants dans une image.

Notre but dans ce mémoire est d'appliquer quelques méthodes de résolution du problème de restauration d'image basées sur les équations aux dérivées partielles.

Nous mettrons en pratique l'étude théorique que nous avons abordée, En effet, nous appliquerons les méthodes de Bertalmio, la diffusion isotrope et anisotrope pour reconstruire nos différentes images. Plusieurs tests seront effectués sur différentes images (niveaux de gris et couleurs) pour évaluer ces méthodes. L'implémentation de notre application se fera avec le langage de programmation MatlabR2009 sur un HP Compaq n×7400 (Processeur Intel(R) core (TM) 2CPU T5500@1.66 GHZ et 1528MB de RAM.

## Mot clé

Inpainting, reconstruction de données manquantes dans une image, restauration de zones perdues dans une image.