

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE  
DEPARTEMENT D'INFORMATIQUE

## Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Domaine : **Mathématiques et Informatique**

Filière : **Informatique**

Spécialité : **Conduite de Projet Informatique**

*Présenté par*

**M<sup>r</sup> : YOUNSI Mouloud**

**M<sup>r</sup> : BELLIL Abdeslam**

Thème

# Application du modèle variationnel non local à la segmentation d'images

*Mémoire soutenu publiquement le 21/09/2016 Devant le jury composé de :*

**Président : M<sup>me</sup> SINI.G**

**Encadreur : M<sup>elle</sup> AIT AIDER.M**

**Examineur : M<sup>r</sup> HABET.M.S**

**Examineur : M<sup>me</sup> DJAMAH.T**

# *Remerciements :*

**Je tiens à remercier en tout premier lieu DIEU le tout puissant de m'avoir donné la patience, la force et le courage, pour élaborer ce travail.**

**Nous tenons à remercier vivement notre promotrice Mlle Ait Aider qui nous a fait confiance et s'est mis à notre entière disposition pour le parfait déroulement de notre travail.**

**Nous remercions les membres du jury pour l'honneur qu'ils nous font en acceptant de juger ce mémoire de fin d'études.**

**Nous remercions aussi toutes les personnes ayant contribué de près ou de loin au bon accomplissement de ce modeste travail.**

# Dédicaces

*Je* dédie ce travail à la mémoire de mes grands-pères «AHMED-AMMAR» et ma grand-mère «SADIA» qui seront toujours dans mon cœur, que DIEU les accueillent en son vaste paradis.

*A* ma grand-mère «TASSADIT» «YEMAHA» que dieu la garde.

*A* mes chers parents «VAVA» et «YEMA» qui m'ont aidé et soutenu tout au long de mon parcours scolaire et universitaire.

*A* mes chers frères et sœurs.

*A* «NANA OUARDIA»,

*A* «KHALTI OUIZA» et son mari «DA\_MOH» et ces enfants :

*A* ma femme Wassila que dieu la protège,

*A* mon binôme et sa famille,

*A* toute la promotion LMD 2016.

<<MOULOUDD>>

# Dédicaces

*Je dédie ce modeste travail :*

*A Mes parents pour leur aide précieuse et sacrifices*

*A Ma grand-mère à qui je souhaite une longue vie*

*A Mes adorables frères BELAID et MALIK et ma sœur DYHIA à qui je souhaite  
une réussite totale dans leur vie*

*A tous mes proches et ma famille*

*A mon binôme mouloud et toute sa famille*

*A Mes meilleurs amis pour leur aide, leur temps, leur assistance et soutien*

*A Tout se qui ont contribué de près ou de loin a la réalisation de se travail*

*A toute la promotion LMD 2016.*

<<ABDESLAM>>

## Liste des Figures :

Figure 1 : Histogramme et palette associés à une image

Figure 2 : Schéma d'un système de traitement d'images

Figure 3 : Egalisation d'histogramme

Figure 4 : Différents types de contours : marche, toit et pointe

Figure 5 : Cas bidimensionnel

Figure 6 : Orientation du contour

Figure 7 : Les directions de Freeman d'après

Figure 8 : Voisinage d'un pixel

Figure 9 : Découpage par quadtree d'une image 8x8. Traitement avec seuil=100%

Figure 10 : Problème de découpage arbitraire de régions dans la phase de "split"

Figure 11 : Exemples de graphes des k plus proches voisins. (a) : données initiales dans  $\mathbb{R}^2$ ,  
(b) : 3-PPV, (c) : 15-PPV.

Figure 12 : Graphes grille représentant les images. (a) : un graphe grille en 4-voisinage, (b) : un graphe grille en 8-voisinage

Figure 13 : Quatre exemples de patches comme vecteur d'attributs caractérisant les sommets du graphe. Ici, le pixel central (en rouge) est caractérisé par un vecteur de niveaux de gris défini grâce à un patch de taille 13x13.

Figure 14 : le logo de Qt

Figure 15 : Présentations de quelques zones de l'outil de développement

Figure 16 : Interface qui montre le Menu Fichier.

Figure 17 : Interface qui montre l'utilité du bouton Luminance.

Figure 18 : Interface représentant la Détection de contour.

Figure 19 : image original

Figure 20 : transformation de l'image originale en niveaux de gris.

Figure 21 : résultat avec  $\sigma = 2$ ,  $\lambda = 0.4$ ,  $\varepsilon = 0.07$ ,  $p = 3$ ,  $\Delta t = 0.1$ , iter=1.

Figure 22 : résultat avec  $\sigma = 1$ ,  $\lambda = 0.4$ ,  $\varepsilon = 0.07$ ,  $p = 3$ ,  $\Delta t = 0.1$ , iter=1.

Figure 23 : résultat avec  $\sigma = 1.5$ ,  $\lambda = 0.4$ ,  $\varepsilon = 0.07$ ,  $p = 3$ ,  $\Delta t = 0.3$ , iter=1.

Figure 24 : résultat avec  $\sigma = 2$ ,  $\lambda = 0.4$ ,  $\varepsilon = 0.07$ ,  $p = 3$ ,  $\Delta t = 0.1$ , iter=2.

Figure 25 : résultat avec  $\sigma = 2$ ,  $\lambda = 0.4$ ,  $\varepsilon = 0.07$ ,  $p = 3$ ,  $\Delta t = 0.3$ , iter=3.

# Sommaire

Liste des figures.....	
Introduction générale.....	9
<b>Chapitre 1 : Généralité sur le traitement d'image.....</b>	<b>11</b>
<b>1. Introduction.....</b>	<b>12</b>
<b>2. Définition de l'image.....</b>	<b>12</b>
<b>3. Acquisition d'une image.....</b>	<b>12</b>
<b>4. Caractéristiques d'une image numérique.....</b>	<b>13</b>
4.1 Dimension.....	13
4.2 Résolution.....	13
4.3 Bruit.....	13
4.4 Histogramme.....	13
4.5 Luminance.....	14
4.6 Contraste.....	14
4.7 Images à niveaux de gris.....	15
4.8 Images en couleurs .....	15
<b>5. Système de traitement d'images.....</b>	<b>15</b>
5.1 Prétraitement d'images.....	16
5.1.1 Modification d'histogramme.....	16
5.1.1.1 Expansion de dynamique.....	16
5.1.1.2 Egalisation d'histogramme.....	17
5.1.2 Réduction du bruit par filtrage.....	17
<b>6. Extraction de contours.....</b>	<b>18</b>
6.1 Approche gradient.....	19
6.1.1 Opérateurs Prewitt et Sobel.....	20
6.1.2 Opérateur de Kirsh.....	20
6.2 Approche Laplacien.....	21
<b>7. Segmentation .....</b>	<b>21</b>
7.1 Accroissement de région.....	22
7.2 Méthode de type 'split and merge'.....	23
Conclusion.....	25

## **Chapitre 2 : Théorie des graphes et opérateur.....26**

<b>1. Introduction.....</b>	<b>27</b>
<b>2. Définitions et notations.....</b>	<b>27</b>
<b>3. Structure et topologie.....</b>	<b>28</b>
<b>3.1 Domaines non organisés.....</b>	<b>28</b>
<b>3.2 Domaines organisés.....</b>	<b>30</b>
<b>3.3 Similarités entre les sommets.....</b>	<b>32</b>
<b>3.3.1 Vecteurs d'attributs.....</b>	<b>32</b>
<b>3.3.2 Fonctions de poids.....</b>	<b>33</b>
<b>4. Opérateurs définis sur des graphes.....</b>	<b>34</b>
<b>4.1 Fonctions définies sur des graphes.....</b>	<b>34</b>
<b>4.2 Opérateur de différence et adjoint.....</b>	<b>35</b>
<b>4.2.1 Opérateur de différence .....</b>	<b>35</b>
<b>4.2.2 Opérateur adjoint.....</b>	<b>36</b>
<b>4.3 Opérateur gradient.....</b>	<b>36</b>
<b>4.4 Opérateur p-Laplacien isotrope.....</b>	<b>37</b>
<b>5. P-Laplacien anisotrope.....</b>	<b>39</b>
<b>Conclusion.....</b>	<b>40</b>

## **Chapitre 3 : description de la méthode.....41**

<b>1. Introduction.....</b>	<b>42</b>
<b>2. Généralités sur les approches variationnelles.....</b>	<b>42</b>
<b>3. Segmentation.....</b>	<b>42</b>
<b>4. Fonctionnelle de régularisation.....</b>	<b>43</b>
<b>5. Le modèle variationnel proposé.....</b>	<b>44</b>
<b>Conclusion .....</b>	<b>47</b>

## **Chapitre 4 : Mise en œuvre.....49**

<b>1. Introduction.....</b>	<b>50</b>
<b>2. Installation du logiciel.....</b>	<b>50</b>
<b>2.1 Environnement de travail.....</b>	<b>50</b>
<b>2.2 Le langage de codage.....</b>	<b>50</b>

<b>3. Description de l'interface du logiciel.....</b>	<b>52</b>
<b>4. définition des classes d'application.....</b>	<b>53</b>
<b>5. Quelques interfaces de l'application.....</b>	<b>54</b>
<b>5.1 Menu Fichier.....</b>	<b>54</b>
<b>5.2 Bouton Luminance.....</b>	<b>54</b>
<b>5.3 Bouton Détection des contours.....</b>	<b>55</b>
<b>5.4 Bouton Gradient.....</b>	<b>55</b>
<b>5.5 Bouton Initialiser.....</b>	<b>55</b>
<b>5.6 Bouton Quitter.....</b>	<b>55</b>
<b>6. Tests et résultats.....</b>	<b>55</b>
<b>6.1 Influence des différents paramètres sur les résultats .....</b>	<b>56</b>
<b>Conclusion.....</b>	<b>59</b>
<b>Conclusion générale.....</b>	<b>60</b>
<b>Bibliographie.....</b>	<b>62</b>

# *Introduction générale*

# Introduction générale

---

Le traitement d'images qui est un processus comprenant plusieurs étapes, est né de la nécessité de remplacer l'observateur humain par la machine. Il intervient dans bon nombre de domaines tels que la médecine, l'agriculture et l'industrie.

Une étape incontournable dans tout processus d'analyse d'images est celle de la segmentation. En fournissant une description compacte de l'image, plus exploitable que l'ensemble des pixels, la segmentation d'images permet de faciliter l'interprétation automatique d'une image de façon similaire à une interprétation humaine. En effet, elle a été inspirée du système de perception visuel humain qui utilise la notion de similarité et de différence afin de localiser et de délimiter les objets d'une scène.

Cette tâche bien que relevée avec facilité par le système visuel humain, est en réalité complexe et reste un véritable défi pour la communauté du traitement d'images malgré plusieurs décennies de recherche. Ainsi, plusieurs méthodes de segmentation ont été proposées dans la littérature par exemple on trouve les méthodes variationnelles qui sont basées sur les équations aux dérivées partielles. Les solutions de ces modèles définis dans le domaine continu sont obtenues par discrétisation d'opérateurs différentiels afin de pouvoir disposer de schémas numériques applicables. Cette discrétisation constitue un obstacle non négligeable lorsque les données sont discrètes et de grande dimension. En effet, les opérateurs différentiels sont difficilement discrétisable lorsque les données sont définies sur des domaines non uniformes ou possèdent une grande dimensionnalité.

Pour pallier à cet inconvénient, d'autres travaux de recherche ont été développés en proposant de transcrire et d'adapter certains de ces modèles dans un cadre discret en utilisant les graphes.

Dans ce mémoire, nous proposons une méthode de segmentation binaire exploitant le formalisme des équations aux différences partielles, le bénéfice des méthodes basée sur les graphes et les outils mathématiques définis dans le domaine continu. Elle est basée sur un nouveau formalisme de modèle variationnel qui est basé sur les équations aux dérivées partielles.

Pour mener à terme notre travail, nous avons adopté la structure suivante:

- chapitre 1 fera l'objet de généralités sur le traitement d'images.
- Chapitre 2 donnera un aperçu sur la théorie des graphes et operateurs.
- Chapitre 3 sera consacré à la description de la méthode adoptée.
- Chapitre 4 sera réservé à la présentation du logiciel ainsi les résultats obtenus.

Pour finir, nous terminerons par une conclusion générale.

# ***Chapitre 1 :***

## ***Généralités sur le traitement d'images***

## 1. Introduction

Le traitement d'images est un domaine très vaste qui a connu, et qui connaît encore, un développement important depuis quelques dizaines d'années.

On désigne par traitement d'images numériques l'ensemble des techniques permettant de modifier une image numérique afin d'améliorer ou d'en extraire des informations.

De ce fait, le traitement d'images est l'ensemble des méthodes et techniques opérant sur celles-ci, dans le but de rendre cette opération possible, plus simple, plus efficace et plus agréable, d'améliorer l'aspect visuel de l'image et d'en extraire des informations jugées pertinentes.

Dans ce chapitre, nous abordons les notions de base nécessaires à la compréhension des techniques de traitement d'images. Ensuite, nous allons donner un aperçu sur les différentes techniques connues dans ce domaine.

## 2. Définition de l'image

Une image est plutôt difficile à décrire d'une façon générale. Une image est une représentation du monde. En traitement d'image, la majorité du temps, on considère qu'il s'agit d'une fonction mathématique de  $\mathbb{R} \times \mathbb{R}$  dans  $\mathbb{R}$  où le couplet d'entrée est considéré comme une position spatiale, le singleton de sortie comme l'intensité (couleur ou niveaux de gris) du phénomène physique. Il arrive cependant que l'image soit dite 3D donc la fonction est de  $\mathbb{R} \times \mathbb{R} \times \mathbb{R}$  dans  $\mathbb{R}$ . Les images couleurs peuvent être représentées soit par trois images représentant les trois couleurs fondamentales, soit par une image de  $\mathbb{R} \times \mathbb{R}$  dans  $\mathbb{R} \times \mathbb{R} \times \mathbb{R}$ .

L'image numérique est l'image dont la surface est divisée en éléments de tailles fixes appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleurs prélevé à l'emplacement correspondant dans l'image réelle, ou calculé à partir d'une description interne de la scène à représenter.

## 3. Acquisition d'une image

L'acquisition d'images constitue un des maillons essentiels de toute chaîne de conception et de production d'images. Pour pouvoir manipuler une image sur un système informatique, il est avant tout nécessaire de lui faire subir une transformation qui la rendra lisible et manipulable par ce système. Le passage de cet objet externe (l'image d'origine) à sa représentation interne (dans l'unité de traitement) se fait grâce à une procédure de numérisation. Ces systèmes de saisie, dénommés optiques, peuvent être classés en deux

catégories principales [Bénallal.M]:

- les caméras numériques ,
- les scanners.

A ce niveau, notons que le principe utilisé par le scanner est de plus en plus adapté aux domaines professionnels utilisant le traitement de l'image comme la télédétection, la médecine, etc. Le développement technologique a permis l'apparition de nouveaux périphériques d'acquisition appelés cartes d'acquisition, qui fonctionnent à l'instar des caméras vidéo, grâce à un capteur *C.C.D.* (Charge Coupled Device). La carte d'acquisition reçoit les images de la camera, de la T.V. ou du scanner afin de les convertir en informations binaires qui seront stockées dans un fichier [Chakib.K].

## 4. Caractéristiques d'une image numérique

L'image est un ensemble structuré d'informations caractérisé par les paramètres suivants:

### 4.1 Dimension

C'est la taille de l'image. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image.

### 4.2 Résolution

C'est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les moniteurs d'ordinateurs, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre). On utilise aussi le mot résolution pour désigner le nombre total de pixels affichables horizontalement ou verticalement sur un moniteur; plus grand est ce nombre, meilleure est la résolution.

### 4.3 Bruit

Un bruit dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur.

### 4.4 Histogramme

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image. Il permet de donner un grand nombre d'information sur la distribution des niveaux de gris (couleur) et de voir

entre quelles bornes est répartie la majorité des niveaux de gris (couleur) dans le cas d'une image trop claire ou d'une image trop foncée.

Il peut être utilisé pour améliorer la qualité d'une image (Rehaussement d'image) en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci.

Pour diminuer l'erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image, on modifie souvent l'histogramme correspondant.

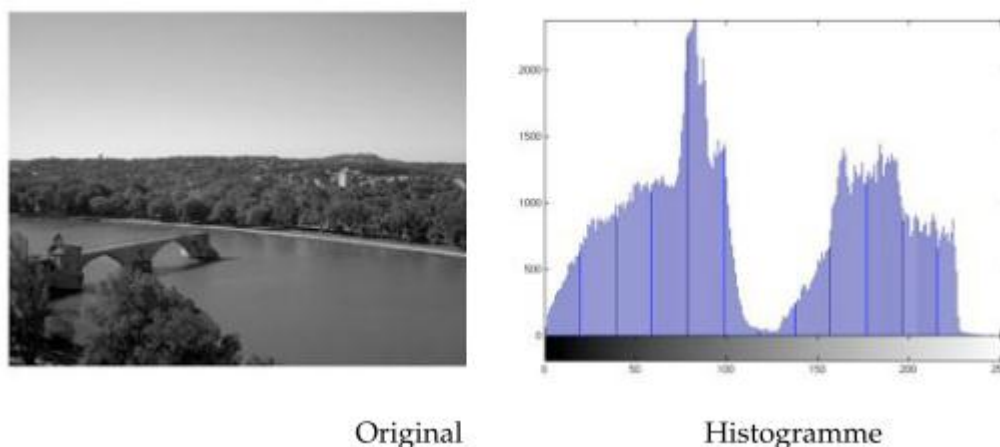


Figure 1. Histogramme et palette associés à une image

#### 4.5 Luminance

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet.

Une bonne luminance se caractérise par [Nicolas.L] :

- Des images lumineuses (brillantes).
- Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir; ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- L'absence de bruit.

#### 4.6 Contraste

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images. Si  $L_1$  et  $L_2$  sont les degrés de luminosité respectivement de deux zones voisines  $A_1$  et  $A_2$  d'une image, le contraste  $C$  est défini

par le rapport proposé par Michelson :

$$C = \frac{L1-L2}{L1+L2} \quad (1)$$

#### 4.7 Images à niveaux de gris

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter les images à niveaux de gris, on peut attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise par exemple entre 0 et 255. Chaque pixel n'est donc plus représenté par un bit, mais par un octet. Pour cela, il faut que le matériel utilisé pour afficher l'image soit capable de produire les différents niveaux de gris correspondant.

Le nombre de niveaux de gris dépend du nombre de bits utilisés pour décrire la couleur de chaque pixel de l'image. Plus ce nombre est important, plus les niveaux possibles sont nombreux.

#### 4.8 Images en couleurs

Même s'il est parfois utile de pouvoir représenter des images en noir et blanc, les applications multimédias utilisent le plus souvent des images en couleurs. La représentation des couleurs s'effectue de la même manière que les images monochromes avec cependant quelques particularités. En effet, il faut tout d'abord choisir un modèle de représentation. On peut représenter les couleurs à l'aide de leurs composantes primaires. Les systèmes émettant de la lumière (écrans d'ordinateurs,...) sont basés sur le principe de la synthèse additive : les couleurs sont composées d'un mélange de rouge, vert et bleu (modèle R.V.B.).

### 5 Système de traitement d'images

Un système de traitement numérique d'images est composé de :

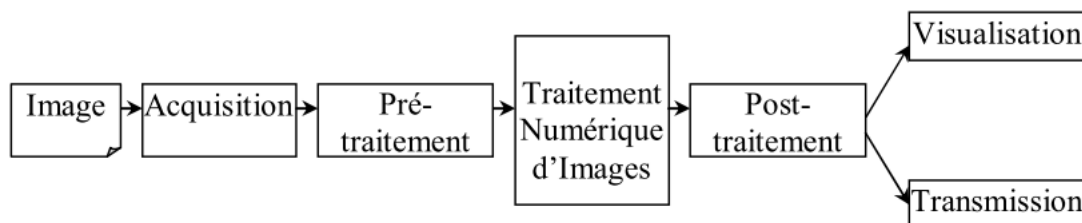


Figure 2. Schéma d'un système de traitement d'images

## 5.1 Prétraitement d'images

Cette phase a lieu juste après l'acquisition des images et a pour objectif d'améliorer la qualité de l'image en vue de sa segmentation.

Le temps de traitement (temps CPU) est très important. C'est un facteur décisif, il doit être le plus petit possible. Ceci implique que les opérateurs doivent être locaux, c'est-à-dire qu'ils doivent intervenir sur un nombre limité de pixels et essentiellement sur des pixels dans le voisinage proche du pixel courant.

Les techniques de prétraitements les plus courantes qu'on va présenter sont [Bensrhair.A] :

- La modification d'histogramme.
- La réduction du bruit par filtrage.

### 5.1.1 Modification d'histogramme

On cherche à améliorer l'image en appliquant une transformation ponctuelle d'intensité; à tout pixel  $a_s(x, y)$  on associe une intensité par la transformation  $T$  croissante de façon à conserver des contrastes entre régions (claire sur fond sombre, apparaît plus claire après transformation). Le pixel après transformation est le pixel  $a'_s(x, y)$  :

$$a'_s(x, y) = T(a_s(x, y)) \quad (2)$$

Dans ce qui suit, on va présenter deux types de modification d'histogramme :

- Expansion de dynamique.
- Egalisation d'histogramme.

#### 5.1.1.1 Expansion de dynamique

Cette méthode consiste à utiliser au mieux l'échelle des niveaux de gris disponible sur le système d'acquisition d'images.

Soit  $A[i, j]$  l'image de départ,  $A'[i, j]$  image après transformation et  $[a_0, a_1]$  l'intervalle des intensités présentes dans l'image et soit  $[a_{min}, a_{max}]$  l'intervalle disponible.

L'expansion de la dynamique correspond à la transformation :  $a_{s'} = \alpha + \beta \cdot a_s$  telle que :

$$\forall a \in [a_0, a_1] a_s \xrightarrow{T} a'_s \in [a_{min}, a_{max}]$$

$$\alpha = \frac{a_{min} \cdot a_1 - a_{max} \cdot a_0}{a_1 - a_0}, \quad \beta = \frac{a_{max} - a_{min}}{a_1 - a_0} \quad [Gonzales, Woods] \quad (3)$$

#### 5.1.1.2 Egalisation d'histogramme

Cette transformation consiste à rendre le plus plat possible l'histogramme de niveaux de gris de l'image. L'image  $A[i, j]$  est considérée comme un ensemble de réalisation d'une variable

aléatoire  $A$ , admettant une densité de probabilité  $f(a)$  et une fonction de répartition  $F(a)$ ,  $f$  représente l'histogramme normalisé de  $A[i, j]$ .

On cherche une transformation  $T$  continue dérivable par morceaux et strictement croissante telle que la variable aléatoire  $B = T(A)$  soit uniformément répartie dans l'intervalle  $[b_{min}, b_{max}]$ .

Soit  $g(b)$  la densité de probabilité de  $B$ , et  $T'$  la transformation dérivée de  $T$  :

$$g(b) = \begin{cases} f(a) \frac{1}{T'(a)} = \frac{1}{b_{max} - b_{min}} & \text{si } b_{min} < b < b_{max} \\ 0 & \text{ailleurs} \end{cases} \quad (4)$$

$$T'(a) = (b_{max} - b_{min}) \cdot f(a), b = T(a), a \in [a_{min}, a_{max}] \quad [\text{Gonzales, Woods}] \quad (5)$$

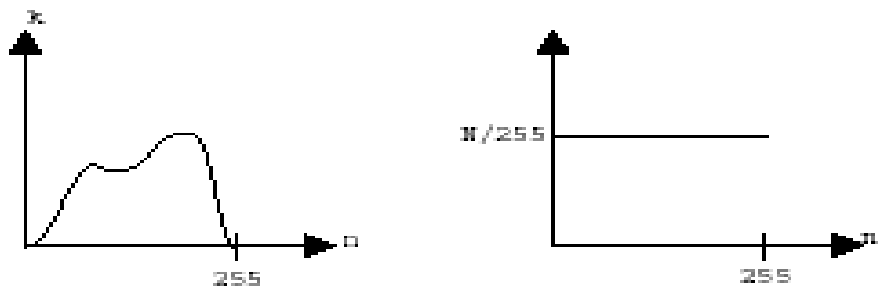


Figure 3. Egalisation d'histogramme

### 5.1.2 Réduction du bruit par filtrage

Le bruit peut provenir de différente source :

- Environnement lors de l'acquisition,
- Qualité du capteur,
- Qualité de l'échantillonnage.

En général, le bruit d'images est considéré comme un champ aléatoire caractérisé par sa densité de probabilité  $f$  et sa fonction de répartition  $F$ . La nature plus ou moins impulsionnelle du bruit peut être décrite à l'aide de densité de probabilités de la forme [Cocquerez.J.P, Philipp.S]:

$$f(a) = C \cdot \exp(-K|a|^\alpha) \quad (6)$$

Plusieurs filtres sont utilisés pour la réduction de bruit, ils sont divisés en deux catégories :

- Filtres linéaires stationnaires,
- Filtres non linéaires stationnaires.

La première catégorie comprend tous les opérateurs pouvant exprimer leur résultat comme une combinaison linéaire des niveaux de gris d'un voisinage de l'image. Ces filtres possèdent des caractéristiques spectrales, on parle ainsi de filtre passe-bas (l'image devient floue) ou de filtre passe-haut (les contours ressortent).

Voici quelques exemples de filtres réducteurs de bruit :

- ✓ Filtre moyenneur
- ✓ Filtre gaussien
- ✓ Filtre exponentiel de Shen

Le principal inconvénient des filtres linéaire est que la réduction de bruit s'accompagne d'un étalement des transitions entre régions. Ce problème peut être surmonté par l'utilisation des filtres non linéaires.

Les opérateurs les plus courants en prétraitement d'images sont :

- Les filtres d'ordre
- Les filtres homomorphique
- Les filtres morphologiques.

## 6. Extraction de contours

La détection de contour est une étape préliminaire à de nombreuses applications de l'analyse d'images. Les contours constituent en effet des indices riches, au même titre que les points d'intérêts, pour toute interprétation ultérieure de l'image. Les contours dans une image proviennent des :

- discontinuités de la fonction de réflectance (texture, ombre),
- discontinuités de profondeur (bords de l'objet),

et sont caractérisés par des discontinuités de la fonction d'intensité dans les images [Cocquerez.J.P, Philipp.S].



Figure 4. Différents types de contours : marche, toit et pointe

Le principe classique de la détection de contours repose sur l'étude des dérivées de la fonction d'intensité dans l'image : les extrema locaux du gradient de la fonction d'intensité et les passages par zéro du Laplacien.

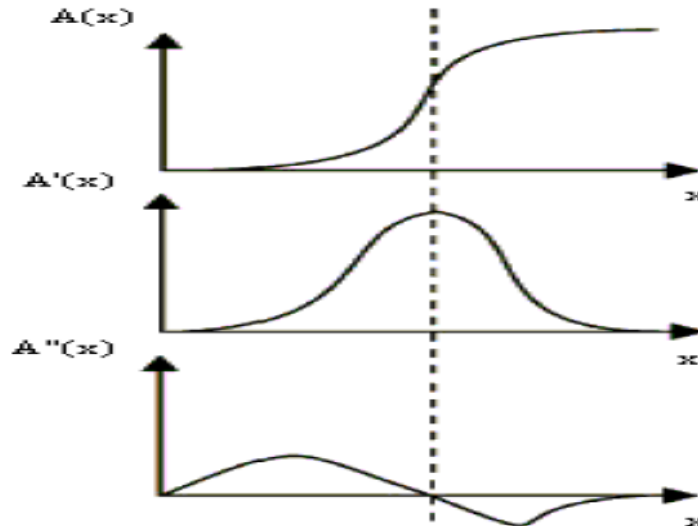


Figure 5. Cas bidimensionnel

### 6.1 Approche Gradient

La recherche de la transition peut être réalisée en recherchant les maxima locaux du gradient. Dans le cas bidimensionnel, le vecteur gradient est défini au point  $M(x, y)$  par :

$$\vec{\nabla} A(x, y) = {}^t \left( \frac{\partial A}{\partial x}, \frac{\partial A}{\partial y} \right) \tag{7}$$

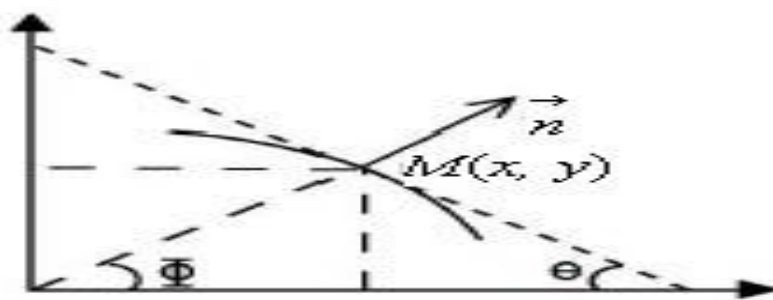


Figure 6. Orientation du contour

$\vec{\nabla} A(x, y)$  est normal à la courbe de niveau donnée par  $A(x, y) = cste$  qui passe par  $M$ .

Au point  $M(x, y)$  l'orientation du gradient est donnée par :  $\Phi = \arctan \left( \frac{\frac{\partial A}{\partial y}}{\frac{\partial A}{\partial x}} \right)$  et le vecteur unitaire  $\vec{n}$  normal au contour :  $\vec{n} = (\cos\Phi, \sin\Phi)$ . (8)

L'orientation du contour a pour expression :  $\theta = \frac{\pi}{2} - \Phi$ . (9)

Pour une image numérique, on a :

$$\frac{\partial A}{\partial y} = \frac{\Delta A}{\Delta i} = A_i[i, j] = A_i[i + 1, j] - A_i[i, j] \quad (10)$$

$$\frac{\partial A}{\partial x} = \frac{\Delta A}{\Delta j} = A_j[i, j] = A_j[i, j + 1] - A_j[i, j] \quad (11)$$

La norme du gradient est :

$$|\nabla A[i, j]| = \sqrt{A_j^2[i, j] + A_i^2[i, j]} \quad (12)$$

Ou :

$$|\nabla A[i, j]| = \max\{A_j[i, j], A_i[i, j]\} \quad (13)$$

Ou bien

$$|\nabla A[i, j]| = |A_j[i, j], A_i[i, j]| \quad (14)$$

### 6.1.1 Opérateurs Prewitt et Sobel

$$h_j = \begin{pmatrix} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{pmatrix} \quad (15)$$

$$A_j[i, j] = h_j * A[i, j] \quad (16)$$

$$A_i[i, j] = h_i * A[i, j] \quad (17)$$

$$h_i = \begin{pmatrix} 1 & c & -1 \\ c & 0 & -c \\ -1 & -c & -1 \end{pmatrix} \quad (18)$$

Les matrices  $h_j$  et  $h_i$  sont appelées masques. Elles sont les noyaux de convolution de filtres à réponse impulsionnelle finie. Pour Prewitt, on prend  $c = 1$ . Pour Sobel, on prend  $c = 2$ .

### 6.1.2 Opérateur de Kirsh

C'est un opérateur à 8 masques correspondant chacun à une direction préférentielle et obtenue par rotation de  $\frac{\pi}{4}$  de l'opérateur de base  $h_0$  (la numérotation des masques est faite dans l'ordre des directions de Freeman en 8-connexité).

$$h_0 = \begin{pmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix} \quad (19)$$

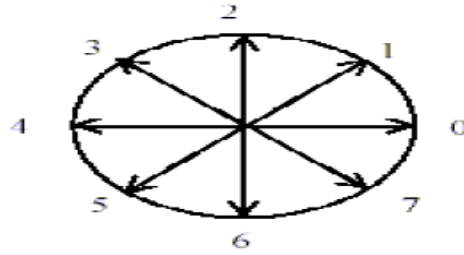


Figure 7. Les directions de Freeman

## 6.2 Approche Laplacien

La recherche des contours dans l'image  $A(x, y)$  peut aussi être réalisée en recherchant les passages par zéro du Laplacien. On dispose donc, de la valeur du Laplacien en chaque point de l'image soit la fonction  $\Delta$ .

$$\Delta A(x, y) = \frac{\partial^2 A}{\partial x^2} + \frac{\partial^2 A}{\partial y^2} \quad (20)$$

On considère que les points de contours sont localisés aux passages par zéro de Laplacien. Si le calcul du Laplacien était exact il suffirait de sélectionner les points  $M$  tels que  $\Delta(M) = 0$ .

## 7. Segmentation

La segmentation d'image est un problème fondamental en vision par ordinateur et en traitement des images.

L'objectif de segmentation d'image est de diviser l'image en région d'intérêt ou d'extraire un ou plusieurs objets dans l'image à traiter.

Parmi les nombreuses méthodes de segmentation existante dans la littérature on trouve :

- La segmentation par approche pixellaire
- La segmentation par approche régions
- La segmentation par approche contours

La première catégorie travaille sur des histogrammes de l'image. Par seuillage, clustering ou clustering flou, l'algorithme construit des classes de couleurs qui sont ensuite projetées sur l'image. La segmentation est implicite puisqu'on suppose que chaque cluster de l'histogramme correspond à une région dans l'image. En pratique, ce n'est pas forcément le cas et il faut séparer les régions de l'image qui sont disjointes bien qu'appartenant au même cluster de couleur. Ces algorithmes sont assez proches des algorithmes de réduction de

couleur.

La deuxième catégorie correspond aux algorithmes d'accroissement ou de découpage de région. L'accroissement de région est une méthode bottom-up : on part d'un ensemble de petites régions uniformes dans l'image (de la taille d'un ou de quelques pixels) et on regroupe les régions adjacentes de même couleur (similaires) jusqu'à ce qu'aucun regroupement ne soit plus possible. Cependant le découpage de région top-down : on part de l'image entière que l'on va subdiviser récursivement en plus petites régions tant que ces régions ne seront pas suffisamment homogènes.

La troisième catégorie s'intéresse aux contours des objets dans l'image. La plupart de ces algorithmes d'accroissement ou de découpage de région. L'accroissement de région est une méthode bottom-up : on part d'un ensemble de petites régions uniformes dans l'image (de la taille d'un ou de quelques pixels) et on regroupe les régions adjacentes de même couleur jusqu'à ce qu'aucun regroupement ne soit plus possible. Le découpage de région des méthodes d'accroissement : on part de l'image entière que l'on va subdiviser récursivement en plus petites régions tant que ces régions ne seront pas suffisamment homogènes.

Algorithmes sont locaux, c'est à dire fonctionnent au niveau du pixel.

Les algorithmes que nous allons présenter sont :

- Accroissement de région fonctionnant au niveau du pixel.
- Split and merge classique.

Des études détaillées de ces algorithmes et d'autres sont données dans [Baillie.J-C].

## 7.1 Accroissement de région

Les méthodes d'accroissement de régions sont les méthodes de segmentation les plus simples. Le principe est basé sur une approche bottom-up : l'algorithme part de petits éléments de l'image qu'il va tenter de regrouper en éléments plus importants.

Nous présentons ici la version de base de l'algorithme d'accroissement de région qui fonctionne en agrégeant des pixels. Supposons une région de couleur homogène  $R$ . Initialement,  $R = 1$  pixel. On va étendre la région  $R$  en incluant les pixels situés sur la frontière et dont la couleur est proche de celle de  $R$  (la variation de couleur est inférieure à un seuil  $\delta$  caractéristique de ce type d'algorithmes). En répétant cette procédure jusqu'à ce qu'il n'y a plus de pixels de couleur assez proche sur la frontière, on obtient une région de couleur homogène maximale autour du pixel de départ. La région initiale "gonfle" en absorbant des pixels de la frontière, jusqu'à stabilité par rapport à la propriété d'homogénéité. Afin d'assurer un critère de connexité, on utilise les voisins en 4-connexités pour

l'accroissement de région et la 8-connexités pour l'accroissement de contour (cas de détection de contours).

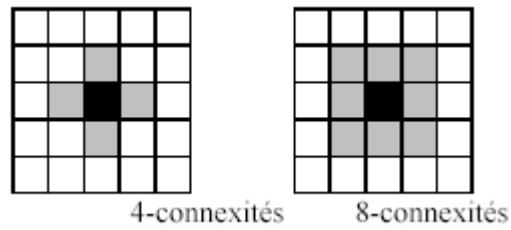


Figure 8. Voisinage d'un pixel

Cette méthode présente deux limitations:

- Les régions obtenues dépendent fortement des pixels d'amorçage choisis et de l'ordre dans lequel les pixels de la frontière sont examinés.
- Le résultat final est très sensible à la valeur du seuil  $\delta$ .

Cet algorithme fait partie de la classe d'algorithmes de segmentation dit "locaux". L'opération élémentaire consiste à manipuler des pixels adjacents et l'algorithme n'a aucune vision globale du résultat qu'il obtient.

## 7.2 Méthodes du type division et fusion (split and merge)

L'algorithme division et /ou fusion "split and merge" a été présenté la première fois en 1974 par Pavlidis et Horowitz [Baillie.J-C]. Cet algorithme s'apparente dans son principe à l'algorithme d'accroissement de région que nous venons de présenter.

La différence principale provient de la nature des régions élémentaires agrégées. Dans l'algorithme "split and merge", les régions agrégées proviennent d'une première phase (split) de traitement de l'image qui construit de manière récursive des régions carrées de taille variable mais homogènes.

### ➤ *Division (split)*

La méthode de découpage de l'image utilisée dans cet algorithme est basée sur la notion de "quadtree". Cette structure de données est un arbre quaternaire qui permet de stocker l'image à plusieurs niveaux de résolution. On part d'une région initiale qui est l'image tout entière. Si cette image vérifie un critère d'homogénéité de couleur, l'algorithme s'arrête. Sinon, on découpe cette région en quatre parties de même taille et on lance la procédure récursivement dans ces quatre parties. La région initiale va être stockée comme un nœud dans un graphe et les sous parties comme des fils de ce nœud. La figure 9 montre une image en noir et blanc 8x8 et le découpage correspondant à chaque niveau, ainsi que la structure d'arbre associée à

ce découpage.

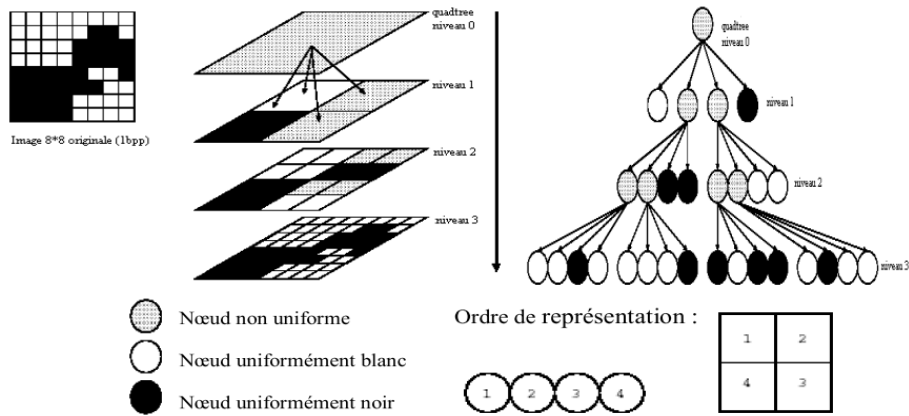


Figure 9. Découpage par quadtree d'une image 8x8.

➤ **Fusion (merge)**

La procédure de découpage décrite précédemment aboutit à un nombre de régions trop élevé. La cause fondamentale de cette sur-segmentation est que l'algorithme découpe les régions de manière arbitraire. Il se peut qu'il coupe de cette façon une zone homogène en deux ou quatre parties (Figure 10).

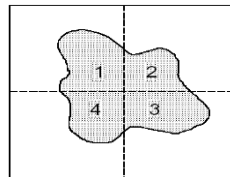


Figure 10. Problème de découpage arbitraire de régions dans la phase de fusionner "split"

Les parties 1, 2, 3 et 4 de la (Figure 10) par exemple, appartiennent à des branches différentes du quadtree. Elles sont donc considérées comme des régions différentes bien que leur couleur soit identique.

La solution, qui correspond à la phase division "merge" de l'algorithme, est de procéder à une fusion de régions après le découpage. L'implémentation la plus simple de cette fusion cherche tous les couples de régions adjacentes dans l'arbre issu du découpage et cherche à les fusionner si leur couleur est assez proche [Baillie.J-C], [Duchesnay.E].

**Conclusion**

Nous avons introduit dans ce chapitre les notions de base qui servent de fondement à la compréhension de différentes techniques de traitement d'images. Plusieurs méthodes classiques de traitement ont été proposés dans la littérature, nous avons présenté quelques-unes qui nous semble les plus courantes dans le processus du traitement et analyse d'image. Les prétraitements d'images permettent d'améliorer la qualité de l'image en vue de traitements ultérieurs. Deux grandes approches peuvent être envisagées pour extraire les zones pertinentes des images l'approche contour et l'approche région.



***Chapitre 2 :***  
***Théorie des graphes et operateur***

## 1. Introduction

Dans ce chapitre, nous introduisons les notions et les définitions nécessaires à la bonne compréhension de la suite de ce mémoire. Dans une première partie, nous rappelons de manière brève les notations et les définitions de la théorie des graphes. Dans une seconde partie, à partir des définitions d'opérateurs définis sur des graphes nous introduisons une nouvelle famille d'opérateurs différentiels qui sont tels que les différences et une classe de Laplaciens. Ces opérateurs, basés sur des équations aux différences partielles [Bougleux.S, Melkemi.M], [Elmoataz.A, Lézoray.O].

## 2. Définitions et notations

D'une manière générale, nous pouvons considérer que toutes données définissent un domaine discret peuvent être représentées par un graphe pondéré  $G = (V, E, w)$ . Dans ce manuscrit, nous ne considérons que des graphes simples (sans arêtes multiples et sans boucles) non-orientés et pondérés.

Un graphe non-orienté  $G$  est un couple  $(V, E)$ . L'ensemble fini  $V$  est composé d'éléments appelés sommets et  $E$  est un ensemble de paires non-ordonnées d'éléments où chaque élément est appelé arête. L'ensemble des arêtes  $E$  est un sous ensemble de  $V \times V$  défini par

$$E = \{uv \subseteq V \times V \text{ avec } u \neq v \text{ et } u, v \in V\} \quad (1)$$

Avec  $uv$  désignant une arête appartenant à l'ensemble  $E$ . Si  $uv$  est une arête du graphe non-orienté  $G$  alors :

- l'arête  $uv$  est incidente aux sommets  $u$  et  $v$  ;
- l'arête  $uv$  relie les sommets  $u$  et  $v$  ;
- les sommets  $u$  et  $v$  sont adjacents ou voisins.

Par la définition de l'ensemble des arêtes  $E$ , il ne peut y avoir d'arête qui relie un sommet à lui-même. Le graphe  $G$  est donc considéré sans boucle.

Le voisinage  $N(u)$ , d'un sommet  $u$ , désigne l'ensemble des voisins de ce sommet.

Il est défini par :

$$N(u) = \{v \in V \setminus u \mid uv \in E\} \quad (2)$$

Dans ce mémoire, nous considérons des graphes pondérés  $G = (V, E, w)$  où  $w : E \rightarrow \mathbb{R}^+$  désigne la fonction de poids du graphe. Cette fonction de pondération est appliquée à l'ensemble des arêtes  $E$ . Pour une arête  $uv \in E$ , la notation  $w_{uv}$  désigne alors la pondération appliquée à l'arête  $uv$  qui satisfait entre autres :

$$w_{uv} \begin{cases} \geq 0 \text{ si } v \in \mathcal{N}(u) \\ = 0 \text{ si } v \notin \mathcal{N}(u) \\ = w_{vu} \end{cases} \quad (3)$$

Le degré  $deg : V \rightarrow \mathbb{R}^+$  d'un sommet  $u$  est la somme des poids des arêtes incidentes à ce sommet :

$$deg_v(u) = \sum_{v \sim u} w_{uv} \quad (4)$$

Où la notation  $v \sim u$  signifie que les sommets  $u$  et  $v$  sont adjacents (ce qui revient à considérer l'ensemble des sommets voisins à  $u$ , c'est-à-dire  $\forall v \in \mathcal{N}(u)$ ).

Nous pouvons remarquer que dans le cas où le graphe  $G$  n'est pas pondéré, alors le degré d'un sommet correspond au nombre d'arêtes incidentes à ce sommet.

À partir de la définition du degré d'un sommet, le volume d'un ensemble de sommets  $A \subseteq V$ ,  $vol : V \rightarrow \mathbb{R}^+$  se définit par :

$$vol(A) = \sum_{u \in A} deg_A(u) = \sum_{u \in A} \sum_{v \sim u} w_{uv} \quad (5)$$

Où  $deg_A$  correspond au degré des sommets relativement à l'ensemble  $A$ .

### 3. Structure et topologie

La structure de graphe est naturellement adaptée pour représenter n'importe quel domaine discret. Ainsi, chaque donnée de ce domaine est représentée par un sommet  $u$  du graphe. Le domaine considéré peut être organisé ou non organisé et les fonctions  $f : V \subset \mathbb{R}^m \rightarrow \mathbb{R}^m$  définies sur les sommets du graphe représentent les données à traiter.

#### 3.1 Domaines non organisés

Considérons le cas général d'un domaine non organisé. Dans cette situation, l'ensemble des points  $V \subset \mathbb{R}^m$  de ce domaine peut être vu comme une fonction définie par  $f : V \subset \mathbb{R}^m \rightarrow$

$\mathbb{R}^m$ . Chaque sommet du graphe est associé à un vecteur d'attributs (ou de caractéristiques) de  $\mathbb{R}^m$ .

Construire un graphe  $G = (V, E, w)$  à partir des données de ce domaine consiste donc à définir l'ensemble des arêtes  $E$  en modélisant le voisinage de chaque sommet du graphe.

Cette modélisation est basée sur des relations de similarité entre les vecteurs caractéristiques des données considérées.

La similarité entre deux sommets du graphe dépend généralement d'une mesure de distance  $\mu : E \rightarrow \mathbb{R}^m$ . Dans le cas des domaines non organisés, la mesure de distance usuelle est généralement la distance Euclidienne.

$$\mu(uv) = \sqrt{\sum_{i=1}^m (f_i(u) - f_i(v))^2} \quad \text{avec } f: V \subset \mathbb{R}^m \rightarrow \mathbb{R}^m \quad (6)$$

D'autres mesures de distances peuvent être également considérées. Le choix de cette mesure dépend de l'application et des données considérées. Elle peut, par exemple, être robuste aux transformations géométriques ou prendre en compte les variations des données à traiter. De plus, il n'y a pas de règle générale qui définisse la structure de graphe la plus adaptée à un domaine donné. Cette décision dépend de l'application considérée et des données à traiter [Hein.M].

Dans ce manuscrit, nous allons présenter différents types de graphes :

**Graphe complet** : le graphe complet est un graphe pondéré tel qu'il existe une arête entre toute les paires de sommet.

Il possède la structure la plus simple à mettre en œuvre, à partir d'un domaine non organisé, ainsi l'ensemble des arêtes sont définies tel que :

$$E = \{V * V\} \quad (7)$$

### Graphe de v-voisinage

Le graphe de v-voisinage, noté  $G_\varepsilon = (V, E, W)$ , est un graphe pondéré dont le voisinage pour un sommet donné  $u$  est défini comme étant l'ensemble des voisins dont la mesure de distance  $\mu$  est inférieure ou égale au paramètre de seuillage  $\varepsilon$ .

Le voisinage  $N_\varepsilon(u)$  pour un sommet  $u$  est alors

$$N_\varepsilon(u) = \{V \setminus \mu(uv) \leq \varepsilon \} \quad (8)$$

$$\text{Est l'ensemble des arrêtes est défini par : } E = \{(u, v)/v \in N_\varepsilon(u)\} \quad (9)$$

### Graphe des $k$ -plus proches voisins

Un graphe des  $k$ -plus proches voisins ( $k$ \_ppv) est graphe pondéré dont chaque sommet est relié à ses plus proches voisins, selon une métrique  $\mu$ . L'ensemble des  $k$  plus proches voisins d'un sommet  $u$  est noté  $K(u)$ . cette définition produit un graphe dirigé l'ensemble des arrêtes  $E$  est défini tel que

$$E = \{(u, v)/u \in K(v) \text{ ou } v \in K(u)\} \quad (10)$$

Cette construction garantit que chaque sommet du graphe est connectée a au moins  $k$  autre sommets.

En contrepartie, un sommet peut posséder jusqu'à  $N - 1$  voisins (ou  $N$  est le nombre totale de sommets).

La figure 11 présente deux exemples de ( $k$ .ppv) sur un ensemble de données discrètes définies dans  $\mathbb{R}^2$ , pour deux valeurs différentes du paramètre  $k$ .

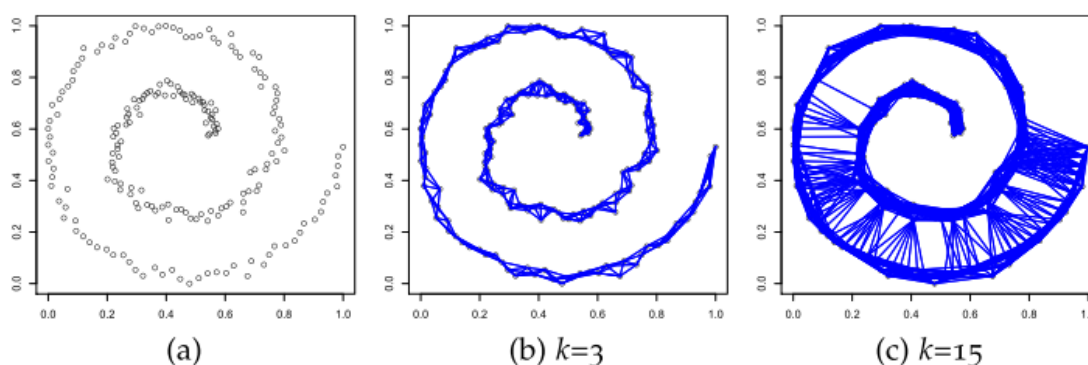


Figure 11. Exemples de graphes des  $k$  plus proches voisins. (a) : données initiales dans  $\mathbb{R}^2$ , (b) : 3-ppv, (c) : 15-ppv.

### 3.2 Domaines organisés

Considérons maintenant le cas des domaines organisés. Nous appelons domaines organisés, des données de type signal ou images (2D ou 3D) pour lesquelles l'organisation spatiale est

connue à priori. Ainsi, ces domaines peuvent être vus comme des fonctions  $f : V \subset \mathbb{Z}^q \rightarrow \mathbb{R}^m$  où  $q = 1, 2$  ou  $3$ , correspondant aux différents cas mentionnés.

Dans le cas des domaines organisés, la mesure de distance  $\mu$  utilisée pour construire un graphe de voisinage correspond généralement à une distance mesurée entre les coordonnées spatiales associées à chaque sommet. Nous pouvons par exemple citer la distance de « city block » (aussi appelée la distance de Manhattan) ou encore celle de Chebychev.

Prenons le cas des images 2D. Si chaque sommet  $u \in V$  est associé à un pixel de l'image (avec les coordonnées spatiales), alors les distances de « city-block » et de Chebychev entre deux sommets  $u = (x_u, y_u)$  et  $v = (x_v, y_v)$  de  $V$  sont respectivement

$$\mu(uv) = |x_u - x_v| + |y_u - y_v| \quad (11)$$

$$\mu(uv) = \max(|x_u - x_v|, |y_u - y_v|) \quad (12)$$

- avec la distance de « city block » et  $v \leq 1$ , le graphe grille en 4-voisinage (connexité), noté  $G_0$  ;

- avec la distance de Chebychev et  $v \leq 1$ , le graphe de grille en 8-voisinage, noté  $G_1$ .

La figure 12 montre deux exemples de graphe grille en 4 et en 8-voisinage. De manière plus générale, les graphes grilles en  $((2s + 1) \times (2s + 1))$ -voisinage sont obtenus à partir de la distance de Chebychev avec  $v \leq s$  et  $s \geq 1$ . Cela correspond à ajouter des arêtes entre le pixel (sommet) central et les autres pixels contenus dans une fenêtre de taille  $(2s + 1) \times (2s + 1)$ .

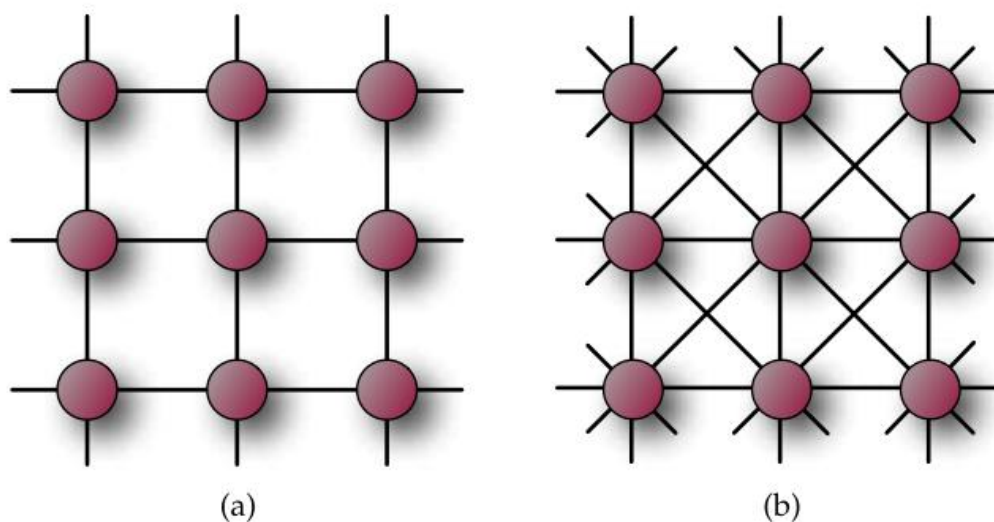


Figure 12. Graphes grille représentant les images. (a) : un graphe grille en 4-voisinage, (b) : un graphe grille en 8-voisinage.

### 3.3 Similarités entre les sommets

Les similarités entre les sommets d'un graphe  $G = (V, E, w)$  peuvent être incorporées dans la fonction de poids  $w$  des arêtes  $E$  par le biais d'une fonction de similarité  $g : E \rightarrow \mathbb{R}^+$  de telle manière que  $w(uv) = g(uv)$  pour une arête  $uv \in E$ .

#### 3.3.1 Vecteurs d'attributs

Les similarités entre les sommets du graphe reposent sur une comparaison des attributs associés à chacun de ses sommets. Ces attributs dépendent généralement d'une fonction initiale  $f^0 : V \subset \mathbb{R}^m \rightarrow \mathbb{R}^m$  définie en chaque sommet. Ainsi, nous définissons le vecteur d'attributs  $F(f^0, u) \in \mathbb{R}^m$  en chaque sommet  $u \in V$  du graphe dépendant de la fonction initiale  $f^0$ .

Il existe différents choix possibles pour  $F_v(f_0, u)$  définir le vecteur d'attributs initial. La définition la plus simple étant de considérer la fonction initiale  $f^0$  elle-même et de définir  $F(f^0, \cdot) = f^0$ . D'autres définitions ont été proposées notamment dans le cas particulier des images 2D.

#### Cas particulier des images 2D

Dans le contexte du traitement des images, une grande famille de vecteurs d'attributs est définie par les patches de l'image. Dans ce cas, le vecteur d'attributs  $F$  est défini en un sommet  $u \in V$  à l'aide du voisinage  $N_v$  tel que

$$\mathcal{F}_v(f^0, u) = (f^0(v) \in V : v \in \mathcal{N}_v(u) \cup \{u\})^T. \quad (13)$$

Dans le cas d'une image en niveaux de gris, le vecteur d'attributs est de taille  $(2v + 1)^2$  et correspond aux valeurs de la fonction  $f^0$  contenues dans une fenêtre de taille  $(2v + 1) \times (2v + 1)$  centrée sur le sommet  $u$ . Dans le cas d'une image multi variée dans  $\mathbb{R}^m$  alors le vecteur d'attributs associé aux sommets du graphe sera de taille  $m \times (2v + 1)$  [Tschumperlé .D].

La figure 13 illustre la définition des vecteurs d'attributs par des patches pour une image en niveaux de gris. Le pixel central (entouré en rouge), au lieu d'être uniquement défini par sa valeur de niveau gris, est caractérisé par un vecteur défini par un patch de l'image. Dans le cas de la figure 13, les patches sont définis comme des fenêtres symétriques de taille  $13 \times 13$ . Cette figure présente quatre patches différents de l'image. Avec cette nouvelle représentation, l'espace des patches est alors un espace de grande dimension ( $\mathbb{R}^{169}$ ).

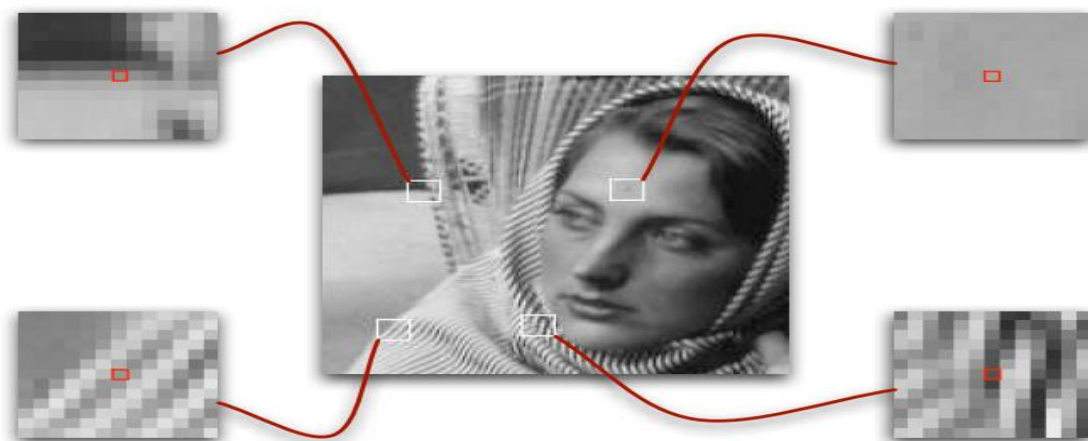


Figure 13. Quatre exemples de patches comme vecteur d'attributs caractérisant les sommets du graphe. Ici, le pixel central (en rouge) est caractérisé par un vecteur de niveaux de gris défini grâce à un patch de taille  $13 \times 13$ .

Afin de pouvoir calculer les poids entre les sommets du graphe qui sont définis par des patches de l'image, la fonction de distance  $\rho : \mathbb{R}^m \times p \times \mathbb{R}^m \times p \rightarrow \mathbb{R}^+$  avec  $p$  la taille du vecteur d'attributs  $F_v(f^0, .)$ . Peut être utilisée

$$\rho^2(F_v(f^0, u), F_v(f^0, v)) = \sum_{x=-v}^v \sum_{y=-v}^v G_a((x, y)) * ||f^0(x_u + x, y_u + y) - f^0(x_v + x, y_v + y)||_2^2 \quad (14)$$

Avec  $G_a$  un noyau Gaussien de variance  $a$  ;  $F_v(f^0, u)$  et  $F_v(f^0, v)$  correspondent aux patches centrés sur les deux sommets (pixels)  $u = (x_u, y_u)$  et  $v = (x_v, y_v)$ .

### 3.3.2 Fonctions de poids

L'utilisation d'une fonction de poids rend compte des similarités entre les données. Cela dépend de l'application considérée et des données elles-mêmes.

Avec la définition des vecteurs d'attributs en chaque sommet d'un graphe, nous pouvons utiliser les définitions des fonctions de poids suivantes.

Soient un graphe pondéré  $G = (V, E, w)$ , la mesure de distance  $\rho$  permettant de comparer les vecteurs d'attributs  $F(f^0, u)$  et  $F(f^0, v)$  des sommets  $u, v \in V$  et

$f^0 : V \subset \mathbb{R}^m \rightarrow \mathbb{R}^m$  Une fonction initiale. Pour une arête  $uv \in E$  donnée, nous considérons les fonctions de poids suivantes :

$$g_1(uv) = 1 \text{ (poids constant)} \quad (15)$$

$$g_2(uv) = (\rho(F(f^0, u), F(f^0, v)) + \varepsilon)^{-1} \quad (16)$$

$$g_3(uv) = 1 - \left( \frac{\rho(F(f^0, u), F(f^0, v))}{\max} u(F(f^0, \cdot), F(f^0, \cdot)) \right) \quad (17)$$

$$g_4(uv) = \exp\left(-\rho\left(F(f^0, u), \frac{F(f^0, v)}{v^2}\right)\right) \quad (18)$$

L'utilisation d'une fonction de poids (avec ou sans patches) permet de capturer les détails des images et de rendre compte des similarités et des interactions entre les pixels.

## 4 Opérateurs définis sur des graphes

Dans cette section, nous rappelons les définitions des opérateurs différentiels sur graphes introduites par [Elmoataz.A, Lézoray.O], [Hein.M]. Ces définitions peuvent être vues comme étant les versions sur graphes pondérés des opérateurs de géométrie différentielle classiques définis dans le domaine continu. Des démarches similaires ont été proposées en apprentissage, en segmentation d'images, en analyse fonctionnelle sur graphe ou encore en théorie spectrale des graphes.

## 4.1 Fonctions définies sur des graphes

Soit  $f : V \rightarrow \mathbb{R}$  une fonction scalaire qui associe à chaque sommet  $u \in V$  une valeur réelle  $f(u)$ . Pour un graphe  $G = (V, E, w)$ , cette fonction peut être vue comme un vecteur colonne défini comme  $f = (f(u))_{u \in V}^T$ . Par analogie avec l'espace des fonctions continues, l'intégrale discrète de la fonction  $f$  sur  $G$  est définie par

$$\int_G f = \sum_{u \in V} f(u) \quad (19)$$

Nous définissons l'espace des fonctions sur des graphes. Soit  $H(V)$  l'espace de Hilbert des fonctions à valeurs réelles définies sur les sommets  $V$  du graphe  $G$ .

L'espace de Hilbert  $H(V)$  est muni du produit scalaire.

$$\langle f, h \rangle_{H(V)} = \sum_{u \in V} f(u)h(u) \quad \text{avec } f, h : V \rightarrow \mathbb{R} \quad (20)$$

Et la norme  $L_2$ ,  $\|f\|_2 = \sqrt{\langle f, f \rangle_{H(V)}}$ .

De la même manière, soit  $H(E)$  l'espace des fonctions  $F : E \rightarrow \mathbb{R}^+$  définies sur les arêtes et également muni du produit scalaire.

$$\langle F, H \rangle_{H(E)} = \sum_{u \in V} \sum_{v \sim u} F(uv)H(uv) \quad \text{avec } F, H : E \rightarrow \mathbb{R} \quad (21)$$

Et la norme  $\mathcal{L}_2$

$$\|F\|_2 = \sqrt{\langle F, F \rangle_{H(E)}}.$$

Dans ce mémoire, nous ne considérons que des fonctions scalaires  $f : V \rightarrow \mathbb{R}$  afin de simplifier les notations et la compréhension de ce chapitre.

## 4.2 Opérateur de différence et adjoint

### 4.2.1 Opérateur de différence

L'opérateur de différence pondérée  $d_w$  d'une fonction  $f \in H(V)$  est une application de  $H(V) \rightarrow H(E)$ . Elle est définie en un sommet  $u$  selon une arête  $uv \in E$  [Bougleux.S, Melkemi.M], [Elmoataz.A, Lézoray.O]:

$$(d_w f)(uv) = \frac{f(v) - f(u)}{h_{uv}} \quad (22)$$

Où  $h_{uv}$  un pas d'échantillonnage entre  $f(u)$  et  $f(v)$  dépendant du poids  $w$  entre les sommets  $u$  et  $v$  de  $V$ . Ce pas peut être  $vu$  comme la version discrète sur graphe du taux d'accroissement d'une fonction continue.

Par la suite, nous utilisons un pas  $h_{uv}$  faisant intervenir le poids  $w_{uv}$  d'une arête  $uv$  telle que  $h_{uv} = 1/\sqrt{w_{uv}}$ , donnant ainsi la définition suivante pour une arête  $uv \in E$  et une fonction  $f \in H(V)$

$$(d_w f)(uv) = \sqrt{w_{uv}}(f(v) - f(u)) \quad (23).$$

Ce choix permet de retrouver les versions discrètes et pondérés des opérateurs gradients et Laplace définis sur des graphes et d'intégrer les interactions non locales dans le contexte des images.

Par analogie avec le calcul différentiel en continu, nous pouvons voir la différence pondérée comme la version discrète définie sur un graphe de l'opérateur de dérivée appliquée à une arête  $uv \in E$  :

$$\left. \frac{\partial f}{\partial uv} \right|_u = \partial_v f(u) = (d_w f)(uv) \quad (24)$$

Cette définition est consistante avec la définition de la dérivée d'une fonction continue définie dans l'espace Euclidien et elle possède également les propriétés suivantes

$$\partial_v f(u) = \begin{cases} -\partial_u f(v) \\ 0 & \text{si } v = u \\ 0 & \text{si } f(v) = f(u) \end{cases} \quad (25)$$

### 4.2.2 Opérateur adjoint

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

$$\langle d_w f, H \rangle_{\mathcal{H}(E)} = \langle f, d_w^* H \rangle_{\mathcal{H}(V)}$$

Pour toutes les fonctions  $f \in \mathcal{H}(V)$  et  $H \in \mathcal{H}(E)$ . En utilisant les produits scalaires, l'expression de l'opérateur adjoint d'une fonction  $F \in H(E)$  est [Bougleux.S,Melkemi.M], [Elmoataz.A, Lézoray.O].

$$(d_w^* F)(u) = \sum_{v \sim u} \sqrt{w_{uv}} (F(vu) - F(uv)). \quad (26)$$

Où la notation  $v \sim u$  signifie que  $v$  est voisin de  $u$ , c'est-à-dire  $v \in N(u)$ .

Par analogie avec les opérateurs différentiels continus, l'opérateur de divergence d'une fonction  $F \in H(E)$  est

$$\operatorname{div} F = -d_w^* F. \quad (27)$$

### 4.3 Opérateur gradient

L'opérateur gradient pondéré  $\nabla_w: \mathcal{H}(V) \rightarrow \mathbb{R}^m$  d'une fonction  $f \in \mathcal{H}(V)$  en un sommet  $u \in V$  est défini comme étant le vecteur de toutes les différences pour l'ensemble des arêtes de  $E$

$$(\nabla_w f)(u) = ((d_w f)(uv))_{uv \in E}^T. \quad (28)$$

La définition de ce gradient ne dépend pas de la structure du graphe. Par conséquent, le gradient aura toujours la même expression quelles que soient les données qui seront représentées par le graphe considéré. Cet opérateur est de premier ordre, défini localement en un sommet  $u \in V$  et dépend de son voisinage  $N(u)$ . Afin de mesurer la variation locale en un sommet  $u \in V$ , différentes normes peuvent être utilisées telles que les normes  $L_q$ . Ainsi pour  $0 < q < +\infty$ ,

$$\|(\nabla_w f)(u)\|_q = \left[ \sum_{v \sim u} w_{uv}^{q/2} |f(v) - f(u)|^q \right]^{1/q} \quad (29)$$

Et pour  $q = \infty$

$$\|(\nabla_w f)(u)\|_\infty = \max_{v \sim u} (\sqrt{w_{uv}}) |f(v) - f(u)| \quad (30)$$

**Remarque.** Le gradient pondéré défini sur un graphe intègre naturellement les interactions non locales dans sa formulation. Avec la définition de la fonction de poids  $w$  (pour  $u \in V, w_{uv} = 0$  pour tout  $v \notin N(u)$ ) les normes (8) et (9) peuvent se réécrire où la notation  $v \sim u$  est remplacée par  $v \in V$ .

$$\|(\nabla_w f)(u)\|_q = \left[ \sum_{v \in V} w_{uv}^{q/2} |f(v) - f(u)|^q \right]^{1/q} \quad (31)$$

$$\|(\nabla_w f)(u)\|_\infty = \max_{v \sim u} \sqrt{w_{uv}} |f(v) - f(u)| \quad (32)$$

Où la notation  $v \sim u$  est remplacée par  $v \in V$ .

#### 4.4 p-Laplacien isotrope

Une expression du p-Laplacien pondéré isotrope peut être obtenue [9] à partir de l'opérateur de différence et de son adjoint.

Soit  $p$  un nombre réel tel que  $0 < p < +\infty$ . Le p-Laplacien pondéré isotrope

$\Delta_{w,p}^i: H(V) \rightarrow H(V)$  D'une fonction  $f \in H(V)$  est défini par

$$\Delta_{w,p}^i f = d_w^* (\|\nabla_w f\|_2^{p-2} d_w f). \quad (33)$$

L'expression de cet opérateur en un sommet  $u \in V$ , pour la fonction  $f$  est

$$(\Delta_{w,p}^i f)(u) = \sum_{v \sim u} \sqrt{w_{uv}} \left( \frac{(d_w f)(vu)}{\|\nabla_w f(v)\|_2^{2-p}} - \frac{(d_w f)(uv)}{\|\nabla_w f(u)\|_2^{2-p}} \right) \quad (34)$$

$$= \sum_{v \sim u} w_{uv} \left( \frac{f(u) - f(v)}{\|\nabla_w f(v)\|_2^{2-p}} - \frac{f(v) - f(u)}{\|\nabla_w f(u)\|_2^{2-p}} \right) \quad (35)$$

$$= \sum_{v \sim u} w_{uv} (\|\nabla_w f(v)\|_2^{p-2} + \|\nabla_w f(u)\|_2^{p-2}) (f(u) - f(v)). \quad (36)$$

Le p-Laplacien isotrope (12) est relié à différentes formulations de l'opérateur de Laplace [Hein.M], [Osher.S, Shen.J].

Dans le cas où le paramètre  $p=2$ , le p-Laplacien isotrope (12) se réduit à l'expression suivante

$$\Delta_{w,2}^i f = d_w^* (d_w f) = \frac{1}{2} \Delta_w f \quad (37)$$

Définissant ainsi le Laplacien pondéré d'une fonction  $f \in H(V)$  sur graphe. En un sommet  $u \in V$  son expression est

$$(\Delta_w f)(u) = 2 \sum_{v \sim u} w_{uv} (f(u) - f(v)) \quad (38)$$

Ce Laplacien est linéaire et correspond au Laplacien combinatoire [Chung.F].

Dans le cas où le paramètre  $p = 1$ , le  $p$ -Laplacien isotrope (12) se réduit à l'expression suivante

$$\Delta_{w,1}^i f = d_w^* (\|\nabla_w f\|^{-1} d_w f) = k_w f \quad (39)$$

Définissant ainsi l'opérateur de courbure pondéré d'une fonction  $f \in H(V)$  sur graphe. En un sommet  $u \in V$  son expression est

$$(K_w f)(u) = \sum_{v \sim u} w_{uv} \left( \frac{1}{\|(\nabla_w f)(v)\|_2} + \frac{1}{\|(\nabla_w f)(u)\|_2} \right) (f(u) - f(v)). \quad (40)$$

L'opérateur de courbure sur graphe peut être vu comme étant la version discrète de l'opérateur de courbure d'une fonction continue  $f: \Omega \subset \mathbb{R}^m \rightarrow \mathbb{R}$  définie comme

$$k f = d_w^* \left( \frac{\nabla f}{\|\nabla f\|_2} \right) = -\operatorname{div} \left( \frac{\nabla f}{\|\nabla f\|_2} \right) \quad (41)$$

Dans le cas d'un graphe non pondéré, l'expression de la courbure peut être considérée comme étant la version discrète de l'opérateur de courbure utilisé dans le contexte du traitement des images [Osher.S, Shen.J], [Chan.T.F].

Finalement, le  $p$ -Laplacien isotrope peut être vu comme étant une généralisation du Laplacien combinatoire et de l'opérateur de courbure

$$\Delta_{w,p}^i f = -\operatorname{div} \left( \frac{\nabla f}{\|\nabla f\|_2^{2-p}} \right) \quad (42)$$

## 5 $p$ -Laplacien anisotrope

À partir du  $p$ -Laplacien isotrope (12), une extension possible est le  $p$ -Laplacien pondéré anisotrope  $\Delta_{w,p}^a: \mathcal{H}(V) \rightarrow \mathcal{H}(V)$  appliqué à une fonction  $f \in \mathcal{H}(V)$  pour toutes valeurs réelles  $0 < p < +\infty$ . Il se définit par

$$\Delta_{w,p}^a f = d_w^* (|d_w f|^{p-2} d_w f) \quad (43)$$

Avec  $|\cdot|$  désignant la valeur absolue.

Comme pour le  $p$ -Laplacien isotrope (12), nous pouvons déduire l'expression de l'opérateur de  $p$ -Laplace anisotrope en un sommet  $u \in V$  à l'aide la définition de la différence pondérée et de son adjoint.

$$(\Delta_{w,p}^a f)(u) = \sum_{v \sim u} \sqrt{w_{uv}} \left( \frac{(d_w f)(vu)}{|(d_w f)(vu)|^{2-p}} - \frac{(d_w f)(uv)}{|(d_w f)(uv)|^{2-p}} \right) \quad (44)$$

$$= \sum_{v \sim u} w_{uv}^{p/2} \left( \frac{(f(u) - f(v))}{|f(u) - f(v)|^{2-p}} - \frac{(f(v) - f(u))}{|f(v) - f(u)|^{2-p}} \right) \quad (45)$$

$$= \sum_{v \sim u} w_{uv}^{p/2} (f(u) - f(v)) |f(u) - f(v)|^{p-2}. \quad (46)$$

Lorsque le paramètre  $p=2$ , les  $p$ -Laplacien isotrope et anisotrope ont la même expression.

$$(\Delta_{w,2}^a f)(u) = (\Delta_{w,2}^i f)(u) = \sum_{v \sim u} w_{uv} (f(u) - f(v)) \quad (47)$$

Lorsque  $p=1$ , l'opérateur de  $p$ -Laplace anisotrope peut se réécrire de la manière suivante :

$$(\Delta_{w,1}^a f)(u) = \sum_{v \sim u} \sqrt{w_{uv}} \operatorname{sign}(f(u) - f(v)) \quad (48)$$

Où (*sign*) désigne la fonction signe avec l'identité suivante  $\operatorname{sign}(a) = a/|a|$ .

## Conclusion

Dans ce chapitre, nous avons rappelé les types de graphes et les définitions nécessaires pour la compréhension de la suite de ce manuscrit. Nous avons rappelé des définitions sur la structure de graphe et les opérateurs différentiels qui lui sont associé.

La méthode que nous aborderons dans la suite de ce mémoire utilise les opérateurs différentiels et la structure de graphe présentés dans ce chapitre.

# ***Chapitre 3 :***

## **Description de la méthode**

### 1. Introduction :

Dans ce chapitre, nous développons une approche variationnelle pour la détection d'objet(s) en utilisant la méthode de segmentation binaire non locale. Le modèle que nous définissons met en jeu un ensemble de régions et de contours. Le critère proposé contient des termes liés à l'information sur les régions et au niveau des contours. Dans ce qui suit, nous présenterons la méthode proposée.

### 2. Les méthodes variationnelles :

Les méthodes variationnelles font appel à des outils mathématiques liés au calcul des variations ainsi qu'à l'analyse fonctionnelle. Le principe consiste à trouver la solution d'un problème par la minimisation fonctionnelle qui joue le rôle d'une énergie. Cette minimisation s'effectue en général en déterminant une équation d'évolution représentant une descente de gradient qui converge vers un minimum. Plusieurs énergies discrètes sont étudiées et des algorithmes linéaires et non linéaires sont proposés afin d'approximer leurs solutions. L'énergie choisie est la combinaison de fonctions coût qui favorisent la présence de propriétés désirées, comme par exemple la régularité sur les objets étudiés.

### 3. Segmentation

Rappelons que la segmentation d'images est un problème fondamental en vision par ordinateur et en traitement des images. L'objectif de la segmentation d'image est de diviser l'image en régions d'intérêts ou d'extraire un ou plusieurs objets dans l'image à traiter. Parmi les nombreuses approches qui ont été proposées pour résoudre les problèmes rencontrés en traitement d'images, les méthodes variationnelles sont très exploitées dans ce domaine.

Par exemple, dans le cas de la segmentation des images, les méthodes variationnelles basées sur la régularisation et les équations aux dérivées partielles sont très utilisées. Cependant, peu de travaux intégrant les configurations non locales dans les modèles de segmentation existent dans la littérature. Les travaux qui ont tout de même abordé ceci reposent principalement sur une extension de la variation totale vers des schémas non locaux [Szlam.A.D], [Houhou.J].

Pour segmenter une image, Mumford et Shah ont proposé de minimiser la fonctionnelle [Mumford.D]:

$$E(f) = \int_{\Omega} |\nabla f(u)|^2 du + \frac{\lambda}{2} \int_{\Omega} |f(u) - f_0(u)|^2 du \quad (1)$$

Où  $f_0$  l'image originale définie sur  $\Omega \subset \mathcal{R}^2$ ,  $f$  image reconstruite et  $\lambda$  un réel positif. Le premier terme de cette fonctionnelle, mesure la régularité (légalité) de  $f$  à l'intérieur d'une région, le second la qualité de l'approximation.

Quand  $f_0$  est une image en niveaux de gris, la norme  $|f(u) - f_0(u)|$  est la valeur absolue de la différence des deux niveaux de gris. Si  $f_0$  est une image couleur, une norme pondérée sur les différents canaux est utilisée.

Dans ce qui suit, nous présentons une autre classe de modèle variationnel basée sur le modèle anisotrope.

#### 4. Fonctionnelle de régularisation

Dans cette section, nous présentons une fonctionnelle de régularisation définie sur les graphes. Cette fonctionnelle est basée sur les normes des gradients présentés dans le chapitre 2. De plus, elle peut être vue comme étant une version discrète de la fonctionnelle de régularisation non locale définie dans le domaine continu et proposée par Gilboa et Osher [Gilboa.G, Osher.S]. Cette fonctionnelle définie sur un graphe pondéré  $G = (V, E, w)$  est donnée par :

$$\min_{f:V \rightarrow \mathcal{R}} \left\{ E_1(f, f_0, \lambda) = R_{w,q}(f) + \sum_{u \in V} \frac{\lambda}{2} \|f(u) - f_0(u)\|_2^2 \right\} \quad (2)$$

Où le terme de régularisation  $R_{w,q}$  utilise l'opérateur de sommation et le gradient symétrique  $\nabla_w$  tel que :

$$R_{w,q}(f) = \sum_{u \in V} \phi(\|(\nabla_w f)(u)\|_q) \quad (3)$$

Cette fonctionnelle est basée sur la norme  $\mathcal{L}_q$  du gradient de la fonction  $f \in H(V)$  et fait intervenir la fonction  $\phi(\cdot)$  (noyau de régularisation) permettant de pénaliser les larges variations de la fonction  $f$  dans le voisinage de chaque sommet.

Différentes expressions du noyau  $\phi(\cdot)$  ont été proposées dans la littérature. Nous pouvons par exemple citer les cas suivants.

- La fonctionnelle de régularisation de Tikhonov [Tikhonov.A.N] :

$$\sum_{u \in V} \|(\nabla_w f)(u)\|_q^2 \quad (4)$$

Avec  $\phi(s) = s^2$

- La fonctionnelle de régularisation basée sur la variation totale :

$$\sum_{u \in V} \|(\nabla_w f)(u)\|_q \quad (5)$$

Avec  $\phi(s) = s$

Rappelons que la norme  $\mathcal{L}_q$  avec  $0 < q < \infty$  du gradient symétrique  $\nabla_w$  d'une fonction  $f \in H(V)$  sur un graphe pondéré  $G = (V, E, w)$  en un sommet  $u \in V$  est :

$$\|(\nabla_w f)(u)\|_q = \left[ \sum_{v \sim u} w_{uv}^{\frac{q}{2}} |f(v) - f(u)|^q \right]^{\frac{1}{q}} \quad (6)$$

Dans notre travail, nous optons pour un autre noyau de régularisation  $\phi(s) = \frac{1}{2p} s^p$ , sa dérivé  $\phi'(s) = \frac{1}{2} s^{p-1}$  et pour  $q = p$ , nous obtenons le modèle p\_Tv anisotrope suivant :

$$\min_{f: V \rightarrow R} \left\{ E_a(f, f_0, \lambda) = \frac{1}{2p} \sum_{u \in V} \|(\nabla_w f)(u)\|_p^p + \sum_{u \in V} \frac{\lambda}{2} \|f(u) - f_0(u)\|_2^2 \right\} \quad (7)$$

Qui peut se réécrire de la manière suivante :

$$\min_{f: V \rightarrow R} \left\{ E_a(f, f_0, \lambda) = \frac{1}{2p} R_{w,p}(f) + \sum_{u \in V} \frac{\lambda}{2} \|f(u) - f_0(u)\|_2^2 \right\} \quad (8)$$

(8)

Où le terme de régularisation  $R_{w,p}$  utilise l'opérateur de sommation et le gradient symétrique  $\nabla_w$  tel que :

$$R_{w,p}(f) = \sum_{u \in V} \|(\nabla_w f)(u)\|_p^p = \sum_{u \in V} \sum_{v \sim u} w_{uv}^{\frac{p}{2}} |f(v) - f(u)|^p \quad (9)$$

## 5. Modèle variationnel proposé

Cette section présente le modèle variationnelle non local que nous proposons pour régulariser des fonctions définies sur les sommets de graphes pondérés. Soit  $G = (V, E, w)$  un graphe pondéré, et  $f_0: V \rightarrow R$  une fonction donnée de  $H(V)$ . La régularisation de la fonction  $f_0$

consiste à trouver une fonction  $f \in H(V)$  qui soit proche de  $f_0$ , et suffisamment régulière sur  $G$ . Nous formalisons ce problème d'optimisation comme la minimisation de la somme de trois fonctionnelles de la forme :

$$\min_{f \in H(V)} \left\{ \begin{aligned} E_w(f, f_0, \lambda, \varepsilon) &= \frac{1}{2p} \sum_{u \in V} \|\nabla_w f(u)\|_p^p + \frac{\lambda}{2} \sum_{u \in V} (f(u) - f_0(u))^2 \\ &+ \frac{1}{4\varepsilon^2} \sum_{u \in V} (1 - (f(u))^2)^2 \end{aligned} \right\} \quad (10)$$

avec,  $p \in ]0, \infty[$ .

La première fonctionnelle mesure la régularité de la fonction  $f$  sur le graphe  $G$ . La seconde fonctionnelle mesure l'écart entre la fonction  $f$  et la fonction initiale  $f_0$ . La compétition entre les deux termes est contrôlée par le paramètre de fidélité  $\lambda \geq 0$ , ou multiplicateur de Lagrange. Quant au troisième terme, il permet de réaliser une partition de l'image en deux régions homogènes.  $\varepsilon$  est un paramètre qui caractérise la taille de la transition entre les deux régions -1 et 1. Ces deux états stables joueront un rôle essentiel dans l'identification des régions noires ou blanches. La transition entre les états proche de -1 et 1 est soumise à une compétition entre le terme diffusif et le terme non linéaire.

La solution du problème de minimisation (10) en un sommet  $u \in V$  peut être obtenue en résolvant le système d'équations :

$$\frac{\partial E_w(f, f_0, \lambda, p, \varepsilon)}{\partial f(u)} = \frac{\partial}{\partial f(u)} \left( \frac{1}{2p} R_{w,p}(f) + \frac{\lambda}{2} \|f - f_0\|_2^2 + \frac{1}{4\varepsilon^2} (1 - (f(u))^2)^2 \right) = 0, \forall u \in V \quad (11)$$

La dérivée de la fonctionnelle  $R_w(f, p)$  par rapport à  $f$  en chaque sommet du graphe, correspond à l'expression du p-Laplacien anisotrope, tel que :

$$\begin{aligned} \frac{1}{2p} \frac{\partial R_{w,p}(f)}{\partial f(u)} &= \frac{\partial}{\partial f} \left( \frac{1}{2p} \sum_{u \in V} \|\nabla_w f(u)\|_p^p \right) = \frac{\partial}{\partial f} \left( \frac{1}{2p} \sum_{u \in V} \sum_{v \sim u} w^{\frac{p}{2}} |f(v) - f(u)|^p \right) \\ &= \sum_{v \sim u} w^{\frac{p}{2}} |f(u) - f(v)|^{p-2} (f(u) - f(v)) = (\Delta_{w,p}^a f)(u) \end{aligned} \quad (12)$$

D'où :

$$\frac{1}{2p} \frac{\partial R_{w,p}(f)}{\partial f(u)} = \sum_{v \sim u} w^{\frac{p}{2}} |f(u) - f(v)|^{p-2} (f(u) - f(v)) = (\Delta_{w,p}^a f)(u) \quad (13)$$

La solution du problème (11) devient :

$$(\Delta_{w,p}^a f)(u) + \lambda(f(u) - f_0(u)) - \frac{1}{\varepsilon^2} f(u) (1 - (f(u))^2) = 0, \forall u \in V \tag{14}$$

Cette dernière équation peut être considérée comme l'équation d'Euler-Lagrange associée au problème de minimisation (10). Elle est basée sur le p-Laplacien anisotrope, et pour  $p = 2$ , nous retrouvons le cas linéaire basé sur le Laplacien combinatoire. En remplaçant l'expression du p-Laplacien anisotrope dans l'équation (14), nous obtenons le système d'équations :

$$\sum_{v \sim u} w^{\frac{p}{2}} |f(u) - f(v)|^{p-2} (f(u) - f(v)) + \lambda(f(u) - f_0(u)) - \frac{1}{\varepsilon^2} f(u) (1 - (f(u))^2) = 0 \tag{15}$$

Dans notre travail, nous proposons d'utiliser la méthode de descente de marche infinitésimale (méthode de diffusion semi discrète) pour approximer la solution du système d'équations (15) :

$$\begin{cases} f = f_0 \\ \frac{df^{(t)}}{dt} = -(\Delta_{w,p}^a f)(u) + \lambda(f_0(u) - f(u)) + \frac{1}{\varepsilon^2} f(u) (1 - (f(u))^2), \quad \forall u \in V \end{cases} \tag{16}$$

Où  $f(t)$  est la paramétrisation de la fonction  $f$  par un temps artificiel  $t$ . C'est un système d'équations différentielles ordinaires, qui décrit le processus de diffusion. La solution de ce système peut être établie en utilisant différentes méthodes. Nous proposons d'utiliser la méthode itérative de Runge-Kutta d'ordre 1 pour sa qualité d'approximation et sa simplicité.

Une itération de la méthode est donnée par l'expression suivante :

$$\begin{cases} f = f_0 \\ f^{(t+1)}(u) = f^{(t)}(u) + \Delta t \left( \sum_{v \sim u} w^{\frac{p}{2}} |f(v) - f(u)|^{p-2} (f(v) - f(u)) + \lambda(f_0(u) - f^{(t)}(u)) + \frac{1}{\varepsilon^2} f^{(t)}(u) (1 - (f^{(t)}(u))^2) \right), \quad \forall u \in V \end{cases} \tag{17}$$

Où  $\Delta t$  est la taille de la marche infinitésimale. Dû à l'introduction de ce dernier paramètre, le processus de diffusion est semi discret. La diffusion de la fonction  $f_0$  à l'instant  $t$  est la fonction  $f^t$ , initialisée en  $t = 0$  à  $f_0$ . Le nombre d'itérations dépend généralement du contexte, résultat souhaité à obtenir. Néanmoins, pour satisfaire le système (14), le processus nécessite d'être itéré jusqu'à ce qu'il converge, c.à.d.  $f^{(t+1)} = f^{(t)}$ . Une façon simple de

déterminer la convergence est de tester à chaque itération comment la fonction varie par rapport à l'itération précédente.

$$\|f^{(t+1)} - f^{(t)}\|_2 < \vartheta \|f^{(t+1)}\|_2, \quad \vartheta \rightarrow 0 \quad (18)$$

## 6. Conclusion

Dans ce chapitre, nous avons présenté un modèle variationnel non local pour régulariser des fonctions définies sur des graphes pondérés. Nous avons proposé un type de fonctionnelle de régularisation basé sur la norme  $\mathcal{L}_p$  du gradient discret. Dans le prochain chapitre, nous appliquerons le modèle proposé pour la segmentation binaire en vue d'extraire des régions d'intérêts.

L'implémentation de la méthode sous forme d'un logiciel, ainsi que des expérimentations et des résultats seront présentés dans le chapitre suivant.

***Chapitre 4 :***  
***Mise en oeuvre***

## 1. Introduction

Dans ce chapitre, nous allons aborder l'aspect pratique (logiciel) de la méthode de segmentation que nous avons adoptée et détaillée au cours du chapitre précédent. Nous commençons par la description de la base utilisée, le choix de l'environnement de travail.

Par la suite, nous proposerons une série d'expérimentations montrant le comportement du modèle proposé pour la segmentation binaire en utilisant des images réelles. Ces expérimentations permettent d'illustrer l'efficacité de notre modèle pour la détection d'objet(s) dans une image en niveaux de gris en utilisant la segmentation binaire. Ici, nous nous intéressons aux images en niveaux de gris. Dans le cas des images couleurs, une transformation permettant le passage d'un espace couleur à un espace monochrome (niveaux de gris) sera nécessaire.

## 2. Installation du logiciel

### 2.1 Environnement de travail

Pour que notre travail atteigne l'objectif qu'on visait, on a pris l'initiative d'exploiter et d'implémenter notre algorithme sur la version : Windows 7 (4 Go de RAM et un processeur i3 M380 @2.53 GHz). Ce choix se traduit par l'efficacité de cet environnement, en ce qui concerne la structure d'interaction événementielle qu'elle dispose pour communiquer avec des applications actives, ainsi que les ressources de la machine qu'il offre aux différentes applications, enfin, son système d'allocation de mémoire qui est un des meilleurs présents dans ce domaine.

### 2.2 Le langage de codage

Cette application a été codée en sa globalité par le langage C++ à travers la plateforme Qt (prononcez "Quioute" venu du mot anglais « Cute » ce qui signifie "Mignonne") ce choix repose sur le fait que Qt est une bibliothèque Framework, Qt est donc pour créer des GUI (programmes de fenêtres). Elle est écrite en C++, à la base elle est faite pour être utilisée qu'en C++, néanmoins il est possible de l'utiliser dans d'autres langages comme Java, Python [[https://fr.wikipedia.org/wiki/Qt\\_Creator](https://fr.wikipedia.org/wiki/Qt_Creator)].



Figure 16. Le logo de Qt

Comme nous l'avons dit, Qt n'est pas une simple bibliothèque pour développer des applications graphiques, mais elle est un Framework qui veut dire qu'elle regroupe plusieurs bibliothèques (modules) de domaines différents, comme :

- ✓ **Module GUI:** C'est la partie dédiée à la création des fenêtres.
- ✓ **Module OpenGL:** Le module qui a la tâche de 3D et gérer par OpenGL.
- ✓ **Module réseau:** Qt toute une boîte d'outils pour accéder au réseau, que ce soit pour créer un logiciel de Chat, un client FTP, un client Bittorrent, un lecteur de flux RSS...
- ✓ **Module de script :** La partie Javascript (ou ECMAScript), qu'on peut l'utiliser dans des applications pour ajouter des fonctionnalités, sous forme de plugins.
- ✓ **Module XML:** Une partie puissante pour le XML, afin d'échanger des données avec des fichiers formés à l'aide de balises, presque comme le XHTML.
- ✓ **Module SQL:** un module qui permet un accès aux bases de données (MySQL, Oracle, PostgreSQL...).

En plus , un autre critère nous a poussé vers ce choix , c'est que Qt est multiplateforme ,c'est-à-dire, que vous n'avez pas besoin de coder 3 fois le même code pour 3 différents OS (système d'exploitation), d'ailleurs la principale idée à laquelle Qt s'appuie , c'est « codez une seule fois (sous n'importe quel OS) , et vous pouvez le compiler dans tous les OS pour générer l'exécutable correspondant » ce qui explique la rapidité de ses applications par rapport aux autres langages comme java.

Qt dispose d'un nombre important d'outils de développement d'applications, comme :

- ✓ **QtDesigner:** Qui est un outil pour le dessin d'applications graphiques visuellement en glissant les composantes (Widgets) en question sur la fenêtre, c'est-à-dire vous n'aurez pas besoin de coder tout le code qui génère la fenêtre.

- ✓ **QtLinguist:** Un très bon moyen pour distribuer vos applications en plusieurs langues, ce que vous devrez le faire c'est de respecter certaines normes de nommages et à la fin de votre développement vous passez votre code dans cet outil pour générer un petit fichier linguistique, qui va servir comme moyen afin de traduire votre application dans la langue que vous choisissez.
- ✓ **QtAssistant:** Un autre performant style pour accompagner les développeurs pendant et durant toute la période de la réalisation des applications, grâce à Qt Assistant vous pouvez trouver toute la documentation de Qt , bien structurée d'une manière qui ne laisse pas le doute sur une fonctionnalité d'une telle fonction ou autre, ainsi que une grande liste des exemple bien commentés pour donner un aperçu sur la fonction en question.

Il est à signaler que Qt n'utilise pas que le langage C++, mais elle a son propre langage , ses propres structures ,conventions de nommage, ses propres Classes . . .etc. dont celle regroupe entre la rapidité et la simplicité de C/C++ ainsi que ses avantages d'interface graphique pour développer des bonnes applications.

### 3. Description de l'interface du logiciel

Le logiciel réalisé est représenté par la figure (17). Cette interface est composée de :

- 1- Barre des Menus.
- 2- Zone des classes.
- 3- Espace de travail.
- 4- La console (affichage des stdout ou des erreurs).
- 5- Boutons standards (compilation, exécution,...etc.).
- 6- Alerte en cas d'erreur avant même de lancer son programme.
- 7- Anglets (Tabs) de fichier ouvert.

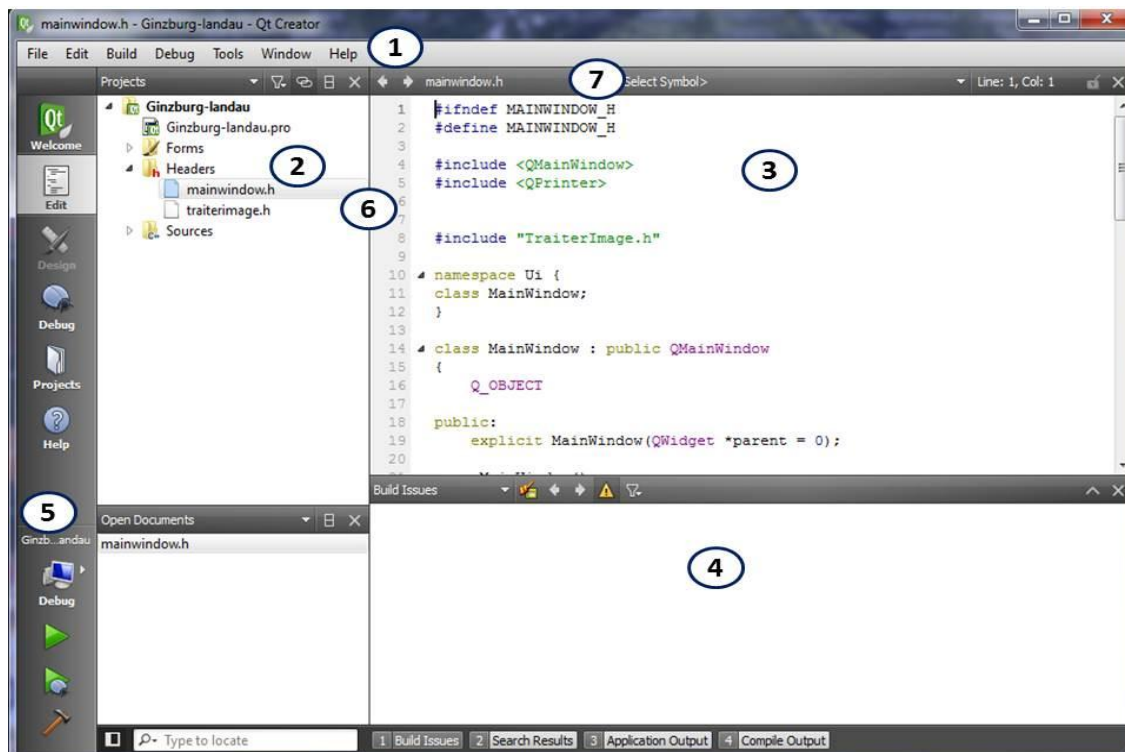


Figure 17. Présentations de quelques zones de l’outil de développement

#### 4. définition des classes d’application :

Rappelons tout d’abord que ce travail est focalisé principalement sur la segmentation des images en niveaux de gris. La méthode utilisée est une méthode non locale, elle est basée sur un modèle variationnel.

En ce qui concerne la répartition des classes de notre application on l’a réparti en 5 classes :

- ✓ « Classe mainwindow.h » et « Classe traiterimage.h » contient tous les méthodes qui seront utilisés dans le dossier sources.
- ✓ « Classe main.cpp » responsable de l’exécution du programme.
- ✓ « Classe mainwindow.cpp » contient tous les méthodes implémentés qui sont en interaction avec l’utilisateur (interface graphique).
- ✓ « Classe traiterimage.cpp » contient tous les méthodes implémentées qui gère les différentes fonctions de l’application.

Ainsi une interface qui gère l’interface graphique.

## 5. Quelques interfaces de l'application

### 5.1 Menu Fichier

Ce menu comporte deux fenêtres (voir figure 18).

#### a. Ouvrir

Cette opération permet de charger l'image à partir du disque.

#### b. Enregistrer

Cette opération permet de sauvegarder l'image sous son nom actuel.

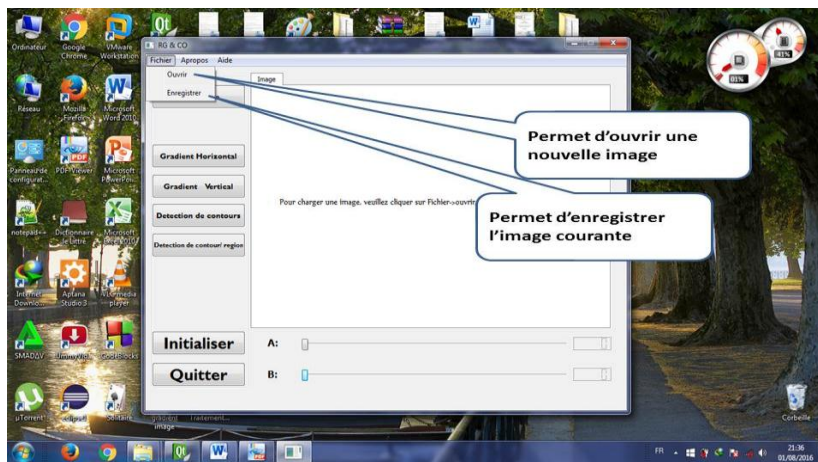


Figure 18. Interface qui montre le Menu Fichier

### 5.2 Bouton Luminance

Cette opération permet de convertir une image couleur en une image en niveaux de gris. La figure 19 montre le résultat obtenu de cette transformation.

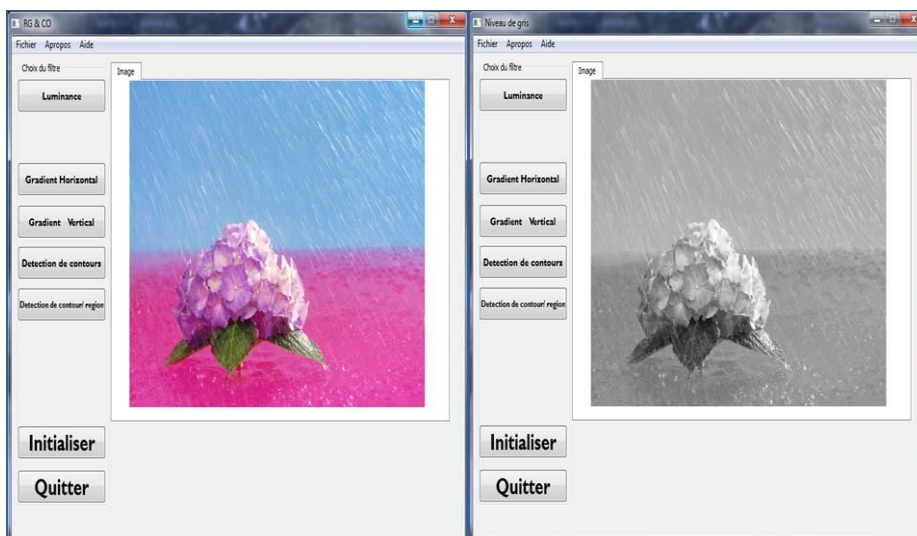


Figure 19. Interface qui montre l'utilité du bouton Luminance.

### 5.3 Bouton Détection des contours :

Cette option permet de détecter les contours d'une image par l'opérateur de Sobel (figure 20).

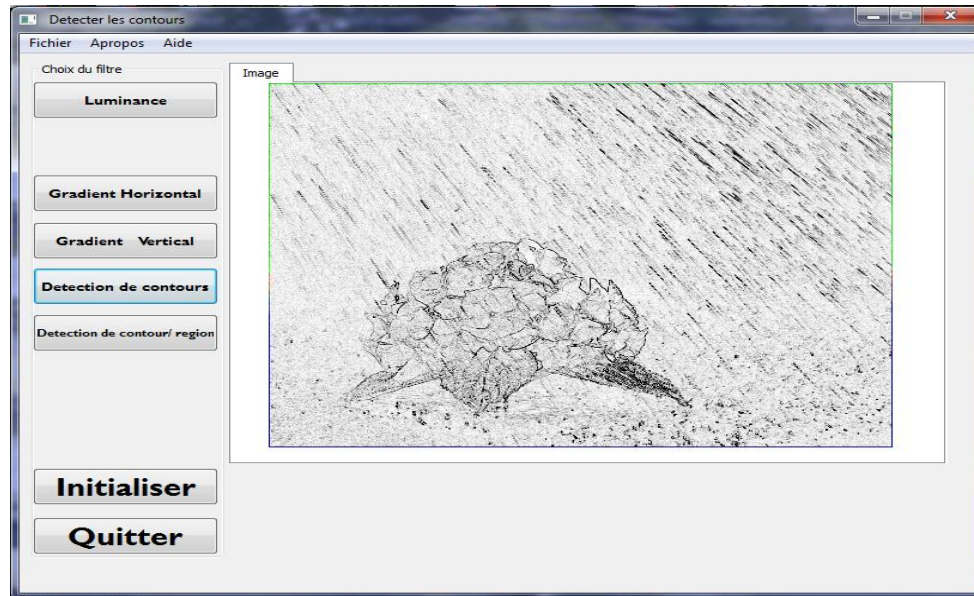


Figure 20. Résultat de la détection de contour.

### 5.4 Bouton Gradient

Permet de calculer les composantes horizontale et verticale du gradient d'une image.

### 5.5 Bouton Initialiser

Permet d'afficher l'image initiale.

### 5.6 Bouton Quitter

Permet de fermer l'application.

## 6. Tests et résultats :

Afin d'étudier les performances de notre méthode, nous avons effectué plusieurs tests en faisant varier les valeurs réelles des paramètres  $\Delta t$ ,  $\lambda$ ,  $\varepsilon$ ,  $\sigma$ , et un nombre d'iteration. En effet, plusieurs résultats sont obtenus mais nous avons retenu que ceux qui sont jugés satisfaisants. Pour montrer ces tests, nous avons choisi une image réelle figure (21). Cette image contient 3 régions (la rose, les pétales et le fond). Notre objectif ici, c'est de détecter la région représentée par la couleur claire (rose).

Avant d'appliquer notre méthode, on doit convertir cette image couleur en une image en niveaux de gris (figure 22) qu'on transformera par la suite en une image intensité en utilisant la formule :  $f(x) = (-1)^k((2 * f(x)/255) - 1)$ ,  $k = 0$  ou  $k = 1$ .

Si l'objet à détecter est plus claire que le fond,  $k = 1$ . Autrement,  $k = 0$ .

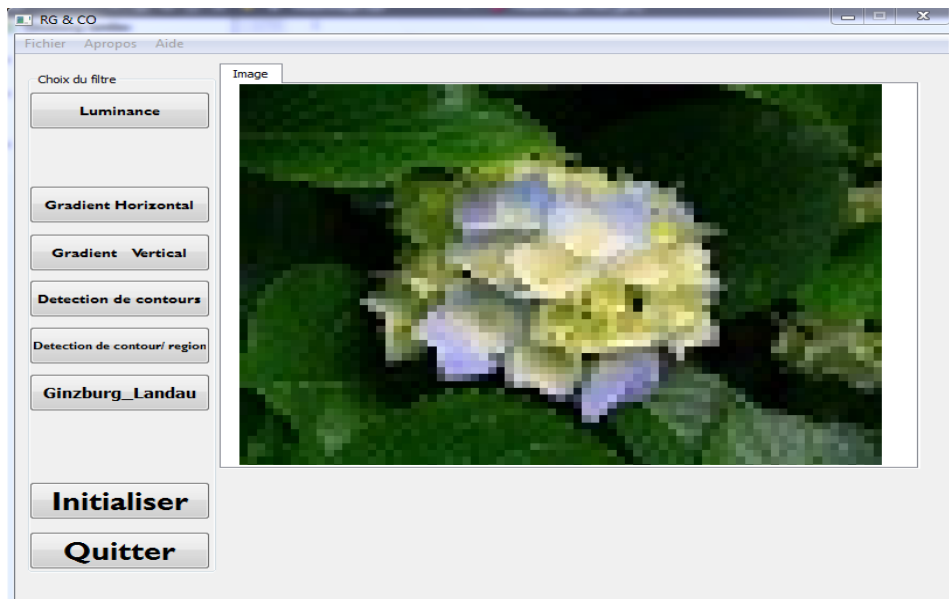


Figure 21. Image originale.

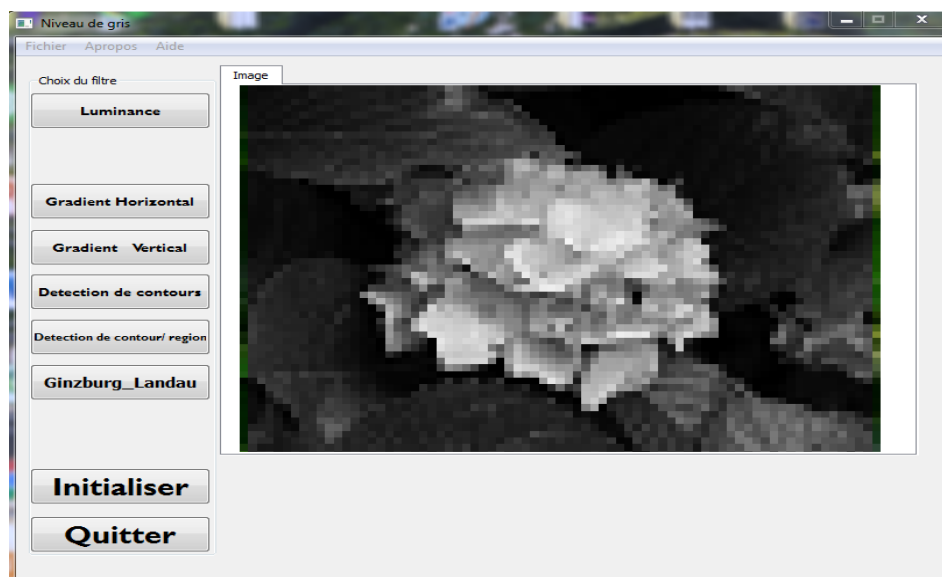


Figure 22. Conversion de l'image couleur en une image en niveaux de gris.

### 6.1 Influence des différents paramètres sur les résultats

- Les résultats présentés par les figures (23) et (24) correspondent aux différentes valeurs choisies pour les différents paramètres mais seulement la valeur du paramètre  $\sigma$  a varié d'un test à un autre en la fixant à 1 ensuite 2. Dans ces deux cas, on remarque que l'objet a été détecté et les deux autres régions sont presque fusionnées en une seule région.

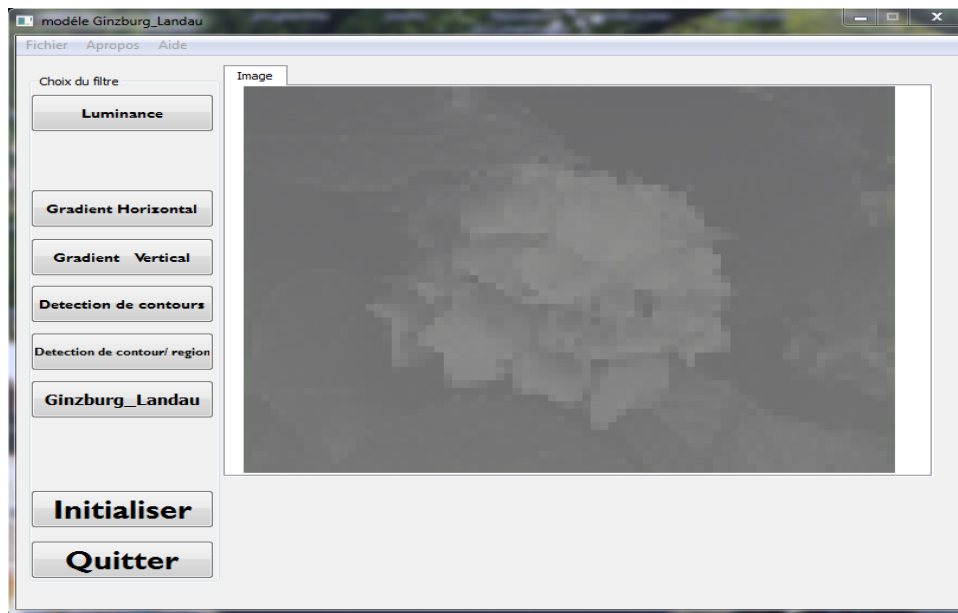


Figure 23. Résultat avec  $\sigma = 2$ ,  $\lambda = 0.4$ ,  $\varepsilon = 0.07$ ,  $p = 3$ ,  $\Delta t = 0.1$ ,  $iteration = 1$ .

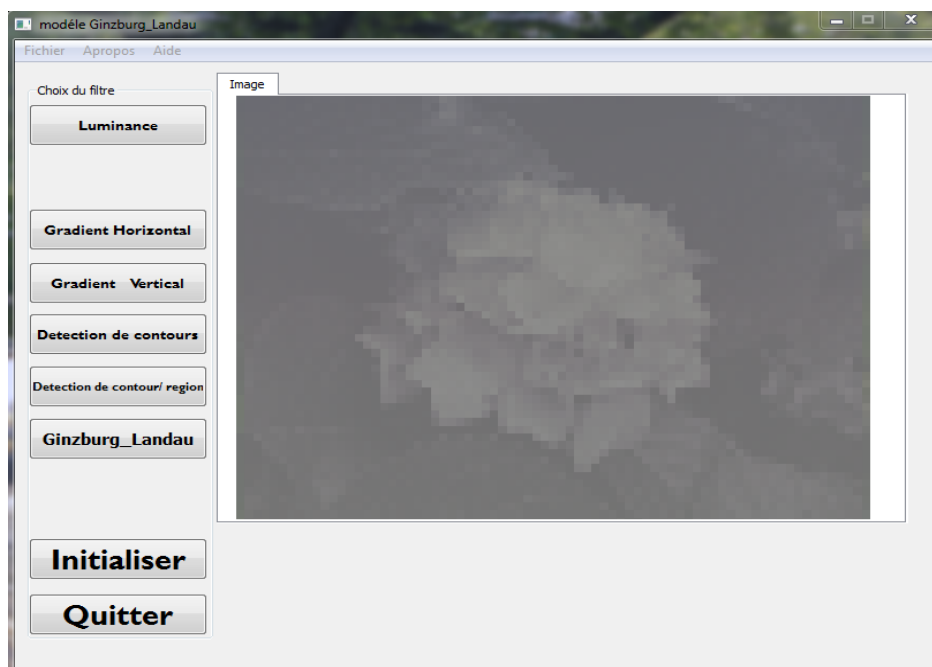


Figure 24. Résultat avec  $\sigma = 1$ ,  $\lambda = 0.4$ ,  $\varepsilon = 0.07$ ,  $p = 3$ ,  $\Delta t = 0.1$ ,  $iteration = 1$ .

- La figure (25) présente le résultat obtenu en variant uniquement le paramètre  $\Delta t$  en passant de 0.1 à 0.3. En augmentant sa valeur, on constate que la région représentée par les pétales a réapparu un peu par rapport au résultat trouvé précédemment. Toutefois, les contours sont un peu plus accentués dans ce cas.

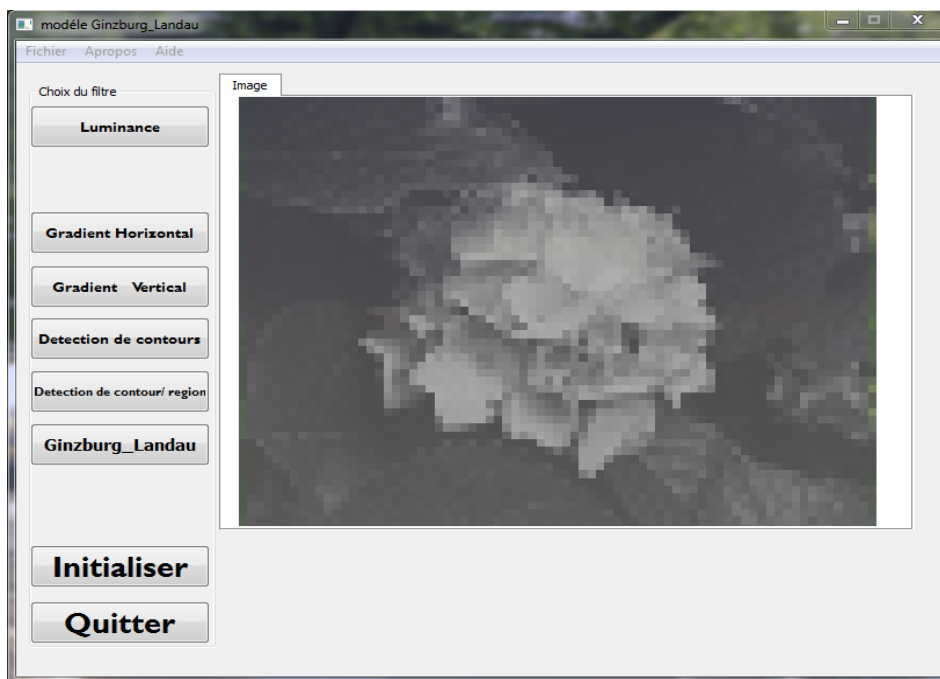


Figure 25. Résultat avec  $\sigma = 1.5$ ,  $\lambda = 0.4$ ,  $\varepsilon = 0.07$ ,  $p = 3$ ,  $\Delta t = 0.3$ ,  $iteration = 1$ .

- Les figures (26) et (27) montrent l'influence du nombre d'itérations sur les résultats obtenus. On remarque bien que plus le nombre d'itérations augmente, plus certaines régions s'effacent ou disparaissent. Ainsi, on obtient une région très lissée dépourvue de ses contours.

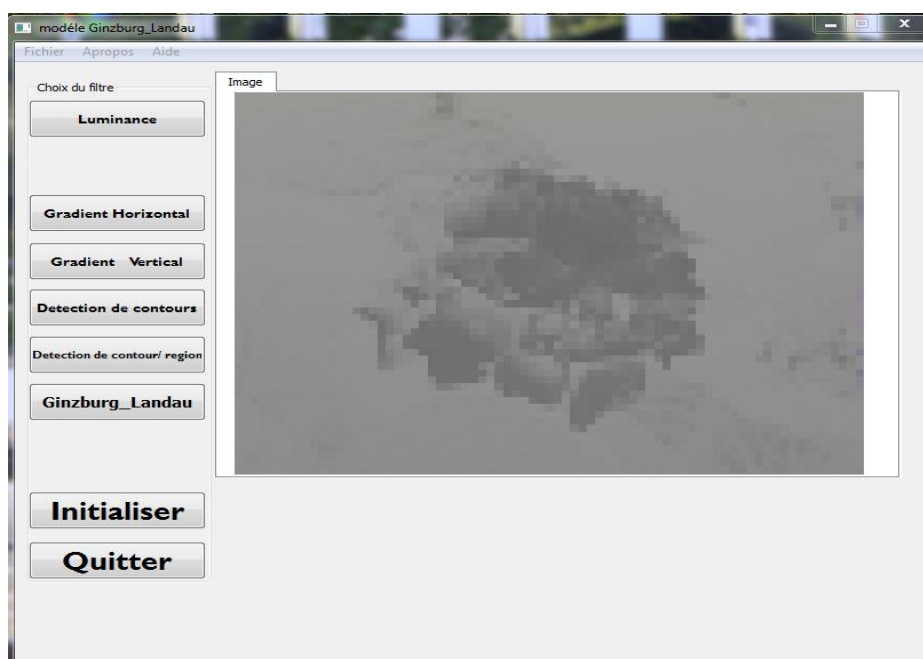


Figure 26. Résultat avec  $\sigma = 2$ ,  $\lambda = 0.4$ ,  $\varepsilon = 0.07$ ,  $p = 3$ ,  $\Delta t = 0.1$ ,  $iteration = 2$ .

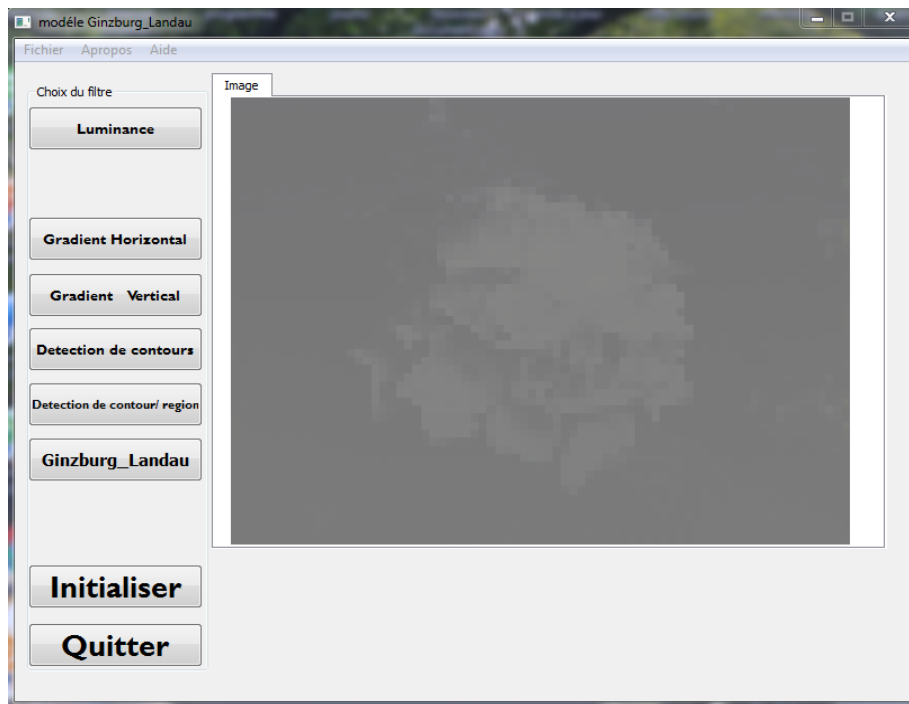


Figure 27. Résultat avec  $\sigma = 2$ ,  $\lambda = 0.4$ ,  $\varepsilon = 0.07$ ,  $p = 3$ ,  $\Delta t = 0.3$ ,  $iteration = 3$ .

## Conclusion

Ce dernier chapitre a été consacré à la présentation de différentes étapes de la réalisation de notre application.

Nous avons présenté les outils logiciels qui nous ont permis de réaliser cette application, à savoir l'environnement de développement, le langage de programmation C++. Pour illustrer les principales fonctionnalités de notre application, nous avons donné quelques exemples d'interfaces.

Nous avons présenté également une méthode de segmentation binaire dans le cas d'une image réelle. Différentes expérimentations ont été réalisées et les résultats obtenus sont satisfaisants montrant très clairement le bénéfice de notre méthode pour la détection d'objet(s) contenus dans l'image.

Un des apports de notre approche est que notre modèle est basé sur des graphes. Cet avantage permet de traiter n'importe quel type de données même celles définies sur des domaines irréguliers ou de grande dimension.

# *Conclusion générale*

## Conclusion générale

---

L'évaluation des algorithmes de traitement d'images et notamment ceux de segmentation constitue un problème d'importance, tant pour le choix d'un algorithme et de son paramétrage pour un utilisateur que pour la comparaison avec l'existant d'un nouvel algorithme par un chercheur. Le nombre important de critères quantitatifs témoigne d'un besoin de toute la communauté en traitement d'images.

Dans ce travail présenté concerne le domaine du traitement d'images et plus précisément celui de la segmentation d'images en niveaux de gris basée sur un modèle variationnelle.

Nous avons étudié dans ce mémoire un nombre exhaustif de ces variantes, qui ont été proposées dans la littérature.

Notre travail peut se résumer en deux parties :

- Dans la première partie, nous avons fait une étude théorique sur les techniques de traitement d'image et les bases mathématiques de la théorie des graphes.
- Dans la deuxième partie, nous avons concrétisé notre méthode conceptuelle en réalisant une application de traitement d'image qui répond aux objectifs fixés.

La réalisation de ce travail nous a permis d'élargir aussi bien nos connaissances théoriques que pratiques en rapport avec le travail collaboratif, et d'acquérir de nouvelles connaissances sur le langage C++ ainsi que l'utilisation de logiciel Qt Creator.

Après le passage par les différentes étapes de traitement et de développement, l'application a abouti à un logiciel fonctionnel qui répond aux critères imposés dans ce modèle.

# *Bibliographie*

## Bibliographie

---

- Baillie.J-C segmentation module D9 : Traitement d'Image et Vision Artificielle, ENSTA 2003.
- Bénallal.M, Système de calibration de caméra localisation de forme polyédrique par vision monoculaire, 2002.
- Bensrhair.A contribution à la réalisation d'un capteur de vision 3D par stéréovisions passive mise au point d'algorithmes rapide et auto-adaptatifs de mise en correspondance d'images stéréoscopique pp (683-686).
- Bougleux.S, Elmoataz. A, Melkemi.M, Discrete regularization on weighted graphs for image and mesh filtering. In Proceedings of International Conference on Scale Space and Variational Methods in Computer Vision (Ssvm), LNCS 4485, Springer, pp (128–139), 2007.
- Chakib.K, "La Compression des Images Fixes par les Approximations Fractales Basée sur la Triangulation de Delaunay et la quantification Vectorielle", mémoire de fin d'étude, 1999.
- Chan.T.F, Osher.S. et Shen.J, The digital tv filter and nonlinear denoising. IEEE Transactions on Image Processing, Vol 10(n°2): pp (231–241), 2001.
- Chung.F. R. K. Spectral Graph Theory. CBMS Regional Conference Series in Mathematics, 1997.
- Cocquerez.J.P, Philipp.S, Analyse d'Images et Segmentation. Masson, 1995.
- Duchesnay.E, Agents situés dans l'image et organisés en pyramide irrégulière : contribution à la segmentation par une approche d'agrégation coopérative et adaptative. PhD thesis, Université Rennes-1, 2001.
- Elmoataz.A, Lézoray.O, Bougleux.S, Non-local discrete regularization on weighted graphs: A framework for image and manifold processing. IEEE Transactions on Image Processing, Vol 17(n°7), pp (1047–1060), 2008.
- Gilboa.G, Osher.S, Nonlocal linear image regularization and supervised segmentation. SIAM Multiscale Modeling and Simulation, Vol 6(n°2) : pp(595–630),2007.
- Gonzales.C.Rafael ,Woods.E.Richard, Digital Image Processing, Pearson Prentice Hall,2008.
- Hein.M, Von Luxburg.U, et Maier.M, Influence of graph construction on graph-based clustering measures. In Advances in Neural Information Processing Systems,pp(1025–1032), 2008.

## Bibliographie

---

- Houhou.J, Lartiges.BS, Hofmann.A, Frappier.G, Ghanbaja.J, Temgoua.A, Phosphate dynamics in an urban sewer: A case of Nancy, France. Water research , Vol 43:pp (1088-1100) ,2009.
- [https://fr.wikipedia.org/wiki/Qt\\_Creator](https://fr.wikipedia.org/wiki/Qt_Creator). (visiter le 02/04/2016).
- Mumford.D et Shah.Boundary.J detection by minimizing functionals, Image understanding, Ed.S. Ullman and W.Richards, 1988.
- Nicolas.L, Extraction de Connaissances à partir d'images, Extraction des Connaissances : Etat et Perspectives, RNTI-E-5, Ed.Cépaduès, novembre 2005.
- Osher.S, Shen.J, Digitized PDE method for data restoration. Analytical-Computational Methods in Applied Mathematics, pp (751–771), 2000.
- Szlam.A.D, Maggioni.M et Coifman.R.D, Regularization on graphs with function-adapted diffusion processes. Journal of Machine Learning Research, Vol 9 : pp (1711–1739), 2008.
- Tikhonov.A.N, et Arsenin.V.Y, Solutions of Ill-Posed Problems. Wiley, 1977.
- Tschumperlé .D, Lissage non local d'images multi-values par diffusion anisotrope dans l'espace des patches. In Groupe de recherche sur le traitement du signal et des Images (Gretsi), 2009.