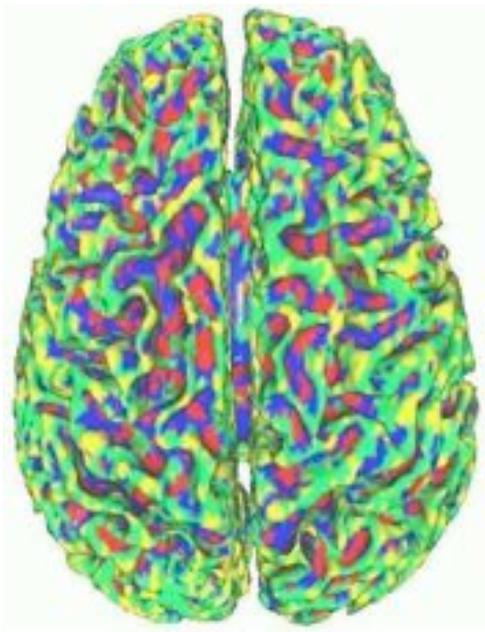


Rapport de Stage



Analyse multi résolution sur graphes:

Application à la Compression et l'analyse de données de grandes dimensions.

Lieu du stage : ENSICAEN, GREYC UMR 6072, Équipe Image

Durée du stage : du 4 mars au 4 septembre

Responsables : O. Lezoray, A. Elmoataz

Je tiens avant tout à remercier mon encadreur O.Lezorey, qui a été très sympathique et très patient avec moi. Je le remercie également de m'avoir accordé ce stage qui m'a permis de découvrir des travaux récemment élaboré par A.Elmoataz et O.Lezorey.

Je remercie particulièrement A.Elmotaz pour ces explications et ces idées, qui m'ont aidé à comprendre et à simplifier l'idée globale et les principes de mon sujet.

Je remercie aussi tous mes collègues, les stagiaires, et tous les membres de l'équipe Image(Greyc) pour leur bonne humeur et leur accueil.

Et je remercie mes parents ainsi que toute ma famille qui m'a soutenue dans toutes mes démarches.

I. Introduction.....	3
II. Traitement d'image par modèle discret sur les graphes.....	4
II.1. Préliminaires sur les images	4
II.2. Les cartes de régions	5
II.3. Quelques notions sur les graphes.....	6
II.4. Représentations par graphes.....	7
II.5. Graphe Pondéré	9
II.6. L'analyse multi-résolution	9
II.6.1. Algorithmes d'analyse et de synthèse de Mallat.....	11
II.6.2. Pyramide Irrégulière.....	12
III. Régularisation discrète sur graphes pondérés.....	13
III.1 Introduction et problématique.....	13
III.2 Géométrie différentielle sur graphes.....	14
III.2.1 Opérateurs gradient et divergence.....	14
III.2.2 Opérateur p-Laplacien.....	15
III.3 Principe de la Segmentation sur graphe.....	16
III.4 Régularisation du graphe.....	16
III.4.1. Principe général.....	16
III.4.1.1 Régularisation pour $p = 2$	17
III.4.1.2 Régularisation pour $p = 1$	18
III.4.2 Représentation associée aux nœuds du graphe.....	19
III.4.3 Poids des arêtes du graphe.....	19
IV. Application	20
IV.1 Application sur L'image.....	20
IV.1.1 Estimation des poids	20
IV.1.2 Choix des paramètres Optimaux	20
IV.1.3 Choix de la fonction de poids	21
IV.1.4 Filtrage	22
IV.1.5 Fusion	23
IV.1.6 Critère d'arrêt.....	25
IV.1.7 Le Schéma Diffusion-Fusion.....	25
IV.2 Application sur Donnée quelconque	26
IV.3 Illustration.....	29
V. Conclusion	41
<i>Référence</i>	42

I. Introduction

La représentation par graphe est une représentation structurelle puissante qui a prouvé sa flexibilité en traitement d'une grande variété de types d'images (les anciens documents, les plans électriques et architecturaux, les images naturelles, les images médicaux ...). Les travaux, qui ont fait recours aux graphes, ont visé une représentation qui préserve l'information topographique de l'image ainsi que les relations entre les composantes. Actuellement, plusieurs approches dérivées des graphes pour l'analyse d'images ont été proposées notamment pour les systèmes de recherche par contenu, la segmentation, compression et l'indexation etc...

Mais à l'ère du numérique, de grandes masses de données sont apparues qui nécessitent une approche simplificatrice pour entamer une segmentation, restauration ou même une compression, d'où le principe d'analyser ou de réduire l'image à plusieurs niveaux de résolution. Ces données sont cependant irrégulières et sont représentées par des graphes dont la topologie est totalement arbitraire ; ce qui rend très difficile l'utilisation des théories basées sur des espaces réguliers.

Mon stage consiste à proposer une méthode d'analyse multi résolution sur graphes de topologies arbitraires. Ce graphe pourra aussi par la suite représenter des données quelconques (graphe d'images, graphe d'objets, etc.).

Je dois élaborer et travailler sur un schéma d'analyse multi résolution exploitant une diffusion basée sur le p -Laplacien sur graphes couplée à une décimation. Les applications visées concernent la conception de représentations résumées d'images ou de bases de Données (d'images ou d'objets).

II. Traitement d'image par modèle discret sur les graphes

II. 1. Préliminaires sur les images :

Les images informatiques sont des signaux numériques dépendant de deux variables (les deux dimensions du plan). Une image peut être interprétée comme la variation dans un espace à deux dimensions d'une information lumineuse. La figure.1 donne deux représentations possibles de la même image, la représentation à deux dimensions (2D) classique et la représentation spatiale en trois dimensions : la largeur, la hauteur et la variation de la luminosité (ou niveau de gris).

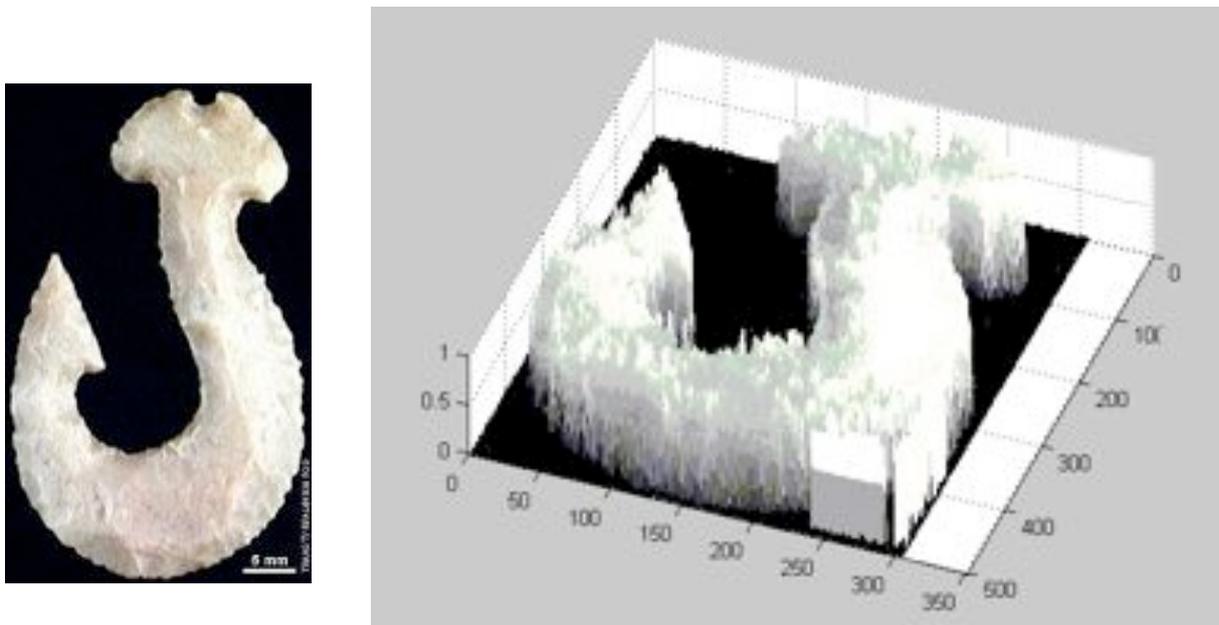


Figure.1 Une image et sa représentation spatiale en termes de variation de luminosité.

Pour coder les images couleur, on utilise en général trois plans en niveaux de gris : rouge, Vert et bleu dans lesquels les couleurs sont des quantités numériques. Ainsi, il faut trois fois plus d'informations pour coder une image en couleurs qu'une image en niveaux de gris. La figure.1 donne la représentation numérique d'une image en niveaux de gris et d'une image couleur. L'image en niveaux de gris n'utilise qu'un plan pour stocker l'information de luminosité. L'image couleur, quant à elle, utilise trois plans, un rouge (R), un vert (V) et un bleu (B) pour stocker l'information couleur de l'image. Pour des contraintes informatiques de stockage mémoire, on code souvent un niveau de gris sur un octet (soit huit bits), ce qui équivaut donc à une dynamique de $2^8 = 256$ niveaux de gris possibles. Pour une image couleur, on code souvent chaque plan couleur avec un octet, ce qui équivaut $(2^8)^3 = 16\,777\,216$ couleurs possibles.

11.2. Les cartes de régions :

Afin de représenter l'image en graphe, une représentation en carte de région s'impose (phase intermédiaire). Une méthode classique consiste en la recherche des différentes régions, qui composent l'image. Dans la plupart des cas, ces régions correspondent aux objets présents dans la scène.

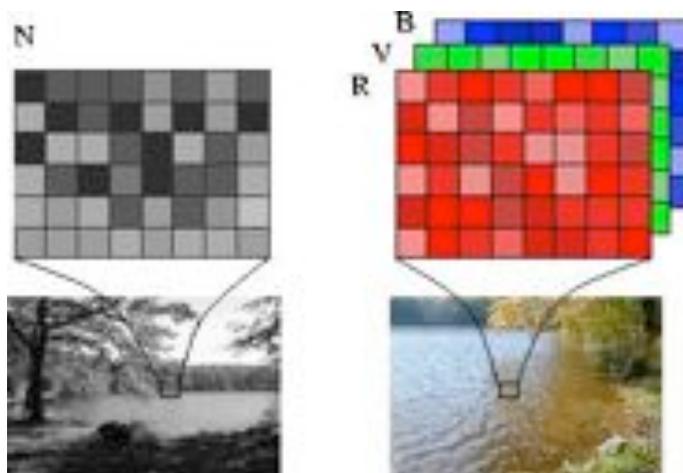


Figure 2. La vision informatique sur l'image.

La recherche des régions qui composent une image est une segmentation. [BCC+95].

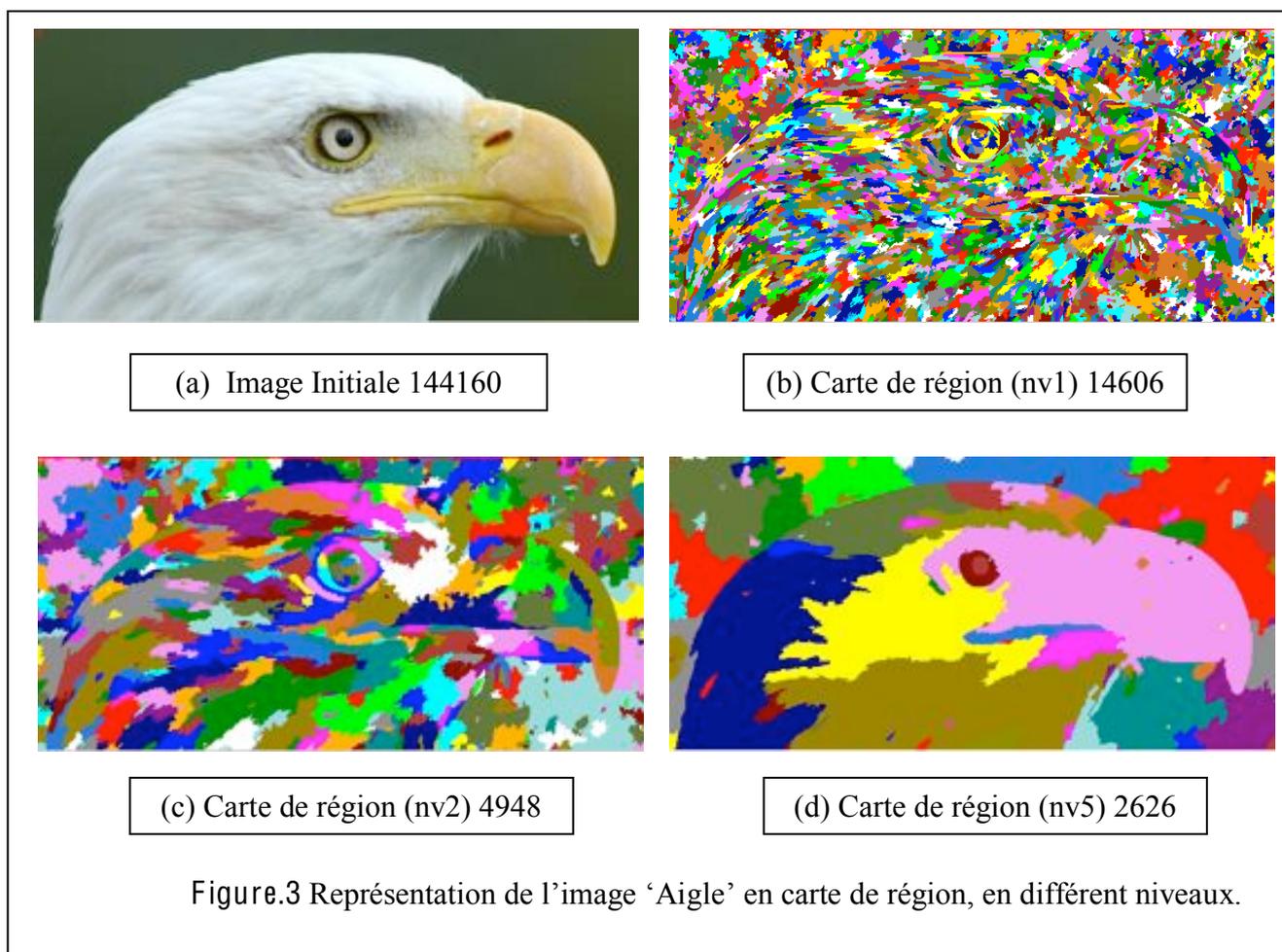


Figure.3 Représentation de l'image 'Aigle' en carte de région, en différent niveaux.

La représentation en régions d'une image permet de décomposer l'image en zones homogènes, afin de calculer les attributs sur des régions, et non sur l'image tout entière.

Cette représentation utilise des méthodes structurelles telles que la croissance de régions ou les algorithmes de type ligne de partage des eaux (*watershed*). La figure.3 donne un exemple de segmentation d'une image, au niveau (1,2 et 5).

11.3. Quelques notions sur les graphes

Nous rappelons ici des notions de base de la théorie des graphes (voir dans [DIESTE05] pour plus de détails). Un graphe \mathcal{G} est un couple $\mathcal{G} = (V, E)$ où V est un ensemble fini de nœuds (de taille $|V|$) et E un ensemble d'arêtes (de taille $|E|$) : $E \in V \times V$. Deux nœuds u et v dans un graphe sont adjacents si l'arête (u, v) appartient à E , les deux nœuds sont alors appelés des nœuds voisins.

Le degré (simple) $\delta(v)$ d'un nœud v est le nombre d'arêtes incidentes en ce nœud. $\delta : V \rightarrow N^+$ est défini par $\delta(v) = |u \sim v|$ où $|\cdot|$ désigne le cardinal d'un ensemble. La relation $u \sim v$ désigne l'ensemble des nœuds u adjacents au nœud v via les arêtes $(u, v) \in E$: u est un voisin de v . Si un nœud a un degré simple de un, il est appelé une feuille.

Un chemin C est un ensemble de nœuds $C = (v_1, v_2, \dots, v_k)$ tel qu'il existe une arête pour tout couple de nœuds successifs du chemin : $\forall i \in [1, k[$, l'arête $(v_i, v_{i+1}) \in E$. La longueur d'un chemin correspond à son nombre d'arêtes. Un chemin est simple s'il ne passe qu'une et une seule fois par toute arête. Un chemin est Hamiltonien s'il passe par tous les nœuds du graphe et ceci une et une seule fois (ce problème est NP-complet).

Un graphe complet est un graphe où chaque paire de nœuds est reliée par une arête. Un graphe complet contenant n nœuds a donc exactement $n(n - 1)/2$ arêtes et le degré simple de chaque nœud est $(n - 1)$.

Un graphe est connexe si, pour chaque paire de nœuds u et v , il existe un chemin tel que $v_1 = u$ et $v_k = v$. Un graphe est non-orienté quand l'ensemble des arêtes est symétrique, c'est à dire que pour chaque arête $(u, v) \in E$, nous avons également $(v, u) \in E$. Nous considérerons toujours des graphes simples où au maximum une arête relie deux nœuds, ces graphes seront également connexes et non orientés [DIESTE05].

Nous pouvons à présent définir l'espace des fonctions sur graphes. Soit $\mathcal{H}(V)$ l'espace de Hilbert des fonctions sur les nœuds, dans lequel chaque fonction $f : V \rightarrow \mathbb{R}^p$ associe un vecteur $f(v)$ à un nœud v . L'espace des fonctions $\mathcal{H}(V)$ peut être muni du produit scalaire usuel $\langle f, g \rangle_{\mathcal{H}(V)} = \sum_{u \in V} f(u)g(u)$ où f et g sont deux fonctions de $\mathcal{H}(V)$. La norme d'une fonction f obtenue à partir du produit scalaire est $\|f\| = \sqrt{\langle f, f \rangle}$. De la même manière, nous pouvons définir $\mathcal{H}(E)$ comme l'espace de Hilbert des fonctions sur les arêtes, dans lequel chaque fonction $g : E \rightarrow \mathbb{R}^+$ associe une valeur réelle à une arête e . Cet espace de fonctions peut être muni du produit scalaire usuel $\langle h, l \rangle_{\mathcal{H}(E)} = \sum_{u,v \in E} h(u, v)l(u, v)$ où h et l sont deux fonctions de $\mathcal{H}(E)$.

Un graphe sera dit pondéré si une fonction de poids $\omega \in \mathcal{H}(E)$ lui est associée. Cette fonction ω satisfait en outre $\omega(u, v) > 0$ si $(u, v) \in E$, $\omega(u, v) = 0$ si $(u, v) \notin E$ et $\omega(u, v) = \omega(v, u)$ pour toutes les arêtes de E puisque nous ne considérerons que des graphes non orientés. Un graphe peut être pondéré par une distance ou une similarité entre nœuds. Si l'on considère la 2.1. Graphes et traitement d'images 5 matrice d'adjacence associé à un graphe, on notera W cette matrice si les poids sont des distances et S si les poids sont des similarités. Une fonction de distance classique est la norme L_2 :

$$w_{uv} = w(u, v) = \|f(u) - f(v)\|_2. \text{ Deux exemples classiques de fonctions de similarité sont :}$$

$$w(u, v) = \exp\left(-\frac{\|f(v)-f(u)\|_2}{2\sigma^2}\right) \text{ et } (u, v) = \frac{1}{\|f(v)-f(u)\|_2}.$$

Le degré d'un nœud $v \in V$ peut être défini par $d_i = d(v_i) = \sum_{u \sim v} w(u, v_i)$. La matrice des degrés D est définie comme la matrice diagonale avec les degrés d_i des nœuds $v_i \in V$ sur la diagonale. A partir d'un sous ensemble $A \in V$, nous notons \bar{A} son complément. La taille de cet ensemble peut être mesurée en considérant soit $|A|$, son nombre de nœuds, ou bien $vol(A) = \sum_{v_i} d(v_i)$, en terme de poids de ses arêtes.

Un arbre est un graphe simple connexe acyclique. Un arbre de recouvrement d'un graphe connexe non orienté \mathcal{G} est un arbre composé de tous les nœuds du graphe et de quelques arêtes de \mathcal{G} . Un arbre de recouvrement minimum (un Minimum Spanning Tree (MST)) est un arbre de recouvrement dont la somme des poids des arêtes est inférieure à celle de n'importe quel autre arbre de recouvrement. Ainsi, un arbre de recouvrement minimum $T(\mathcal{G})$ d'un graphe \mathcal{G} est un graphe connexe pondéré $T(\mathcal{G}) = (V', E')$ où la somme des poids des arêtes $\sum_{(u,v) \in E'} w(u, v)$ est minimale. Pour un graphe \mathcal{G} de n noeuds, son arbre de recouvrement minimum (ou MST) $T(\mathcal{G})$ a exactement $(n - 1)$ arêtes. Le MST peut être calculé efficacement en $O(|E| \log |V|)$ par l'algorithme de Prim avec des structures de données appropriées

II.4. Représentations par graphes

Les images sont généralement segmentées en régions. Chaque sommet du graphe, représente alors une région de l'image, et les arcs représentent les différentes relations possibles entre les régions (connectivité, distance...). Sachant que, les graphes sont tout naturellement adaptés à la représentation des données quelconques. L'intérêt principal, de l'utilisation des graphes pour la représentation d'images est l'intégration d'informations spatiales dans la modélisation. En effet les représentations classiques, telles que les histogrammes de couleurs, les descripteurs de textures, ne donnent aucune information sur la façon, dont les régions d'intérêt de l'image sont agencées. Au contraire, la représentation par les graphes, permet de décrire la structure de l'image, comme la façon dont les régions, sont disposées les unes par rapport aux autres. En outre, selon les types de relations choisies, cette représentation en graphe, peut être invariante à certaines transformations telles que, les rotations de l'image, ou les translations de certaines parties de l'image.

En traitement d'image, on utilise habituellement le support de l'image, comme une grille où les relations de connexité spatiale entre les pixels déterminent le voisinage considéré. Par conséquent, sur une grille 2D, le graphe de voisinage associé à une image, correspond par exemple, à l'utilisation d'une 4 ou d'une 8 connexité au niveau pixel (Figures.5(a) et 5(b)). Une grille est une tessellation régulière dont les éléments sont des carrés, mais il existe d'autres tessellations constituées de polygones dont la plus célèbre est la tessellation de Voronoï (Figure.5(c)). A cette tessellation est associée également un graphe de voisinage nommé le graphe de Delaunay (Figure.5(d)).

Plus spécifiquement, lorsque que l'on segmente une image en régions, on peut associer à la carte de segmentation un graphe de voisinage, nommé le graphe d'adjacence de régions, où chaque noeud représente une région et, chaque arête représente une relation de voisinage entre ces régions (Figure.4).

De manière similaire, à un ensemble quelconque de données, nous pouvons associer un graphe de proximité, afin de modéliser les relations spatiales de voisinage entre les données. On citera par exemple [VONLUX06] le graphe complet, le graphe des k plus proches voisins, le graphe de $"$ -voisinage, l'arbre de recouvrement minimum. La figure.6 présente quelques-uns de ces graphes de similarité sur une base de données d'images.

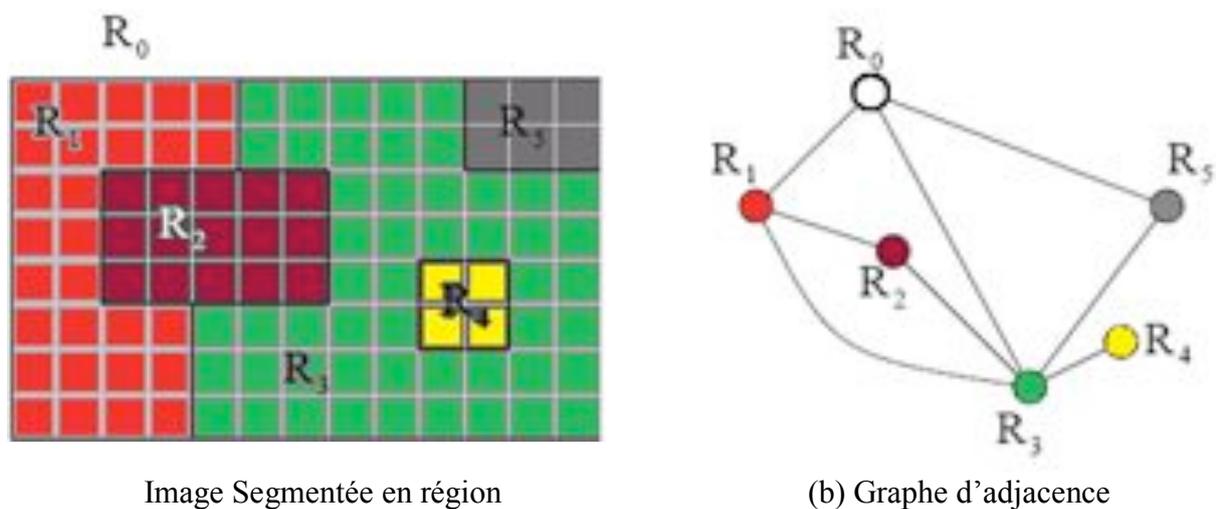


Figure.4 Exemple de représentation d'un graphe d'adjacence

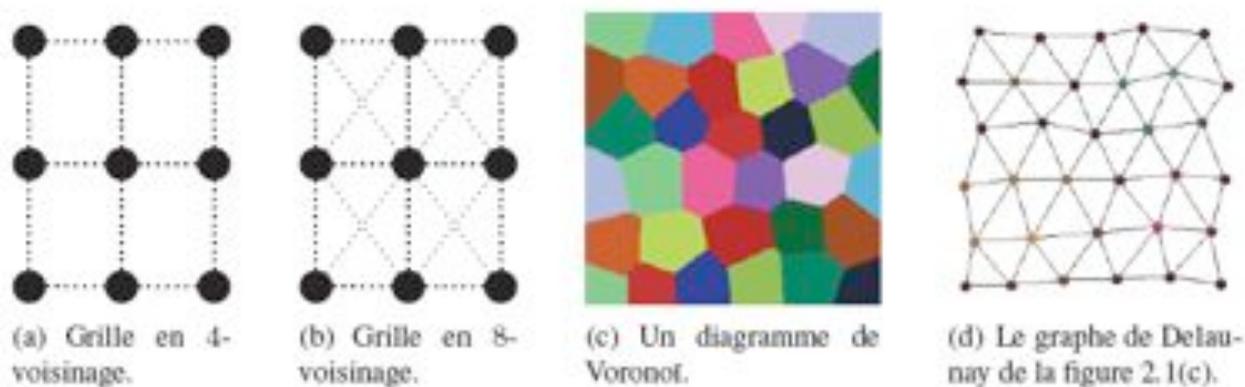


Figure.5 Exemple de graphe ou de type grille ou de Delaunay

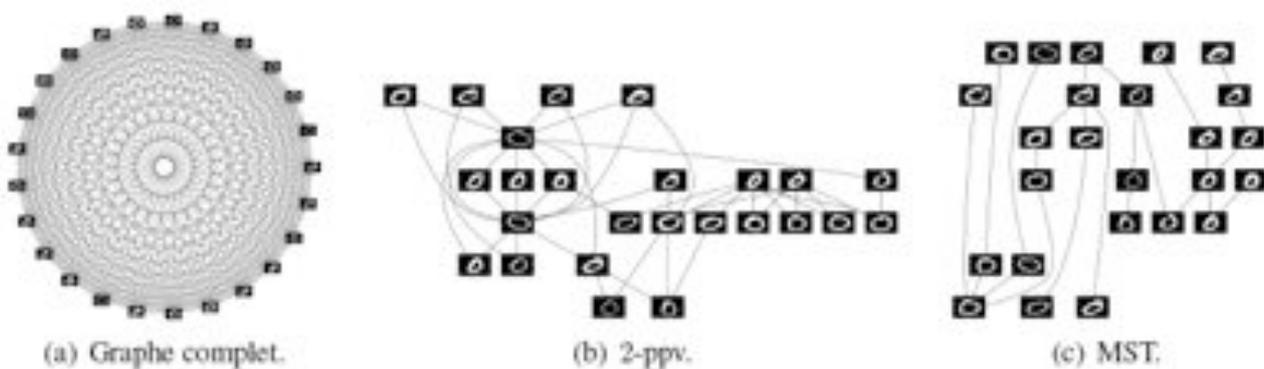


Figure.6 Exemple de graphe de proximité utilisé généralement pour représenté des bases de données quelconque (image ou Objet).

11.5. Graphe Pondéré

Ces graphes décrits précédemment, peuvent être pondérés, si une fonction de pondération w est associée à chaque arête,

$$w : E \rightarrow \mathbb{R}^+ \text{ satisfaisant } \omega(u, v) > 0 \text{ Si } (u, v) \in E, \omega(u, v) = 0 \text{ Si } (u, v) \notin E$$

et $w(u, v) = w(v, u)$. Une fonction de pondération mesure donc la similarité entre deux noeuds.

Le degré d'un nœud $\delta : V \rightarrow \mathbb{R}^+$ est défini comme $\delta(v) = \sum_{u \sim v} w(u, v)$ où $u \sim v$ désigne

l'ensemble des nœuds u adjacents au nœud v . Nous pouvons maintenant définir l'espace des fonctions sur graphes. Soit $\mathcal{H}(V)$ l'espace de Hilbert des fonctions à valeurs réelles sur les nœuds dans lequel chaque fonction $f : V \rightarrow \mathbb{R}^+$ attribue une valeur réelle $f(v)$ à chaque nœud v .

L'espace des fonctions $\mathcal{H}(V)$ peut être muni du produit scalaire usuel

$$\langle f, g \rangle_{\mathcal{H}(V)} = \sum_{v \in V} f(v) g(v) \text{ où } f \text{ et } g \text{ sont deux fonctions de } \mathcal{H}(V). \text{ Une fonction } f \text{ de } \mathcal{H}(V)$$

peut être vue comme un vecteur de $R|V|$. La norme d'une fonction f obtenue à partir du produit

scalaire est $\|f\| = \sqrt{\langle f, f \rangle}$. De même, nous pouvons définir $\mathcal{H}(E)$ l'espace des fonctions à valeurs

réelles sur les arêtes dans lequel chaque fonction $h : E \rightarrow \mathbb{R}^+$ attribue une valeur réelle à chaque

arête e . Cet espace de fonctions peut être muni du produit scalaire usuel $\langle h, l \rangle_{\mathcal{H}(E)} = \sum_{(u,v) \in E} h(u, v) l(u, v)$ où h et l sont deux fonctions de $\mathcal{H}(E)$. Nous verrons cette pondération sur graphe, en

détail un peu plus loin lors de sa régularisation.

11.6. L'analyse multi-résolution

L'analyse multi-résolution, introduite par Meyer et Mallat [Mal89], est un outil de traitement du signal, qui permet de décomposer un signal à plusieurs échelles (résolutions), et de le reconstruire à partir des éléments de cette décomposition.

Une analyse multi-résolution est un partitionnement de l'espace des fonctions d'énergie finie $L^2(\mathbb{R})$, par une famille de sous-espaces vectoriels V_j , emboîtés les uns dans les autres tels que le passage de l'un à l'autre, soit le résultat d'un changement d'échelle. Ces sous-espaces sont appelés des espaces d'approximation à l'échelle j ($j \in \mathbb{Z}$) et vérifient les propriétés suivantes :

Soit un ensemble de sous-espaces de $L^2(\mathbb{R})$ (l'ensemble des signaux à énergie finie) tels que :

$$\dots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \dots \subset V_{j+1} \subset V_j \dots$$

$$\bigcup_{j \in \mathbb{Z}} V_j = L^2(\mathbb{R})$$

$$\bigcap_{j \in \mathbb{Z}} V_j = 0$$

$$\forall j \in \mathbb{Z}, f(x) \in V_j \Leftrightarrow f(2^{-1}x) \in V_{j+1}$$

$$\forall k \in \mathbb{Z}, f(x) \in V_0 \Leftrightarrow f(x - k) \in V_0$$

Ces propriétés définissent une analyse multi-résolution dyadique sur $L^2(\mathbb{R})$

L'analyse multi-résolution a été définie par Mallat [Mal89]. L'idée est de projeter un signal $f(t) \in L^2(\mathbb{R})$ appartenant à un espace V_j sur un sous-espace V_j et un sous-espace W_{j+1} dans

le but de réduire la résolution de moitié. Le schéma est donné en figure.7. Il existe donc un opérateur de projection A_j et un opérateur de projection D_j qui projettent respectivement le

signal $f(t)$ sur V_{j+1} et W_{j+1} . V_{j+1} est le sous-espace d'approximation et W_{j+1} le sous-espace

de détails. On peut démontrer qu'il existe une fonction d'échelle $\phi(t) \in L^2(\mathbb{R})$ qui engendre par

dilatation et translation une base orthonormée de V_{j+1} et une fonction d'ondelettes $\psi(t) \in L^2(\mathbb{R})$

qui engendre par dilatation et translation une base orthonormée de W_{j+1} . Les espaces obtenus

ne sont pas quelconques, ils possèdent des propriétés intéressantes. Par construction, les espaces

d'approximation V_{j+1} et de détails W_{j+1} sont complémentaires : $V_j = V_{j+1} \oplus W_{j+1}$. De plus, si

les bases sont orthogonales, ils sont orthogonaux : $V_{j+1} \perp W_{j+1}$.
 Les fonctions de bases dilatées sont données par les relations :

$$\phi_{j,n}(t) = 2^{-j/2} \phi(2^{-j}t - n) \text{ avec } n \in \mathbb{Z} \text{ et } \psi_{j,n}(t) = 2^{-j/2} \psi(2^{-j}t - n) \text{ avec } n \in \mathbb{Z}$$

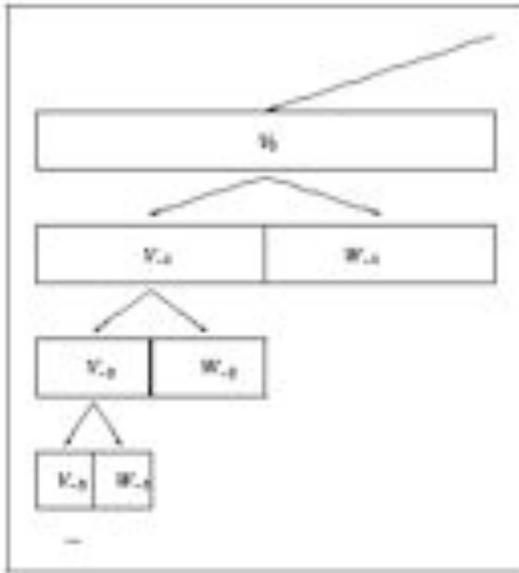
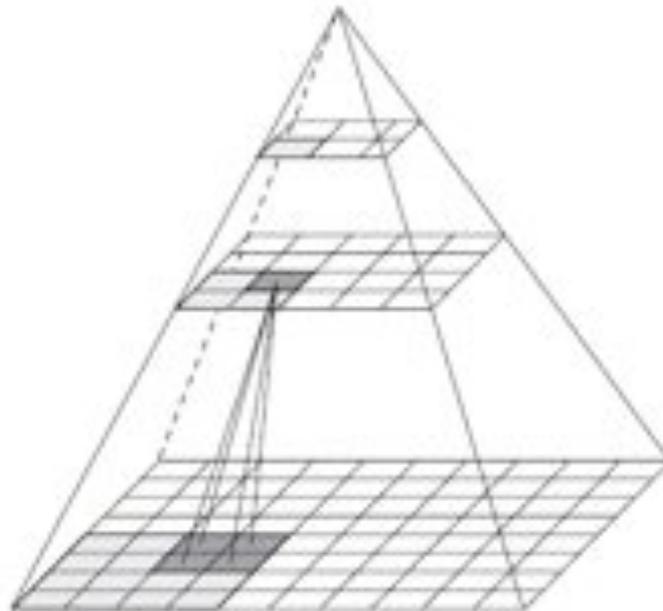


Figure.7 – Principe de l'analyse multi-résolution

On a donc $A_j f = \sum_n \langle f, \phi_{j,n} \rangle \phi_{j,n}$ et $D_j f = \sum_n \langle f, \psi_{j,n} \rangle \psi_{j,n}$ où $\langle f(t), g(t) \rangle$ désigne le produit scalaire de $f(t)$ par $g(t)$: $\langle f(t), g(t) \rangle = \int_{-\infty}^{+\infty} f(t)g(t)^* dt$

Puisque les signaux analysés sont réels, on a $g(t)^* = g(t)$. On pose $a_{j,n} = \langle f, \phi_{j,n} \rangle$ et $d_{j,n} = \langle f, \psi_{j,n} \rangle$. $a_{j,n}$ et $d_{j,n}$ sont respectivement les coefficients d'approximation et de détails de la transformée en ondelettes de la fonction f .

Cette analyse peut être représentée en pyramide introduit par (Hummel, 1987;Klinger, 1984; Peleg, Federbusch, & Hummel, 1987; Rosenfeld, 1984b; Samet,1984) comme suite :



Représentation pyramidale

L'image de base représente l'image à pleine résolution. Donc la grandeur d'image dans cette représentation diminue d'un facteur de 4 à mesure qu'on diminue la résolution.

11.6.1. Algorithmes d'analyse et de synthèse de Mallat

Stéphane Mallat a donné un algorithme d'analyse (ou décomposition) en ondelettes, qui permet d'obtenir une analyse multi-résolution du signal. Cet algorithme travaille par filtrage de l'image suivant les lignes puis les colonnes par deux filtres, \tilde{g} passe-haut et \tilde{h} passe-bas. \tilde{h} va permettre de repérer les basses fréquences dans l'image (l'approximation) et \tilde{g} les hautes fréquences (les détails). \tilde{h} et \tilde{g} sont construits à partir des fonctions ψ et ϕ . La figure.8 donne le schéma d'analyse (de décomposition) de Mallat.

La reconstruction des signaux analysés est effectuée à l'aide d'un banc de filtres h et g qui sont les filtres conjugués de \tilde{h} et \tilde{g} . En fonction de l'ondelette et du type de base (orthogonale ou bi-orthogonale) choisis pour l'analyse, les filtres d'analyse et de synthèse peuvent être de même taille, symétriques ou bien de taille différente, non symétriques. La figure.10 donne le schéma de synthèse (de reconstruction) de Mallat.

Dans la reconstruction, on travaille alternativement sur les colonnes puis sur les lignes lorsque les ondelettes sont séparables. Il existe un autre algorithme de transformation en ondelette comme « le lifting scheme ».

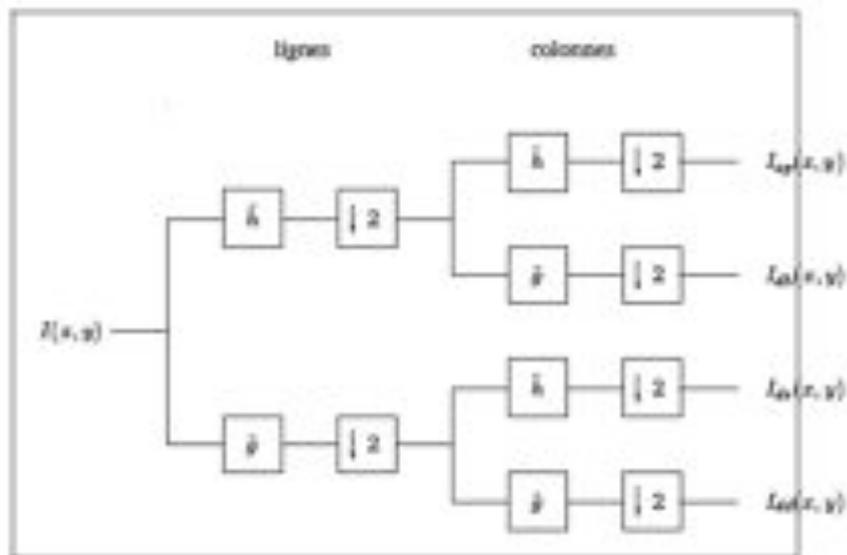


Figure.8 Algorithme d'analyse de Mallat.



Figure.9 Exemple de décomposition en ondelette d'une image au premier niveau de résolution

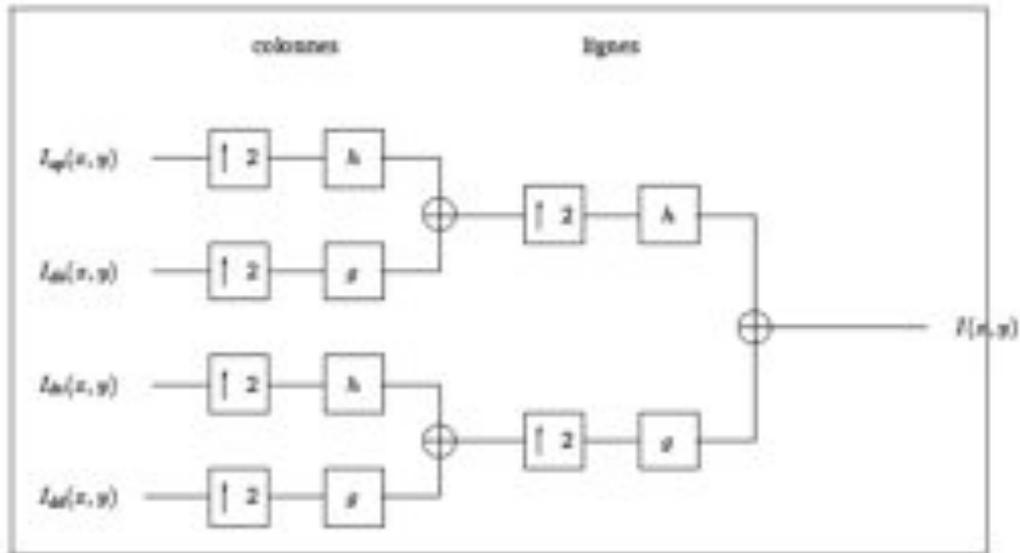


Figure.10 Algorithme de reconstruction de mallat

11.6.2. Pyramide Irrégulière

La pyramide Irrégulière est utilisée pour modéliser et segmenter ou simplifier ou même compresser une image ou des données quelconques représentées par un graphe. Dans le cas d'image, le nœud représente un pixel ou une région, dans le cas de donné quelconque ce nœud représente un objet ou un élément dans une base de donnée (image ou objet). Le niveau 0 (la base), est le graphe qui représente la totalité de l'image, ou dans le cas d'une base de donnée la base toute entière.

Une décimation appliquée sur le graphe de similarité permet de choisir localement le sommet le plus représentatif selon un critère donné. Dans le cas d'une image, ces sommets survivants composent le graphe de régions du niveau suivant. Les sommets non survivants, quant à eux, fusionnent au sommet survivant voisin le plus similaire. Chaque sommet survivant met à jour ses attributs en fonction de ceux de ses fils. Les adjacences du nouveau niveau sont alors créées par rapport aux adjacences du niveau précédent et aux fusions effectuées.

A l'aide du schéma décrit brièvement ci-dessus le nouveau niveau est traité à son tour, jusqu'au dernier nœud qui représente l'Apex. La figure.11, nous montre l'irrégularité de la pyramide l'or de la fusion.

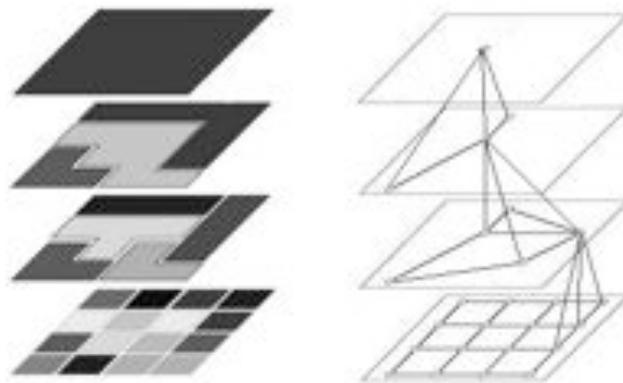


Figure.11 exemple de Pyramide Irrégulière d'un graphe d'adjacence 4-connexité

III. Régularisation discrète sur graphes pondérés

III.1 Introduction et problématique

En traitement d'image et par exemple en restauration d'image, on peut répartir les méthodes selon trois différents modèles de traitement [PEYR07] :

Traitement local : ce modèle est le modèle traditionnel où seules les interactions locales autour d'un pixel sont prises en compte. Beaucoup de méthodes de traitement d'images sont conformes à ce modèle. On peut citer les filtres gaussien, médian [LUKAC06B], anisotropique [PERONA90], basés sur des ondelettes [MALLAT98] ou la minimisation de la variation totale par équations aux dérivées partielles [CHAN05B, RUDIN92].

Traitement semi local : ce modèle correspond à prendre en compte des interactions de voisinage plus large qui peuvent être favorisées par la géométrie de l'image. On peut citer le filtre de YAROSLAVSKY [YAROSL85], le filtre bilatéral [TOMASI98], le filtre de SUSAN [SMITH97],

Traitement non local : ce modèle a été récemment proposé par BUADES et al. [BUADES05A, BUADES05B] qui ont proposé de supprimer les contraintes spatiales locales et de les remplacer par des contraintes sur des blocs de pixels (des patches). Ce modèle est devenu très populaire et on le retrouve à présent dans de nombreux travaux [GILBOA07, KERVRA06, PEYR07, SZLAM06].

On trouve deux notions sous-jacentes dans tous ces modèles : le voisinage de traitement et le type de variété à traiter. Nous avons vu dans la section précédente que les graphes sont des structures de données tout à fait adaptées pour ce type de problématique. Nous proposons donc ici un formalisme général de traitement de données d'images de dimensions quelconques sur des graphes de topologies arbitraires, diffusion sur graphes d'adjacence de régions pour la création de hiérarchies de partition. Parmi les modèles de traitement local, les méthodes variationnelles basées sur une régularisation fournissent un cadre formel qui a été extrêmement étudié pour une grande variété de problèmes de vision par ordinateur. Les solutions de modèles variationnels peuvent être obtenues en minimisant des fonctions d'énergies appropriées et cette minimisation est habituellement réalisée à l'aide d'équations aux dérivées partielles (EDP) [AUBERT02]. Les EDP sont définies en continu et lorsque l'existence et l'unicité d'une solution ont été prouvées, elles sont discrétisées afin d'obtenir une solution numérique [CHAN05B, TANG01, TSCHUM05, WEICKE99]. Nous proposons de nous inspirer de ces travaux définis en continu pour définir un cadre formel de régularisation discrète sur graphes de topologies arbitraires. Ainsi, contrairement aux EDP, notre méthode de régularisation pourra s'appliquer sur des données ayant une organisation spatiale différente d'une grille et elles ne nécessiteront aucun schéma de discrétisation puisqu'exprimées directement en discret. De plus, notre approche permet naturellement d'effectuer des traitements selon n'importe lequel des précédents modèles (local, semi local ou non local) en modifiant simplement la topologie du graphe et les poids des arêtes de celui-ci. Un certain nombre des approches de la littérature peuvent donc être vues comme un cas particulier de notre approche et elle présente également des liens avec la théorie spectrale sur graphe.

Nous formalisons la régularisation discrète d'une fonction $f^0 \in \mathcal{H}(V)$ par le problème suivant de minimisation sur graphe :

$$\min_{f \in \mathcal{H}(V)} \left\{ E_p(f, f^0, \lambda) = \frac{1}{p} \sum_{v \in V} \|\nabla_v f\|^p + \frac{\lambda}{2} \sum_{v \in V} \|f - f^0\|^2 \right\}$$

où $p \in [1, +\infty[$ est le degré de régularité, λ un terme d'attache aux données et ∇f représente un gradient pondéré de la fonction f sur un graphe. La solution de ce problème de minimisation amène à la définition d'une famille de filtres non-linéaires. Nous commençons par définir des opérateurs de géométrie différentielle sur graphe, puis nous présentons le problème de régularisation discrète et nous montrons les liens avec d'autres approches. Enfin nous présentons des résultats sur des données de dimensions quelconques réparties sur des graphes de topologies arbitraires.

III.2 Géométrie différentielle sur graphes

Ce que nous présentons ici est une discrétisation de la géométrie différentielle classique en continu et des définitions similaires ont été proposées pour l'apprentissage semi-supervisé [ZHOU04] ainsi que pour le calcul différentiel sur graphes [BENSOU05].

III.2.1 Opérateurs gradient et divergence

Nous rappelons tout d'abord que nous considérons que les fonctions $f \in \mathcal{H}(V)$ associent un vecteur de dimension q à chaque nœud : $f : V \rightarrow \mathbb{R}^q$. L'opérateur de différence $df : \mathcal{H}(V) \rightarrow \mathcal{H}(E)$ sur un graphe $G = (V, E)$ d'une fonction $f \in \mathcal{H}(V)$ sur une arête (u, v) reliant deux nœuds u et v est défini pour toute arête $(u, v) \in E$ par :

$$(df)(u, v) = (df)_{uv} = \sqrt{\omega(u, v)}(f(v) - f(u)) \quad (1)$$

Dans le cas d'un graphe représentant une grille discrète, la différence revient à $(f(v) - f(u))$ qui est l'opérateur de différence classique en analyse numérique. La dérivée $\left. \frac{\partial f}{\partial e} \right|_v : \mathcal{H}(V) \rightarrow \mathbb{R}^+$ d'une fonction f au nœud v selon une arête $e = (u, v)$ est définie par

$$\left. \frac{\partial f}{\partial e} \right|_v = (df)(u, v)$$

Nous la noterons également $\partial_v f(u)$. Cette définition est cohérente avec la définition continue de la dérivée d'une fonction, par exemple si $f(v) = f(u)$ alors $\left. \frac{\partial f}{\partial e} \right|_v = 0$. Nous avons donc

$$\left. \frac{\partial f}{\partial e} \right|_v = -\left. \frac{\partial f}{\partial e} \right|_u \quad (\partial_v f(u) = -\partial_v f(v)) \quad \text{Et } \partial_u f(u) = 0.$$

Pour une fonction $f \in \mathcal{H}(V)$ et un nœud v , le gradient pondéré de f au nœud v est l'opérateur vectoriel défini comme $\nabla : V \rightarrow \mathbb{R}^N$ par

$$\nabla f(v) = (\nabla_v f(u) : (u, v) \in E, u \sim v)^T \quad (2)$$

Ainsi, la norme du gradient pondéré sur graphe notée $\|\nabla f\|$ en un nœud v est définie comme $\|\nabla\| : \mathbb{R}^N \rightarrow \mathbb{R}^+$ par

$$\|\nabla_v f\| = \|\nabla f(v)\| = \sqrt{\sum_{u \sim v} \partial_u f(u)^2} = \sqrt{\sum_{u \sim v} \omega(u, v)(f(v) - f(u))^2} \quad (3)$$

La norme du gradient mesure la régularité d'une fonction autour d'un nœud. Soit \mathcal{R} une fonctionnelle sur $\mathcal{H}(V)$ définie par $\mathcal{R}_p(f) = \sum_{v \in V} \|\nabla_v f\|^p$, pour chaque $p \in [1; +\infty[$. Cette fonctionnelle \mathcal{R}_p peut être vue comme une mesure de régularité de f puisque c'est la somme des variations locales en chaque nœud.

La divergence sur graphe est l'opérateur $div : \mathcal{H}(E) \rightarrow \mathcal{H}(V)$ qui satisfait

$$\langle df, h \rangle_{\mathcal{H}(E)} = \langle f, -div(h) \rangle_{\mathcal{H}(V)}$$

avec $f \in H(V)$ et $h \in H(E)$. Cet opérateur $-div$ est donc l'opérateur adjoint d^* de l'opérateur de différence d . A partir des définitions des produits scalaires dans $H(V)$ et $\mathcal{H}(E)$ et de l'équation (2.1), on peut prouver que la divergence sur graphe d'une fonction $h \in \mathcal{H}(E)$ en un nœud v peut s'exprimer comme

$$(d^*h)(v) = (-div(h))(v) = \sum_{u \sim v} \sqrt{\omega(v, u)}(h(u, v) - h(v, u)) \quad (4)$$

III.2.2 Opérateur p-Laplacien

Le Laplacien sur graphe est un opérateur $\Delta : \mathcal{H}(V) \rightarrow \mathcal{H}(V)$ défini par

$$\Delta f = -div(df) = d^*(df).$$

Le Laplacien sur graphe est un opérateur linéaire car le gradient et la divergence sont des opérateurs linéaires. De plus, le Laplacien est auto-adjoint :

$$\langle \Delta f, g \rangle_{\mathcal{H}(V)} = \langle df, dg \rangle_{\mathcal{H}(E)} = \langle df, \Delta g \rangle_{\mathcal{H}(V)}$$

et positif semi-défini :

$$\langle \Delta f, f \rangle_{\mathcal{H}(V)} = \langle df, df \rangle_{\mathcal{H}(E)} = \mathcal{R}_p(f) \geq 0$$

ce qui implique que

$$\Delta f = \frac{\partial \mathcal{R}_p(f)}{\partial f} \quad (5)$$

La courbure sur graphe est un opérateur non linéaire $\kappa : \mathcal{H}(V) \rightarrow \mathcal{H}(V)$ défini par :

$$\kappa f = -div\left(\frac{df}{\|\nabla f\|}\right) = d^*\left(\frac{df}{\|\nabla f\|}\right) \quad (6)$$

Nous pouvons généraliser le Laplacien et la courbure par un opérateur qui peut être considéré comme un analogue discret du p-Laplacien dans le cas continu [TROYAN00]. Le p-Laplacien sur graphe est un opérateur $\Delta_p : \mathcal{H}(V) \rightarrow \mathcal{H}(V)$ avec $p \in [1; +\infty[$ défini par

$$\Delta_p f = -\frac{1}{p} div(\|\nabla f\|^{p-2} df) = \frac{1}{p} d^*(\|\nabla f\|^{p-2} df) \quad (7)$$

En conséquence, nous avons $\Delta_1 = \kappa$ et $\Delta_2 = \Delta$. En substituant (1) et (4) dans la définition (2.7) de $\Delta_p f$, nous obtenons alors :

$$(\Delta_p f)(v) = \frac{1}{p} \sum_{u \sim v} \gamma(u, v)(f(v) - f(u)) \quad (8)$$

où $\gamma(u, v)$ est la fonction définie par

$$\gamma(u, v) = \omega(u, v)(\|\nabla f(v)\|^{p-2} + \|\nabla f(u)\|^{p-2}) \quad (9)$$

qui généralise les opérateurs classiques de Laplacien et courbure sur graphe. En général Δ_p est

non linéaire (sauf pour le cas de $p = 2$) et est positif semi-défini.

Afin d'éviter le cas d'un dénominateur nul pour le calcul de la courbure ($p = 1$), le gradient $\|\nabla_v f\|$ est remplacé par sa version régularisée $\|\nabla_v f\|_\beta = \sqrt{\beta^2 + \|\nabla_v f\|^2}$

Où β est un paramètre ayant une faible valeur positive. Il est appelé le paramètre de régularisation par CHAN [CHAN01B].

III.3 Principe de la Segmentation sur graphe :

Soit $\mathcal{G} = (V, E)$ un graphe non-orienté. Les nœuds $v \in V$ sont l'ensemble des éléments à segmenter et les arcs (v, u) correspondent aux paires des nœuds voisins. Chaque arc (v, u) a un poids correspondant $\omega(v, u)$, qui est une mesure de dissimilarité positive entre les éléments voisins v et u . Concrètement, les éléments dans V sont les pixels de l'image et le poids est une certaine mesure de dissimilarité entre deux pixels connectés par un arc (différence d'intensité, couleur, localisation ...). Dans les approches de segmentation basées sur les graphes, une segmentation S est une partition de V aux composants tels que chaque composant (ou région) $C \subseteq S$ correspond à un composant connexe dans un graphe $\mathcal{G}' = (V, E')$, où $E' \geq E$. Autrement dit, chaque segmentation est induite par un sous-ensemble des arcs de E . Il existe plusieurs façons de mesurer la qualité d'une segmentation mais en général nous voulons que les éléments dans une région soient similaires, et les éléments dans des régions différentes soient dissimilaires. Cela veut dire que les arcs entre deux nœuds dans le même composant doivent relativement avoir un poids faible, et les arcs entre les nœuds dans des composants différents doivent avoir un poids élevé. Notant que la segmentation est sensible au choix de la mesure entre les pixels p . Un choix de mesure de dissimilarité faible a tendance à produire une sur-segmentation de l'image alors qu'une mesure élevée à produire une sous-segmentation de l'image.

III.4 Régularisation du graphe

Etant donné qu'un graphe est une représentation discrète de données dont la dimensionnalité et la topologie sont arbitraires, nous proposons ici en premier lieu un cadre général pour la régularisation sur graphe.

III.4.1. Principe général

Étant donné un graphe $\mathcal{G}(V, E)$ auquel est associé une fonction de pondération $\omega : E \rightarrow \mathbb{R}^+$, nous désirons effectuer la régularisation p-Laplacienne d'une fonction $f^0 \rightarrow \mathcal{H}(V)$ en utilisant le p-Laplacien. Cela revient à considérer des problèmes variationnels généraux sur graphes. Le but est alors, étant donné une fonction $f^0 \rightarrow \mathcal{H}(V)$ de trouver une fonction $f \rightarrow \mathcal{H}(V)$ qui est non seulement lisse sur \mathcal{G} mais suffisamment proche de f^0 . Ce problème d'optimisation peut être formalisé comme la minimisation d'une somme pondérée de deux termes d'énergies :

$$f^* = \min_{f \in \mathcal{H}(V)} \left\{ E_p = \frac{1}{p} \mathcal{R}_p(f) + \frac{\lambda}{2} \|f - f^0\|^2 = \frac{1}{p} \sum_{v \in V} \|\nabla_v f\|^p + \frac{\lambda}{2} \sum_{v \in V} \|f - f^0\|^2 \right\} \quad (10)$$

Le premier terme est le terme de régularité qui impose que f ne change pas trop entre des nœuds proches. Le second terme est l'attache aux données qui impose que f ne soit pas trop éloignée de f^0 $\lambda \geq 0$. Est un paramètre de fidélité aux données initiales, appelé multiplicateur de Lagrange, qui spécifie le compromis entre ces deux termes d'énergie. Les deux termes de l'énergie E_p sont des fonctions strictement convexes de f [CHAN01B, CHAMBO97], et ce problème a une

solution unique pour $p = 1$ ou $p = 2$ qui satisfait :

$$\left. \frac{\partial E_p}{\partial f} \right|_v = \frac{1}{p} \frac{\partial}{\partial f} \|\nabla_v f\|^p + \lambda(f^*(v) - f^0(v)) = 0, \forall v \in V. \quad (11)$$

En utilisant la propriété (5) du p-Laplacien pour calculer la dérivée du premier terme de E_p , l'équation (11) peut être réécrite ainsi :

$$(\Delta_p f^*)(v) + \lambda(f^*(v) - f^0(v)) = 0, \forall v \in V \quad (12)$$

La solution f^* de (10) est également la solution de (12). En substituant l'expression du p-Laplacien dans (12), nous obtenons :

$$\left(\lambda + \frac{1}{p} \sum_{u \sim v} \gamma_{uv} \right) f^*(v) - \frac{1}{p} \sum_{u \sim v} \gamma(u, v) f^*(u) = \lambda f^0(v), \forall v \in V. \quad (13)$$

Parmi les méthodes existantes [BRONSH97, HAGEMA81] qui peuvent être utilisées pour résoudre (13), nous utilisons un algorithme itératif de type Gauss-Jacobi [GREENB97, KINCAI02]. Soit t le numéro d'itération, et $f^{(t)}$ la solution de (13) à l'itération t . La fonction initiale $f^{(0)}$ peut être initialisée avec f^0 . L'itération correspondante est donnée par :

$$\begin{cases} f^{(0)} = f^0 \\ \gamma^{(t+1)}(u, v) = \omega(u, v) \left(\|\nabla f^{(t+1)}(v)\|^{p-2} + \|\nabla f^{(t+1)}(u)\|^{p-2} \right), \forall (u, v) \in E \\ f^{(t+1)}(v) = \frac{p\lambda}{p\lambda + \sum_{u \sim v} \gamma(u, v)} f^0(v) + \frac{\sum_{u \sim v} \gamma(u, v) f^t(u)}{p\lambda + \sum_{u \sim v} \gamma(u, v)}, \forall v \in V, \end{cases} \quad (14)$$

où $\gamma^{(t)}$ est la fonction (u, v) à l'itération t . Nous définissons la fonction $\varphi: V \rightarrow \mathbb{R}^+$ par

$$\varphi_{uv}^{(t)} = \frac{\gamma^t(u, v)}{p\lambda + \sum_{u \sim v} \gamma^t(u, v)} \text{ Si } u \neq v \text{ Et } \varphi_{uv}^{(t)} = \frac{p\lambda}{p\lambda + \sum_{u \sim v} \gamma^t(u, v)} \text{ Sinon.}$$

Ainsi, une itération de notre algorithme de régularisation 14 peut être réécrite par :

$$f^{(t+1)}(v) = \varphi_{uv}^{(t)} f^0(v) + \sum_{u \sim v} \varphi_{uv}^{(t)} f^{(t)}(u) \quad (15)$$

On peut noter que la valeur de $f(v)$ à une itération donnée $(t + 1)$ dépend de deux quantités : la valeur originale de f au noeud v (i.e. $f^0(v)$) et une somme pondérée des valeurs à l'itération t dans le voisinage de v . Ainsi, itérer (15) permet d'obtenir un filtre passe bas dont les coefficients sont mis à jour au fur et à mesure des itérations. En modifiant la topologie du graphe, la représentation associée à chaque nœud du graphe et les poids des arêtes de celui-ci, nous obtenons une famille de filtres non linéaires simples et rapides.

III.4.1.1 Régularisation pour $p = 2$

Quand $p = 2$, en utilisant (12) on en déduit que la solution de (10) est basée sur le Laplacien et satisfait :

$$\Delta f^* + 2\lambda(f^* - f^0) = 0 \quad (16)$$

Ceci peut être vu comme un analogue discret de l'équation d'Euler-Lagrange. Dans ce cas le filtre de la section 6.a est linéaire et si $\lambda \neq 0$, il converge vers la solution de (10). Le schéma itératif utilisé pour résoudre ceci est exprimé par, $\forall v \in V$:

$$\begin{cases} f^0 = f \\ f^{(t+1)}(v) = \frac{2\lambda}{2\lambda + \sum_{u \sim v} \omega(u, v)} f^0(v) + \frac{\sum_{u \sim v} \omega(u, v) f^t(u)}{2\lambda + \sum_{u \sim v} \omega(u, v)} \end{cases} \quad (17)$$

où t indique le numéro de l'itération. Nous définissons la fonction $c : V \rightarrow \mathbb{R}^+$ par $c(v) = \frac{1}{2\lambda + \sum_{u \sim v} \omega(u, v)}$ et le schéma itératif s'exprime alors ainsi, $\forall v \in V$:

$$f^{(t+1)} = 2\lambda c(v) f^0(v) + c(v) \sum_{u \sim v} \omega(u, v) f^t(u) \quad (18)$$

On peut constater qu'à chaque itération, la nouvelle valeur d'un nœud est obtenue à partir des valeurs pondérées de ses voisins et de la sienne. Le filtre correspondant est un filtre passe-bas dont le comportement s'adapte à l'image à traiter grâce aux valeurs de $\omega(u, v)$ calculées à partir de f^0 . Quand $\lambda = 0$ et $\omega(u, v) = 1 \forall (u, v) \in E$, c'est l'analogue discret de la diffusion sur des variétés Riemanniennes [SAPIRO96].

III.4.1.2 Régularisation pour $p = 1$

Quand $p = 1$, en utilisant (12) on en déduit que la solution de (10) est basée sur l'opérateur non linéaire de courbure κ et satisfait

$$\kappa f^* + 2\lambda(f^* - f^0) = 0 \quad (19)$$

Le schéma itératif utilisé pour résoudre ceci est exprimé par, avec $f^0 = f, \forall (u, v) \in E, \forall v \in V$:

$$\begin{cases} f^0 = f \\ \gamma^{(t+1)}(u, v) = \omega(u, v) \left(\frac{1}{\varepsilon + \|\nabla f^{(t+1)}(v)\|} + \frac{1}{\varepsilon + \|\nabla f^{(t+1)}(u)\|} \right) \\ f^{(t+1)}(v) = \frac{2\lambda}{2\lambda + \sum_{u \sim v} \gamma^t(u, v)} f^0(v) + \frac{\sum_{u \sim v} \gamma^t(u, v) f^t(u)}{2\lambda + \sum_{u \sim v} \gamma^t(u, v)} \end{cases} \quad (20)$$

De même que pour $p = 2$, nous pouvons définir la fonction $c : V \rightarrow \mathbb{R}^+$ par $c^t(v) = \frac{1}{2\lambda + \sum_{u \sim v} \gamma^t(u, v)}$ et le schéma itératif utilisé pour calculer les nouvelles valeurs de f^t s'exprime ainsi, $\forall v \in V$:

$$f^{(t+1)}(v) = 2\lambda c^t(v) f^0(v) + c^t(v) \sum_{u \sim v} \gamma^t(u, v) f^t(u) \quad (21)$$

Si l'on compare ceci avec l'algorithme itératif dans le cas de $p = 2$, c'est également un filtre passe-bas mais dont les coefficients sont mis à jour de façon adaptative au cours des itérations en plus de mettre à jour la fonction f . Pour $p = 1$, si $\forall(u, v) \in E, \omega(u, v) = 1$, ce filtre itératif sur un graphe de type grille régulière correspond au TV digital filter [CHAN01B] ($TV + L^2$).

III.4.2 Représentation associée aux nœuds du graphe

Comme nous l'avons précédemment mentionné, à chaque nœud v du graphe est associé un vecteur $f(v)$. Par exemple, dans le cas d'un graphe de type grille pour une image couleur, $f(v)$ est le vecteur RGB associé au pixel. Dans le cas des images couleur RGB classiques, nous avons donc $q = 3$. On peut également considérer par exemple une représentation de type Luminance-Chromaticité [CHAN01A] et restaurer les deux séparément pour ensuite les re-combiner. Dans le cas où les données associées à un nœud sont multi-variées, nous effectuons quelques schémas de régularisation indépendants (un par composante). Pour le cas de $p = 1$, le p -Laplacien est alors différent pour chaque composante. Cependant, afin de prendre en compte le couplage existant entre les différentes composantes, les régularisations des composantes ne doivent pas être basées sur des géométries locales indépendantes mais sur une géométrie vectorielle commune. Ceci est nécessaire afin de prendre en compte la corrélation qui existe entre les différentes composantes du vecteur $f(v)$. En conséquence, nous considérons la norme suivante

$$\|\nabla f(v)\| = \sqrt{\sum_{i=1}^q \|\nabla f_i(v)\|^2}$$

Où $f_i(v)$ désigne la $i^{\text{ème}}$ composante de $f(v)$.

III.4.3 Poids des arêtes du graphe

Même si nous avons associé à chaque nœud v du graphe un vecteur $f(v)$, nous pouvons intégrer des informations supplémentaires (extraites du voisinage de v ou de $f(v)$) dans le calcul des poids des arêtes. Nous noterons $Ff(v)$ un vecteur de caractéristiques calculé à partir de $f(v)$ et de $f(u)$ tel que $u \sim v$. Dans le cas le plus simple, on peut avoir $Ff(v) = f(v)$. Plus généralement, à chaque nœud $v \in V$ du graphe, nous pouvons associer un vecteur d'attributs $Ff(v) \in \mathcal{R}_m$ avec $m \in \mathbb{N}^+$. Ce vecteur peut donc être de dimension quelconque et incorporer des attributs variés décrivant un nœud du graphe (couleur, texture, etc.) à partir de mesures locales, semi locales ou non locales selon la topologie du graphe considéré.

Nous pouvons ensuite associer une fonction de poids à chaque arête du graphe. Cette fonction de poids détermine le type de régularisation induite par $\mathcal{R}_p(f)$. Les poids sont positifs et symétriques; ils permettent de quantifier la similarité entre les données multi-variées associées à deux nœuds reliés par une arête. La formulation générale d'une fonction de poids est alors la suivante $\omega(u, v) = g(Ff(u), Ff(v)), \forall(u, v) \in E$. A titre d'exemples, nous pouvons en considérer plusieurs :

1. $g_1(Ff(u), Ff(v)) = \frac{1}{\varepsilon + \|Ff(u), Ff(v)\|}$
2. $g_2(Ff(u), Ff(v)) = \exp\left(-\frac{\|Ff(u), Ff(v)\|^2}{\sigma^2}\right)$
3. $g_3(Ff(u), Ff(v)) = \exp\left(-\frac{\|Ff(u), Ff(v)\|^2}{\sigma^2}\right) \exp\left(\frac{\|u-v\|^2}{2\sigma_d^2}\right)$

Pour g_1 , le paramètre ε est nécessaire pour éviter une division par zéro et a une très petite valeur (10^{-4} dans notre cas). Pour g_2 et g_3 , le paramètre σ est généralement fixé a priori, mais il est préférable de l'estimer.

IV. Application

IV.1 Application sur L'image

Nous présentons ici les potentialités de la méthode, qui permet un filtrage/segmentation qui résulte naturellement après la régularisation du graphe.

Afin d'aboutir a un bon résultat, il faut souligner quelque points fondamentaux, que nous avons abordés précédemment, et qu'on a plus au moins modifiés ou adaptés dans notre traitement comme :

1. **Estimations des poids** : cela intervient directement sur le terme σ pour les fonctions de poids g_2, g_3 comme étant des poids (fixe/global/local) au fur et à mesure des itérations.
2. **Choix des paramètres optimaux ρ, λ** : selon le filtrage ou la segmentation, des essais et illustrations un peu plus bas, montrerons l'impact/l'effet de ces paramètres sur l'image traitée.
3. **Choix de la fonction de poids** : cela dépend de la nature du bruit (Impulsionnel ou Gaussien).
4. **Filtrage** : En premier lieu, on filtre le graphe en automatisant le nombre d'itération, en introduisant un critère d'arrêt (la différence entre deux images résultantes après deux itérations est ≤ 0.01).
5. **Fusion** : Enfin la fusion des zones homogènes selon un critère (seuil), peut être fixe ou évolutif.
6. **Critère d'arrêt** : déterminer automatiquement le nombre d'itération.
7. **Schéma Diffusion-Fusion**: alterner entre le filtrage et la Fusion, et entre les paramètres ρ, λ adéquate pour chaque traitement.

IV.1.1 Estimation des poids : hors le poids fixe, deux types d'estimations :

Dans notre traitement, nous avons choisi de l'estimer localement, pour chaque nœud, l'écart type mesuré sur l'image originale f^0 [LEZORAY07] autour de celui-ci.

Le deuxième type d'estimation est le poids global mais il ne s'avère pas très utile dans notre traitement. Son principe consiste à prendre le max de la distance entre deux nœuds (u, v) mesuré sur l'image originale.

Qu'elle soit fixe, globale ou locale cette valeur est affectée directement au terme σ , des fonctions g_2, g_3 .

IV.1.2 Choix des paramètres Optimaux : Afin de déterminer les paramètres optimaux ρ, λ , tout en sachant que $p = 2$ et $p = 1$, agissent comme un filtre passe-bas et TV digital filter respectivement, nous avons voulu aller plus loin et procéder à des tests avec des $p < 1$ et $\lambda \in [0.01, 0.1]$,

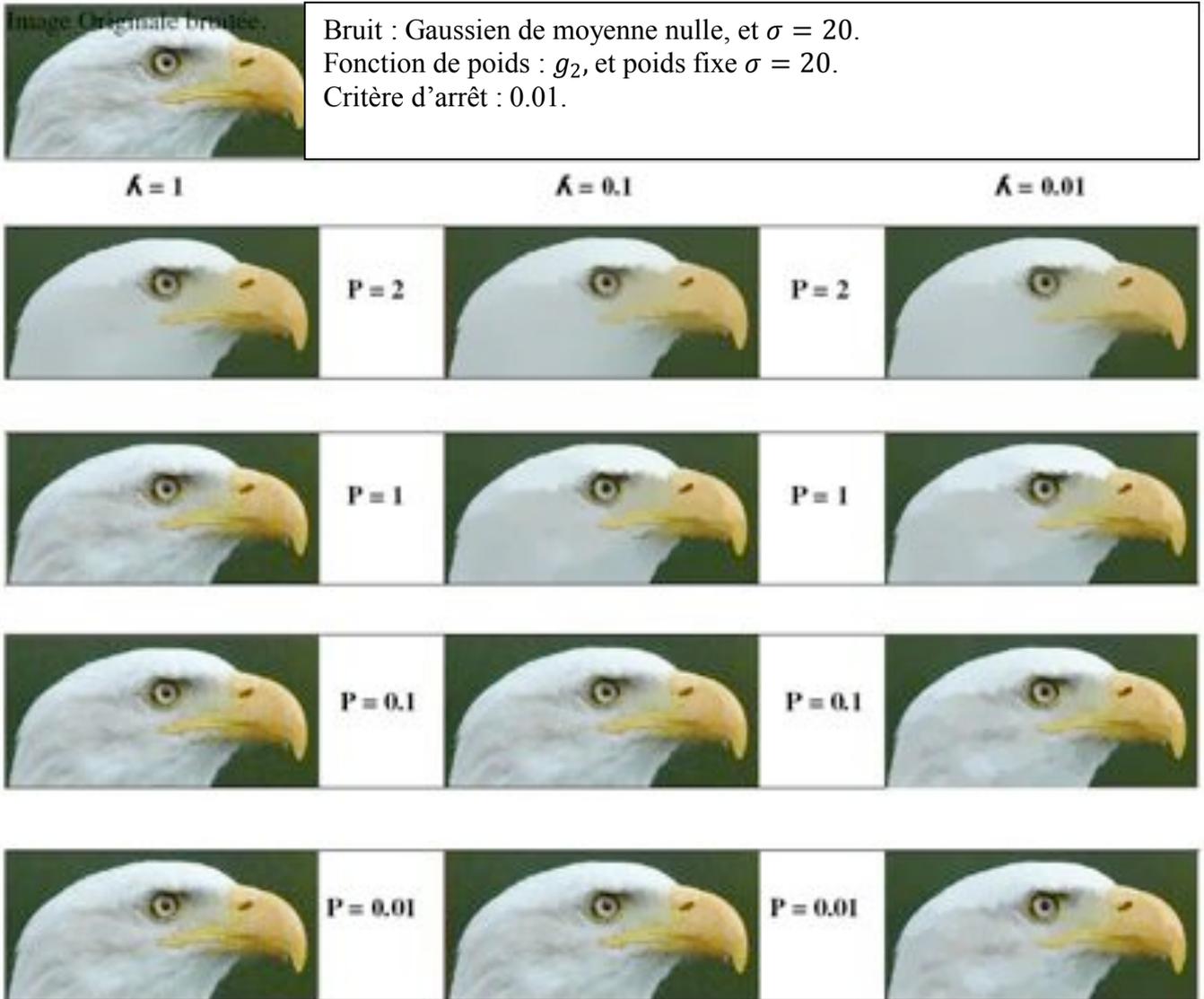


Figure 12. Régularisation du graphe avec différent paramètre sur une image bruitée.

Le filtrage est constaté avec les paramètres $p = 0.1$ ou $p = 0.01$ et $\lambda = 0.01$, avec ces derniers le bruit est nettement éliminé et la texture conservée, contrairement à $p = 1$ ou $p = 2$ et $\forall \lambda$ qui semble segmenté l'image. Cette vue globale, nous sera très utile par la suite pour élaborer un schéma diffusion-fusion.

IV.1.3 Choix de la fonction de poids : Ce choix dépend de la nature du bruit. Effectivement, si c'est un bruit Impulsionnel, la fonction g_1 s'avère très efficace, contrairement au bruit gaussien qui nécessite la fonction g_2 Ou g_3 , préservant mieux la texture que la fonction g_1 . La figure.14 montre que la fonction g_1 est mieux adapté au bruit Impulsionnel que la fonction g_2 et la figure13 montre les faiblesses de la fonction g_1 sur le bruit gaussien. Par contre, la fonction g_2 convient parfaitement au bruit gaussien voir figure.12.

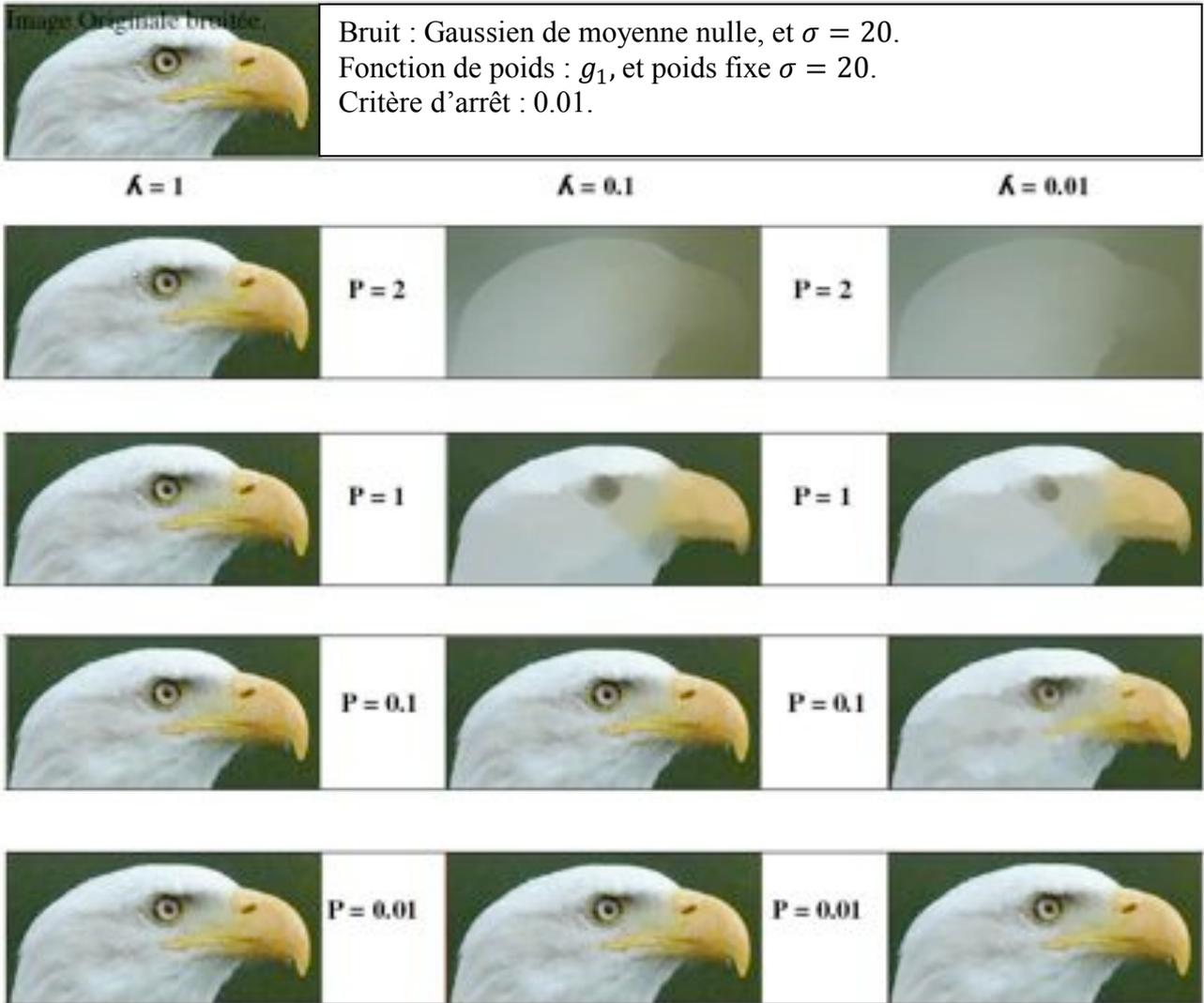


Figure.13 Restauration de l'image 'Aigle' avec la fonction de poids g_1 différent paramètre p, λ .



(a) Image originale.



(b) Image bruitée par 15% de bruit impulsionnel.

(c) Restauration de (a) avec g_2 .(d) Restauration de (b) avec g_1 .

Figure.14 Comparaison des deux fonctions g_1 et g_2
Pour la restauration d'image avec un bruit Impulsionnel

IV.1.4 Filtrage : Pour la restauration, nous considérons des graphes de type grille régulière en 4-connexité. A chaque nœud est associé la couleur du pixel correspondant, donc pour $f \in \mathcal{H}(V)$, nous avons $f : V \rightarrow \mathbb{R}^3$. La régularisation est effectuée sur chaque composante de l'image couleur indépendamment et donc le p-Laplacien est différent pour chaque composante, on a :

$$\gamma_i(u, v) = \omega(u, v) \|\nabla f_i(v)\|^{p-2} + \|\nabla f_i(u)\|^{p-2}$$

Pour la $i^{\text{ème}}$ composante. Dans le cas de $p = 2$ il est identique pour toutes les composantes, mais pour $p \leq 1$, il est différent. Afin d'éviter ce problème de non-couplage de la restauration entre les composantes et prendre en compte l'aspect vectoriel des données colorimétriques, le p-Laplacien est considéré comme étant le même pour les trois régularisations, mais en utilisant une norme vectorielle dans le cas où $p \leq 1$:

$$\gamma(u, v) = \omega(u, v) (\|\nabla f(v)\|_{3D}^{p-2} + \|\nabla f(u)\|_{3D}^{p-2})$$

$$\|\nabla f(v)\|_{3D}^{p-2} = \sqrt{\|\nabla f_1(v)\|^2 + \|\nabla f_2(v)\|^2 + \|\nabla f_3(v)\|^2}$$

La méthode de régularisation s'applique donc sur chaque composante indépendamment avec une pondération des arêtes et un gradient vectoriel qui agissent tous deux comme un couplage entre les composantes. Nous avons considéré la même fonction de pondération pour toutes les applications présentées dans ici, à savoir $(u, v) = \exp\left(-\frac{\|Ff(u), Ff(v)\|^2}{\sigma^2}\right)$,

$$\text{Et } \|f(u) - f(v)\| = \sqrt{\sum_{i=1}^3 \|f_i(u) - f_i(v)\|^2}$$

La figure.12 présente un exemple de restauration d'images couleur dans le cas de bruit gaussien. En comparaison avec les EDP, notre méthode de régularisation est aussi efficace tout en étant plus simple et plus rapide.

IV.1.5 Fusion : La régularisation peut s'avérer également extrêmement utile pour la simplification d'images. Partant d'une partition fine d'une image (une partition sur-segmentée mais proche de l'image originale), nous pouvons, en construisant le graphe d'adjacence des régions, opérer une régularisation directement sur ce graphe. A chaque nœud est associé la couleur moyenne de la région et à chaque arête la distance entre deux régions connexes. Ceci permet d'effectuer une simplification d'image sans déplacement des contours. Utiliser la régularisation sur graphe, est une alternative rapide et intéressante, à la simplification classique d'images. Enfin, la régularisation peut servir à la construction de hiérarchies de partitions. La régularisation tend à faire se rapprocher des modèles similaires, et ceux-ci fusionner au fur et à mesure des itérations. Le principe (voir [NOCK04]) revient à alterner diffusion (régularisation sans attache aux données, soit $\lambda = 0$) et fusion de régions selon un critère donné (voir dans [NOCK04] pour différents critères). Les critères peuvent être fixes, évolutifs. Ceci permet de générer des partitions de finesses différentes à partir d'une partition initiale, ce qui constitue une hiérarchie de partitions utile pour la segmentation. La figure.16 (b),(e) présente des graphiques de psnr sur la segmentation obtenue pour différents critères de fusion à savoir un critère fixe (seuil égal à 1), un critère évolutif (qui augmente selon le niveau de la pyramide irrégulière).

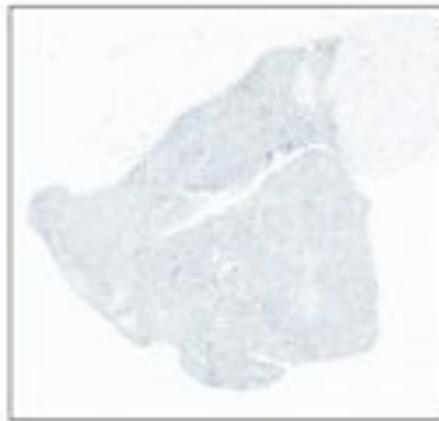
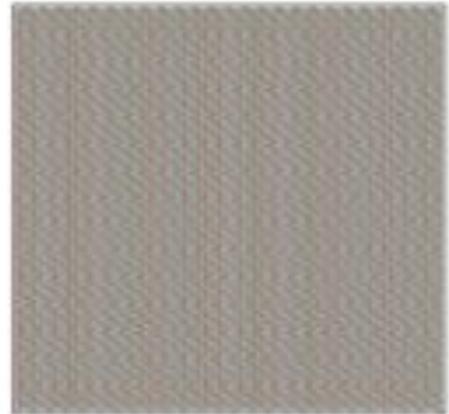


Image Initiale Histo3



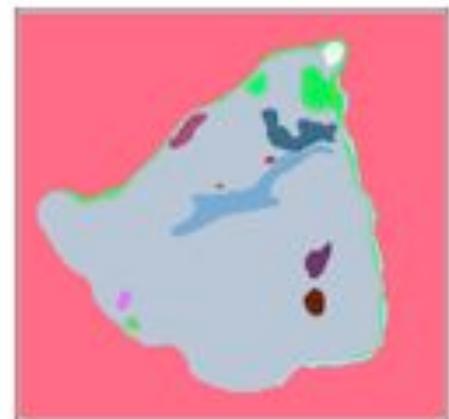
Régions(13216)

Debruitage :
 $P = 0.1$, $\text{Lambda} = 0.01$,
Itérations = 3 , Seuil fixe = 1.



Régions(1143)

Segmentation :
 $P = 1$, $\text{Lambda} = 0$.
Itérations = 1 , Seuil fixe = 10.



Régions(18)

Figure.15(a) exemple de Diffusion-Fusion (Simplification + segmentation).

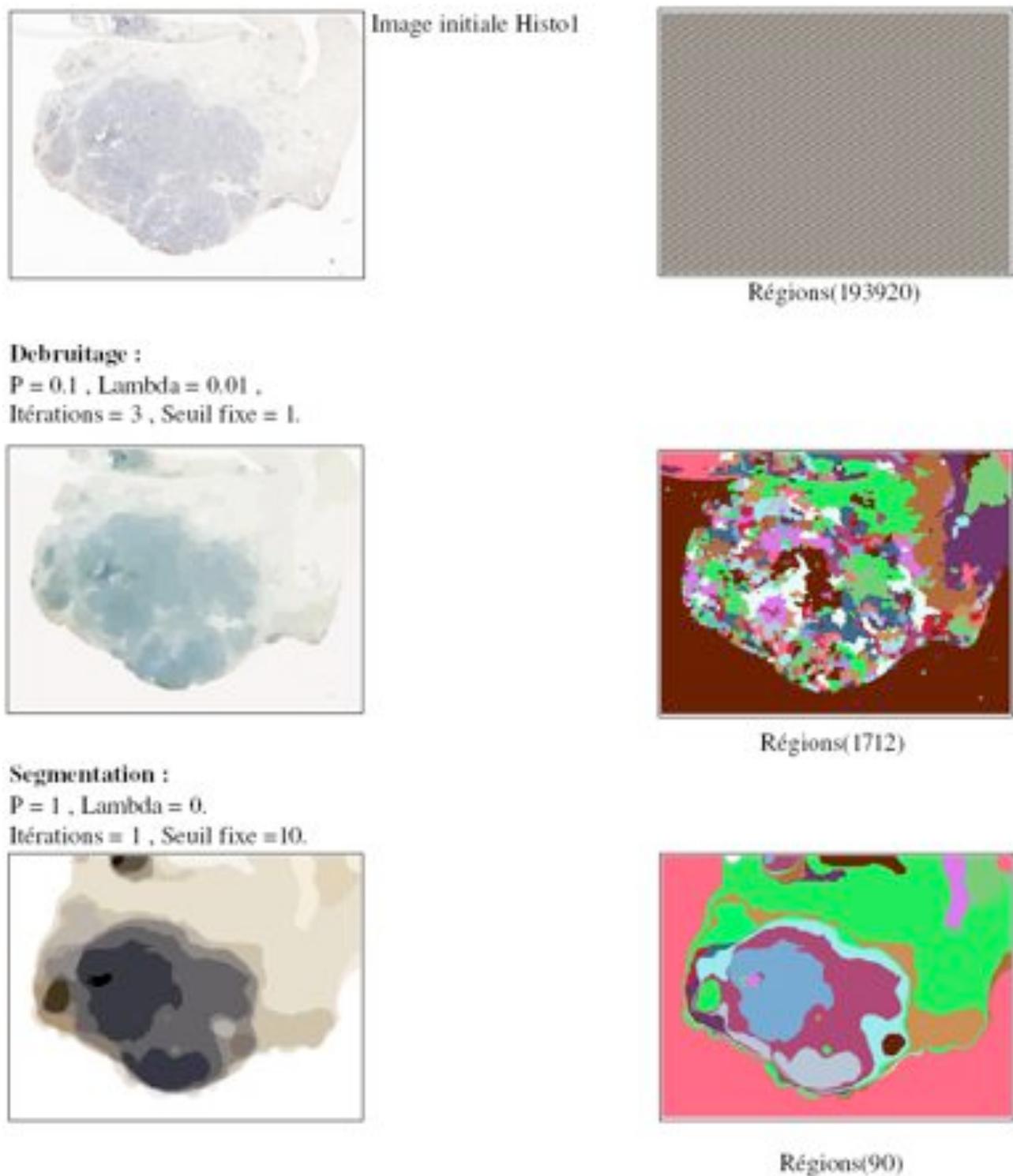


Figure.15(b) exemple de Diffusion-Fusion (Simplification + segmentation).

IV.1.6 Critère d'arrêt : Afin de déterminer automatiquement le nombre d'itérations, nous pouvons utiliser un critère de terminaison défini. Ceci permet de stopper les itérations lorsque peu de modifications sont constatées entre f^t et f^{t+1} .

IV.1.7 Le Schéma Diffusion-Fusion : Une alternation entre deux régularisations, avec différent p et avec ou sans attache aux données (λ) voir figure.15,16, nous donne en premier lieu une image simplifiée qu'on considère comme une étape de débruitage ou simplification. En revanche, la seconde image est plus une segmentation qu'un filtrage et donc, considérée comme une étape de segmentation. La carte de région générée est une carte de région type (Région/pixel), qui facilite la simplification. Le critère de fusion, qu'il soit fixe ou évolutif, pour la première étape sera minime (0.5 ou 1), et l'incrément du seuil évolutif par (+0.5 ou +1), au fur et à mesure des itérations. Pour la seconde étape, une des particularités de cette carte de région (Région/pixel) est qu'elle accentue la sur-segmentation, en fixant simplement un seuil fixe ≥ 5 , ou un seuil évolutif avec un grand pas.

Sachant qu'une pré-segmentation de l'image Original avant tout traitement, ne peut qu'accroître la qualité des deux étapes diffusion-Fusion. La figure.15,16 montre les performances de notre méthode, ce schéma de diffusion-fusion a été déduit lors des essais illustrés dans La section IV.3.

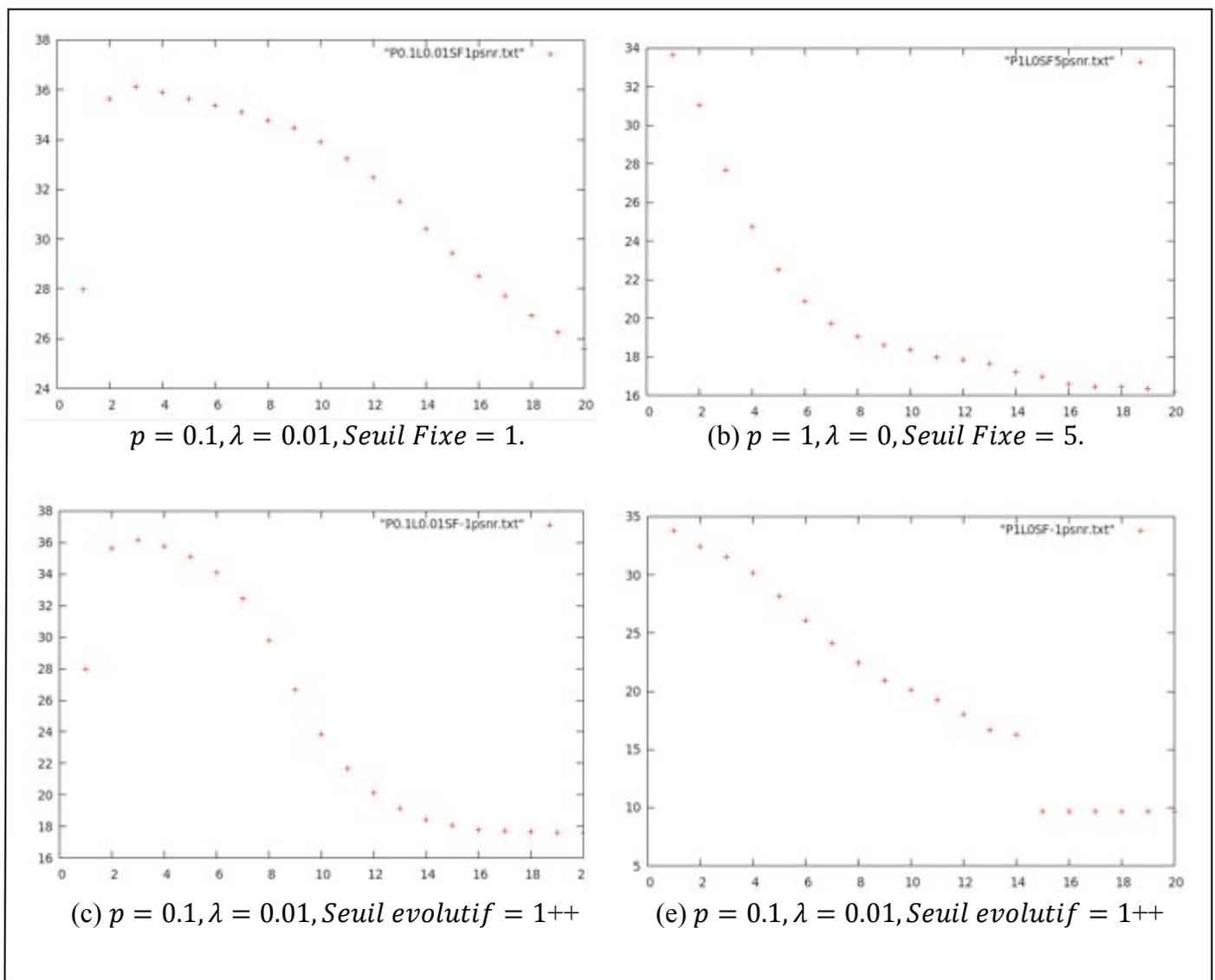


Figure.17 graphique du Psnr, Seuil Fixe et évolutif.

Ces résultats étaient effectués sur image « aigle » bruitée (gaussien (0,20)). On constate qu'on obtient des images avec un psnr relativement bon dès les premières itérations.

IV.2 Application sur Donnée quelconque

Avant toute chose, nous avons une formulation exprimée directement en discret, ce qui permet de considérer des graphes de topologies arbitraires. Comme la topologie peut être arbitraire, nous pouvons naturellement effectuer une régularisation locale, semi locale ou non locale en modifiant simplement la connexité du graphe. Par conséquent, nous pouvons alors effectuer une régularisation discrète non locale sur des variétés. A titre d'exemple, nous considérons un ensemble d'images représentant le chiffre manuscrit zéro dans la base USPS. Nous pouvons effectuer une régularisation non locale sur ces variétés, en considérant un graphe complet et une distance euclidienne simple entre les images. Chaque image est considérée comme un vecteur de taille 64 qui est régularisé soit 64 régularisations effectuées en parallèle. La figure. 18(a) présente l'ensemble des images.

Notant ici, que l'application d'une pré-segmentation telle que sur l'image, ne peut qu'être avantageuse, pour le découpage de la base par exemple ou l'analyse multi-résolution dans le cas où la base est de grande dimension, cette dernière sera simplifiée et/ou compressée les données de la base.

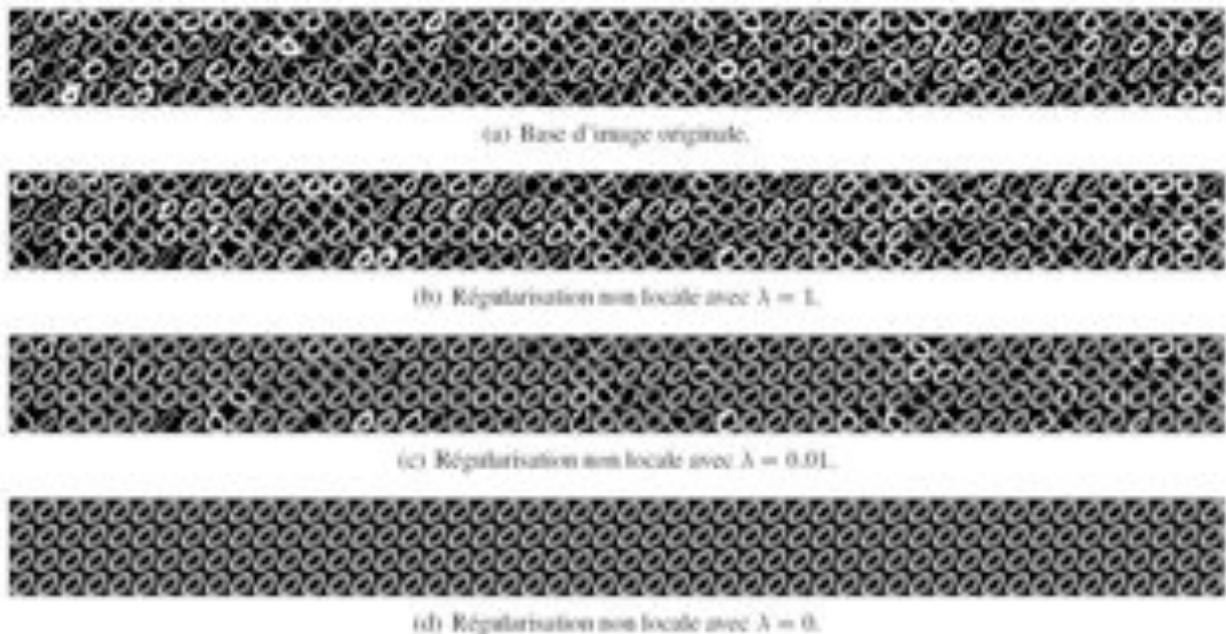


Figure.18 Régularisation discrète non local sur une base de données,
Qui représente le chiffre manuscrit Zéro.

On remarque bien évidemment que le terme d'attache aux données permet de ne pas trop s'éloigner des données initiales. Notre méthode permet donc de réaliser une régularisation discrète non locale sur des variétés, ce qui n'est le cas d'aucune méthode de la littérature jusqu'à présent.

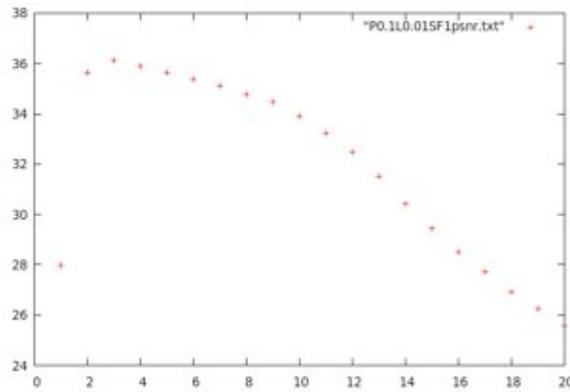
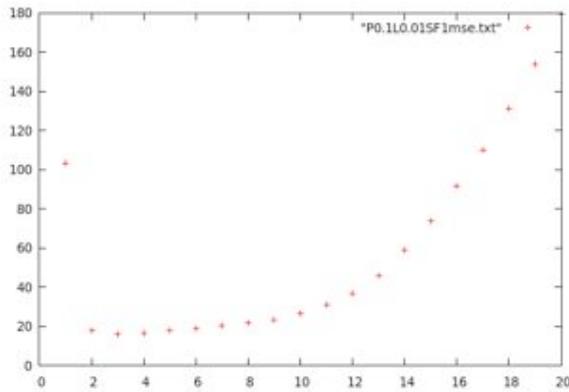
IV. Illustration

Afin de comprendre comment on a pu fixer certains paramètres comme, p , λ ou le seuil ou encore le schéma diffusion-fusion, et d'avoir une idée général sur l'approche, voici les illustrations entamées Lors de ces essais, cette série d'essais est effectuées avec différents paramètres et seuils fixe/évolutif sur une image 'aigle' bruitée (gaussien (0,20)),

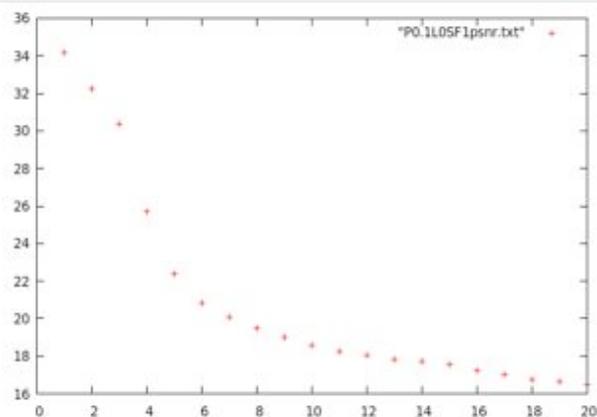
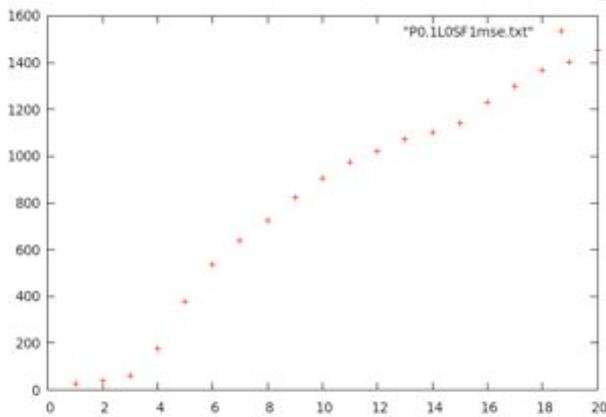
Les essais illustrés ci-dessous, sont accompagnées de leur graphique psnr et mse.

Malgré que nous somme allé plus loin dans les itérations (20), nous nous somme rendu compte que l'approche ne nécessite que 3 itérations pour obtenir des bons résultats.

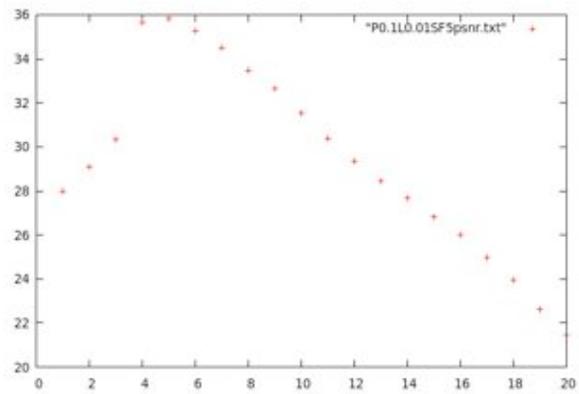
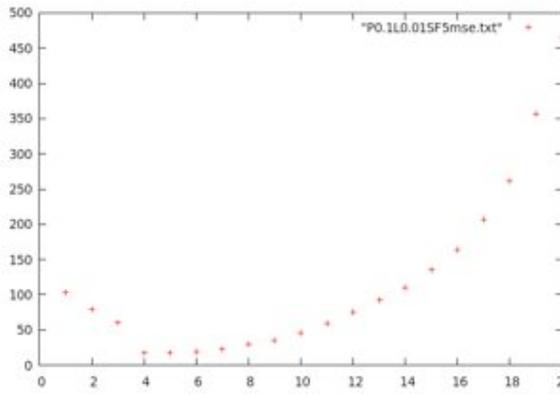
$$p = 0,1 \text{ Et } \lambda = 0.01 \text{ Et Seuil Fixe} = 1$$



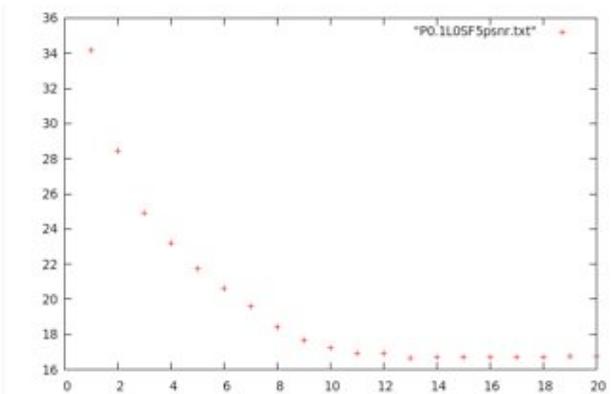
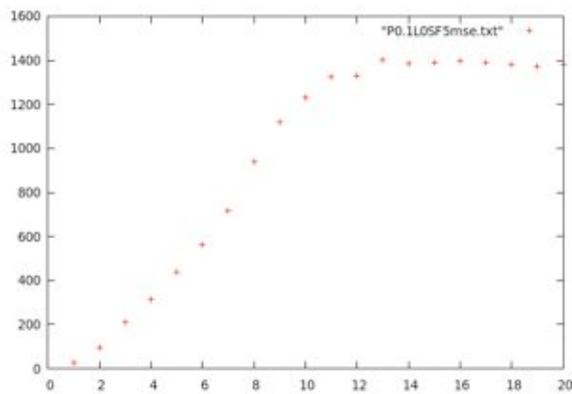
$$p = 0,1 \text{ Et } \lambda = 0 \text{ Et Seuil Fixe} = 1$$



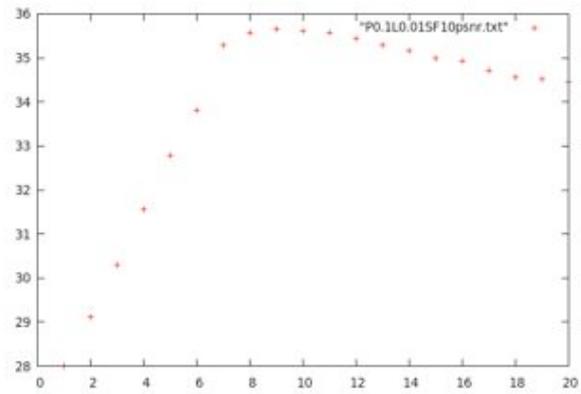
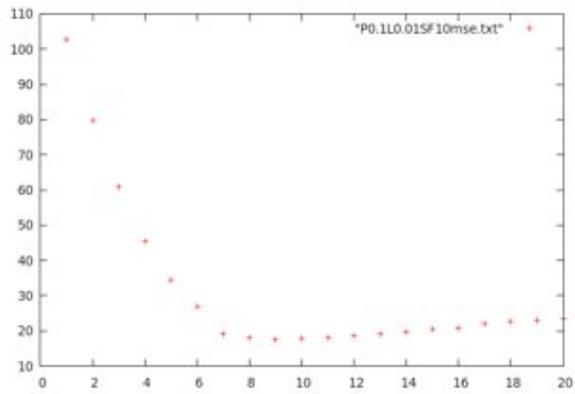
$p = 0,1$ Et $\lambda = 0,01$ Et Seuil Fixe = 5



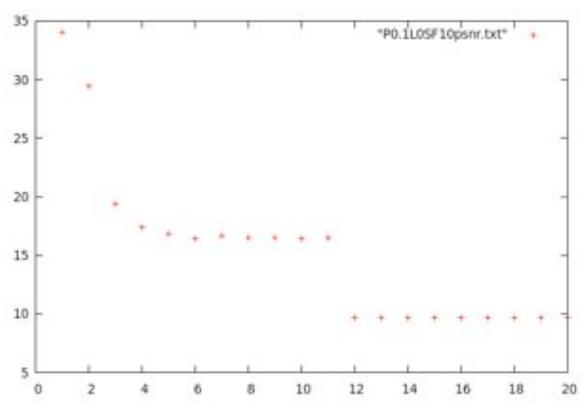
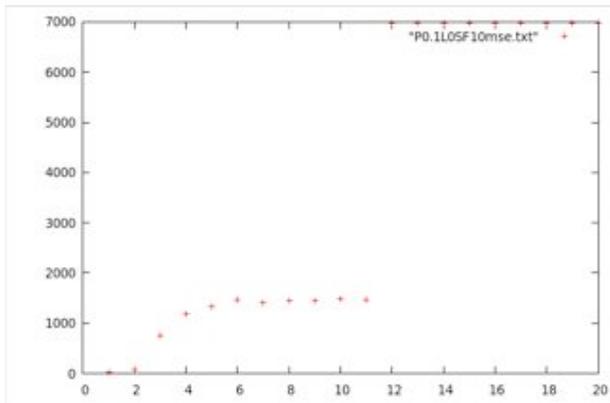
$p = 0,1$ Et $\lambda = 0$ Et Seuil Fixe = 5



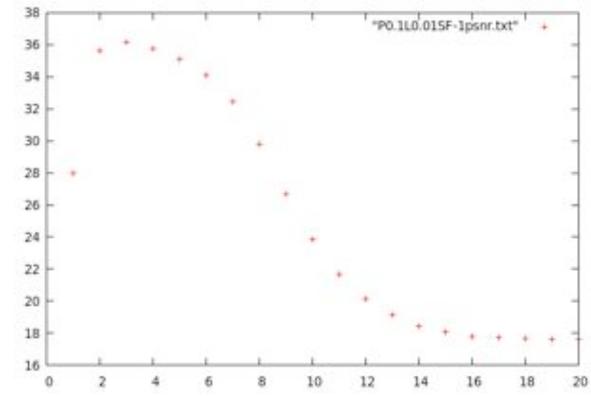
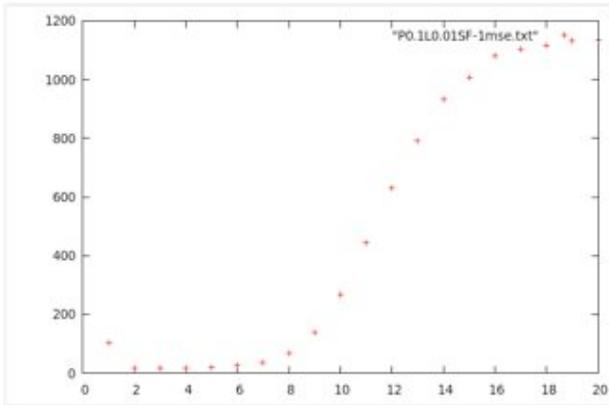
$p = 0,1$ Et $\lambda = 0,01$ Et Seuil Fixe = 10



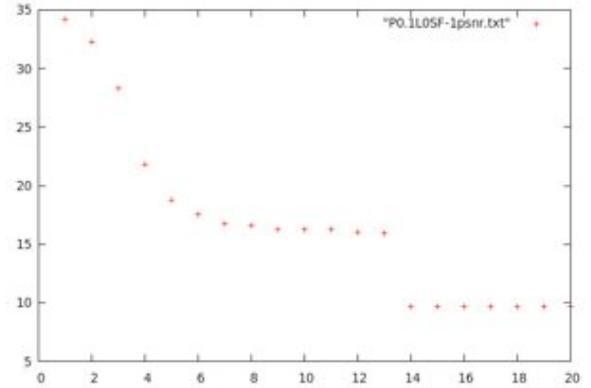
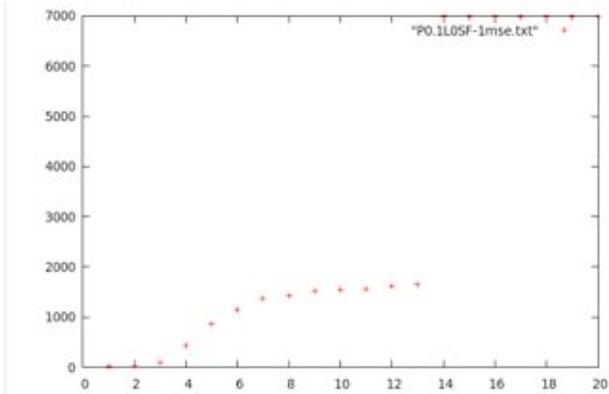
$p = 0,1$ Et $\lambda = 0$ Et Seuil Fixe = 10



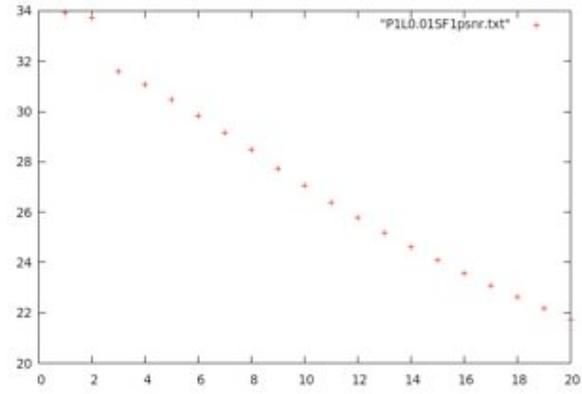
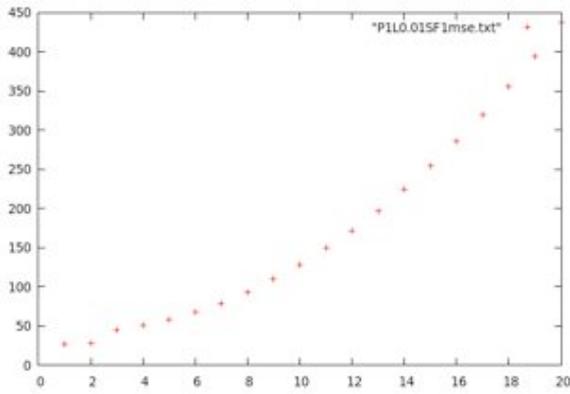
$p = 0,1$ Et $\lambda = 0,01$ Et Seuil évolutif = 1 + +



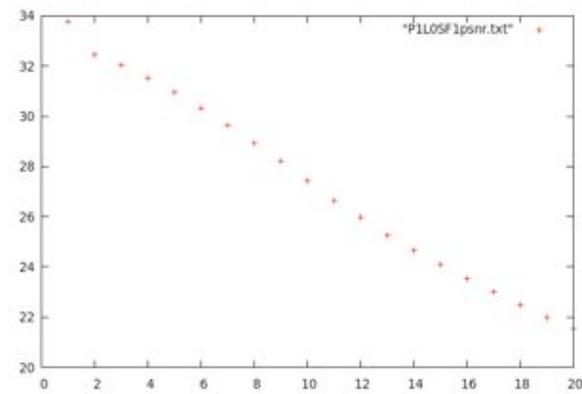
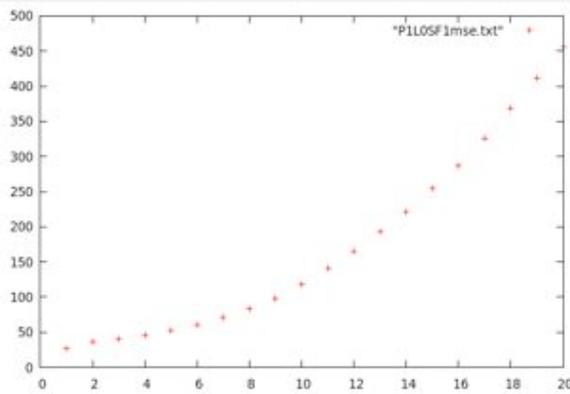
$p = 0,1$ Et $\lambda = 0$ Et Seuil évolutif = 1 + +



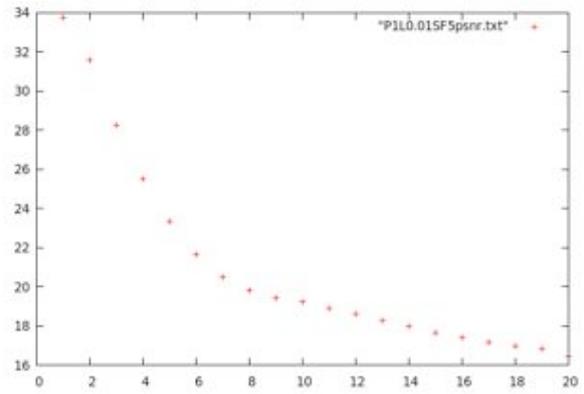
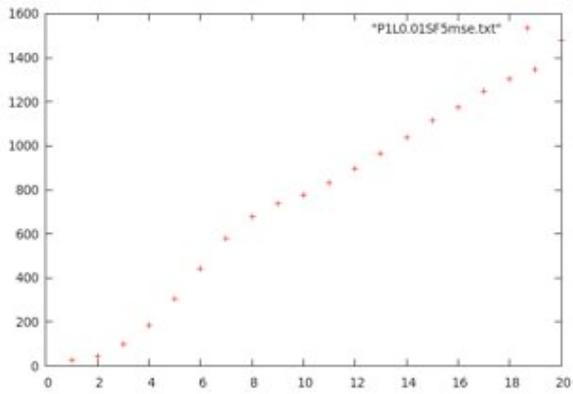
$p = 1$ Et $\lambda = 0,01$ Et Seuil fixe = 1



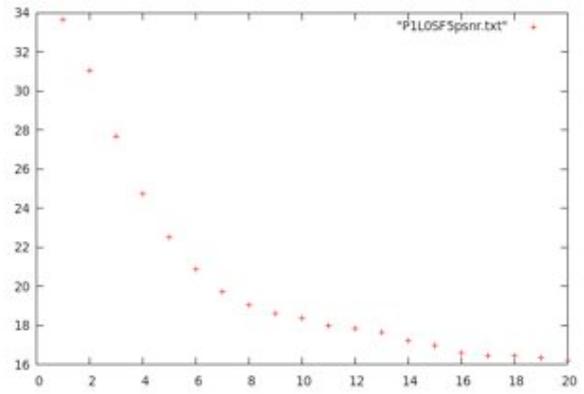
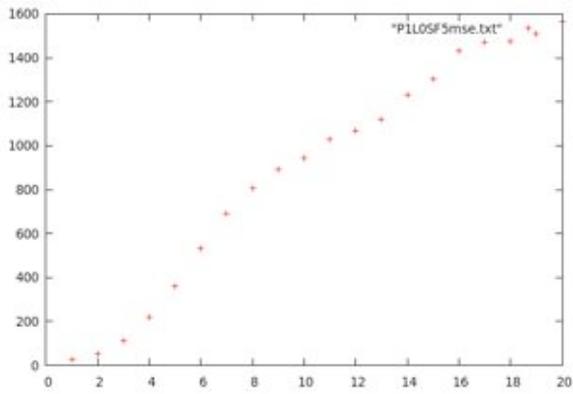
$p = 1$ Et $\lambda = 0$ Et Seuil fixe = 1



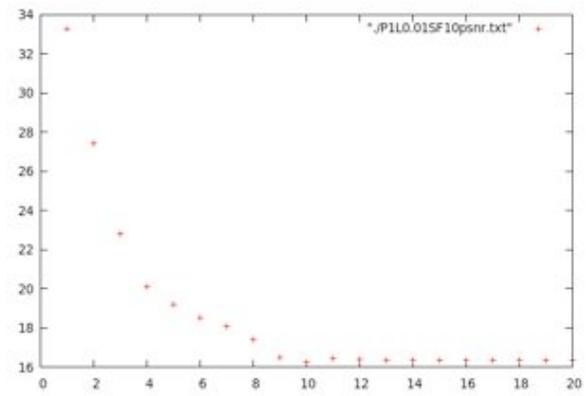
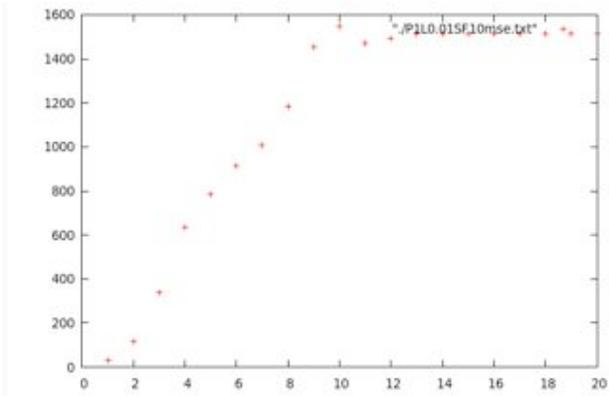
$p = 1$ Et $\lambda = 0,01$ Et Seuil fixe = 5



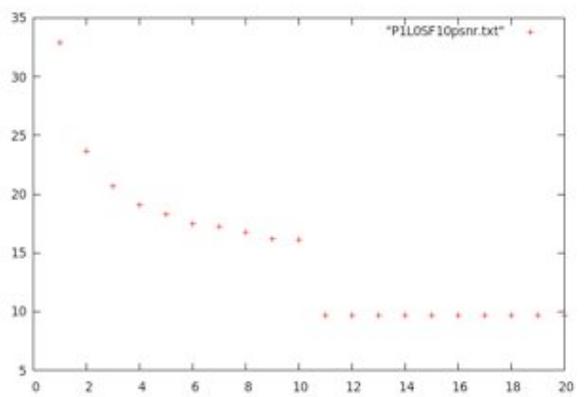
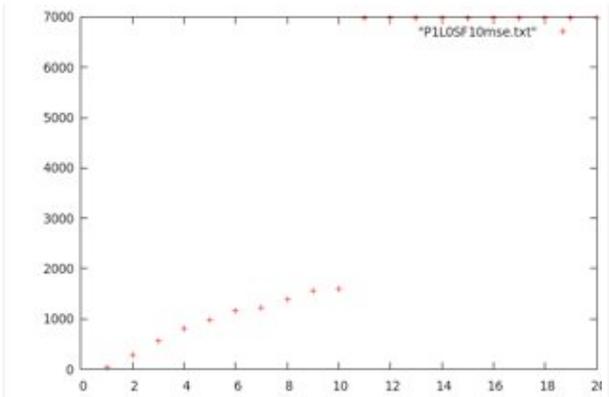
$p = 1$ Et $\lambda = 0$ Et Seuil fixe = 5



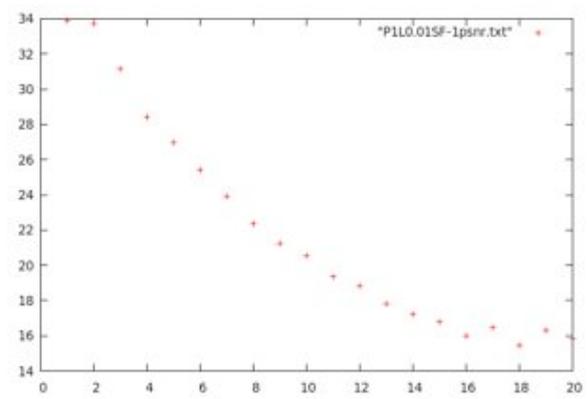
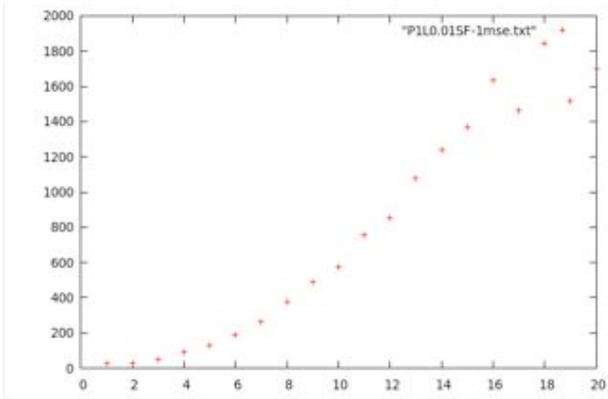
$p = 1$ Et $\lambda = 0,01$ Et Seuil fixe = 10



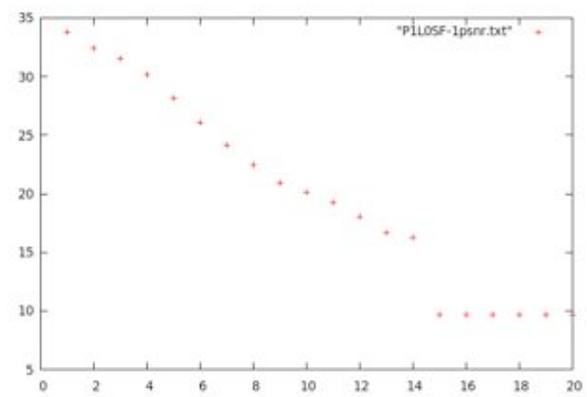
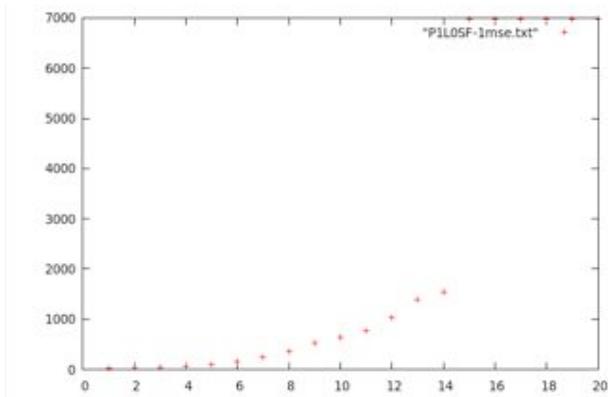
$p = 1$ Et $\lambda = 0$ Et Seuil fixe = 10



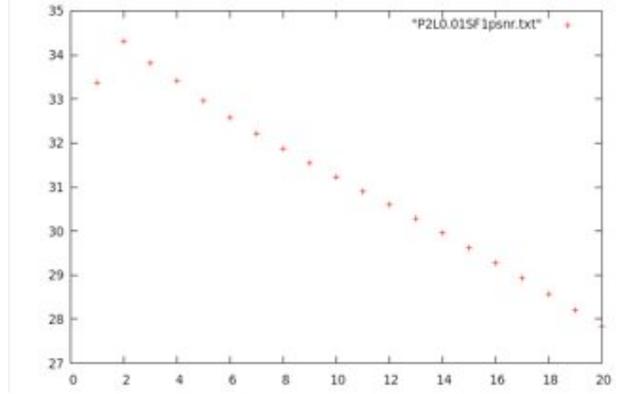
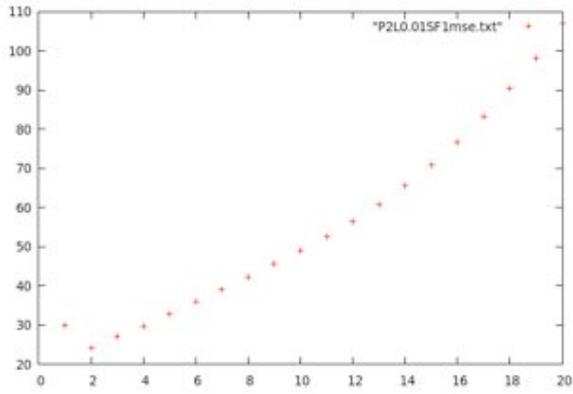
$p = 1$ Et $\lambda = 0,01$ Et Seuil évolutif = 1 + +



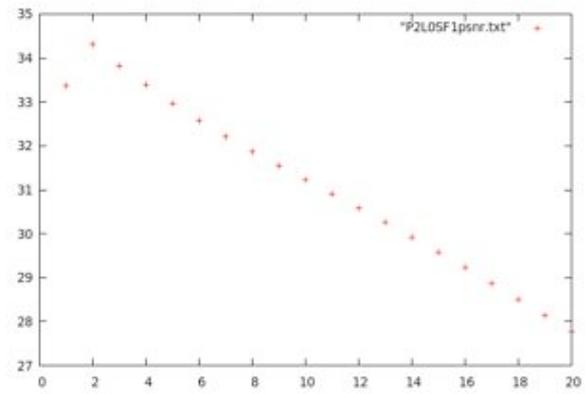
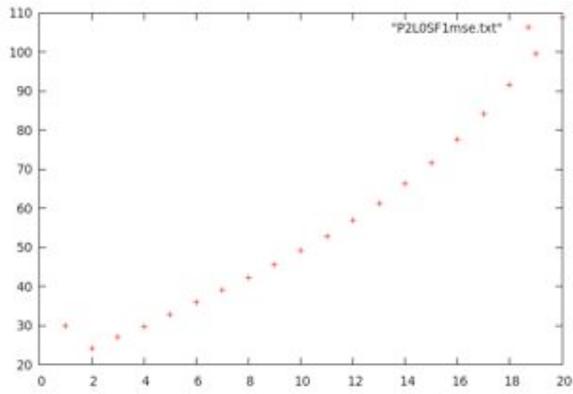
$p = 1$ Et $\lambda = 0$ Et Seuil évolutif = 1 + +



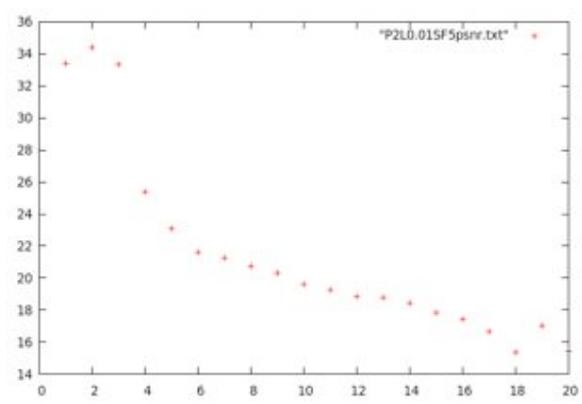
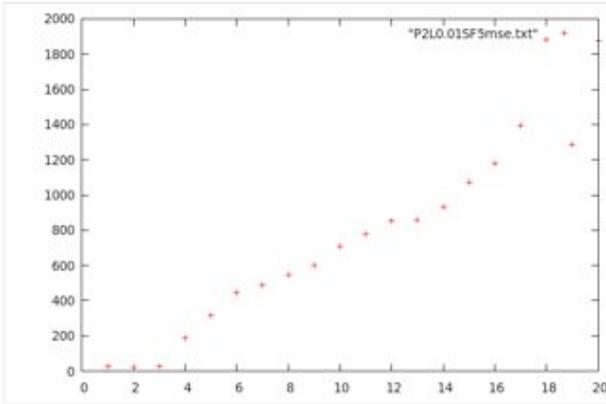
$p = 2$ Et $\lambda = 0,01$ Et Seuil fixe = 1



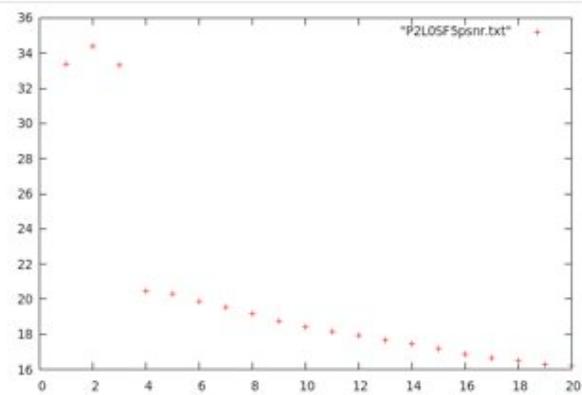
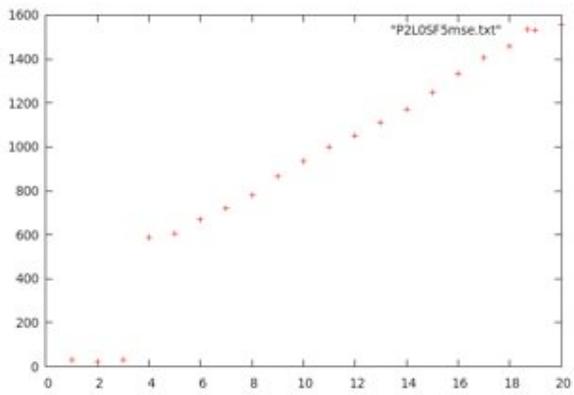
$p = 2$ Et $\lambda = 0$ Et Seuil fixe = 1



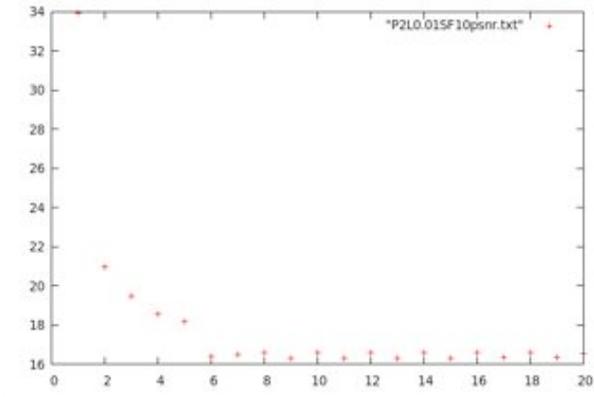
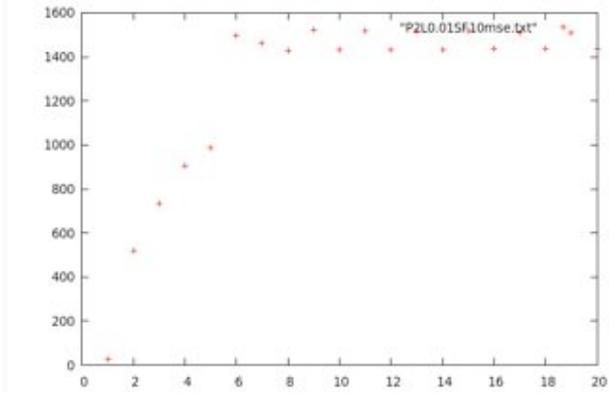
$p = 2$ Et $\lambda = 0,01$ Et Seuil fixe = 5



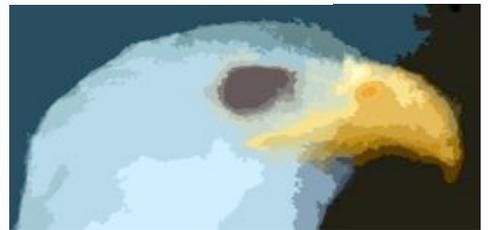
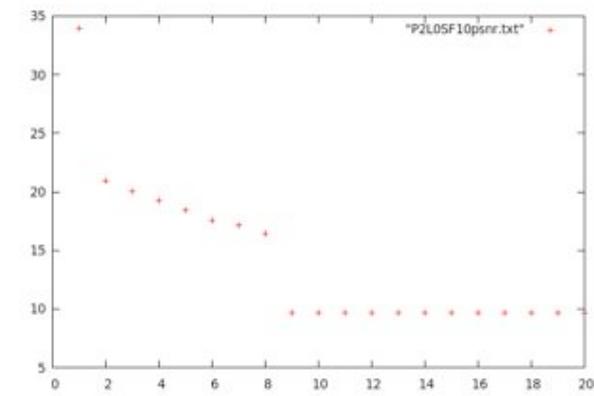
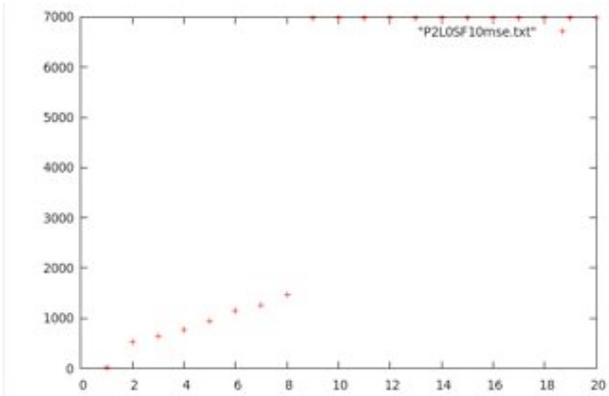
$p = 2$ Et $\lambda = 0$ Et Seuil fixe = 5



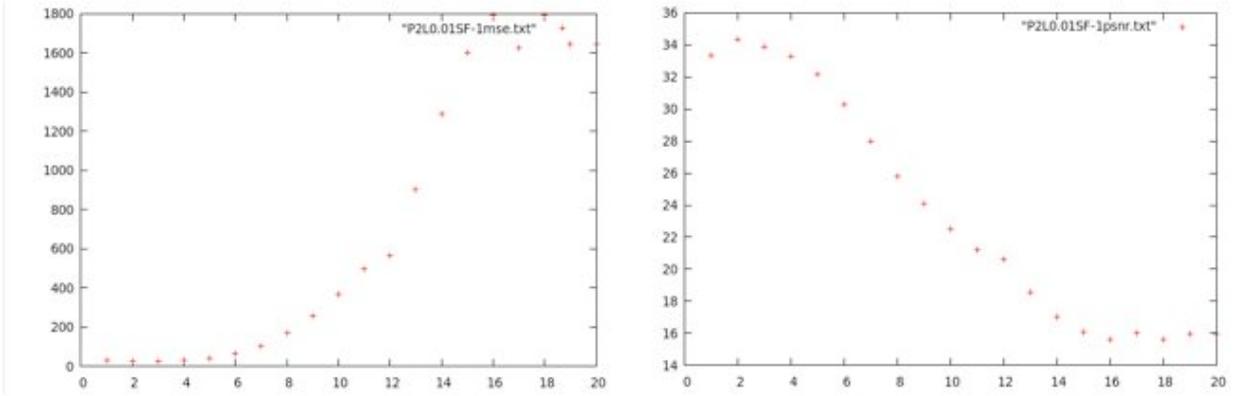
$p = 2$ Et $\lambda = 0,01$ Et Seuil fixe = 10



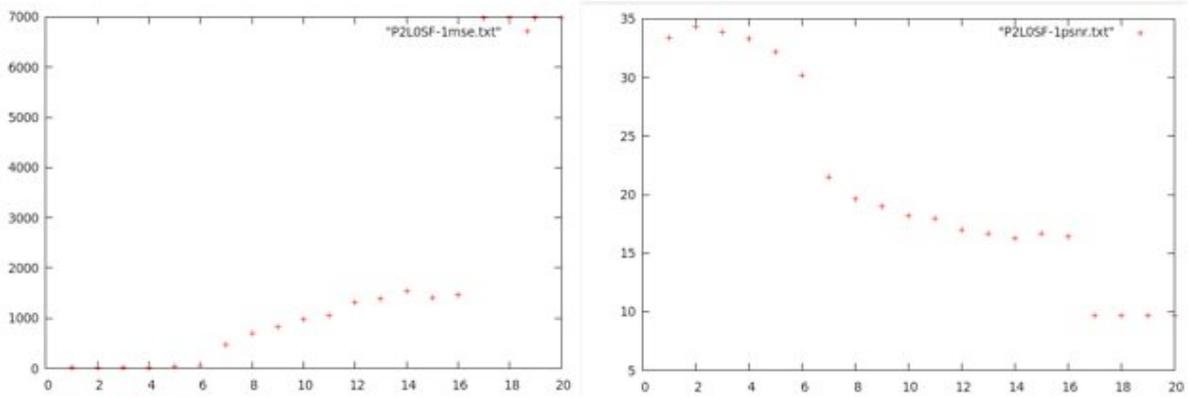
$p = 2$ Et $\lambda = 0$ Et Seuil fixe = 10



$p = 2$ Et $\lambda = 0$ Et Seuil évolutif = 1 + +



$p = 2$ Et $\lambda = 0$ Et Seuil évolutif = 1 + +



V. Conclusion

La régularisation discrète basée sur une géométrie différentielle sur graphe, a naturellement comme résultat un filtrage, une simplification, une restauration ou même une segmentation de l'image à la fin du traitement. Cette régularisation mène à une famille de filtres anisotropiques linéaires et non-linéaires. Les capacités de cette approche est équivalente aux approches classiques simples et rapides.

Nous avons montré aussi les différentes régularisations qui se font en local, non-local et semi-local sur des images et des variétés. Cela est du à la topologie qui peut être arbitraire. Nous pouvons naturellement effectuer une régularisation locale, semi locale ou non locale en modifiant simplement la connexité du graphe. Et la résolution du problème est simple. Notant bien que chaque type de régularisation est spécifique aux données ; image ou objet, un filtrage selon un modèle non local permet de mieux préserver les informations textures. Ainsi, la fonction de poids est aussi plus ou moins spécifique au type de bruit (ex. Impulsionnel ou Gaussien).

Les perspectives de ces travaux sont nombreuses. Tout d'abord, le filtrage de variétés telles que des bases de données (d'images ou non) présente un intérêt certain qui n'a pratiquement jamais été exploré en fouille de données où les données sont considérées brutes et jamais filtrées.

Cependant, si l'on considère le traitement d'image, la simplification est un pré-traitement important facilitant la segmentation. On peut donc supposer que la simplification de bases de données devrait également faciliter leur découpage. Donc, grâce a cette a approche combinée à l'analyse multi-résolution, on peut dors et déjà entamer d'autre problèmes de vision tels que la coalescence colorimétrique (connexité dans l'espace colorimétrique), les contours actifs, la retouche d'image, la décomposition structure-texture, et cela dorénavant même en grande dimension.

Références

- [AUBERT02] G. AUBERT & P. KORNPORST. « Mathematical Problems in Image Processing ». Springer-Verlag, 2002.
- [BCC+95] P. Bolon, J.-M. Chassery, J.-P. Cocquerez, D. Demigny, C. Graffigne, A. Montanvert, S. Philipp, R. Zédoudj, and J. Zérubia. *Analyse d'images, filtrage et segmentation*. Masson, 1995.
- [BENSOU05] A. BENSOUSSAN & J.-L. MENALDI. « Difference Equations on Weighted Graphs ». *Journal of Convex Analysis*, vol. 12, pages 13–44, 2005.
- [BUADES05A] A. BUADES, B. COLL & J. MOREL. « A non local algorithm for image denoising ». Dans *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, vol. 2, pages 60–65, 2005.
- [BUADES05B] A. BUADES, B. COLL & J. MOREL. « A review of image denoising algorithms, with a new one ». *Multiscale Modeling and Simulation (SIAM interdisciplinary journal)*, 4(2), pages 490–530, 2005.
- [BRONSH97] I. BRONSHTEIN & K. SEMENDYAYEV. « Handbook of Mathematics ». New York : Springer-Verlag, 3rd ed. édition, 1997.
- [CHAN01A] T. CHAN, S. KANG & J. SHEN. « Total Variation Denoising and Enhancement of Color Images Based on the CB and HSV Color Models ». *J. of Visual Communication and Image Representation*, vol. 12, pages 422–435, 2001.
- [CHAN01B] T. Chan, S. Osher, et J. Shen. The digital tv filter and nonlinear denoising. *IEEE Transactions on Image Processing*, 10 :231–241, 2001.
- [CHAN05B] T. CHAN & J. SHEN. « Image Processing and Analysis - Variational, PDE, wavelet, and stochastic methods ». SIAM, 2005.
- [CHAMBO97] A. CHAMBOLLE & P.-L. LIONS. « Image recovery via total variation minimization and related problems ». *Numerische Mathematik*, 76(2), pages 167–188, 1997.
- [DIESTE05] R. DIESTEL. « Graph Theory », vol. 173. Springer-Verlag, 2005.
- [CHAN05B] T. CHAN & J. SHEN. « Image Processing and Analysis - Variational, PDE, wavelet, and stochastic methods ». SIAM, 2005.
- [GREENB97] A. GREENBAUM. « Iterative Methods for Solving Linear Systems », vol. 17 de *Frontiers in Applied Mathematics*. SIAM, Philadelphia, 1997.
- [GILBOA07] G. GILBOA & S. OSHER. « Nonlocal Linear Image Regularization and Supervised Segmentation ». *SIAM - Multiscale Modeling and Simulation (MMS)*, vol. to appear, 2007.

- [HAGEMA81] L. HAGEMAN & D. YOUNG. « Applied Iterative Methods ». New York : Academic Press, 1981.
- [KERVRA06] C. KERVVANN & J. BOULANGER. « Unsupervised Patch-Based Image Regularization and Representation. ». Dans ECCV, vol. 4, pages 555–567, 2006.
- [KINCAI02] D. KINCAID & W. CHENEY. « Numerical Analysis : Mathematics of Scientific Computing ». Brooks/Cole, Pacific Grove, 3rd edition édition, 2002.
- [LEZORAY06] O. Lezoray, A. Elmoataz, S. Bogleux, Régularisation sur graphe pour le traitement d'images couleur, CORESA 2006, pp. 236-241, Caen
- [LEZORAY07] O. Lezoray, S. Bogleux, and A. Elmoataz. Parameterless discrete regularization on graphs for color image filtering. In ICIAR (International Conference on Image Analysis and Recognition), volume Lecture Notes in Computer Science 4633, pages 46–57, 2007.
- [LUKAC06B] R. LUKAC & K. PLATANIOTIS. « A taxonomy of color image filtering and enhancement solutions ». Dans P. HAWKES, rédacteur, *Advances in Imaging and Electron Physics*, vol. 140, pages 187–264. Elsevier, 2006.
- [Mal89] S. Mallat. A theory for multiresolution signal decomposition : The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7) :674–693, July 1989.
- [MALLAT98] S. MALLAT. « A Wavelet Tour of Signal Processing ». Academic Press, San Diego, 1998
- [NOCK04] R. Nock et F. Nielsen. Statistical region merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11) :1452–1458, 2004.
- [PERONA90] P. PERONA & J. MALIK. « Scale-space and edge detection using anisotropic diffusion ». *IEEE Transactions on Pattern Analysis Machine Intelligence.*, vol. 12, pages 629–639, 1990.
- [PEYR07] G. PEYRÉ. « Manifold Models for Signals and Images ». preprint 17, ceremade, 2007.
- [RUDIN92] L. RUDIN, S. OSHER & E. FATEMI. « Nonlinear total variation based noise removal algorithms ». *Physica D*, vol. 60, pages 259–268, 1992.
- [SAPIRO96] G. Sapiro et D. Ringach. Anisotropic diffusion of multivalued images with applications to color filtering. *IEEE Transactions on Image Processing*, 10(5) :1582–1586, 1996.
- [SMITH97] S. M. SMITH & J. M. BRADY. « SUSAN—A New Approach to Low Level Image Processing ». *Int. J. Comput. Vision*, 23(1), pages 45–78, 1997.

- [SZLAM06] A. SZLAM, M. MAGGIONI & R. COIFMAN. « A general framework for adaptive regularization based on diffusion processes on graphs ». Rapport technique, Yale, 2006.
- [TANG01] B. TANG, G. SAPIRO & V. CASELLES. « Color image enhancement via chromaticity diffusion ». IEEE Transactions on Image Processing, vol. 10, pages 701–707, 2001.
- [TOMASI98] C. TOMASI & R. MANDUCHI. « Bilateral Filtering for Gray and Color mages. ». Dans ICCV, pages 839–846, 1998.
- [TSCHUM05] D. TSCHUMPERLÉ & R. DERICHE. « Vector-Valued Image Regularization with PDEs : A Common Framework for Different Applications ». IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(4), pages 506–517, 2005.
- [VONLUX06] U. VON LUXBURG. « A Tutorial on Spectral Clustering ». Rapport technique 149, Max Planck Institute for Biological Cybernetics, 2006. To appear in Statistics and Computing.
- [WEICKE99] J. WEICKERT. « Coherence-enhancing diffusion of colour images. ». Image Vision Comput., 17(3-4), pages 201–212, 1999.
- [YAROSL85] L. P. YAROSLAVSKY. « Digital Picture Processing—an Introduction ». Springer, Berlin,, 1985.
- [ZHOU04] D. ZHOU & B. SCHOLKOPF. « A Regularization Framework for Learning from Graph Data ». Dans ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields, pages 132–137, 2004.