

Segmentation par approche contours

Tuteurs :

Romain RAVEAUX



SOMMAIRE

1.	Introduction-----	3
2.	Objectifs du projet ALPAGE-----	4
3.	Définitions-----	6
4.	Les différentes approches de segmentation-----	8
5.	Segmentation par approche contours-----	13
6.	Résultats de la segmentation pour le projet ALPAGE -----	23
7.	Fusion de régions (graphes de voisinages)-----	28
8.	Evaluation de la segmentation -----	31
9.	Bilan et autres approches -----	37
	Conclusion -----	40
	Bibliographie -----	41
	Annexes -----	42

1. Introduction

Dans le cadre du projet de l'année, ce document a pour but de présenter la segmentation d'images par approche contours et plus particulièrement à l'aide du gradient de DiZenzo. Ce projet est la suite du projet de l'an dernier qui était basé sur la détermination d'Espaces Hybrides Décorrélés (EHD) et cela appliqué sur des plans cadastraux parisiens du 14^{ème} au 19^{ème} siècle (projet ALPAGE). Par conséquent, le but est de réaliser une segmentation par approche contours sur ces mêmes plans cadastraux afin de pouvoir déterminer des zones automatiquement (régions). Pour mettre en évidence ce projet, nous respecterons le plan suivant, à savoir :

- une présentation du projet ALPAGE
- une présentation de la segmentation d'images avec la méthode de DiZenzo
- la fusion d'informations (graphe de voisinage)
- une évaluation de la segmentation

2. Objectifs du projet ALPAGE

Le projet ALPAGE [1][2](AnaLyse diachronique de l'espace urbain Parisien : Approche Géomatique) est un projet qui a pour mission de développer les recherches concernant l'espace urbain de Paris au 14^{ème} et au 19^{ème} siècle. Le projet ALPAGE est un projet pluridisciplinaire regroupant historiens et géographes spécialisés dans différents domaines, mais aussi, des informaticiens spécialisés dans le traitement de l'image. Ce projet est dirigé par Jean-marc OGIER et regroupe plusieurs laboratoires comme le LAMOP (Université de Paris I Panthéon Sorbonne, UMR 8589) ou l' OTELO (université de la Rochelle).

ALPAGE se positionne dans le cadre d'une valorisation du patrimoine Français, plus précisément, une mise en exergue des plans cadastraux de la ville de Paris issus de l'atlas VASSEROT du 19^{ème} siècle. Afin de rendre exploitable l'information spatiale contenue dans les documents cadastraux, il est nécessaire d'individualiser informatiquement des objets cohérents en l'occurrence les rues, les îlots, les parcelles pour ensuite insérer ces éléments dans un système d'information géographique (SIG). Ces étapes d'extraction de connaissance s'appuient sur les avancées de domaines scientifiques tel que la reconnaissance de formes, la vectorisation d'objets linéaires, le traitement d'images couleur : les espaces de représentation, l'analyse d'histogrammes couleur 3D, les invariants colorimétriques et les signatures dynamiques pour le suivi spatial et/ ou temporel d'objets.

Les objectifs scientifiques du projet ALPAGE sont multiples:

- Construire des outils de reconnaissance des formes innovants et adaptés aux plans cadastraux anciens
- Produire des états des lieux de l'espace urbain parisien en fonction d'une échelle variable
- Intégrer la dimension géographique et physique dans les relations sociétés/milieux
- Utiliser des modèles explicatifs pour expliquer la répartition géographique des objets
- Analyser la morphologie du parcellaire à l'échelle de la ville

C'est d'ailleurs cette morphologie parcellaire des plans cadastraux qui va nous intéresser dans le projet ALPAGE et plus précisément dans la segmentation par une approche contours car chaque plan cadastral est divisé en différentes parcelles qui sont elles-mêmes séparées à l'aide de contours noirs.

Voici un exemple de plan cadastral :

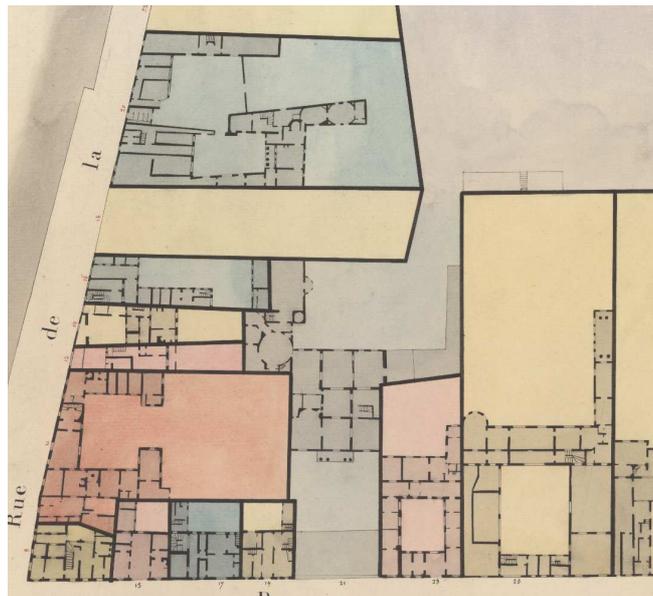


Figure 1 : Ref.tif

Sur cette image, on voit bien les parcelles mentionnées auparavant, et on peut constater qu'elles sont bien toutes entourées et séparées les unes des autres par des contours noirs qui vont faciliter la détection de contours lors de la segmentation.

Cependant, il est à noter que ces plans cadastraux sont des documents très anciens et donc altérés par le temps. Par conséquent, il est d'abord nécessaire de les numériser pour pouvoir faire un travail automatique dessus et d'appliquer ce que l'on appelle les pré traitements afin d'éviter certains problèmes comme l'affadissement par exemple, avant de faire l'étape de la segmentation.

Cette étape de pré traitement est vraiment importante car elle va permettre de réduire le nombre très élevé de couleurs exprimées dans l'image à analyser et de former les premières coalitions de pixels ayant des couleurs semblables. Le but final de cette étape de pré traitements est de former des régions compactes de pixels ayant des caractéristiques colorimétriques similaires afin de pouvoir les traiter par paquets et non plus individuellement. Ces régions serviront alors de base pour l'étape de la segmentation.

Lors du projet de l'année dernière, des pré traitements étaient déjà effectués sur les plans afin de pouvoir déterminer un Espace Hybride Décorrélé. Ces pré traitements étaient les suivants :

- théorie du monde blanc
- anti-affadissement
- extension dynamique

On pourrait également ajouter la technique dite du clustering.

3. Définitions

3.1. Définition d'une image

Nous avons déjà vu en exemple d'image que nous allons traiter lors de ce projet, mais il est nécessaire de définir ce qu'est une image numérique avant d'aborder la notion de segmentation. En réalité, une image numérique (ou numérisée) est constituée de zones rectangulaires comme on peut le voir ci-dessous [3]:

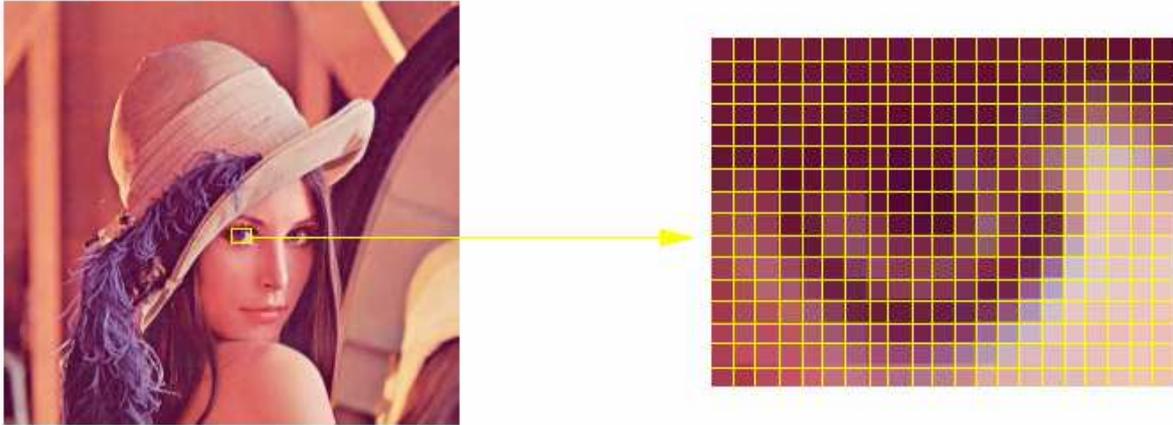


Figure 2 : Image numérique

Ici, après un zoom sur l'œil de Lena, on peut voir cette composition de l'image en zones rectangulaires appelées des pixels.

3.2. Définition d'un pixel

Un pixel est l'unité de base d'une image numérique. Il constitue l'unité minimale adressable par le contrôleur vidéo. A chaque pixel est associée une couleur, elle-même décomposée en 3 composantes primaires qui sont le Rouge, le Vert et le Bleu (RGB).

Les pixels ont une forme rectangulaire proche du carré comme le montre le schéma suivant :

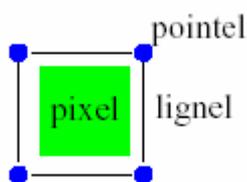


Figure 3 : illustration d'un pixel

Ici, la couleur associée est le vert et on voit bien la forme rectangulaire voire carrée de cet élément de base d'une image numérique.

3.3. Définition de la segmentation

La segmentation consiste à partitionner l'image étudiée en régions disjointes avec des couleurs homogènes. La segmentation peut être basée sur différents critères comme donc la couleur, la texture ou les niveaux de gris d'une image.

Dans le cadre du projet ALPAGE le critère est la couleur des différentes parcelles.

Voici un exemple de segmentation d'images permettant de montrer ce que l'on entend par régions homogènes [4] :

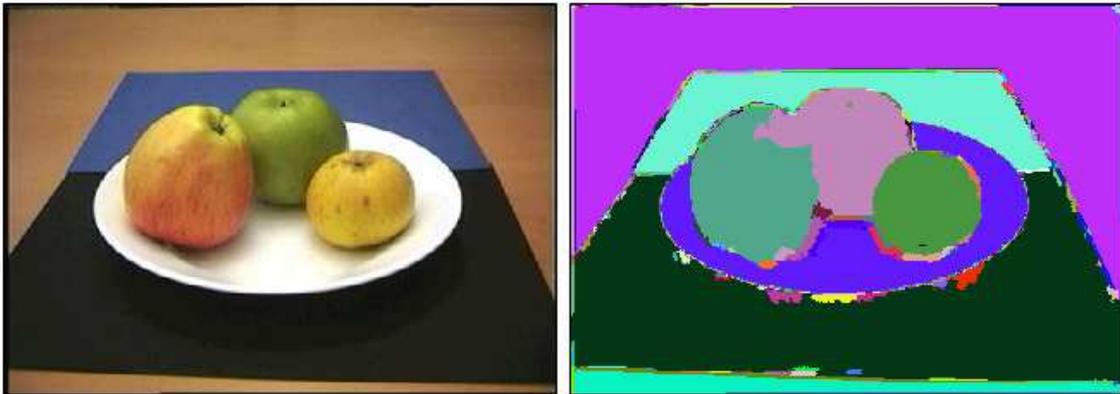


Figure 4 : Segmentation d'une image couleur

Sur cette image, on voit bien que chaque objet de l'image se voit attribué une couleur, d'où la séparation en régions dites homogènes. Cependant on voit qu'il peut y avoir des défauts de reconnaissance et donc y avoir des confusions entre les régions comme c'est le cas ici entre la pomme du milieu et celle se situant à sa gauche car elles ont une zone en violet commune.

3.4. Définition d'une région

La notion de région dans le traitement d'images, comme évoquée ci-dessus, est de regrouper des zones possédant les mêmes caractéristiques (dans notre cas, la couleur). C'est-à-dire que si plusieurs pixels adjacents s'avèrent être de couleur identique alors la zone qu'ils forment est une région. Ci-après, une illustration montrant deux régions de pixels différentes :

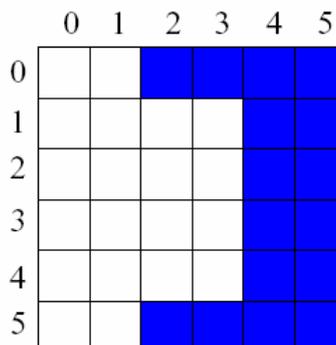


Figure 5 : Différentes régions d'une image

4. Les différentes approches de la segmentation

Il existe plusieurs types d'approches pour réaliser une segmentation. En effet, ce nombre d'approches s'élève au nombre de trois, à savoir :

- les approches globales (histogrammes).
- les approches locales (croissance de régions, détection de contours).
- les approches hybrides (split and merge, CSC).

L'approche qui nous intéresse directement ici est l'approche par détection de contours qui est une approche locale, mais avant de développer cette dernière, nous allons nous focaliser tout d'abord sur les approches globales, puis les approches hybrides afin d'avoir une présentation complète de la segmentation.

4.1. Approche globale de la segmentation

Le principe de ce type de segmentation est de réaliser une segmentation sur un espace dérivé de l'image elle-même et d'en extraire par exemple une couleur. C'est une approche dite globale car l'histogramme calculé est basé sur toute l'image et chaque couleur correspond à un pic à l'intérieur de ce dernier comme l'illustration ci-dessous peut le mettre en évidence :

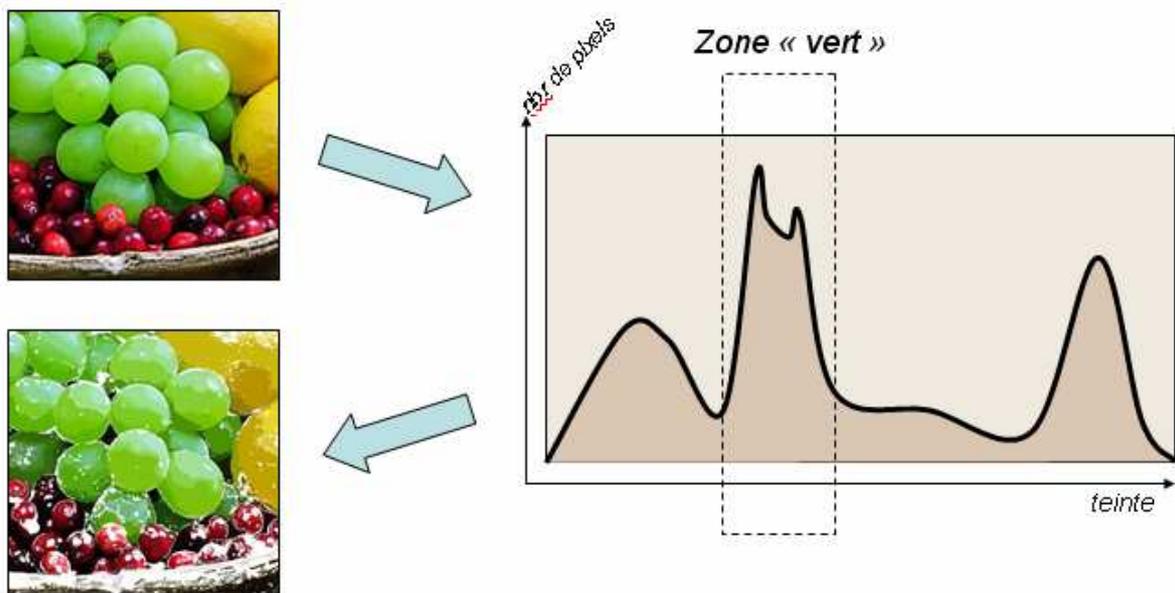


Figure 6 : Histogramme de la répartition des différentes zones couleurs

Ici, on voit bien la zone de la couleur verte sur l'histogramme qui correspond, en effet, à un pic.

Ensuite, chaque pixel de l'image est décrit selon un canal d'un espace couleur. Par exemple, pour l'espace RGB le pixel est projeté dans le canal R, puis le G et enfin le B. Cela est exactement pareil si on choisit l'espace YUV par exemple.

Ainsi, on a les différentes étapes suivantes pour parvenir à isoler la couleur verte de cette image :

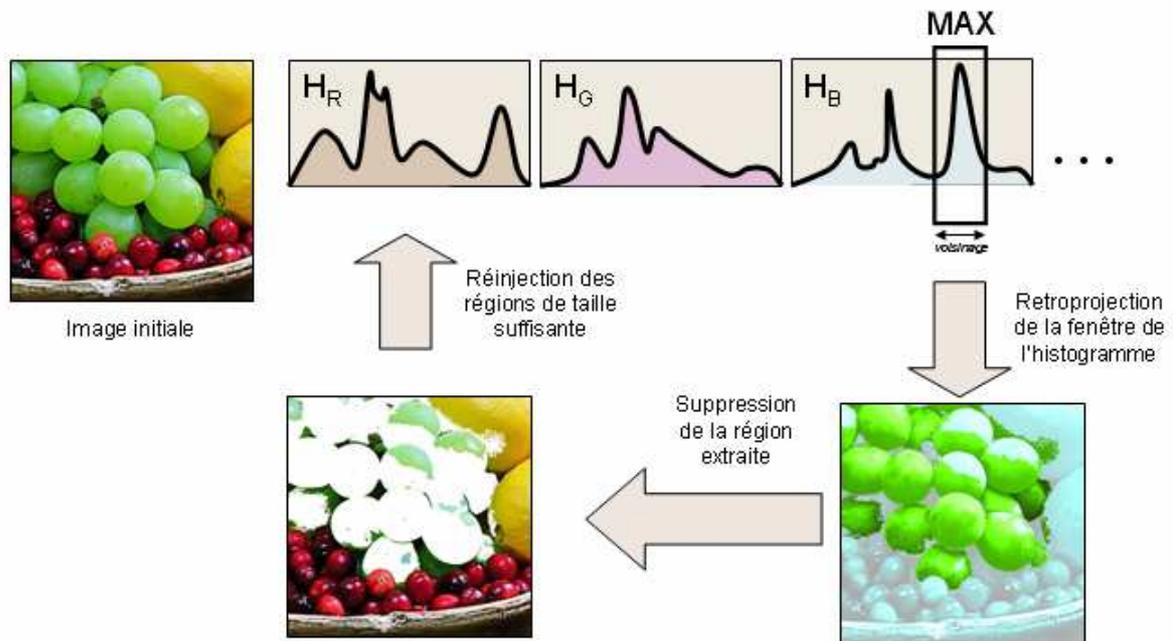


Figure 7 : Etapes à suivre pour la segmentation par histogramme

Ici, on voit comment isoler la couleur verte avant de réinjecter dans l'image les autres couleurs à extraire.

Passons maintenant à une autre approche de la segmentation dite la segmentation par approche hybride.

4.2. Approche hybride de la segmentation

Il existe plusieurs méthodes d'approche hybride de la segmentation mais nous n'allons nous focaliser que sur une seule, à savoir la méthode du split and merge évoquée précédemment.

L'idée de cette méthode qui se divise en deux phases, le split et le merge, est de regrouper les pixels de l'image originale en zones homogènes pré calculées sur l'image.

Tout d'abord, voyons la première phase de cette méthode, le **split** :

Pour cette phase, l'image est stockée dans un arbre avec Initialement :

arbre racine = image complète

Récursivement, chaque feuille F est subdivisée en quatre si elle n'est pas assez homogène, et les quatre sous images sont ajoutée en tant que feuilles de F, comme on peut le voir ci-dessous :

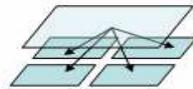


Figure 8 : Subdivision d'une feuille

L'algorithme poursuit tant qu'il reste des feuilles non homogènes à diviser jusqu'à arriver à la fin et d'avoir une structure du type :

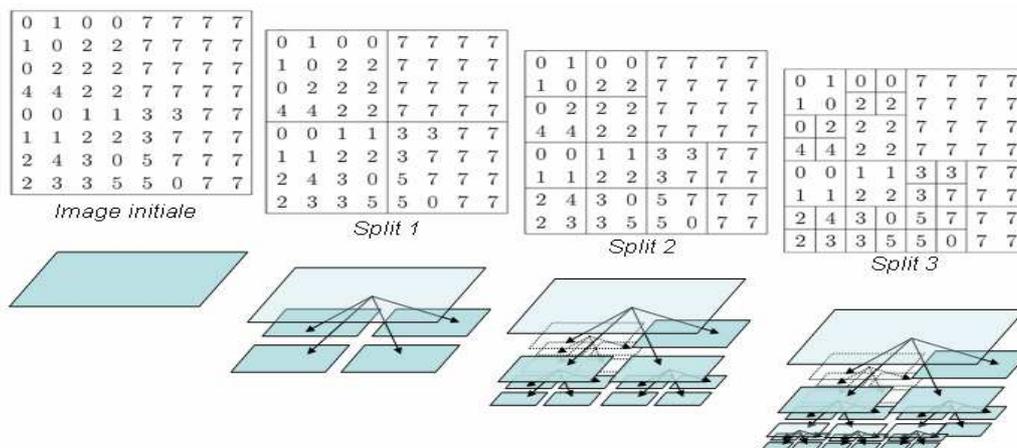


Figure 9 : Subdivisions successives jusqu'à convergence

L'étape suivante consiste à créer un graphe d'adjacence traduisant les découpages réalisés auparavant et qui servira par la suite dans la phase du merge :

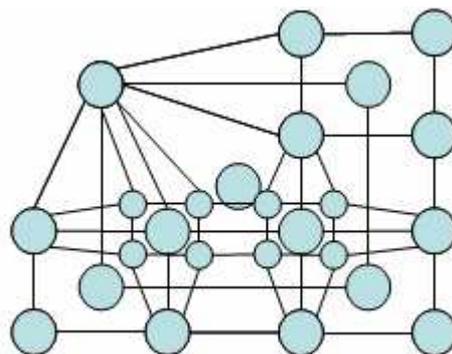


Figure 10 : Region Adjacency Graph

Une fois le graphe d'adjacence déterminé, on peut passer à la phase du merge :

- Chaque nœud du Region Adjacency Graph (graphe d'adjacence) est examiné.
- Si un des voisins de ce nœud est à une distance inférieure à un seuil de regroupement, les deux nœuds fusionnent dans le RAG.

- Lorsque plus aucun nœud ne peut fusionner avec l'un de ses voisins, alors cela entraîne une condition d'arrêt pour l'algorithme comme l'illustre le schéma ci-dessous :

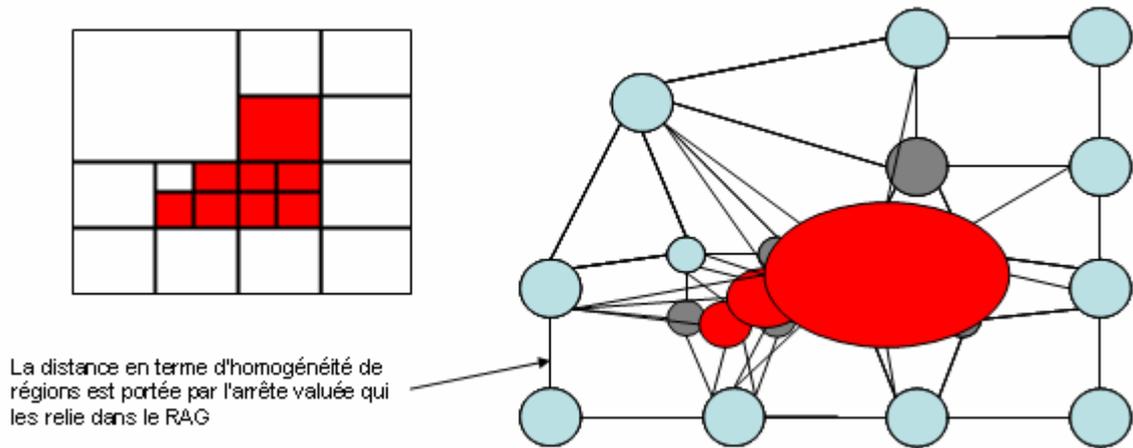


Figure 11 : Fusion de nœuds du RAG

Ici, on voit bien les étapes de fusion avec des voisins (représenté en rouge).

Cette méthode comporte certains avantages comme celui de s'affranchir du problème du gradient, comme nous pourrons le voir dans l'approche locale de la segmentation, mais a pour défaut principal d'être complexe à mettre en œuvre et de provoquer des découpages trop carrés qui sont dus à la nature des graphes.

Voici une illustration mettant en évidence ce dernier problème :



Figure 12 : Résultat du Split and Merge sur une image donnée

En effet, on peut constater que sur l'image ayant « subie » le split and merge, une zone carrée est apparue sur l'image près des panneaux de signalisation alors qu'elle n'apparaissait pas sur l'image originale.

Passons maintenant aux approches locales de la segmentation.

4.3. Approche locale de la segmentation

Il existe plusieurs types d'approches locales de segmentation comme la détection de contours ou l'approche par croissance de régions. Dans cette partie, nous verrons succinctement l'approche par croissance de régions car la détection de contours sera développée de manière plus approfondie dans la partie suivante.

Ainsi, nous nous focalisons sur la segmentation en croissance de régions.

L'idée de ce type de segmentation est de partir d'un point d'amorce (appelé seed) que l'on étend en ajoutant les points de la frontière qui satisfont le critère d'homogénéité afin de constituer des régions.

Voici une illustration montrant ce principe de croissance de régions :



Figure 13 : Principe de la croissance de régions

Il existe deux types de méthodes de croissance de régions, à savoir :

- la méthode par amorce.
- la méthode linéaire.

Cette méthode de croissance par régions a pour avantage d'être rapide et conceptuellement très simple mais elle ne permet pas d'avoir une vision globale du problème et difficilement d'avoir une bonne séparabilité des régions.

En effet, dans une image il y a toujours plus ou moins de similitude entre des pixels adjacents au niveau de la couleur donc il est très difficile de trouver ces fameuses frontières, encore que dans le cas du projet ALPAGE cela s'avère sûrement différent du fait de la séparation des différentes parcelles par des contours noirs.

Par conséquent, dans notre cas la segmentation par croissance de régions peut se trouver intéressante à appliquer.

Enfin, l'algorithme est très sensible au bruit et réalise en général une mauvaise segmentation si la variation des couleurs se fait progressivement. Cet algorithme est donc sujet au problème de dégradé des couleurs ou problème du gradient.

Passons dorénavant à la méthode appliquée dans le cadre de ce projet, à savoir la segmentation par approche contours.

5. Segmentation par approche contours

5.1. Principe

La segmentation par approche contours [6] [8] s'intéresse aux contours de l'objet dans l'image. La plupart des algorithmes qui lui sont associés sont locaux, c'est-à-dire qu'ils fonctionnent au niveau du pixel.

Des filtres détecteurs de contours sont appliqués à l'image et donnent généralement un résultat difficile à exploiter sauf si les images sont très contrastées.

Les contours extraits sont la plupart du temps morcelés et peu précis, il faut alors utiliser des techniques de reconstruction de contours par interpolation ou connaître a priori la forme de l'objet recherché.

Formellement, ce type d'algorithme est proche des méthodes d'accroissement de régions fonctionnant au niveau du pixel. Ces techniques purement locales sont en général trop limitées pour traiter des images bruitées et complexes.

Cependant, ce dernier problème devrait être évité dans le cadre du projet ALPAGE, car après les pré traitements appliqués aux plans cadastraux les images se retrouvent peu bruitées et globalement peu complexes quant à leur contenu.

Dans la segmentation par approches contours, il y a deux problématiques à résoudre, à savoir :

- caractériser la frontière entre les régions :

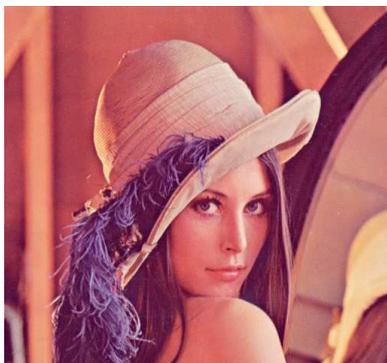


Figure 14 : Détection de contours sur Lena

- fermer les contours :



Figure 15 : Illustration de contours à fermer sur Lena

Nous allons maintenant passer au développement de cette approche.

5.2. La détection de contours

La détection de contours est une étape préliminaire à de nombreuses applications de l'analyse d'images. Les contours constituent 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).

Pour les détecter, il existe deux types d'approches :

- approche gradient : détermination des extrema locaux dans la direction du gradient.
- Approche laplacien : détermination des passages par zéro du laplacien.

Par souci de concision, nous ne développerons que les approches par gradient mais, avant tout, définissons ce qu'est le gradient d'une image.

5.3. Le gradient d'une image

Le gradient d'une image est le vecteur $\nabla I(x, y)$ défini par :

$$\nabla I(x, y) = \left(\frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y} \right)^t.$$

Il est donc caractérisé par un module m et une direction Φ dans l'image :

$$m = \sqrt{\left(\frac{\partial I(x,y)}{\partial x}\right)^2 + \left(\frac{\partial I(x,y)}{\partial y}\right)^2},$$

$$\phi = \arctan\left(\frac{\partial I(x,y)}{\partial y} / \frac{\partial I(x,y)}{\partial x}\right).$$

La direction du gradient maximise la dérivée directionnelle.

La dérivée de $I(x,y)$ dans une direction donnée d s'écrit :

$$\nabla I(x,y) \cdot d.$$

Le gradient d'une image filtrée a pour définition :

$$\nabla I'(x,y) = \nabla(I(x,y) * h(x,y)) = \nabla I(x,y) * h(x,y) = I(x,y) * \nabla h(x,y).$$

Voici une illustration montrant le gradient d'une image découpé en deux zones distinctes :

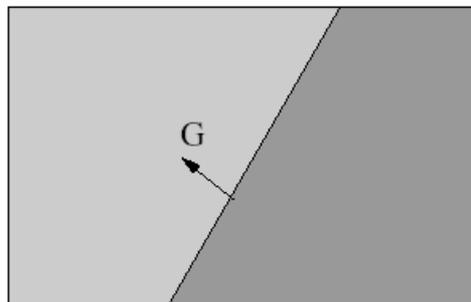


Figure 16 : Gradient d'une image à 2 régions

Le gradient n'est autre qu'une dérivée vectorielle de l'image et permettent de détecter les contours du fait que les contours correspondent à des discontinuités d'ordre 0 de la fonction d'intensité.

Cependant, le calcul de la dérivée nécessite un pré filtrage de l'image étudiée. Dans notre cas, ce pré filtrage va être réalisé à l'aide de l'opérateur de Sobel.

5.4. Pré filtrage par l'opérateur de Sobel

Pour filtrer nos images, nous devons utiliser des filtres que l'on appelle filtres étroits dont l'approche la plus courante est le filtre gradient dans lequel l'opérateur de Sobel est utilisé.

On rappelle que le gradient, en un pixel d'une image numérique, est un vecteur caractérisé par son amplitude et sa direction. L'amplitude est directement liée à la quantité de variation locale des niveaux de gris.

La direction du gradient est orthogonale à la frontière qui passe au point considéré. La méthode la plus simple pour estimer un gradient est donc de faire un calcul de variation monodimensionnelle, *i.e.* en ayant choisi une direction donnée.

On a alors le schéma suivant :

$$G_d(x,y)=(I*W_d)(x,y)$$

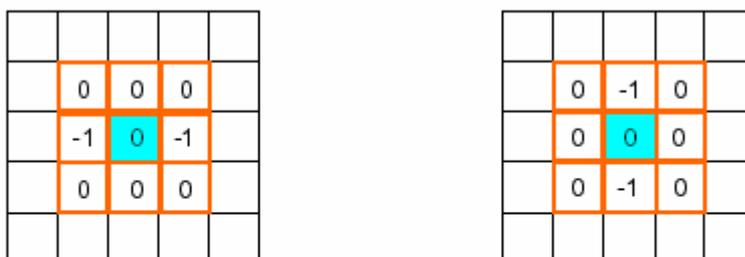
Où W_d désigne l'opérateur de dérivation dans la direction d et $*$ le produit de convolution.

Il existe de très nombreux opérateurs différents (Roberts, Sobel, Prewitt, Kirsch, ...) qui ont globalement les mêmes propriétés.

Le gradient étant un vecteur, l'approche la plus classique pour estimer le gradient consiste à choisir deux directions privilégiées (naturellement celles associées au maillage, *i.e.* ligne et colonne) orthogonales, sur lesquelles on projette le gradient.

En chaque point (x,y) de l'image, on peut donc calculer le vecteur gradient. Sa direction maximise la dérivée directionnelle et sa norme est la valeur de cette dérivée.

Appliquée à une image, considérée comme une fonction de deux variables, on peut définir deux dérivées partielles, suivant x (colonnes) et suivant y (lignes) :



Dérivée horizontale $\left(\frac{\partial I(x,y)}{\partial x}\right)$

Dérivée verticale $\left(\frac{\partial I(x,y)}{\partial y}\right)$

Figure 17 : Dérivées partielles selon x et y

C'est ici que l'opérateur de Sobel entre en jeu car la dérivation augmentant le bruit, il a été nécessaire de créer des filtres plus robustes comme celui de Sobel.

L'opérateur de Sobel, créé en 1972, est un filtre très populaire permettant d'utiliser des masques de dérivation dans les directions horizontales, verticales et obliques.

Voici d'ailleurs les 3 masques de dérivation de l'opérateur de Sobel :

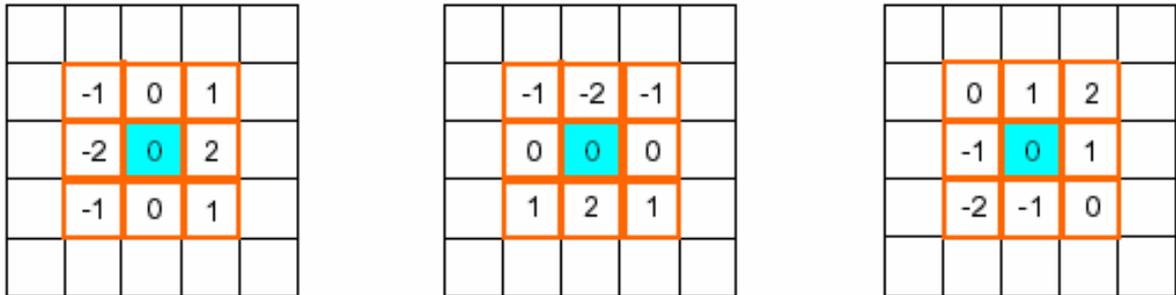


Figure 18 : Les 3 masques de dérivation de Sobel

Une fois l'image filtrée par cet opérateur, on peut alors appliquée la méthode de détection de contours que l'on a choisi. Dans notre cas il s'agit d'une approche par gradient et notamment par le gradient de DiZenno.

5.5. Gradient de DiZenno

Le gradient de DiZenno est un gradient vectoriel. Un gradient multi-composante ou gradient vectoriel prend en compte la nature vectorielle des images dans l'espace de représentation dans lequel il se trouve.

L'image doit tout de même être préalablement filtrée pour être deux fois continûment différentiable, ce qui est le cas puisque nous avons utilisé le filtre de Sobel.

Le vecteur gradient est calculé à partir de l'ensemble des composantes en recherchant la direction pour laquelle les variations sont les plus élevées. Cela s'effectue par maximisation d'un critère de distance, mesuré selon la norme L^2 .

Elles calculent donc un vecteur gradient couleur à partir de vecteurs gradients marginaux, c'est-à-dire qu'elles calculent un vecteur global RGB en calculant chaque composante séparément, le R, le G et le B et fait une fusion des 3 composantes en recherchant la direction pour laquelle les variations sont les plus grandes.

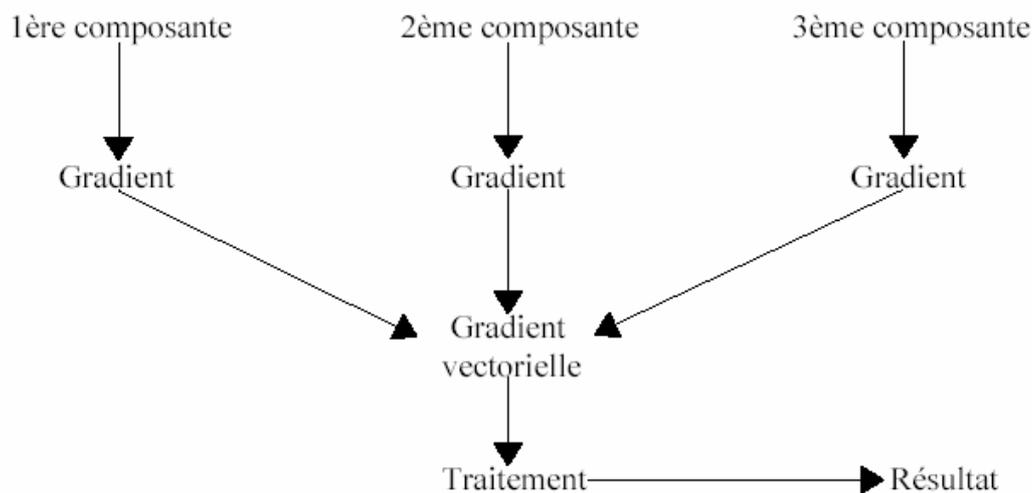


Figure 19 : Schéma des étapes de DiZenko

Entre chaque pixel voisin de l'image, on calcule la variation dI pour un déplacement élémentaire largeur - hauteur par :

$$dI = \left(\frac{\partial I}{\partial x}\right) \cdot dx + \left(\frac{\partial I}{\partial y}\right) \cdot dy$$

dI est considéré comme la norme de la distance entre les deux pixels considérés.

Le carré de la norme s'exprime alors par :

$$dI^2 = \left(\frac{\partial I}{\partial x}\right)^2 dx^2 + 2 \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} dx dy + \left(\frac{\partial I}{\partial y}\right)^2 dy^2 = a \cdot dx^2 + b \cdot dx dy + c \cdot dy^2$$

Avec :

$$a = G_{x,1}^2 + G_{x,2}^2 + G_{x,3}^2 \quad b = G_{x,1} \cdot G_{y,1} + G_{x,2} \cdot G_{y,2} + G_{x,3} \cdot G_{y,3} \quad c = G_{y,1}^2 + G_{y,2}^2 + G_{y,3}^2$$

Où :

$G_{x,1}, G_{x,2}, G_{x,3}$ représente le gradient selon le déplacement dx sur chacune des trois composantes.

On obtient alors la matrice M suivante dont les valeurs propres interviennent dans le calcul de la norme et les vecteurs propres associés traduisent les directions du gradient :

$$M = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$$

On a donc réalisé la détection des contours en réalisant une extraction des extrema locaux du gradient dans la direction du gradient. Cela est donc revenu à déterminer, pour un pixel p , les valeurs du gradient sur la droite passant par p et de direction relative à son gradient. Il suffit désormais de vérifier que le gradient en p est bien maximal localement sur cette droite, d'où le nom d'approche locale pour la détection de contours.

Voici un schéma récapitulatif de l'approche contour par le gradient de DiZenno :

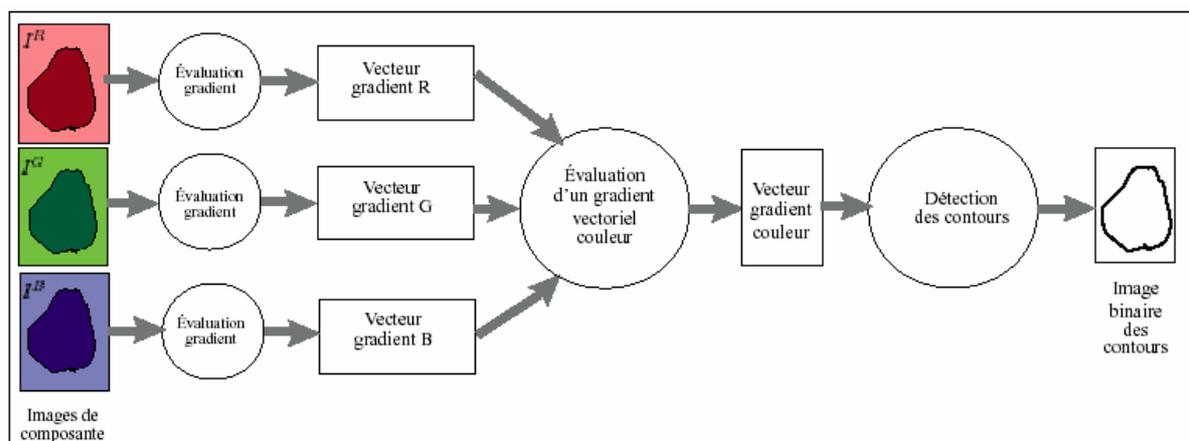


Figure 20 : Schéma récapitulatif des étapes de DiZenno

A ce stade de la segmentation, on a détecté les contours par une approche gradient. Il ne reste donc plus qu'à résoudre la deuxième problématique : la fermeture des contours.

5.6. Seuillage par hystérésis

En effet, l'objectif est de rendre connexes les contours déterminés auparavant afin de pouvoir déterminer des régions. Pour cela, on réalise un seuillage par hystérésis des extrema.

Le principe est d'utiliser deux seuils pour la norme du gradient S_b et S_h et de sélectionner les pixels pour lesquels :

- la norme du gradient est supérieure à S_b
- le pixel donné est connecté, par un chemin constitué de pixels dont la norme du gradient est supérieure à S_b , à un pixel pour lequel la norme du gradient est supérieure à S_h .

Cette notion de seuillage permet notamment, dans le cadre du projet ALPAGE, de distinguer la valeur de niveaux de gris que l'on désire pour définir les valeurs supérieures à ce seuil comme étant des contours.

Ce seuil peut permettre d'éviter une sur-segmentation, notion que nous aborderons au travers les exemples d'applications dans la suite de ce rapport.

5.7. Détermination du seuil par la méthode d'OTSU

Une fois la segmentation par le gradient de DiZenzo terminée, il est nécessaire de déterminer le seuil afin de faire de l'intérieur des contours des régions.

Avant cela, une étape de binarisation est effectuée et celle-ci correspond à la conversion d'une image en niveaux de gris (de 0 à 255) en une image binaire (0 ou 1, 0 ou 255). Dans notre cas, l'image résultat se prête bien à une étape de binarisation car ce qui nous intéresse, ce sont les contours et l'intérieur de ceux-ci sont les régions.

Cependant, pour binariser une image numérique, il est nécessaire de choisir un seuil qui déterminera à partir de quelle valeur nous avons des 0 ou des 1.

Exemple : Voici un vecteur montrant une répartition de niveaux de gris et le seuil fixé est de 205 :

Vecteur = [10 20 50 200 210 255 255] Seuil = 205

Alors, après binarisation avec ce seuil de 205, le vecteur résultant vaut :

Vect_res = [0 0 0 0 255 255 255]

Toutefois, la détermination du seuil est une tâche un peu « ingrate » à déterminer manuellement, c'est pour cela que dans le cadre du projet ALPAGE nous utilisons une méthode permettant de définir ce seuil automatiquement, à savoir la méthode d'OTSU.

La méthode d'OTSU se décompose en plusieurs étapes :

- Le calcul du niveau de gris moyen et de la variance des niveaux de gris pour l'image donnée :

$$g_T = \sum_{g=0}^L g \cdot p(g) \qquad \sigma^2 = \sum_{g=0}^L (g - g_T)^2 \cdot p(g)$$

- Le calcul de la probabilité d'apparition de chaque niveaux de gris dans l'image en vérifiant que :

$$\sum_{g=0}^L p(g) = 1$$

- En supposant que l'image soit divisée en deux classes C1 et C2 comme dans notre cas (C1 : contours, C2 : régions intra-contours) et que C1 correspond aux pixels de niveaux de gris compris entre 0 et t et C2 entre t et L, nous définissons alors :

- la probabilité d'un pixel d'appartenir à la classe C1 ou C2 :

$$\omega_1(t) = \sum_{g=0}^t p(g) \qquad \omega_2(t) = \sum_{g=t+1}^L p(g)$$

- les niveaux de gris moyens de C1 et de C2 :

$$g_1(t) = \frac{\sum_{g=0}^t g \cdot p(g)}{\omega_1(t)} \qquad g_2(t) = \frac{\sum_{g=t+1}^L g \cdot p(g)}{\omega_2(t)}$$

- les variances dans C1 et C2 :

$$\sigma_1^2(t) = \frac{\sum_{g=0}^t (g - g_1(t))^2 \cdot p(g)}{\omega_1(t)} \qquad \sigma_2^2(t) = \frac{\sum_{g=t+1}^L (g - g_2(t))^2 \cdot p(g)}{\omega_2(t)}$$

La méthode d'OTSU assimile le problème de détermination d'un seuil pertinent T pour la binarisation de l'image à un problème de meilleure classification des pixels en deux sous-groupes C_{1T} et C_{2T}.

Une des classes contient le fond et l'autre les objets dans l'image (dans notre cas, les contours).

Nous obtenons alors la valeur de niveaux de gris moyen dans l'image par :

$$g_T = \sum_{g=0}^L g \cdot p(g) = \sum_{j=1,2} \omega_j(t) \cdot g_j(t)$$

Cette valeur obtenue, il ne reste plus qu'à répartir les pixels inférieurs à ce seuil ensembles et ceux supérieurs ensembles, puis donner une valeur binaire à chaque classe.

5.8. Fermeture des contours

En général, après avoir déterminés les contours de l'image, il est nécessaire de chaîner les pixels contours afin qu'ils constituent des lignes fermées correspondant aux frontières des régions.

L'idée est donc de prolonger les segments ouverts afin qu'ils deviennent connexes et qu'ils permettent une partition en régions de l'image segmentée.

Cette prolongation de segments se fait par rapport au voisinage de chaque pixel et doit donner, on l'espère, une fermeture de contours cohérente.

Il est à noter que les méthodes de détection des pixels contours sont particulièrement bien adaptées à la segmentation d'images où les couleurs sont fortement contrastées.

Les résultats sont alors parfois décevants car les contours dans l'image peuvent être difficiles à percevoir, ce qui implique l'oubli de certaines régions.

Cependant, dans notre cas cette fermeture de contours ne s'applique pas car elle est pour ainsi dire faite, de par la nature des plans cadastraux d'ALPAGE, en même temps que l'étape de détection par le gradient de DiZenzo et la binarisation.

6. Résultats de la segmentation pour le projet ALPAGE

Après avoir défini les étapes à suivre et les moyens utilisés pour réaliser la segmentation par approche contours et méthode de DiZenzo, nous allons aborder la partie résultats afin de pouvoir évaluer, par la suite, la qualité de notre segmentation.

Nous rappelons que nous travaillons sur des plans cadastraux du type :

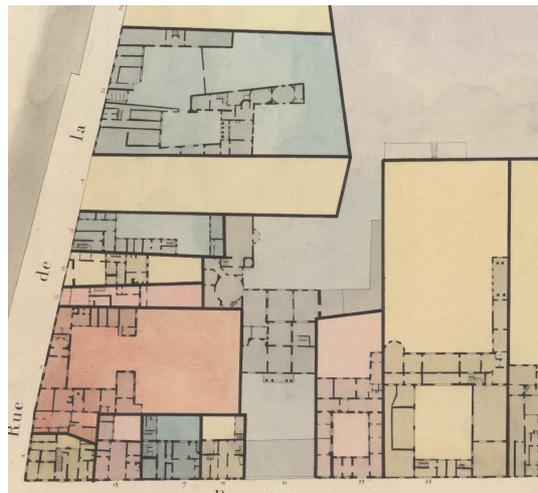


Figure 1 : Ref.tif

Cette image est l'image qui nous a servi de référence pour réaliser les différents tests et donc chronologiquement :

- les pré traitements
- le filtrage de Sobel
- l'utilisation du gradient de DiZenzo
- binarisation
- détermination du seuil à utiliser par la méthode d'OTSU
- étiquetage des composantes connexes blanches formant les régions

Dans cette partie nous ne nous attarderons pas sur les pré traitements car ils ont déjà été abordés au préalable.

Cependant, il est intéressant de voir les résultats concernant les autres étapes.

Voici ce que nous obtenons après avoir appliqué le filtre de Sobel sur l'image de référence Réf.tif :



Figure 21 : RefSobel.tif

Nous pouvons alors constater que cette image a permis de filtrer certains contours mais qu'elle nécessite un réhaussement de ces contours par l'intermédiaire du gradient de DiZeno afin de pouvoir réaliser une bonne segmentation.

L'étape suivante consiste donc à appliquer le Gradient de DiZeno sur l'image filtrée et voici ce que nous obtenons :

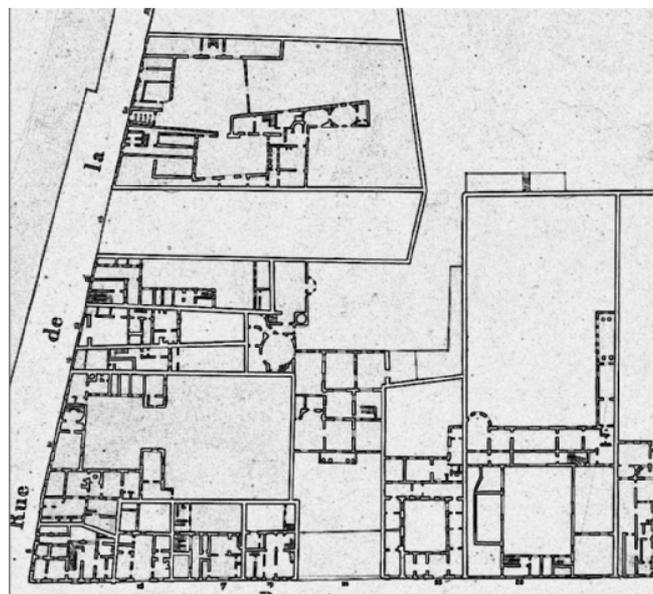


Figure 22 : RefDiZeno220.tif

Voici donc ce que nous avons après l'utilisation du gradient de DiZeno. Ici, nous voyons que nous obtenons une assez bonne détection de contours puisque les parcelles sont bien en accord avec la vérité terrain (Réf.tif) et qu'elle permet une bonne résolution sur les contours fins.

Après un zoom sur l'image **DiZenzo220.tif**, nous pouvons mettre en évidence ces zones dites bruitées :

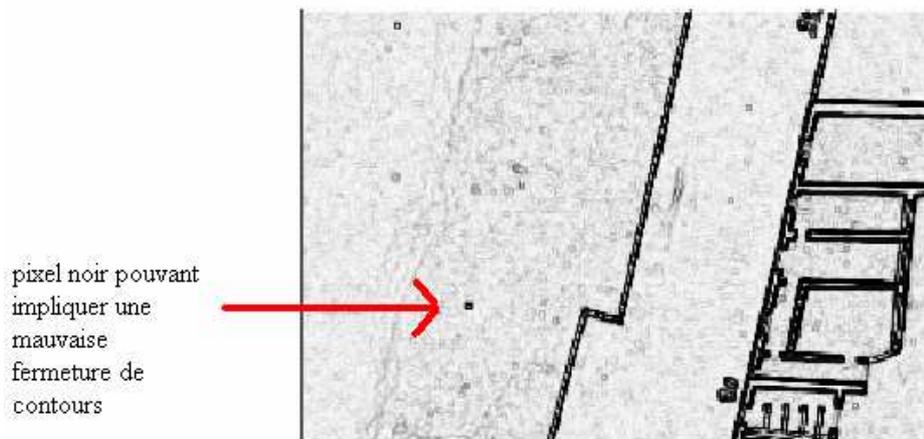


Figure 23 : Zoom sur RefDiZenzo220.tif

Sur cette illustration, nous pouvons remarquer ce pixel noir à gauche qui n'a pas été éliminé, vu la valeur du seuil choisie, et qui aurait pu gêner le bon déroulement de la segmentation.

Ceci montre bien un des défauts de la segmentation par détection de contours car elle est tributaire du seuil choisi et si ce dernier est trop faible (dans le cas d'une recherche des pixels noirs) et si les pré traitements n'ont pas assez filtrés les pixels parasites alors il est possible de rencontrer ce genre de bruit.

Nous pouvons noter que cette image résultante **RefDiZenzo220.tif** de l'analyse de DiZenzo est en niveaux de gris et à partir de celle-ci, une binarisation est effectuée.

Le seuil de la binarisation est calculé de manière automatique en utilisant la méthode d'OTSU. Sur cette image binaire, un étiquetage des contours fermés est réalisé. Chaque contour fermé est donc une région.

Dans notre cas, le seuil de niveaux de gris est de 220 sachant que les niveaux de gris sont répartis de 0 à 255. En effet, cette valeur assez élevée sur l'échelle des niveaux de gris permet alors de réaliser une segmentation cohérente avec les objectifs de la segmentation par contours et plus globalement du projet ALPAGE car les contours séparant les parcelles sont noirs.

Ainsi, le fait de choisir un seuil assez élevé permet de s'affranchir de « faux contours » ou de zones bruitées qui engendreraient une sur segmentation.

Voici l'image après l'étape de binarisation et juste avant étiquetage des des régions (composantes connexes blanches) :

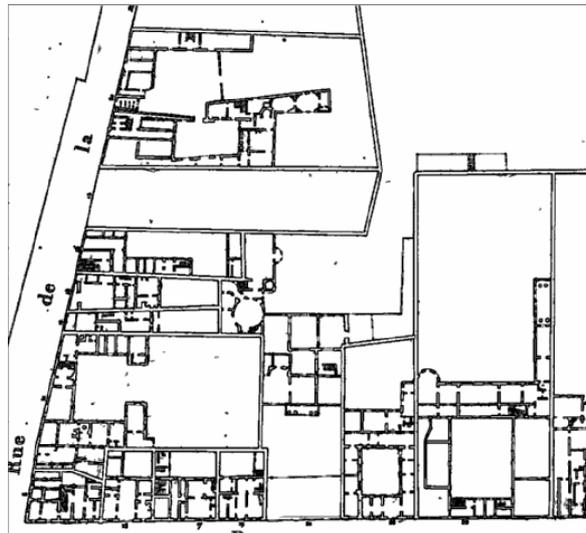


Figure 24 : RefBina.tif

La segmentation par détection de contours terminée, nous pouvons afficher les résultats obtenus pour un seuil de 220 après étiquetage des composantes connexes blanches (régions) et un affichage en fausses couleurs :

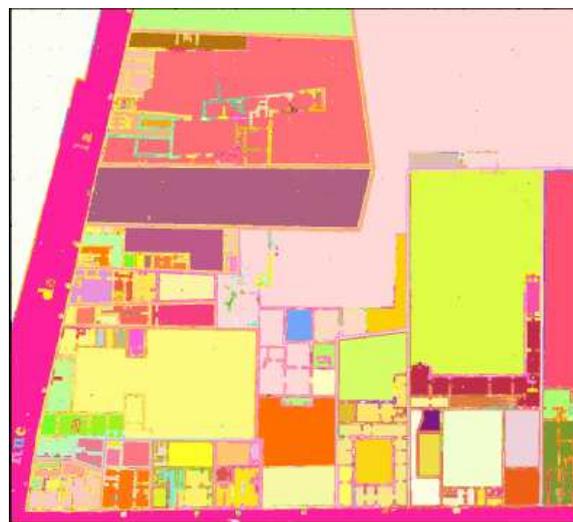


Figure 25 : RefSeg220.tif

Ici, nous pouvons voir les résultats une fois la segmentation terminée et nous pouvons penser qu'il y a d'assez bons résultats car le découpage est fait en régions s'approchant réellement de l'homogénéité. Quelques défauts subsistent mais le rendu est satisfaisant pour cette valeur de seuil.

Toujours concernant ce seuil, si la valeur de celui-ci est trop basse pour une approche qui maximise la valeur du gradient alors la partition en régions devient moins « exigeante » et trop de régions sont créés par rapport à ce que peut nous montrer la vérité terrain. Ce phénomène est appelé la sur segmentation.

Voyons une preuve de sur segmentation sur Réf.tif lorsque nous prenons un seuil de 110 en entrée :

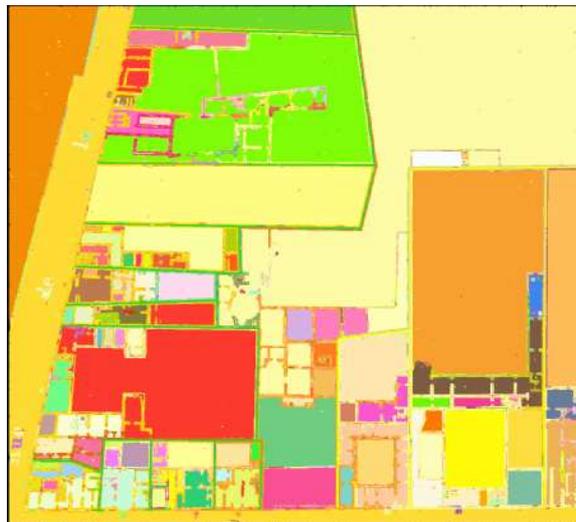


Figure 26 : RefSeg110.tif

Sur cette dernière image, nous pouvons voir que plus de régions ont été créés comme peut le montrer le comparatif suivant sur une zone précise de l'image :

Seuil de 220 :



Figure 27 : Zoom sur RefSeg220.tif

Seuil de 110 :

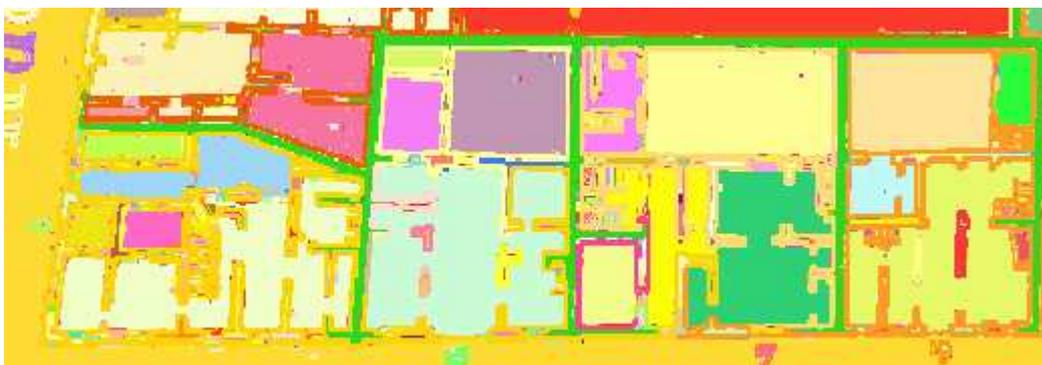


Figure 28 : Zoom sur RefSeg110.tif

En comparant les deux résultats avec des seuils différents nous constatons bien le phénomène de sur segmentation auquel nous pouvons ajouter une plus mauvaise séparation des régions ce qui est du à une moins bonne détection des contours.

7. Fusion de régions (graphe de voisinage)

7.1. Définitions

7.1.2. La fusion de régions

La fusion de régions est une étape qui vient après l'étape vue précédemment, à savoir la segmentation.

Le but de la fusion de régions est faire diminuer le nombre de régions en vue de traitements ultérieurs.

Nous rappelons qu'une région est constituée d'un ensemble de pixels connexes possédant les mêmes caractéristiques et cela selon le même critère.

Ci-dessous, voici un exemple de partition en régions qui sont, dans ce cas, selon le critère des niveaux de gris :

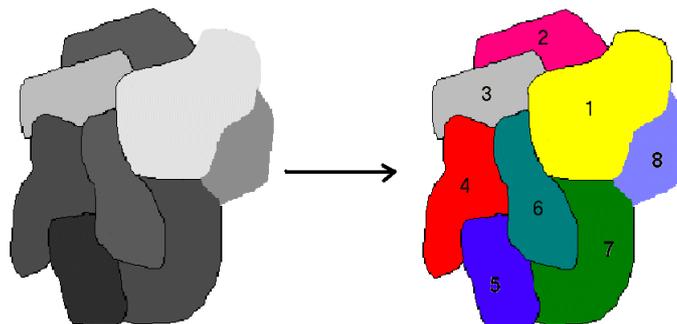


Figure 29 : D'une image en niveaux de gris vers ses régions

Dans l'exemple ci-dessus, nous décidons de vouloir faire la fusion entre les régions 1 et 6 sachant, nous le rappelons que, une fusion revient à supprimer une frontière et à faire absorber une région par l'autre.

Voici le résultat obtenu après fusion des régions 1 et 6 :

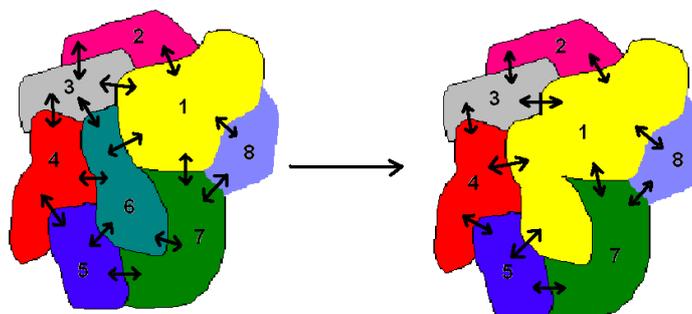


Figure 30 : Fusion de 2 régions

Ici, nous pouvons voir la fusion de régions entre deux régions adjacentes fusionnant selon un critère donné.

L'étape suivante consiste à modéliser les régions et leur étape de fusion par un graphe appelé graphe de voisinage.

7.1.3. Graphe de voisinage

Un graphe est un objet mathématique constitué d'un ensemble de sommets (ou nœuds) et d'un ensemble d'arêtes (ou arcs) reliant certains de ces sommets.

L'idée de l'utilisation des graphes dans ce projet est de modéliser les images segmentées par des graphes car ils s'avèrent adaptés à la représentation de relations entre les éléments d'une ou plusieurs images.

L'objectif des graphes de voisinages (ou graphes spatiaux) est de donner une représentation spatiale de la segmentation.

Il est à noter qu'un graphe, et plus particulièrement un graphe de voisinage, contrairement aux arbres, est non orienté et il possède des nœuds et des arcs qui ont pour caractéristiques respectives dans le cadre de la segmentation :

- Les nœuds : chaque nœud est une région
- Les arcs : Il y a un arc entre deux nœuds si une région est située dans les h plus proches voisins de l'autre région dans l'image correspondante. L'arc est non orienté et étiqueté avec la distance spatiale (x,y) entre les centres des 2 régions.

Les relations spatiales entre les régions sont communément formalisées par la notion de graphe de voisinage, et peuvent être représentées sous forme d'une matrice de voisinage. Cette dernière est une matrice binaire M où $M[i,j]=1$ si i et j sont voisins et $M[i,j]=0$ sinon. La figure ci-dessous montre la matrice de voisinage associé au graphe de voisinage suivant :

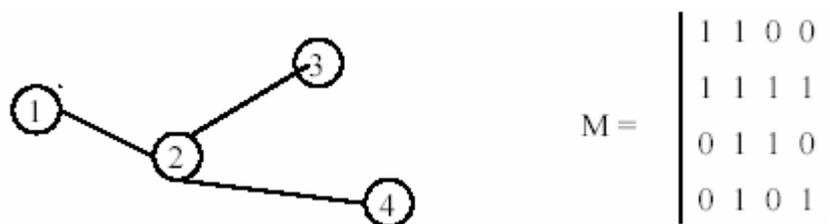


Figure 31 : Association graphe et matrice de voisinage

Ici par exemple, 2 est le voisin de tous les autres alors la ligne 2 de la matrice M est à 1.

7.2. Fusion de régions pour ALPAGE

Cette partie a pour objectif de montrer l'utilisation de graphes de voisinage dans le cadre du projet ALPAGE. Les tests ont également été effectués sur l'image Réf.tif.

Nous rappelons que pour réaliser la fusion de régions dans le cadre de ce projet, nous utilisons les graphes de voisinage [5]. Comme dit auparavant, ces graphes se décomposent en nœuds et en arcs avec chaque nœud représentant une région et chaque arc représentant une liaison entre les régions.

Cette liaison entre deux nœuds existe si une région est située dans les h plus proches voisins de l'autre région liée par l'arc.

Dans le projet ALPAGE, cette liaison entre les nœuds se calcule sur les 10 plus proches voisins, c'est-à-dire que si une région R1 est liée à une région R2 alors cela signifie que la région R2 se situe dans les 10 plus proches voisins de la région R1 au niveau de la distance entre les deux centres des régions.

Une fois, le graphe de voisinage établi (nous ne pouvons pas en donner une représentation graphique ici car de taille trop importante vu le nombre de régions), il faut donc choisir quelles sont les régions à fusionner.

Pour cela, il y a deux critères à respecter qui sont :

- la taille des régions à fusionner : en effet, si les deux régions à fusionner sont trop importantes en terme de taille alors il est peut être préférable de ne pas faire absorber une région par l'autre. Ainsi, il y a un seuil maximal de taille à respecter pour ne pas faire alors une « sous segmentation ».
- une contrainte d'homogénéité couleur : c'est-à-dire que sur notre image résultat segmentée, il ne faut pas fusionner des régions (même si le graphe de voisinage le préconiserait sans un seuil d'homogénéité couleur) qui font partie des 10 plus proches voisins si elles ne sont pas du tout homogènes.

Pour mettre en évidence l'aspect concret de ces critères, voici un agrandissement d'une partie de l'image résultante de la segmentation :



Figure 32 : Zoom2 sur RefDiZenko220.tif

Ici, les deux critères sont mis en évidence. En effet, les régions vertes et roses ne peuvent pas fusionner car de tailles trop importantes et de couleurs trop différentes alors que pourtant leurs centres de régions se retrouvent parmi leurs 10 plus proches voisins.

Ci-dessous, nous pouvons voir un comparatif entre l'image segmentée (et représentée par le graphe de voisinage) et l'image après fusion de régions :

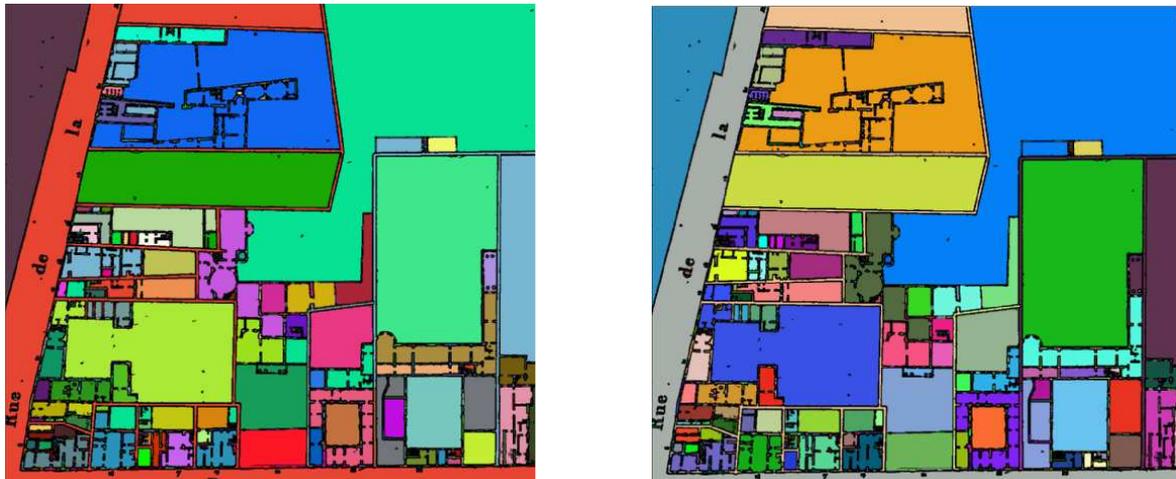


Figure 33 : Comparatif ImSeg.tif et ImGraph.tif

Sur l'image de droite, après fusion, nous pouvons voir que les régions sont plus propres et que les régions voisines entre elles ont des couleurs bien distinctes pour éviter de les confondre a lors que cela n'était pas le cas avant fusion, comme nous pouvons le constater sur la partie en haut à droite de l'image de gauche.

Toutefois, il ressort de ces images d'assez bons résultats a priori mais pour s'assurer de la bonne segmentation et de l'amélioration de cette dernière après l'étape de fusion, il serait intéressant d'utiliser des outils permettant de les évaluer de manière quantitative.

8. Evaluation de la segmentation

L'évaluation pour la segmentation [7] des images est un problème crucial. En effet, il est nécessaire de disposer de méthodes permettant d'évaluer les résultats obtenus, notamment dans un cadre aussi vaste que la segmentation. Il est très difficile de pouvoir se prononcer, si ce n'est à l'œil nu, sur la qualité d'une segmentation par rapport à une autre. Par conséquent, nous utilisons des outils quantitatifs reflétant la bonne segmentation de la méthode en fonction des critères choisis.

Il existe deux grandes classes d'évaluation de la segmentation, à savoir :

- les méthodes avec « vérité-terrain », c'est-à-dire des méthodes où la vérité est parfaitement connue concernant, dans notre cas, la segmentation qui a servi à synthétiser l'image.
- Les méthodes sans vérité-terrain, sans références sur la segmentation utilisée où sur les attributs des images testées.

8.1. Evaluation avec une vérité-terrain

Si l'on veut comparer de manière objective les méthodes, il est plus simple d'utiliser des images de synthèse, pour lesquelles une "vérité" est parfaitement connue, à savoir la segmentation qui a servi à synthétiser l'image. L'inconvénient d'une telle démarche est que ces images ne représentent pas toutes les situations possibles d'une utilisation réelle.

La pertinence d'une région déterminée lors de l'étape de segmentation est liée à l'application que nous voulons en faire. Par exemple, ce qui nous importe dans la segmentation par contours du projet ALPAGE, ce sont les contours encadrant chaque parcelle alors que nous pourrions nous focaliser sur les couleurs de ces parcelles par exemple.

Quand on dispose d'une vérité-terrain, l'évaluation des segmentations s'effectue à l'aide de critères comparant chaque segmentation avec l'image de référence (Ref.tif). Nous pouvons ainsi ordonner les segmentations.

Dans cette partie, nous allons aborder une méthode permettant d'évaluer la segmentation avec une vérité-terrain, à savoir l'utilisation du critère de Vinet.

Critère de Vinet

Nous noterons R une segmentation à évaluer, c'est donc une partition de X (ensemble de sites représentant les coordonnées spatiales des pixels) en régions notées R_i , $i = 1 \dots N$ vérifiant $R_i \cap R_j = \emptyset$ et $\bigcup_{i=1}^N R_i = X$. Nous notons A le nombre de pixels de l'image et A_i le nombre de pixels de la région R_i . Nous avons donc :

$$A = \sum_{i=1}^N A_i = \text{card } \bar{X}$$

Dans cette partie, nous disposons d'une segmentation de référence notée V , dont les régions sont notées V_i , $i = 1 \dots M$.

La mesure de Vinet s'appuie sur un appariement biunivoque entre les régions des deux segmentations à comparer. Pour tout couple de régions (V_i, R_j) , nous définissons leur recouvrement par $t_{ij} = \text{card}(V_i \cap R_j)$. Un couplage de poids maximal du graphe bipartite (V, R, t) fournit alors un appariement biunivoque optimal entre régions des deux segmentations au sens de la somme des recouvrements des parties appariées.

Soit K le nombre de couples obtenus et C_1, \dots, C_k les recouvrements de chacun des couples.

$\frac{1}{A} \times \sum_{k=1}^K C_k$ Représente alors le poids total du couplage normalisé par la surface de l'image.

La mesure de dissimilarité de Vinet est alors :

$$1 - \frac{1}{A} \times \sum_{k=1}^K C_k$$

Bien que le couplage ne soit pas nécessairement unique, cette mesure est une distance.

Une approximation de la mesure de Vinet peut être obtenue efficacement par un algorithme glouton qui consiste à coupler itérativement les deux régions de recouvrement maximal.

Cette mesure de dissimilarité a déjà pu être utilisée pour comparer des segmentations sur des images synthétiques monochromes comportant différents bruits et textures.

8.2. Evaluation sans vérité terrain

De nombreux critères sans vérité-terrain ont été proposés, tous, cherchant à quantifier la lisibilité de l'image. Ces critères peuvent être distingués en deux classes, à savoir :

- les critères de contraste.
- les critères d'adéquation à un modèle.

Les critères de contraste recherchent une variabilité inter-région tandis que les critères d'adéquation à un modèle recherchent une uniformité en intensité à l'intérieur des régions.

Pour montrer un exemple de chaque classe de critère, nous allons montrer les critères déterminés par Levine et Nazif que ce soit pour chacune des classes.

8.2.1. Critère inter-région de Levine et Nazif

Soit le contraste entre deux régions adjacentes R_i et R_j :

$$c_{ij} = \frac{|m_i - m_j|}{m_i + m_j}$$

Le contraste de la région R_i est :

$$c_i = \sum_{R_j} p_{ij} c_{ij}$$

où les R_j sont les régions adjacentes à R_i et le rapport longueur de la frontière commune entre R_i et R_j sur le périmètre de R_i est :

$$p_{ij} = \frac{l_{ij}}{l_i}$$

Le contraste global est alors :

$$\frac{\sum_{R_i} w_i c_i}{\sum_{R_i} w_i}$$

w_i est un poids associé à chaque région, qui peut être l'aire de la région.

8.2.2. Critère intra-région de Levine et Nazif

Ce critère simple est basé sur la somme des variances des régions. Par conséquent, sa valeur doit être faible.

$$\sum_i \sum_{s \in R_i} \left[f(s) - \frac{1}{A_i} \sum_{s \in R_i} f(s) \right]^2 = \sum_i \frac{\sigma_i^2}{C}$$

Le paramètre f peut être l'intensité du pixel s ou tout autre attribut comme la couleur ou la texture.

Le paramètre C est un facteur de normalisation, égal à la variance maximale :

$$\sigma_{\max}^2 = \frac{(f_{\max} - f_{\min})^2}{2}$$

Chaque région, tout comme dans le critère inter-région, peut être pondérée par son nombre de pixels.

L'avantage de ce critère est d'être facilement mis à jour dans les opérations de fusion ou de division des régions.

8.3. Evaluation de la segmentation du projet ALPAGE

Nous rappelons que la segmentation du projet ALPAGE a, dans notre cas, été réalisée sur une image de référence **Ref.tif** dont nous connaissons la vérité-terrain.

Par conséquent, nous utiliserons comme critère d'évaluation de notre segmentation par contours, le critère de Vinet.

Pour réaliser cette évaluation de segmentation, il est nécessaire de comparer une image étiquette de la segmentation (représentative de la segmentation effectuée) avec une image étiquette de la vérité terrain telle qu'elle nous est connue. Ensuite, à l'aide du critère de Vinet on peut évaluer notre segmentation par contours.

Voici les deux images que le critère de Vinet va comparer, en notant qu'ici le graphe spatial et donc la fusion de régions a été réalisée pour les 3 plus proches voisins.

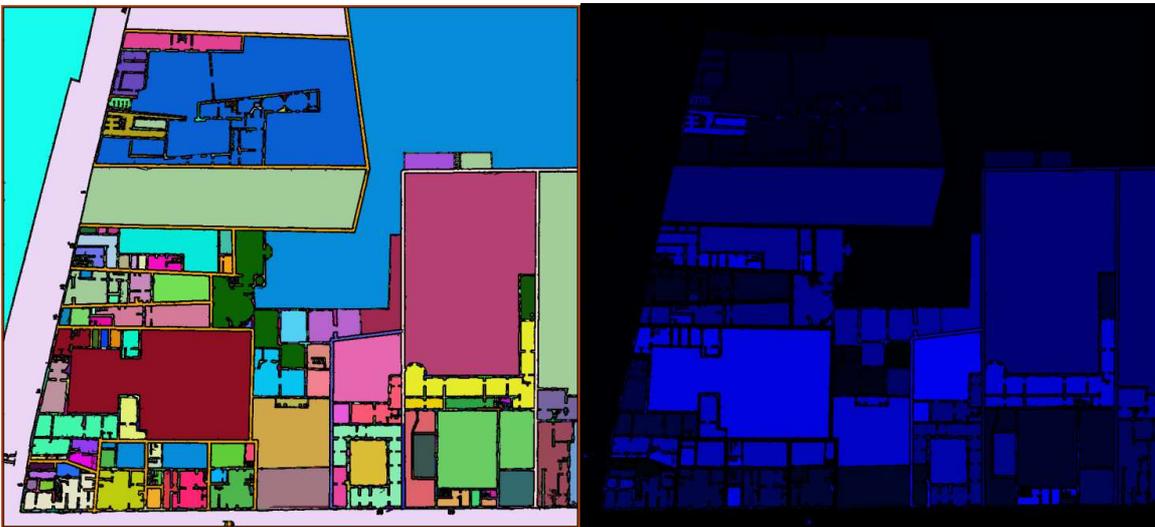


Figure 34 : Comparatif ImSeg.tif et ImVeriteTerrain.tif pour 3 ppv

Après exécution de notre programme, voici la valeur que le critère de Vinet nous renvoie :

0.8115519739224919

Maintenant, si nous réitérons la même expérience avec la fusion de régions réalisée sur les 10 plus proches voisins, voici ce que nous obtenons :

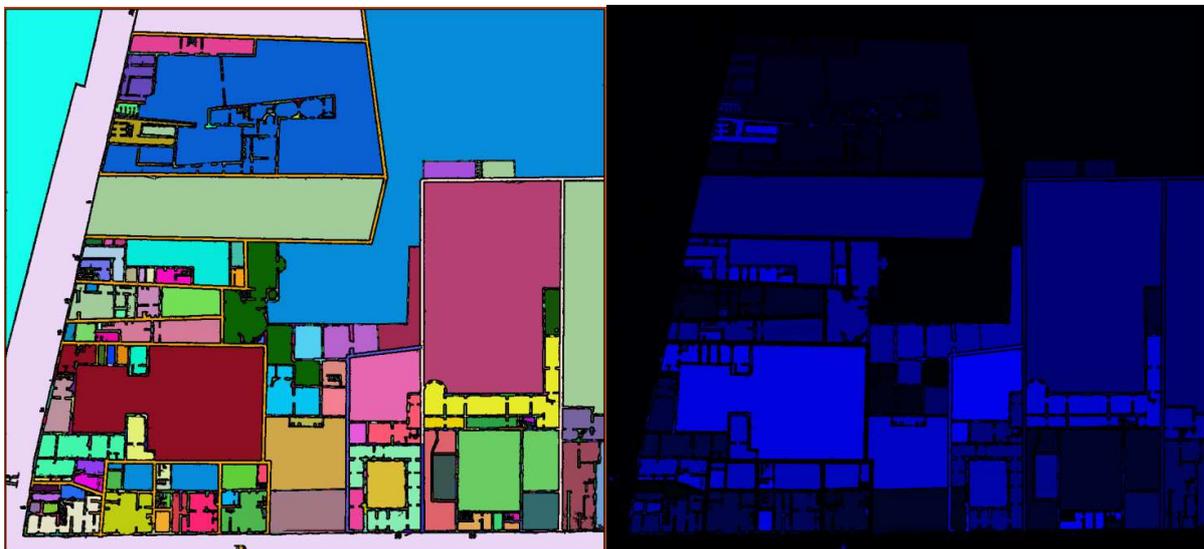


Figure 35 : Comparatif ImSeg.tif et ImVeriteTerrain.tif pour 10 ppv

Critère de Vinet :

0.80423035132198484

Nous constatons ici, que le fait de prendre plus de voisins pour réaliser fait diminuer sensiblement la valeur du critère de Vinet, et par conséquent la qualité de la segmentation.

Un deuxième paramètre, autre que les plus proches voisins, peut permettre d'améliorer la segmentation au sens du critère de Vinet.

En effet, par le biais d'un clustering, il est possible de fusionner les régions d'un certain nombre de pixels avec la région étant le plus proche colorimétriquement parlant.

Dans les exemples précédents, ce nombre de pixels était de 5000. Nous allons donc faire évoluer ce paramètre pour voir son influence sur l'évaluation de la segmentation.

Pour plus de clarté, voici un tableau comparatif montrant l'évolution de critère de Vinet en fonction du nombre de plus proches voisins à l'étape de la fusion de régions et du nombre de pixels pour lequel la région se fusionnera avec la plus proche au niveau colorimétrique :

Nombre de pixels \ Nombre de ppv	1000	3000	5000	7000	10000	Moyenne Vinet lignes
3	0.78665	0.81794	0.81155	0.81921	0.80711	0.80849
7	0.80041	0.73590	0.81390	0.82423	0.82081	0.79905
10	0.81806	0.81506	0.80423	0.82301	0.79485	0.81104
15	0.82253	0.81532	0.81850	0.82171	0.81412	0.81844
20	0.81882	0.80837	0.81717	0.82402	0.80924	0.81552
Moyenne Vinet colonnes	0.80929	0.79852	0.81307	0.82244	0.80922	

Figure 36 : Comparatif entre les deux critères de la segmentation

Selon le critère de Vinet :

- le nombre de pixels pour le clustering de régions qui maximise la segmentation est de 7000 (en moyenne).
- Le nombre de plus proches voisins à l'étape de fusion qui maximise la segmentation est de 15 (en moyenne).

Cependant, le meilleur compromis nombre de pixels / nombre de plus proches voisins est de 7000 / 7 avec la valeur de Vinet la plus élevée.

9. Bilan et autres approches

9.1. Résultats et interprétation

Globalement, nous pouvons affirmer que la segmentation par la détection de contours et la méthode du gradient de DiZenzo est efficace car, au sens du critère de Vinet, elle présente de bons résultats.

Si nous prenons les paramètres « optimaux » par rapport à notre étude préalable, voici l'image segmentée que nous obtenons tout en la comparant à notre image Ref.tif :

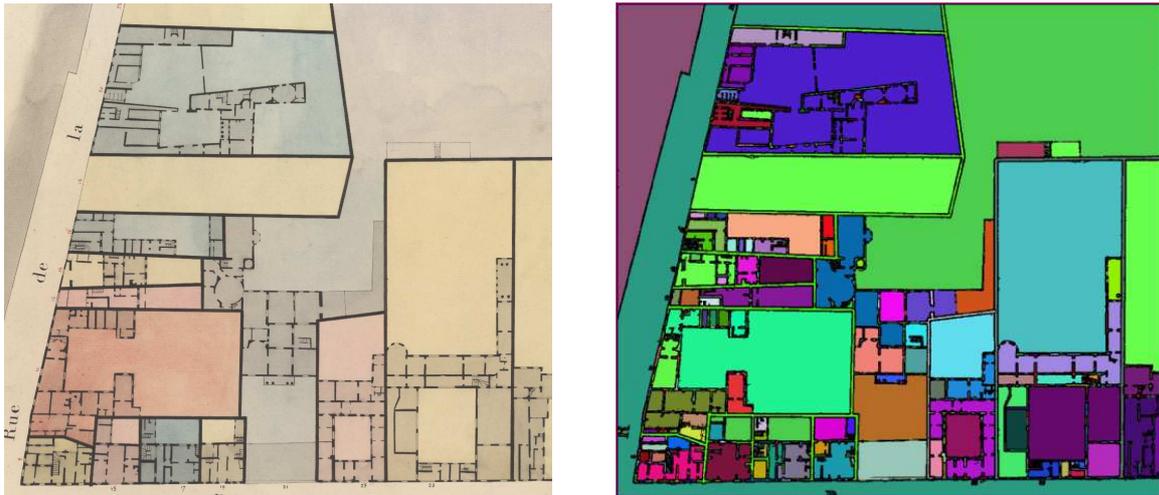


Figure 37 : Comparatif Ref.tif / image segmentée

Sur ce comparatif, nous pouvons voir que la segmentation est assez bonne même si quelques problèmes au niveau du respect de la sémantique de départ persistent.

En effet, le but est d'isoler chaque parcelle comme une région, et cela est globalement réalisé, mais pas seulement. L'autre but est de représenter des parcelles de mêmes couleurs par des régions, au niveau de la segmentation, de mêmes couleurs.

Ici, cela n'est pas réellement le cas. Par conséquent, il serait intéressant de tester d'autres méthodes pour pouvoir améliorer ce dernier point afin que la sémantique du projet ALPAGE soit respectée de la meilleure des manières.

9.2. Perspectives

Ce projet a, comme dit précédemment, permis de montrer que la segmentation par approche contours appliquée au projet ALPAGE donnait d'assez bons résultats en terme de partition en régions.

Cependant, il serait intéressant de voir plusieurs autres approches permettant, peut être d'améliorer les résultats obtenus.

Nous pourrions, ainsi, voir plusieurs perspectives différentes comme la détection de contours par l'intermédiaire du Laplacien par exemple.

Toutefois, l'amélioration la plus probable serait d'utiliser une méthode de segmentation hybride en combinant de la détection de contours et de la croissance de régions.

Le déroulement à suivre serait alors :

- M_a : Etablir la matrice d'adjacence pour le graphe d'adjacence.
- M_s : Etablir la matrice d'adjacence pour le graphe spatial.
- $M_{Color_distance}$: établir la matrice des distances couleurs entre régions.
- $M_{as} = M_a + M_s$. Avec + désignant un 'ou' logique
- $M_{choice} = M_{as} \times M_{color_distance}$

Le résultat de cette combinaison de méthodes pourraient alors être intéressants à voir et pourrait améliorer ce qui a été obtenu par la seule méthode de détection de contours.

Nous pourrions également envisager une autre approche qui consisterait à appliquer le gradient de DiZenzo sur une image définie dans un autre espace colorimétrique que l'espace RGB en entrée.

Cet autre espace colorimétrique pourrait alors être l'espace YUV, XYZ ou bien un Espace Hybride Décorrélé (EHD).

9.3. Utilisation du gradient de DiZenzo avec les EHD

Dans le cadre du projet semestriel de l'année précédente, il nous a été demandé d'étudier les Espaces Hybrides Décorrélés (EHD) afin de déterminer un EHD pour le projet ALPAGE.

L'idée de base d'un Espace Hybride Décorrélé est de combiner 3 composantes couleurs qui proviennent de différents modèles couleurs de bases (RGB, CMY, Lab...) afin de former un nouvel espace à 3 composantes, garantissant une linéarité avec les espaces de base.

Le but était alors, dans notre cas, de discriminer certaines composantes couleurs aux dépends d'autres, pour augmenter la séparabilité des données sur les plans cadastraux parisiens du projet ALPAGE.

Nous ne revenons pas sur la méthode suivie pour déterminer ce fameux EHD, si ce n'est que nous avons utilisé pour cela la méthode de l'Analyse en Composantes Principales (ACP). Cette dernière a permis de calculer un EHD tout en réduisant la corrélation entre les composantes et par conséquent la redondance de l'information.

L'idée de cette évolution du projet de cette année est alors d'utiliser la segmentation par le gradient de DiZenzo sur une image ayant été projetée dans un autre espace couleur que le l'espace traditionnel (et utilisé dans la méthode DiZenzo) RGB.

Nous rappelons que le gradient de DiZenko prend en entrée 3 vecteurs gradient marginaux (un en R, un en G et un en B) et qu'il calcule un vecteur gradient couleur, synthèse des trois et qui permet de faire la détection des contours.

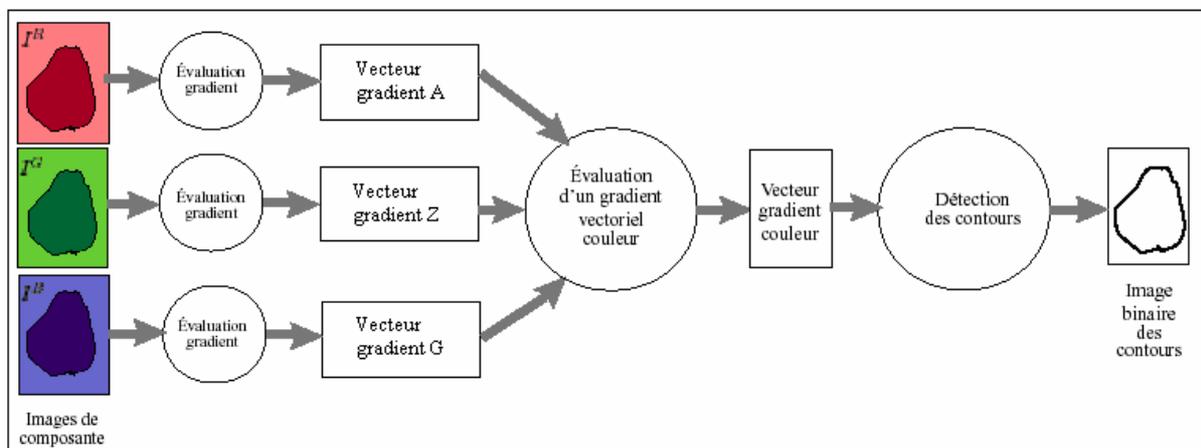
L'année dernière, nous avons déterminé l'Espace Hybride Décorrélé qui maximisait la séparabilité des données sur la même image de référence que dans ce projet, à savoir l'image Ref.tif.

Le nouvel espace que nous avons trouvé pour cette image était :

[A Of AC1C2 Z Of XYZ G Of RGB]

AC1C2, XYZ et RGB étant trois espaces couleurs différents.

Par conséquent, pour mettre en relation les deux projets, il faudrait suivre le schéma suivant quant à l'utilisation du gradient de DiZenko :



Ainsi, nous pourrions comparer les résultats de la segmentation dans l'espace RGB avec ceux qui seraient obtenus dans l'espace AZG pour l'image Ref.tif.

CONCLUSION

La problématique de ce projet était de réaliser une méthode de segmentation d'images dans le cadre du projet ALPAGE. Cette dernière était la méthode de détection de contours par l'intermédiaire du gradient de DiZenzo.

Comme dit auparavant, cette méthode a présenté d'assez bons résultats quant à son application sur ces plans cadastraux de par la nature des plans qui présentent des contours autour de chaque parcelle que nous voulions considérer comme région. Par conséquent, les résultats obtenus en terme de région s'approchaient assez bien de la sémantique de départ qui était d'isoler chaque parcelle.

Cependant, de nombreuses approches différentes restent à appliquer, comme par exemple les approches hybrides telles que la combinaison entre la détection de contours et la croissance de régions ou comme le split and merge.

Une approche par le biais des Espaces Hybrides Décorrélés pourrait tout aussi bien mériter d'être développée afin d'obtenir une meilleure segmentation de ces plans cadastraux anciens.

A titre personnel, ce projet fut très enrichissant car il m'a permis de me familiariser à ce qui peut être demandé dans le monde de la recherche, à savoir réaliser une grosse étude bibliographique sur des concepts nouveaux et une application dans un contexte très intéressant qui est celui du projet ALPAGE.

De plus, la continuité avec le projet de l'an dernier m'a permis de consolider certaines bases dans le traitement d'images et dans l'étude des différents espaces colorimétriques puis, de découvrir un autre domaine très vaste qui est celui de la segmentation d'images.

Enfin, je remercie Romain RAVEAUX, mon tuteur, qui m'a permis de réaliser ce projet très intéressant et aux multiples perspectives.

BIBLIOGRAPHIE

- [1] Romain Raveaux, Jean-Christophe Burie, Jean-Marc Ogier : « A colour document interpretation : Application to ancient cadastral maps ». The 9th International Conference On Document Analysis (ICDAR 2007).

- [2] Romain Raveaux, Jean-Christophe Burie, Jean-Marc Ogier : « A contribution to Ancient Cadastral Maps interpretation through color analysis ». The 7th International Workshop on Pattern Recognition in Information Systems (PRIS 2007).

- [3] Luc Brun :” Modèles Mathématiques et représentation discrètes pour la description des images couleur ».

- [4] Jean-Christophe Baillie : Cours de segmentation.

- [5] Romain Raveaux, Jean-Christophe Burie, Jean-Marc Ogier : « A knowledge-based colour segmentation scheme : Application to ancient cadastral maps ».

- [6] Ludovic Macaire, Sylvie Philipp-Foliguet : « Segmentation d’images couleur ».

- [7] Sylvie Philipp-Foliguet_, Laurent Guigues : « Evaluation de la segmentation d’images : ´etat de l’art, nouveaux indices et comparaison ».

- [8] Alexandre Hamon : « Segmentation couleur de plans cadastraux de la ville de Paris au XIXème siècle ».

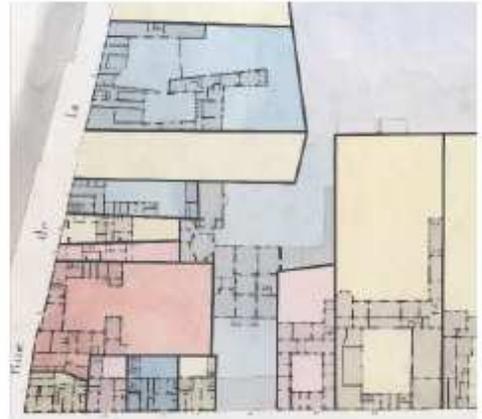
ANNEXES

Annexe 1 : Pré traitements :

- Théorie du monde blanc :



Avant

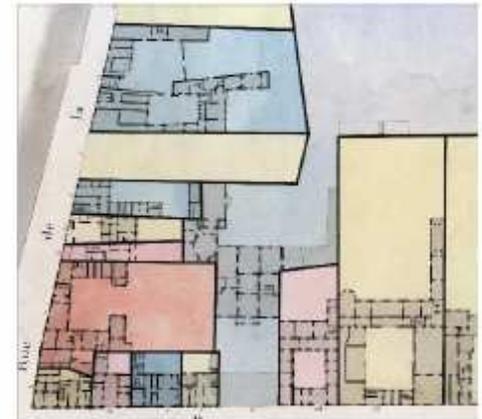


Après

- Extension dynamique :



Avant

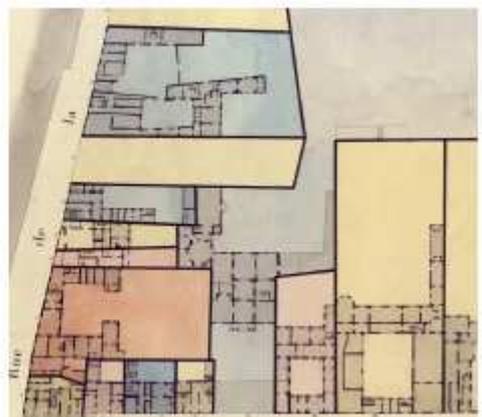


Après

- Anti-affadissement :

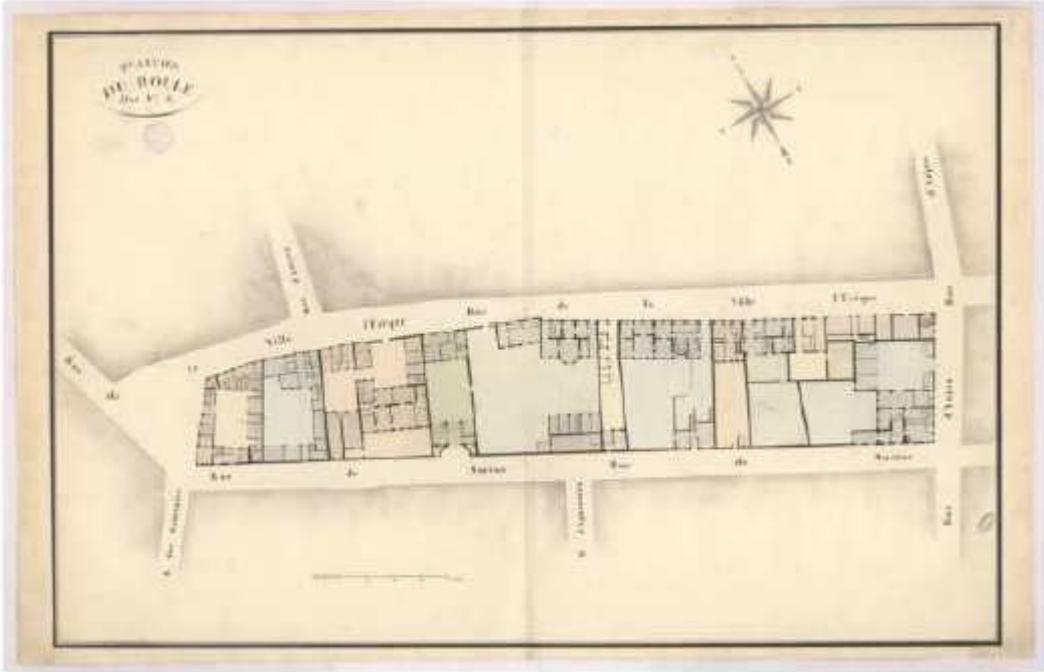
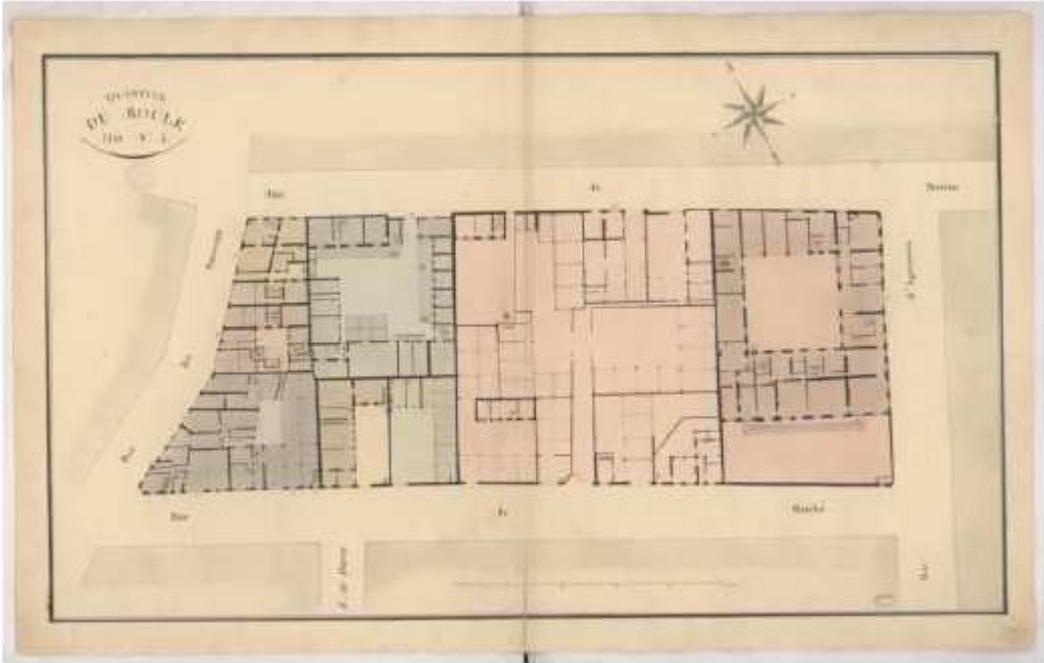


Avant



Après

Annexe 2 : autres plans cadastraux :



Annexe 3 : Exemple Gradient DiZenzo sur Lena :



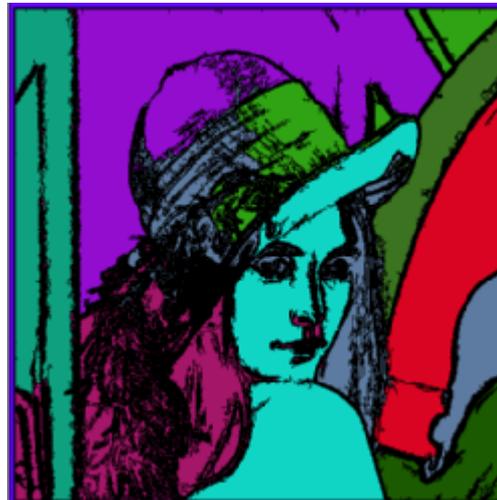
LenaDiZenzo.tif



LenaBinar.tif



LenaSeg.tif



LenaApFusion.tif



LenaLabel.tif