

Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITÉ ABOU BEKR BELKAID TLEMEN



FACULTÉ DE TECHNOLOGIE

DÉPARTEMENT DE TELECOMMUNICATIONS



Mémoire pour l'obtention du diplôme de

Master en Télécommunications

Option : Photonique et Réseaux Optiques de Télécommunications (PROTONS)

Thème

**La géolocalisation basée sur l'utilisation des systèmes
d'informations géographiques**

Présenté par :

Mlle DAHMANI SARRA

Mlle HABIB MERYEM

Soutenu le 26 Mai 2016 devant le jury composé de :

Président :	Mr. CHIKH-BLED MOHAMMED EL-KEBIR	PROFESUR	Univ de Tlemcen
Examineurs :	Mr. MOURAD HADJILA	M C B	Univ de Tlemcen
	Mr. ABDELMALEK ABDELHAFID	M C B	Univ de Tlemcen
Encadreur :	Mr REDA BOUABDALLAH	M AA	Univ de Tlemcen

Table de matière

Table de matière

Remerciement

Dédicace

Listes des Figures

Liste des Tableaux

LISTE DES ACRONYMES

Introduction générale 1

Chapitre I : La géolocalisation

I.1 Introduction 3

I.2 Définition de la géolocalisation 3

I.3 Origine et histoire de la géolocalisation 4

I.4 Les techniques de la géolocalisation 5

 I.4.1 La géolocalisation par satellite 5

 I.4.1.1 La triangulation 6

 I.4.1.2 Le secteur spatial 7

 I.4.1.3 Les stations de contrôle 7

 I.4.2 La géolocalisation par téléphone portable 7

 I.4.2.1 Le positionnement par GSM 7

 I.4.2.2 L'identification par cellules Cell-ID 8

 I.4.2.3 Identification par le temps (E-OTD) 9

 I.4.3 La géolocalisation par adressage IP 9

 I.4.4 La géolocalisation par Wi-Fi 11

 I.4.5 La géolocalisation par RFID 11

I.5 Les techniques hybrides 11

I.6 Les avantages et les inconvénients de chaque technique 11

I.7 Les Services de la Géolocalisation 12

I.8 Conclusion 12

Chapitre II : Le système d'information géographique

II.1 Introduction 14

Table de matière

II.2 La définition d'un SIG.....	14
II.3 Composants d'un SIG	15
II.3.1 Le matériel	15
II.3.2 Le logiciel	15
II.3.3 Les données.....	16
II.3.4 Le personnel.....	16
II.4 Typologie des données SIG	16
II.4.1 Les données géographiques	16
II.4.2 Les données attributaires.....	17
II.5 Représentation des Données SIG.....	18
II.5.1 Représentation en mode vectoriel.....	18
II.5.2 Représentation en mode raster	18
II.5.3 Comparaison entre les deux modes.....	19
II.6 Fonctionnalités d'un SIG	20
II.7 Les domaines d'application des SIG	21
II.8 Avantages de SIG	22
II.9 Conclusion	22
Chapitre III: Traitement d'image	
III.1 Introduction	24
III.2 Traitement d'image	24
III.3 Définition de l'image numérique	24
III.3.1 Types d'image numérique	25
III.3.1.1 Selon la représentation numérique.....	25
III.3.1.2 Selon la représentation des couleurs	26
III.3.2 Format d'une image.....	27
III.3.3 Caractéristiques d'images	28
III.4 Système de traitement d'image	30
III.4.1 Acquisition et numérisation.....	31
III.4.2 Visualisation	31
III.4.3 Filtrage.....	31
III.4.3.1 Filtres linéaires.....	31
III.4.3.2 Filtrage non linéaire	34
III.4.4 Segmentation d'une image	35

Table de matière

III.4.4.1 Définition	35
III.4.4.2 Méthodes de segmentation.....	36
III.5 Conclusion.....	37
Chapitre IV : Description de l'application	
IV.1 Introduction.....	38
IV.2Présentation du logiciel MATLAB	38
IV.2.1Les commandes utilisées sur MATLAB pour réaliser cette application.....	38
IV.2.2 Les domaines d'application de MATLAB	39
MATLAB est utilisé dans plusieurs domaine nous citons :	39
IV.3 La Zone suivie.....	40
IV.4 Les opérations du traitement d'images sur les images satellitaires de barrage BOUGHRARA.....	41
IV.4.1 Analyse de l'image	41
IV.4.1.1 Les images en niveau de gris	41
IV.4.1.2 L'Histogramme.....	41
IV.4.2 Amélioration d'image.....	42
IV.4.2.1Egalisation	42
IV.4.2.2 Filtrage	43
IV.4.2.3 Segmentation	43
IV.5Conclusion	47
Conclusion générale	48
BIBLIOGRAPHIE	
ANNEXE	



بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



REMERCIEMENT

Avant tout, nous remercions Allah de nous avoir aidé à réaliser ce travail.

Nous adressons tout particulièrement à Monsieur « Réda Bouabdelah », nos sincères remerciements pour nous avoir encadré et conseillé pendant la réalisation de ce mémoire. Son aide appréciable, ses encouragements qui nous ont permis d'avancer dans l'élaboration de ce travail et à qui nous témoignons notre profonde reconnaissance.

Nous adressons nos vifs remerciements à Monsieur « CHIKH-BLED MOHAMMED EL-KEBIR » Professeur à l'université Abou-Bakr Belkaid de Tlemcen, de nous avoir fait l'honneur de juger et présider le jury de ce travail.

Nous remercions vivement monsieur « ABDELMALEK ABDELHAFID » M.C.B et monsieur « M.HADJILA » M.C.B à l'Université Abou-Bakr Belkaid de Tlemcen, pour nous avoir honorés de leurs présences en acceptant d'examiner et de juger ce travail. Ainsi que tous ceux qui nous ont assisté de près ou de loin dans le Parcours de la réalisation de ce mémoire.



Dédicace

Que ce travail témoigne de mes respects :

A mes parents :

Grâce à leurs tendres encouragements et leurs grands sacrifices, ils ont pu créer le climat affectueux et propice à la poursuite de mes études. Aucune dédicace ne pourrait exprimer mon respect, ma considération et mes profonds sentiments envers eux. Je prie le bon Dieu de les bénir, de veiller sur eux, en espérant qu'ils seront toujours fiers de moi.

A mes sœurs et à mes frères

A ma famille :

Ils vont trouver ici l'expression de mes sentiments de respect et de reconnaissance pour le soutien qu'ils n'ont cessé de me porter.

A tous mes professeurs :

Leur générosité et leur soutien m'oblige de leurs témoigner mon profond respect et ma loyale considération.

*A tous mes amis et mes collègues (particulièrement **YOUCEFI CHAIMAA**):*

Ils vont trouver ici le témoignage d'une fidélité et d'une amitié infinie.



MERREM HABIB



DEDICACES

A mes parents qui m'ont toujours soutenue tout au long de mes études,

A Mes soeurs,

A mes amis qui ont toujours été là quand j'ai eu besoin d'eux,

A mes professeurs à qui je suis très reconnaissante,

A tous ceux que je n'ai pas cité et qui sont dans mon coeur,

Je dédie ce travail...

SARRA DAHMANI



Liste des Figures et des tableaux

Liste des Figures

Figure I.1 . La triangulation par le satellite.....	6
Figure I.2 . Technique de la localisation par Cell-ID.....	8
Figure I.3 . Technique de la localisation par E-OTD.....	9
Figure I.4 . Technique de localisation par adressage IP.....	10
Figure II.1. Modèle de SIG.....	14
Figure II.2. Composants d'un SIG.....	15
Figure II.3. Les types d'objet.....	17
Figure II.4. Latitude et Longitude d'un lieu sur la terre.....	18
Figure II.5. Représentation des données vectrices.....	18
Figure II.6. Représentation des données Raster.....	19
Figure II.7. La plateforme virtuelle.....	21
Figure III.1. Les pixels d'une image.....	25
Figure III.2. Le spectre électromagnétique.....	26
Figure III.3. L'intervalle du niveau de gris.....	27
Figure III.4. L'histogramme de niveau de gris.....	30
Figure III.5. Application du filtre moyen (lissage).....	32
Figure III.6. Rehaussement de contour.....	34
Figure III.7. Principe du filtre médian.....	35
Figure III.8. Principe du filtre maximum.....	35
Figure III.9. Principe du filtre minimum.....	35
Figure IV.1. Le logiciel de calcul scientifique MATLAB.....	39
Figure IV.2. Barrage Boughrara.....	40
Figure IV.3. Transformation de l'image initiale en niveau de gris.....	41
Figure IV.4. Histogramme d'image.....	42
Figure IV.5 . Egalisation de l'image.....	42
Figure IV.6. L'interface auquel les résultats sont affichés.....	43
Figure IV.7. Filtrage d'image.....	43
Figure IV.8. Segmentation d'image avec présence des pixels isolés.....	44
Figure IV.9. Segmentation d'image avec élimination des pixels isolés.....	44
Figure IV.10. La détection du contour de l'image.....	45
Figure IV .11. La superposition du contour de l'image.....	45

Liste des Figures et des tableaux

Liste des Tableaux

Tableau I.1 : les avantages et les inconvénients de chaque technique.....	12
Tableau II.1 : Tableau représente les avantages et les inconvénients de chaque mode	19
Tableau III.1 : Tableau comparatif de différents formats d'images.....	28
Tableau VI.1 : Les surfaces en pixels.....	46
Tableau VI.2 : Les surfaces en mètre carrée.....	46

LISTE DES ACRONYMES

LISTE DES ACRONYMES

GPS: Global Positioning System.

GSM: Global System for Mobile.

Mac: Medium Access Control.

GPRS: General Packet Radio Service.

EOTD:différence de temps observée.

IP: internet protocol.

Wi-Fi: Wireless Fidelity.

SMS: short message system.

IMEI: Identité Internationale d'Equipement Mobile.

SIM:Subscriber Identity Module.

CELL-ID:cellule identification.

BTS:Base transceiver station.

BSC:Base Station Controller.

RFID: Radio Frequency Identification.

SIG: système d'information géographique.

SGBD:système de gestion de base de données.

FM: frequency modulation.

ISO: International Organization for Standardization.

JPEG:Joint Photographic Experts Group.

GIF: Graphic Interchange Format.

LISTE DES ACRONYMES

PNG: Portable Network Graphics.

TIFF: Tagged Image File Format.

SVG: Scalable Vector Graphics.

Introduction générale

Introduction générale

La géolocalisation est un domaine scientifique de haut technologie, qui existe depuis la préhistoire, elle a été développée et elle se développe continuellement pour mieux répondre aux besoins en informations dans des plusieurs domaines.

D'abord l'homme pensait à trouver des solutions et des moyens afin de faciliter l'utilisation de la géolocalisation. L'être humain a créé des cartes géographiques qui décrivent les lieux et les routes ...etc. Après, il a cherché des solutions plus pratiques, il a arrivé à trouver et développer des technologies faciles à utiliser et à la portée de tout le monde. Parmi ces technologies, nous citons le GPS (global positioning system), GSM....

L'utilisation de la géolocalisation très vaste, par exemple elle nous a servi à la sécurité, le guidage maritime, le transport, etc...

La géolocalisation est basée sur l'analyse des données et leur visualisation, c'est pour cette raison que le système d'information géographique (SIG) joue un rôle très important pour la géolocalisation à cause des fonctions qu'il présente.

Le SIG permet à partir de diverses sources, de rassembler, et d'organiser, de gérer, d'analyser, et de présenter des informations localisées géographiquement contribuant notamment à la gestion de l'espace. Il est appliqué dans des plusieurs domaines comme le transport, planification urbaine, la suivi des forêts Sa principale mission est de mettre à la disposition des utilisateurs les informations sur les éléments d'un territoire (sites ou phénomènes physiques, objet statique ou mobile etc....) ou sur le territoire lui-même.

Pour améliorer les résultats du SIG nous envisagerons par les techniques de traitement d'images qui sont une discipline informatique et de mathématique permettant d'améliorer la qualité d'image, d'extraire, préparation des informations qu'on a besoin.

L'image est constituée des objets et des informations, elle a plusieurs catégories comme exemple l'image satellitaire, qui est l'objectif de ce mémoire.

Notre travail est partagé en quatre chapitres organisés comme suite :

Dans le premier chapitre, nous définissons d'abord la géolocalisation et nous citons son rôle avec ses différentes techniques qui sont devenues nécessaires et très importantes dans tous les domaines à l'heure actuelle. Nous allons présenter par la suite quelques techniques hybrides avec leurs avantages et inconvénients.

Introduction générale

Dans le deuxième chapitre, nous présentons le SIG avec ses composants et ses typologies; ensuite, nous citons les deux modes (vectoriel/raster) suivi par une comparaison entre eux.

Dans le troisième chapitre, nous allons définir l'image avec ces types et citer les formats et les caractéristiques de l'image. Après, nous allons présenter les systèmes de traitement d'images et les différentes opérations de traitements sur les images par exemple « filtrage, segmentation etc... » .

Enfin, le quatrième et le dernier chapitre, nous le consacrons à la présentation d'une application informatique que nous réalisons, développée à l'aide d'outil MATLAB pour le traitement des images satellitaires de la zone qui nous intéresse « HAMMAM BOUGHRARA ».



CHAPITRE I
LA GÉOLOCALISATION

I.1 Introduction

La géolocalisation est le fait de localiser quelque chose ou quelqu'un dans un espace, selon un système de référence. La géolocalisation consiste à localiser un objet dans une surface de la planète (longitude, latitude, altitude) ou dans une carte à l'aide de ses coordonnées géographiques.

Derrière ce mot compliqué se cache une notion assez simple : en donnant une origine et des coordonnées, tout le monde est capable de retrouver un objet en utilisant les technologies de géolocalisation ; là où les choses se compliquent, c'est qu'il faut que la personne à qui on veut situer l'objet doit connaître le lieu de l'origine, il faut donc avoir un langage commun, compréhensible par tout le monde (personne, machine...).

On veut pouvoir situer un objet partout dans le Monde de la même façon, d'où la nécessité de s'entendre sur des coordonnées absolues, acceptées par tous (exemple latitude et longitude).

La plus grosse difficulté qu'a dû surmonter l'homme, c'est d'arriver à situer en temps réel un objet en mouvement, d'où des problématiques telles que : comment détecter un objet ?, comment récupérer des informations sur sa position (en communiquant avec lui ou contre son gré) ?, comment ensuite le situer précisément et rapidement sur une carte ?

I.2 Définition de la géolocalisation

La géolocalisation ou géoréférencement est un procédé permettant de positionner un objet (une personne, un véhicule ... etc.) sur un plan ou une carte à l'aide de ses coordonnées géographiques ; cette opération est réalisée à l'aide d'un terminal capable d'être localisé (grâce à un système de positionnement par satellites et un récepteur GPS par exemple). Les positions enregistrées peuvent être stockées au sein du terminal et être extraites postérieurement, ou être transmises en temps réel vers une plateforme logicielle de géolocalisation. La transmission en temps réel nécessite un terminal équipé d'un moyen de télécommunication de type GSM, GPRS, radio ou satellite lui permettant d'envoyer les positions à des intervalles réguliers. Ceci permet de visualiser la position du terminal au sein d'une carte à travers une plateforme de géolocalisation le plus souvent accessible depuis internet [1].

I.3 Origine et histoire de la géolocalisation

Les technologies de géolocalisation sont très anciennes. En effet, depuis très longtemps, l'être humain avait besoin de se situer dans l'espace et prendre des points de repère pour trouver son chemin et l'envoyer aux autres. Ce sont les vecteurs par lesquels la localisation s'effectue.

Avec l'avènement des guerres de plus en plus sophistiquées, il devient important à partir du 19e siècle de connaître la position d'objets mouvants, que ce soient des troupes, des équipements ou des personnes en particulier et ceci en temps réel. La première moitié du XXe siècle a été marquée par la découverte des ondes radios qui nous ont aidés à transmettre. En effet, une nouvelle technologie, la radiogoniométrie, venait d'être découverte et permit de mesurer la direction d'arrivée d'une onde électromagnétique. Ce qui a permis la découverte d'une nouvelle technique de triangulation maîtrisée depuis 1533.

Dans le même temps, la découverte des antennes directionnelles et des ondes électromagnétiques entraîna la découverte des techniques de localisation par différence de temps observée (EOTD), utilisée actuellement dans le cadre de la géolocalisation par GSM comme exemple. Le premier de ces systèmes fut proposé par la Raytheon Corporation en réponse à un appel d'offre de l'Air Force pour un système de guidage.

C'est le vice-président de l'ingénierie et de la recherche de cette même entreprise, le Dr Ivan Getting qui proposa par la suite d'étudier l'usage de satellite comme base d'un système de navigation pour des véhicules se déplaçant rapidement dans les trois dimensions. Quand le Dr Getting quitta Raytheon en 1960, la technique qu'il avait proposée était parmi les formes les plus avancées au Monde en termes de navigation.

Cela posa les premières bases de la future création du Global Positioning System (aussi appelé GPS).

Cela nous a conduits à la forme de géolocalisation la plus connue du grand public : le GPS, le premier système de positionnement par satellites nommé TRANSIT, Développé en 1958 pour la marine des USA et utilisé pour la première fois en 1964, son utilisation devint civile en 1967.

Dans les années 90, l'avènement des réseaux de téléphonie mobile permit de mettre au point un nouveau type de géolocalisation, dit par GSM. Une personne dotée d'un téléphone portable allumé pouvait désormais être située dans toutes les zones couvertes par le réseau.

Enfin, avec le boom de l'informatique dans les années 2000, il est devenu également possible de localiser une personne sur Internet grâce à son adresse IP.

Depuis 99, et avec la découverte de la technologie WIFI il est possible de localiser une personne connectée à un point d'accès grâce à son adresse MAC[1].

I.4 Les techniques de la géolocalisation

La géolocalisation existe dans des différentes technologies, touchant aux domaines d'activités complémentaires. Dans la partie suivante on va visualiser les plus importantes techniques de la géolocalisation [1].

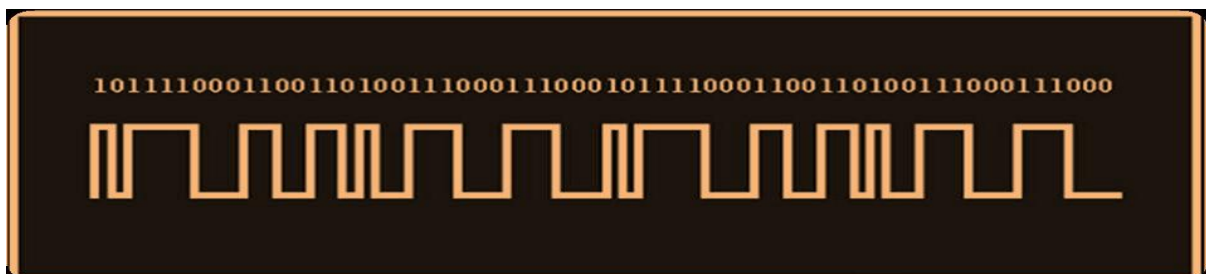
I.4.1 La géolocalisation par satellite

Le système est basé sur la mesure de la distance entre le récepteur et le satellite en orbite. Pour cela le concept global: **distance=temps * vitesse.**

La vitesse correspond à la célérité des ondes transmises, c'est à dire très proche de celle de la lumière = $3 \cdot 10^8$ m/s. Le plus difficile est de déterminer le temps précis. En effet nous aurons affaire à des temps très petits pour détecter un écart d'un kilomètre, ce qui est un écart important par rapport à la précision du système GPS, le récepteur doit pouvoir mesurer un écart de temps égal à $1/300\,000 = 3,33 \cdot 10^{-6}$ soit environ 3 millièmes de secondes.

Le satellite et le récepteur génèrent au même moment une trame pseudo-aléatoire unique. Le récepteur reçoit cette même trame mais décalé.

Trame générée



I.4.1.2 Le secteur spatial

GPS se regroupe 24 satellites, répartis de telle sorte que chaque zone de la terre est couverte par 4 satellites. Ils doivent donc être à environ 20000km d'altitude et avec 6 à 8 satellites répartis sur une orbite inclinée de 50° par rapport à l'équateur. Ces satellites extraient leur énergie de panneaux solaires. Les satellites engendrent deux ondes L1 et L2 respectivement à 1575,42 MHz et 1227,60 MHz. Elles sont modulées par la phase (de 180° lorsque le message ou le code change de parité), et contiennent la position du satellite, l'état de la santé du satellite, et d'autres informations et le code pseudo aléatoire permettant la synchronisation, Il en existe 2, l'un réserve à l'armée, l'autre aux civils.

I.4.1.3 Les stations de contrôle

Une station principale reçoit les données des autres stations, calcule les éphémérides des satellites et surveille le fonctionnement du système.

Un récepteur équipé du système GPS pour pouvoir être géolocalisé doit avoir une fonctionnalité supplémentaire qui est d'envoyer sa localisation via un réseau SMS, GPRS ou autre. En effet, le système GPS permet uniquement de connaître sa position, le récepteur n'est rien obligé d'envoyer pour pouvoir fonctionner. Cette propriété est utile dans d'autres cas d'utilisation comme le guidage GPS, cela permet de protéger ses données.

I.4.2 La géolocalisation par téléphone portable

I.4.2.1 Le positionnement par GSM

La géolocalisation par GSM (téléphones mobiles de deuxième génération actuellement utilisés par la majorité des gens) est basée sur l'IMEI (International Mobile Equipment Identify) obtenu à partir de la carte SIM, qui est un code unique composé de 15 chiffres. Le numéro IMEI identifie l'appareil (le téléphone en lui-même), tandis que le numéro SIM identifie la carte SIM donc le numéro de téléphone demandé. Toutes les informations sont contenues dans la carte SIM. Lorsqu'un utilisateur s'identifie à un réseau mobile, son numéro IMEI est transmis au système de gestion de l'opérateur afin de pouvoir l'autoriser à utiliser les différentes options disponibles.

Il existe plusieurs méthodes, Cell-Id, la plus simple à mettre en œuvre et EODT, un peu plus précise. Aujourd'hui la seule méthode couramment utilisée est Cell-Id, bien qu'elle soit

moins précise elle ne demande aucune infrastructure supplémentaire à celle d'un réseau GSM classique.

Il faut savoir qu'un réseau GSM est composé de plusieurs cellules et que chacune d'elle contient une BTS (Base Transciever Station) qui prend en charge les communications radio des téléphones mobiles. Chaque BTS est relié ensuite à un BSC (Base Station Controler) qui pilote l'ensemble des BTS et gère le basculement d'un portable d'une cellule à une autre. Selon le principe de handover (choisit quelle est la cellule qui prendra en charge le mobile).[1]

I.4.2.2 L'identification par cellules Cell-ID

Cette méthode simple d'identification, elle va s'effectuer à partir de l'adresse de la BTS à laquelle le mobile est connecté. La BTS repère le mobile pour pouvoir prendre la communication, il y a identification de la carte SIM avant de démarrer la communication. Chaque cellule BTS sait donc quels portables sont dans son champ de fonctionnement, ces données sont automatiquement transmises à la BSC puisque que c'est le BSC qui décide quelle BTS est affecté à chaque mobile. Ces données sont ensuite transmises à une base de données, qui sait donc quelles cartes SIM sont dans le champ de chaque cellule. Or cette base de données sait aussi l'adresse exacte de chaque antenne. L'on peut donc connaître la localisation approximative d'une carte SIM.

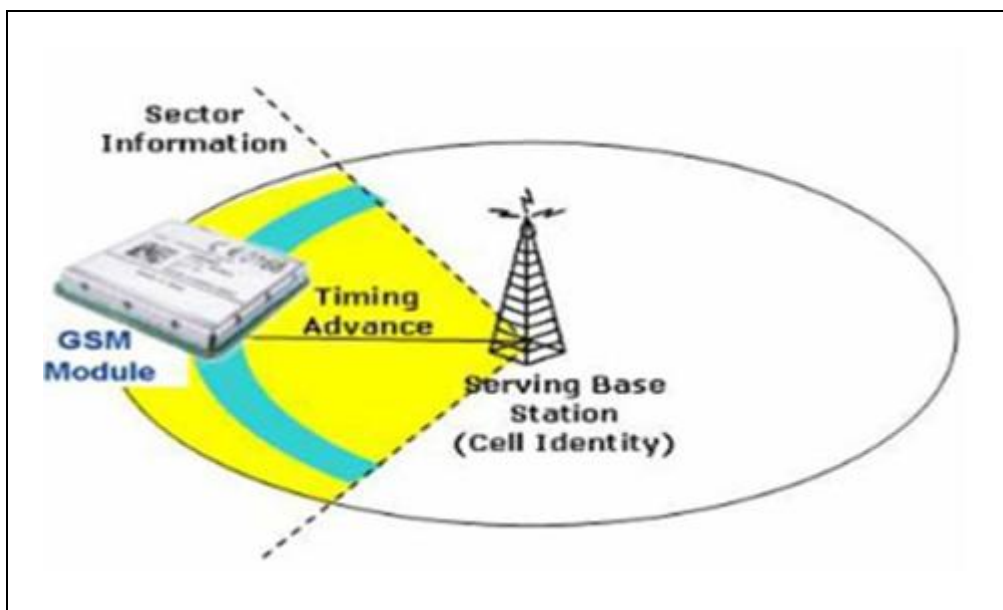


Figure I.2. Technique de la localisation par Cell-ID.

Cette localisation dépend donc fortement de la densité d'antenne, si un récepteur est dans plusieurs champs différents, on commence à pouvoir de localiser de manière précise. En ville l'on peut repérer un portable à 250 mètres près, en zone rurale on peut arriver à une précision de seulement 10 km.

I.4.2.3 Identification par le temps (E-OTD)

La méthode E-OTD (Enhanced Observed Time Difference) nécessite l'envoi d'un signal par le portable. Il faut donc que le mobile soit équipé pour pouvoir être localisé. Le BTS envoie des signaux régulièrement, dès que le mobile reçoit un de ces signaux, il réémet. Le BTS peut donc calculer la distance en mesurant le temps d'aller-retour.

Pour avoir un temps plus précis, on utilise plusieurs cellules BTS pour repérer un mobile. L'idéal serait d'avoir trois cellules dans la portée du mobile pour avoir une localisation optimale.

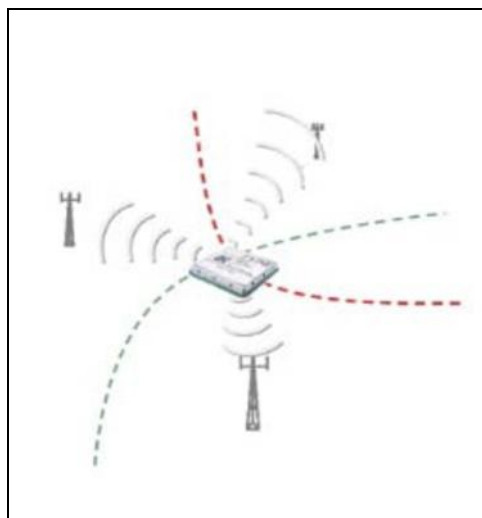


Figure I.3. Technique de la localisation par E-OTD

Toutes ces différentes géolocalisations par GSM sont toutefois moins précises que le GPS, surtout en campagne, elles dépendent essentiellement de la densité d'antenne autour du mobile.

I.4.3 La géolocalisation par adressage IP

La Géolocalisation par adressage IP et implicitement à l'aide des réseaux d'internet par Wifi découle de la manière d'implémentation de l'Internet à l'heure actuelle. Pour comprendre le fonctionnement, il faut tout d'abord savoir comment on a fait jusqu'à présent pour répartir les adresses IP.

Chaque pays se voit attribué une certaine tranche des adresses IP, pouvant être ensuite utilisé par les fournisseurs d'accès à l'Internet du pays respectif. Ensuite, à l'aide d'une base des données, on peut savoir la « nationalité » d'une adresse IP.

Il existe plusieurs sites pour géolocaliser, qui utilise la technique d'adressage IP.

Voici un exemple qui a été réalisé par le site **Hostip.info** [2].

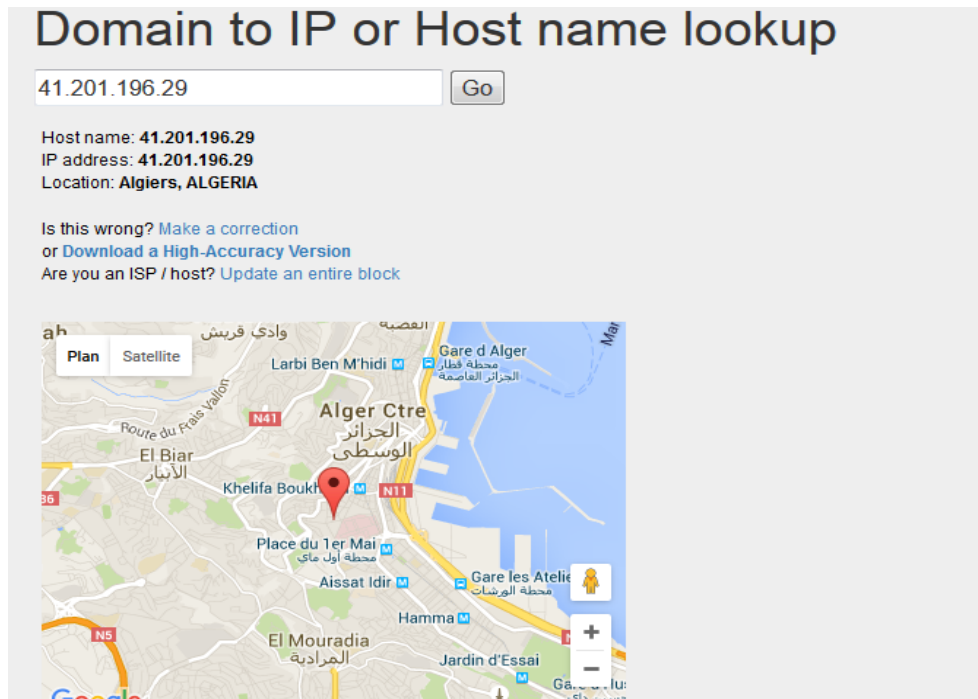


Figure I.4. Technique de localisation par adressage IP.

- **Hostip.info** est un site communautaire qui génère une base de données de géolocalisation.

Les données sont issues des membres qui en s'inscrivant et en soumettant leurs propres adresses postales et adresse IP, renseignent la base. Celle-ci compte déjà plus de 8 millions d'entrées.

Le site propose une API de type REST qui se veut simple et facile à intégrer.

Le service fournit différents types de données comme les noms des villes et les coordonnées géographiques (latitude, longitude).

I.4.4 La géolocalisation par Wi-Fi

De la même façon qu'un terminal GSM peut se localiser par la méthode du Cell ID sur un réseau GSM, un terminal Wi-Fi peut utiliser la même méthode en se basant sur les identifiants des bornes Wi-Fi (adresses MAC) qu'il détecte. Il existe des bases de données recensant une multitude de bornes d'accès Wi-Fi ainsi que leur position géographique. Ces bases peuvent appartenir à des entreprises privées ou à des communautés qui les publient gratuitement. Ces bases de données sont construites en utilisant la méthode appelée WarDriving, qui consiste à parcourir les rues des villes en voiture avec un ordinateur portable équipé du Wi-Fi et relié à un récepteur GPS, afin de recenser un maximum de points d'accès Wi-Fi [1].

I.4.5 La géolocalisation par RFID

La technologie RFID « Identification par Radio Fréquence » permet l'identification des objets, d'en suivre le cheminement et d'en connaître les caractéristiques à distance grâce à une étiquette émettant des ondes radio, attachée ou incorporée à l'objet.

A l'aide de cette technologie, on peut réaliser la lecture des étiquettes même sans ligne de vue directe et le passage par de fines couches de matériaux (peinture, neige, etc.). Ainsi, grâce aux « lecteurs RFID », on peut accéder aux données mémorisées sur des marqueurs appelés « étiquettes radiofréquences »[1].

I.5 Les techniques hybrides

Aucune technique n'est absolue, chacune a son propre défaut, généralement accordée à leur dépendance à un certain réseau. La géolocalisation par GPS est impossible à utiliser à l'intérieur. La géolocalisation par GSM a une couverture géographique limitée. Enfin, la géolocalisation par Wi-Fi souffre, quant à elle d'une dépendance à la présence de bornes d'accès Wi-Fi, plus complexe en milieu rustique.

Il existe certains dispositifs qui combinent ces trois techniques pour limiter leurs faiblesses. Par exemple, de géolocaliser un individu à l'extérieur en utilisant le GPS et de garder sa trace à l'intérieur grâce aux techniques GSM et Wi-Fi.

I.6 Les avantages et les inconvénients de chaque technique

Chaque technique a des inconvénients et des avantages le tableau suivant représente ces derniers [3].

Les théchniquess	Les avantages	Les inconnvenients
GPS	Bonne précision	Extérieur uniquement Difficultés milieu urbain
GSM	Tout le monde à un téléphone	Précision dépendant de la densité des relais GSM Dépendant de l'opérateur
WIFI	Bonne précision Utilise réseau existant Terminaux compatibles	Calibrage Logiciel clients à déployer
RFID	Tage RFID passif peu cher	Déploiement matériel de capteur Position disponible au checkpoints
IP	Gratuit	Précision ville,necessite base de données géographique des IPs (payantes)

Tableau I.1. les avantages et les inconvénients de chaque technique.

I.7 Les Services de la Géolocalisation

La géolocalisation peut se révéler utile, voir primordiale, dans de nombreux secteurs d'activités. Les finalités et objectifs fixés sont différents selon ces secteurs, mais grâce à une adaptabilité forte telle que, la géolocalisation répond aux attentes variées d'un grand nombre de professions [4].

- Garder le contact.
- La surveillance.
- Transport de passagers.
- Suivi de travailleurs mobiles.
- Sécurité.

I.8 Conclusion

Dans ce chapitre, nous avons rappelé quelques éléments concernant les principales techniques existantes utilisées dans les systèmes de localisation ainsi que les différentes applications visées. Puis, nous avons présenté les différentes techniques de localisation existantes.

Dans le chapitre suivant, nous allons présenter les fondements théoriques utilisés pour l'étude du système informatique géographique.



CHAPITRE II
SYSTÈME D'INFORMATION
GÉOGRAPHIQUE

II.1 Introduction

Les systèmes d'informations géographiques (SIG) regroupent différentes méthodes et techniques informatiques, permettant de modéliser, de saisir sous forme numérique, de stocker, de gérer, de consulter, d'analyser, de représenter des objets ou des collections d'objets géographiques, avec la particularité essentielle de prendre en compte les caractéristiques spatiales de ces objets au même titre que les attributs descriptifs qui y sont attachés. En effet, la dénomination « SIG » recouvre une grande variété de réalisations logicielles construites suivant des choix techniques différents, aux fonctionnalités et aux performances très diverses.

Les systèmes d'informations géographiques ont la particularité de faire appel à de nombreux domaines scientifiques et techniques et à de nombreuses méthodes, allant de la géodésie aux systèmes de gestion de bases de données, en passant par le traitement d'images, la modélisation et l'interpolation géométrique, la statistique, la cartographie automatique, l'analyse spatiale, ...etc. Construire un système d'information géographique sans s'éloigner de la rigueur scientifique est une tâche complexe, aussi bien en termes de définition des concepts, d'organisation fonctionnelle, d'architecture logicielle, d'algorithmique, d'ergonomie...etc.

II.2 La définition d'un SIG

Un SIG ou Système d'Information Géographique est un système d'information capable d'organiser et de présenter des données alphanumériques spatialement référencées, ainsi que de produire des plans et des cartes. Ses usages couvrent les activités géomatiques de traitement et diffusion de l'information géographique. On peut enfin définir un SIG comme étant un système informatisé capable de représenter sous forme numérique un territoire ou une partie d'un territoire [5].



Informations localisées (monde réel)

Base de données géoréférencées

Figure II.1. Modèle de SIG.

II.3 Composants d'un SIG

Un SIG est constitué de quatre (04) composantes majeures qui sont Les logiciels, les données, les matériels, les personnels.



Figure II.2. Composants d'un SIG[6].

II.3.1 Le matériel

Il s'agit d'une composante indispensable à un S.I.G ou SIG et l'élément fondamental de cette composante reste l'ordinateur. Les SIG fonctionnent grâce aux ordinateurs et périphériques connectés entre eux ou non et permettant aux utilisateurs d'avoir toutes les fonctionnalités des SIG regroupées autour des 5A (abstraction, analyse, acquisition, affichage, archivage). Etant donné la forte taille des données stockées dans un SIG, les supports de stockage occupent une place de choix dans le matériel des SIG. Des systèmes client-serveur en intranet, extranet voire via Internet facilitent la diffusion des résultats. Ces solutions de diffusion appelées web-mapping ou web-SIG utilisent un serveur cartographique.

II.3.2 Le logiciel

Les logiciels de SIG offrent des outils et des fonctions pour stocker, analyser et afficher toutes les informations. Ce sont des outils pour saisir et manipuler et pour stocker les informations géographiques, les bases de données (SGBD), pour des requêtes, analyse et

visualisation via des interfaces graphiques utilisateurs pour une utilisation facile. Bref les logiciels permettent de rendre cohérents les trois dimensions d'un SIG.

II.3.3 Les données

Les données représentent le contenu même des SIG. Elles peuvent être des cartes géographiques et/ ou des informations relatives à ces objets. C'est la composante la plus importante d'un SIG. Les données géographiques peuvent être, soit importées à partir de fichiers, soit saisies par un opérateur.

II.3.4 Le personnel

Un SIG est avant tout un système et ce sont ses utilisateurs (le personnel qui entretient et gère le système) qui lui permettent de fonctionner pour livrer tout son potentiel au besoin de l'homme.

Les utilisateurs et potentiels utilisateurs d'un SIG sont principalement :

- les techniciens et ingénieurs chargés de la conception, de l'entretien et de la gestion du SIG.
- les techniciens et personnels qualifiés à l'utilisation quotidienne du SIG dans leur travail.
- les décideurs utilisant le SIG comme moyen d'aide à la prise des décisions.

II.4 Typologie des données SIG

Les données SIG sont classées en deux grands types à savoir les données géographiques et les données attributaires [5].

II.4.1 Les données géographiques

Une donnée est dite géographique si on peut la localiser soit directement par ses coordonnées géographiques ou indirectement par son adresse postale ou par son identifiant cadastral. Les données géographiques peuvent être présentées par trois subdivisions à savoir les données géométriques, les données graphiques et les métadonnées.

- ✚ Les données géométriques : Elles renvoient à la forme et à la localisation des objets ou phénomènes.

Toute représentation cartographique passe par la traduction des éléments réels que l'on observe (infrastructure de santé, route, limite administrative,) en objets géométriques qui sont de trois types :

- le point (x, y) : par exemple, les lieux de résidence des sujets atteints de la pathologie étudiée (cas) géoréférencés et représentés sur une carte ou encore les sites industriels, les hôpitaux, etc.
- la ligne $((x_1, y_1), \dots, (x_n, y_n))$: les routes, les cours d'eau, les lignes à haute tension, tout élément d'un réseau, etc.
- le polygone ou surfacique : Exemples les limites administratives, les sites industriels étendus, les nappes d'eau souterraines, etc.

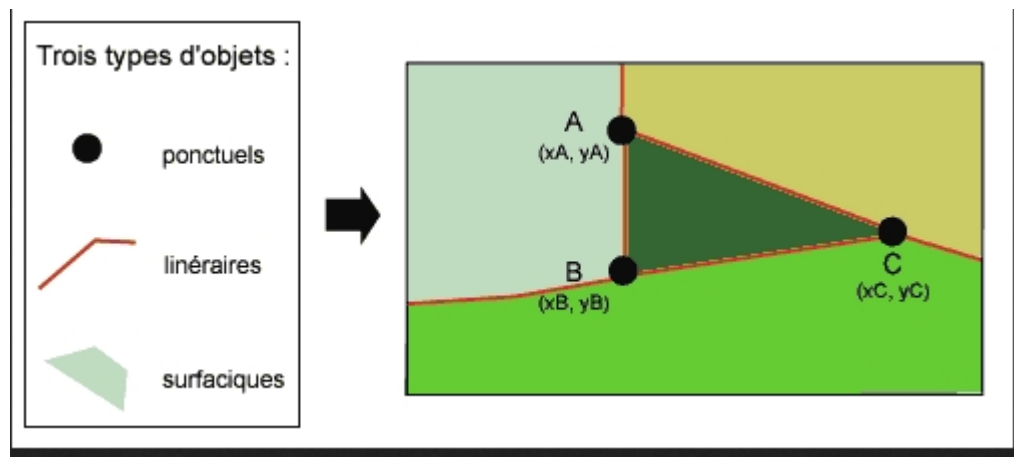


Figure II.3. Les types d'objet [1].

Remarque : X et Y représentent les coordonnées géographiques du point. Il s'agit ici des coordonnées terrestres qui considèrent la terre comme une sphère, mais une sphère imparfaite soit un ellipsoïde. L'ellipsoïde est la surface mathématique qui permet de calculer les coordonnées géographiques d'un lieu en Longitude et en latitude.

- ✚ Les données graphiques : Elles caractérisent la présentation apparente de l'objet.
- ✚ Les métadonnées : Ce sont les informations sur l'origine et le propriétaire d'une donnée géographique.

II.4.2 Les données attributaires

Elles représentent les caractéristiques ou propriétés propres à un objet ou à un phénomène en dehors de sa forme ou de sa localisation. Il peut s'agir par exemple des informations portant sur l'adresse postale d'un site géographique.

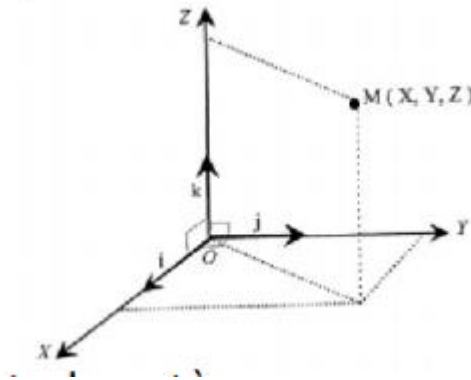


Figure II.4. Latitude et Longitude d'un lieu sur la terre [7].

II.5 Représentation des Données SIG

Il existe deux modes de représentation des données géographiques : le mode vecteur et le mode raster.

II.5.1 Représentation en mode vectoriel

En mode vectoriel, le point avec ses coordonnées est le porteur de l'information géométrique. Les lignes et les surfaces se comprennent comme une suite définie de points caractéristiques. Les données vectorielles sont la plupart du temps le résultat de la numérisation manuelle ou semi-automatique. De façon générale, dans les données vectorielles on distingue les points, les lignes et les surfaces (polygone) qui sont toujours représentés en couches différentes.



Figure II.5. Représentation des données vectorielles[7].

II.5.2 Représentation en mode raster

Les données raster ont comme élément essentiel le pixel (Picture élément). Les pixels sont répartis dans un raster de façon régulière en lignes et colonnes comme indique la figure ci-

dessous. Les lignes et les surfaces ne peuvent être représentées que par l'enchaînement de pixel unique. Un objet ne peut donc être représenté que de façon approximative; c'est ainsi que la taille du pixel-raster (résolution spatiale) conditionne l'exactitude de la représentation.

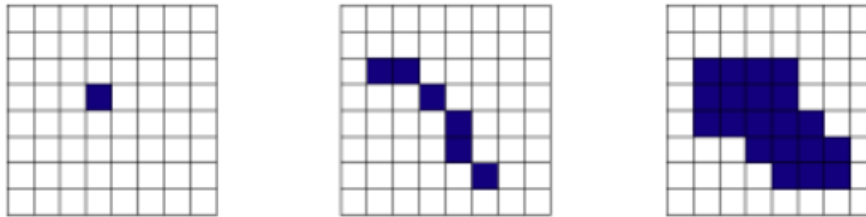


Figure II.6. Représentation des données Raster [7].

Remarque : La donnée raster ou maillée donne une information en chaque point du territoire.

Les données raster sont issues essentiellement de la photographie aérienne, des images satellites, ou d'un plan scanné affichés dans le SIG en tant qu'image.

II.5.3 Comparaison entre les deux modes

Chaque mode a des avantages et des inconvénients comme indique le tableau suivant [8].

	Mode raster	Mode vecteur
Avantages	<ul style="list-style-type: none"> • Structure de données simple. • Superposition et combinaison de données aisée (analyse spatiale). • Représentation des variables spatialement continue. 	<ul style="list-style-type: none"> • Bonne représentation de la structure. • Topologie. • Excellente représentation graphique. • Couplage avec BD relationnelle.
Inconvénients	<ul style="list-style-type: none"> • Gros volume de données. • Résolution dépendante du phénomène étudié. • Aspect visuel médiocre des documents. • Faible précision spatiale. 	<ul style="list-style-type: none"> • Structure de données complexe. • Opération de superposition, analyse spatial limitée. • Faible résolution thématique.

Tableau II.1. Tableau représente les avantages et les inconvénients de chaque mode.

II.6 Fonctionnalités d'un SIG

Les systèmes d'informations géographiques possèdent des fonctionnalités regroupées en cinq familles connues sous les 5A qui sont :

- **Acquisition** : C'est l'opération qui permet d'intégrer des données géographiques dans le système.
- **Archivage** : structuration et stockage de l'information géographique sous forme numérique.
- **Abstraction** : C'est l'opération qui permet de représenter une situation géographique du monde réel et complexe par un système simple et suffisamment précis et compréhensif.
- **Analyse** : C'est l'opération qui permet d'effectuer des traitements ou des interprétations liées à la géométrie des objets (exemple calcul d'itinéraire, croisement des données thématique en couches.)
- **Affichage** : représentation et mise en forme, notamment sous forme cartographique avec la notion d'ergonomie et de convivialité.

Les cinq fonctionnalités de base du SIG lui permettent de créer une plateforme virtuelle des informations du monde réel sous forme de couches thématiques superposables au moyen des cartes géographiques. Cette technologie à la fois et simple et puissant (à la base du fonctionnement de toute application SIG digne de ce nom) a démontré sa puissance de résolution des problèmes complexe du monde réel. Il est illustré par la figure ci-dessous :

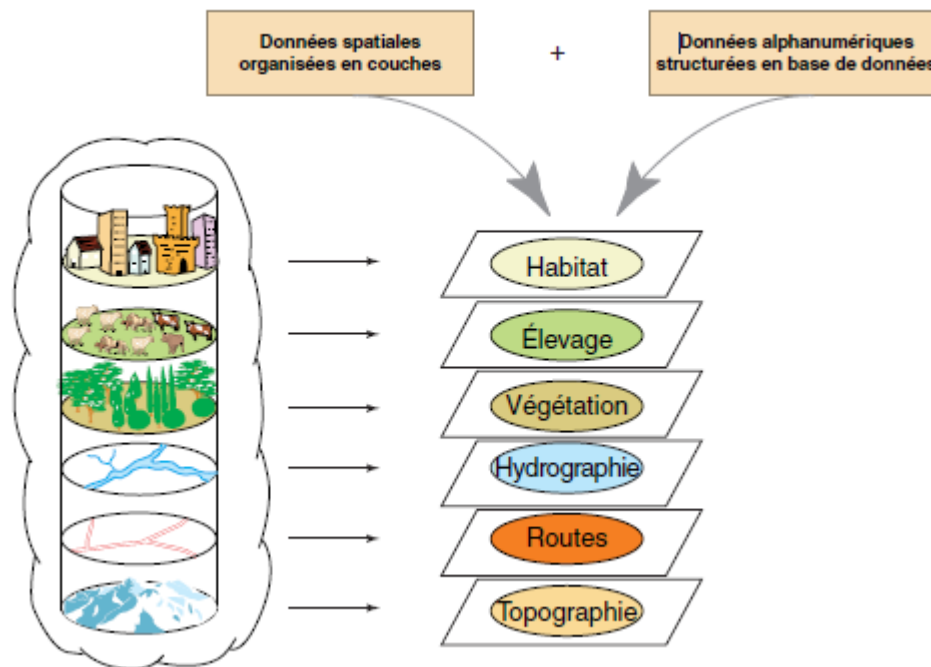


Figure II.7. La plateforme virtuelle [1].

II.7 Les domaines d'application des SIG

L'utilisation d'un SIG sera à priori pertinente dans tous les domaines dans lesquels des entités, objets, informations localisables sur un territoire, permettent par leur connaissance d'aboutir à des processus de décision[1].

Le SIG s'applique à plusieurs domaines, notamment :

- la planification urbaine (cadastre, voirie, réseaux assainissement)
- la gestion des forêts (cartographie pour aménagement, gestion des coupes et sylviculture)
- le transport (planification des transports urbains, optimisation d'itinéraires).
- le tourisme (gestion des infrastructures, itinéraires touristiques).
- la protection civile (gestion et prévention des catastrophes).
- le marketing (localisation des clients, analyse du site).
- les télécommunications (implantation d'antennes pour les téléphones mobiles).
- la biologie (études du déplacement des populations animales).

II.8 Avantages de SIG

Le SIG a plusieurs avantages parmi lesquelles nous citons :

- abaisser les couts de productions des cartes et des plans.
- permet de les établir plus rapidement et à moindre frais.
- permet aussi d'établir des cartes et des plans que l'on ne pouvait pas réaliser à la main grâce à l'information, il est possible de réaliser des produits nouveaux qu'il était impossible de réaliser à la main.
- évite d'avoir à refaire plusieurs fois les mêmes levers.il évite que des services différent procèdent à des levers topographiques sur la même zone et évite les pertes d'informations avec le temps en accumulant l'information recueille sur le terrain.
- Lorsque le SIG est en place, installer une nouvelle application nécessite un investissement modeste et le retour sur investissement est rapide.
- facilite la réalisation d'étude pour tous les projets ayant une composante géographique.il permet de multiplier les représentations visuelles et facilite ainsi la prise de décision tout en diminuant les risques d'erreurs.
- Améliore le service rendu à l'usager en permettant de lui fournir avec rapidité et fiabilité une information de qualité dont il a besoin.par exemple, tous les renseignements délivrés par le service urbanisme seront, en principe, à jour et complets.
- Permet des calculs utiles à la prise de décision.Cela va du calcul simple, la superposition cartographique, au calcul complexe d'analyse spatiale intégrant un grand nombre de paramètres [9].

II.9 Conclusion

De nos jours avec l'augmentation de l'utilisation de l'ordinateur individuel/portable, les SIG sont devenu accessibles.

Dans ce chapitre nous avons détaillé le SIG et les applications qui sont devenus plus facile à utiliser.un SIG est bien plus qu'un logiciel, il se référé à tous les aspects de gestion et d'utilisation de données de domaines par projection géographiques numériques.

L'outil SIG reste jusqu' à lors, un outil préféré pour la conception des idées et des simulations des situations selon des stratégies locales ou globales.

L'utilisation des nouvelles technologies, comme les techniques de traitement d'image, qui feront l'objet du prochain chapitre, permettent d'améliorer les résultats du système d'information géographique.



CHAPITRE III
TRAITEMENTS D'IMAGES

III.1 Introduction

Le traitement numérique des images est utilisé dans beaucoup des domaines, tel que le tri automatique du courrier, le contrôle industriel, la compression, ...etc. La croissance exponentielle des capacités des processeurs est très gourmande en puissance de calcul pour une grande part dans le développement du traitement d'images.

Le but principale de traitement d'image c'est l'amélioration de la qualité d'image et d'extraire les objets voulu pour arriver à ce but il faut appliquer plusieurs opérations comme le filtrage ..., pour faire ces opérations on a besoin d'un logiciel tel que MATLAB (qu'on va utiliser dans le chapitre suivant)

III.2 Traitement d'image

Le traitement d'images est une discipline de l'informatique et des mathématiques appliquées, qui étudie les images numériques et leurs transformations, pour but d'améliorer leur qualité ou d'en extraire de l'information.

- **Traitements de bas niveau** : données de nature numérique suppression du bruit, augmentation de contraste, lissage, rehaussement des contours...
- **Traitements de haut niveau (analyse)** : extraction d'informations.

La compréhension du traitement d'images commence par la compréhension de ce qu'est une image.

III.3 Définition de l'image numérique

Une image numérique est une image (dessin, icône, ...) créée, traitée, stockée sous forme binaire (suite de 0 et de 1).

Une image est composée d'un ensemble des points appelés PIXEL (Picture Élément : le plus petit élément qui compose l'image). La position de chaque pixel peut être exprimée par deux coordonnées sur l'axe horizontal X et l'axe vertical Y comme le montre la figure III.1 ci-dessous [10].

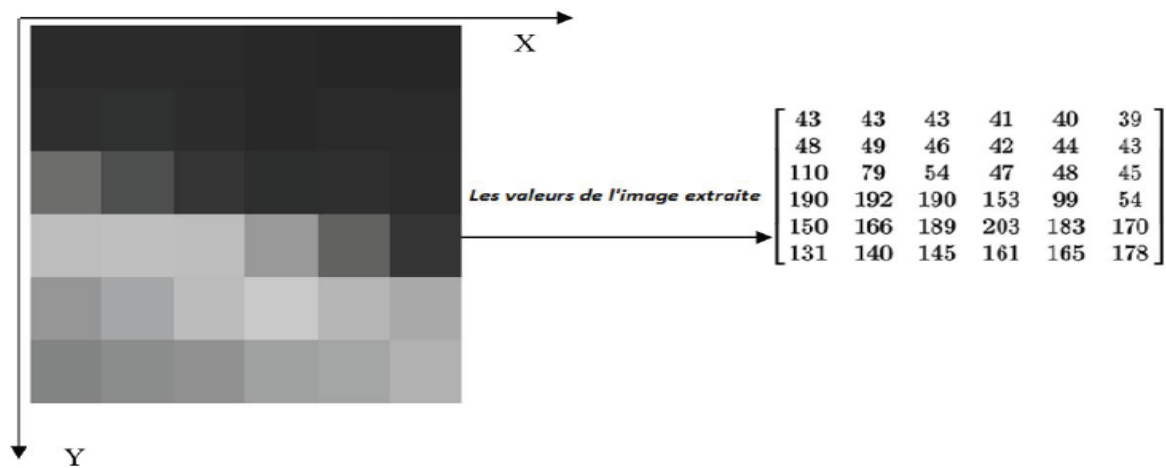


Figure III.1. Les pixels d'une image.

III.3.1 Types d'image numérique

On peut classer les images numériques selon deux critères :

III.3.1.1 Selon la représentation numérique

Ils existent deux types :

➤ Image vectorielle

Dans ce genre d'image, l'image numérique est composée d'objets géométriques individuels comme les polygones, segments de droite ...etc.

Chaque objet est défini par une forme, position, couleur ...etc.

➤ Image matricielle

Comme son nom indique l'image matricielle est composée d'un tableau (matrice) de point à plusieurs dimensions, chaque dimension représente une dimension spatiale (hauteur, largeur, profondeur), temporelle ou autre (un niveau de résolution par exemple).

La différence entre l'image vectorielle et l'image matricielle est que dans cette dernière on travaille sur des pixels.

III.3.1.2 Selon la représentation des couleurs

Il existe plusieurs modes de codage informatique des couleurs parmi lesquels nous citons :

➤ LES MODÈLES DE COULEUR

La couleur est l'une des composantes principales pour la description et l'analyse des images couleurs dans le domaine de la recherche d'images par le contenu. Comme nous le savons, les couleurs perceptibles par l'œil humain représente une petite gamme de l'ensemble du spectre électromagnétique qui représente tous les rayons, depuis les rayons cosmiques et les rayons X jusqu'aux ondes électriques (Figure III.2).

Les couleurs visibles par l'œil humain s'étendent entre les longueurs d'ondes allant de 4000 à 7000 angströms, représentant respectivement les couleurs : violet et rouge et toutes les couleurs entre les deux. Toutes les autres ondes s'étendant des rayons cosmiques des étoiles jusqu'aux ondes FM de nos radios ne peuvent pas être perçues par l'œil humain. Cette petite gamme du spectre est l'espace de couleur perceptible par l'œil humain.

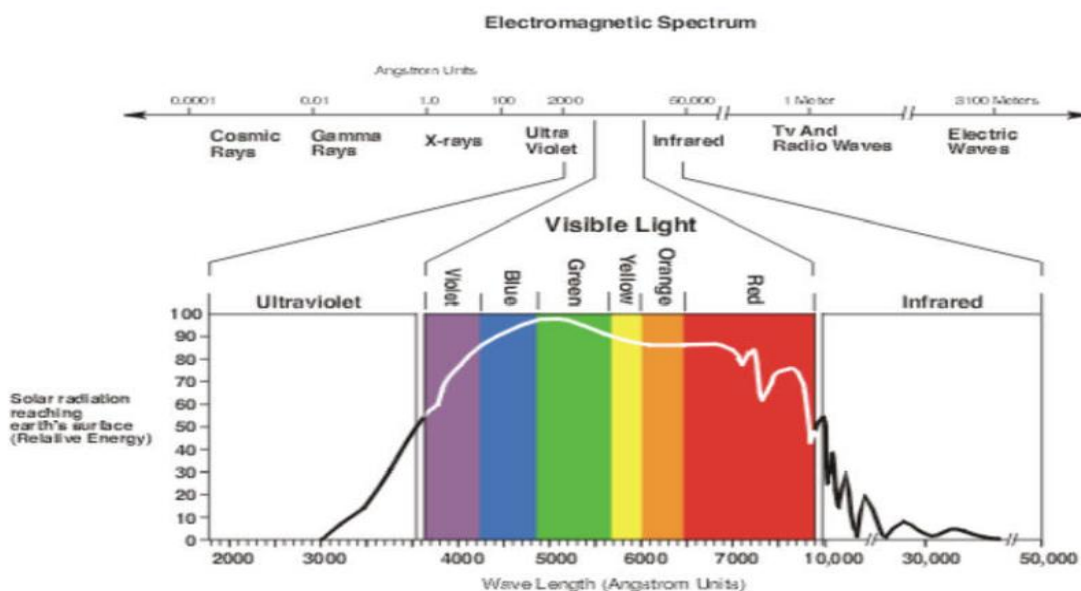


Figure III.2. Le spectre électromagnétique.

➤ **Niveau de gris**

Le niveau de gris minimal est 0. Le niveau de gris maximum dépend de la profondeur de numérisation de l'image. Pour une image de 8 bit c'est 255. Dans une image binaire un pixel peut donc avoir soit la valeur 0 soit la valeur 255. Par contre dans une image des niveaux de gris ou une image en couleurs un pixel peut adopter chaque valeur entre 0 et 255.

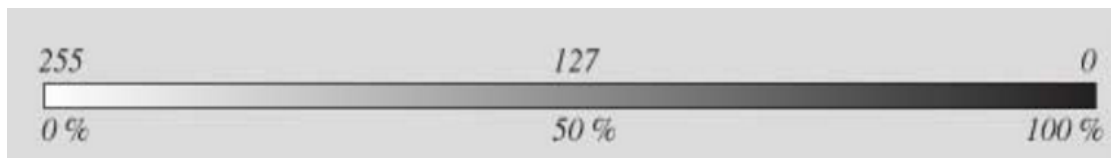


Figure III.3. L'intervalle du niveau de gris

En cas d'une image en couleur le niveau de gris de chaque pixel peut être calculé par la formule suivante :

Niveau de gris = 0,299 * composante rouge + 0,587 * composante verte + 0,114 * composante bleue

La formule tient compte de la sensibilité de l'œil à la couleur. Ainsi la présentation des niveaux de gris devient indépendante de couleur et se limite uniquement à la luminosité des pixels individuels.

Un histogramme de niveau de gris indique combien de pixels d'une image partagent le même niveau de gris. L'axe des x démontre les niveaux de gris (p.ex. de 0 à 255), l'axe des y démontre leur fréquence dans l'image. Cette information peut être utilisée à calculer un seuil.

III.3.2 Format d'une image

Un format d'image est une représentation informatique de l'image, associée à des informations sur la façon dont l'image est codée et fournissant éventuellement des indications sur la manière de la décoder et de la manipuler.

La plupart des formats sont composés d'un en-tête contenant des attributs (dimensions de l'image, type de codage, etc.), suivi des données (l'image proprement dite). La structuration des attributs et des données diffère pour chaque format d'image.

De plus, les formats actuels intègrent souvent une zone de métadonnées (*metadata* en anglais) servant à préciser les informations concernant l'image comme :

- La date, l'heure et le lieu de la prise de vue.
- les caractéristiques physiques de la photographie (sensibilité ISO, vitesse d'obturation, usage du flash...)

Ces métadonnées sont par exemple largement utilisées dans le format .Exif (extension du format . JPEG), qui est le format le plus utilisé dans les appareils photo numériques.

<i>format</i>	<i>Type (matriciel/vectorel)</i>	<i>Compression des données</i>	<i>Nombre de couleurs supportées</i>	<i>Affichage progressif</i>	<i>animation</i>
JPEG	Matriciel	Oui, réglable (avec perte)	16 millions	oui	non
JPEG 2000	Matriciel	Oui, (avec ou sans perte)	32 millions	oui	oui
GIF	Matriciel	Oui, sans perte	256 maxi (palette)	oui	oui
PNG	Matriciel	Oui, sans perte	Palettisé (256 couleurs ou moins) ou 16 millions	oui	non
TIFF	Matriciel	Compression ou pas avec ou sans perte	De monochrome à 16 millions	non	non
SVG	vectorel	Compression possible	16 millions	Ne s'applique pas	oui

Tableau III.1. Tableau comparatif de différents formats d'images.

III.3.3 Caractéristiques d'images

L'image est un ensemble structuré d'informations caractérisé par les paramètres suivants [11] :

- **Pixel**

Le pixel, abrégé px, est une unité de surface permettant de définir la base d'une image numérique. Son nom provient de la locution anglaise Picture élément, qui signifie, « élément d'image » ou « Point élémentaire ».

- **La texture**

Une texture est une région dans une image numérique qui a des caractéristiques homogènes.

Ces caractéristiques sont par exemple un motif basique qui se répète, ou des caractéristiques fréquentielles. Une texture est composée de texels (texture element ou texture pixel), l'équivalent des pixels.

- **Dimension**

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

- **Résolution**

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

- **Bruit**

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

- **Histogramme**

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image. Pour diminuer l'erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image, on modifie souvent l'histogramme correspondant.

Il permet de donner un grand nombre d'information sur la distribution des niveaux de gris (couleur) et de voir entre quelles bornes est répartie la majorité des niveaux de gris (couleur) dans les cas d'une image trop claire ou d'une image trop foncée.

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

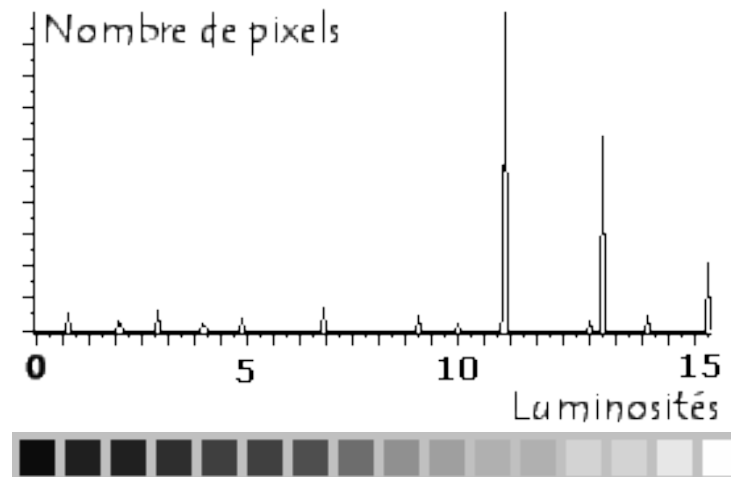


Figure III.4. L'histogramme de niveau de gris.

- **Contours**

Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative. Les textures décrivent la structure de ceux-ci. L'extraction de contour consiste à identifier dans l'image les points qui séparent deux textures différentes.

- **Luminance**

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

- **Contraste**

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images.

III.4 Système de traitement d'image

Dans le contexte de la vision artificielle, le traitement d'images se place après les étapes d'acquisition et de numérisation, assurant les transformations d'images c'est la partie de calcul permettant d'aller vers une interprétation des images traitées. Cette phase d'interprétation est d'ailleurs de plus en plus intégrée dans le traitement d'images, en faisant appel notamment à l'intelligence artificielle pour manipuler des connaissances, principalement sur les informations dont on dispose à propos de ce que représentent les images traitées (connaissance du domaine).

III.4.1 Acquisition et numérisation

C'est le mécanisme qui permet l'obtention d'une image numérique (représentée par une matrice) à deux dimensions à partir d'une scène à trois dimensions, en passant par un système optique, l'image continue $f(x, y)$ est approximée par des échantillons qui sont obtenus par discrétisation des coordonnées (x, y) (ce qu'on appelle l'échantillonnage) et la discrétisation des amplitudes de ces points là (c'est la quantification).

La représentation obtenue ne peut être parfaite à cause du bruit introduit dans l'image lors de son acquisition.

III.4.2 Visualisation

Un dispositif de visualisation permet l'affichage de l'image. Les reconstituteurs permettent de transformer le signal numérique qui est la matrice image en un signal analogique visible à l'œil humain pour cela on dispose d'une multitude de supports qui sont employés (moniteur vidéo, impression sur papier...).

III.4.3 Filtrage

On peut classer les filtres en deux grandes catégories :

III.4.3.1 Filtres linéaires

Les premières et les plus simples méthodes de filtrage sont basées sur le filtrage linéaire, chacun de ses opérateurs est caractérisé par sa réponse impulsionnelle $h(x, y)$, l'expression qui relie les entrées avec les sorties est donnée par la relation suivante :

$$S(i, j) = \sum_{u, v} (E(i, j) \cdot h(i-u, j-v)) = h(i, j) * E(i, j)$$

Où : u, v varient de moins l'infini à plus l'infini.

Le filtrage linéaire est un produit de convolution c à d une combinaison linéaire du voisinage du pixel concerné.

Les différents types des filtres linéaires sont :

➤ **Filtre moyenneur (lissage)**

L'intensité du pixel considéré est remplacée par la moyenne des pixels de son voisinage, la taille de la zone (fenêtre) entourant le pixel est un paramètre important, plus cette dimension est grande, plus Sa sensibilité au bruit diminue, et le lissage devient important (le flou s'accroît).

Le filtre moyenneur est un filtre passe-bas c à d qu'il laisse passer les basses fréquences (les faibles changements d'intensité de l'image) et atténue les hautes fréquences (variations rapides).

Exemple :



Image original

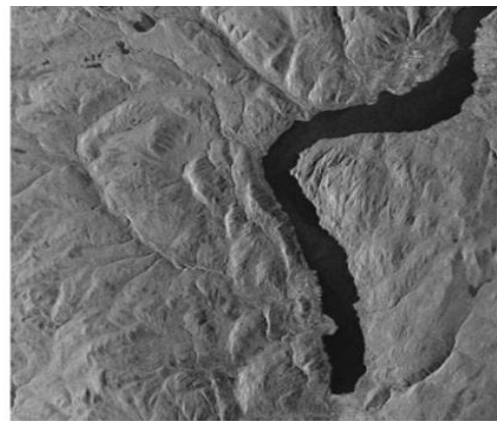


Image filtrée

Figure III.5. Application du filtre moyen (lissage).

➤ **Filtre gaussien**

L'expression gaussienne en deux dimensions est donnée par :

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

L'intérêt de ce filtre est que l'on contrôle facilement le degré de filtrage à travers le paramètre σ (σ : l'écart type de la gaussienne & x, y : sont les positions d'un pixel). Le filtre gaussien donne plus d'importance aux pixels proches du pixel central, et diminue cette importance au fur et à mesure que l'on s'éloigne de celui-ci, il a les mêmes inconvénients que le filtre moyenné c à d il dégrade les contours.

La discrétisation de ce filtre pour un σ égale à 0.6 donne le masque suivant :

$$1/16 \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Masque du filtre.

➤ **Filtre rehausseur de CONTOURS**

C'est un filtre passe haut c à d il met en évidence les changements rapides de l'intensité de l'image (les hautes fréquences) et laisse les zones uniformes inchangées (basses fréquences).

-1	-1	-1
-1	9	-1
-1	-1	-1

Masque du filtre.

Exemple :



Figure III.6. Rehaussement de contour.

III.4.3.2 Filtrage non linéaire

Ils sont conçus pour régler les problèmes des filtres linéaires, sur tout pour ce qui concerne la mauvaise conservation des contours. Leur principe est le même que celui des filtres linéaires, il s'agit toujours de remplacer la valeur de chaque pixel par la valeur d'une fonction calculée dans son voisinage, la seule différence c'est que cette fonction n'est plus linéaire mais une fonction quelconque (elle peut inclure des opérateurs de comparaisons).

➤ Filtre médian

Sur un voisinage à huit, le nouveau niveau de gris du pixel centre est choisi comme étant la valeur médiane de tous les pixels de la fenêtre d'analyse centrée sur ce dernier. Son avantage est qu'il garde la netteté des éléments qui constituent l'image sans étaler les transitions.



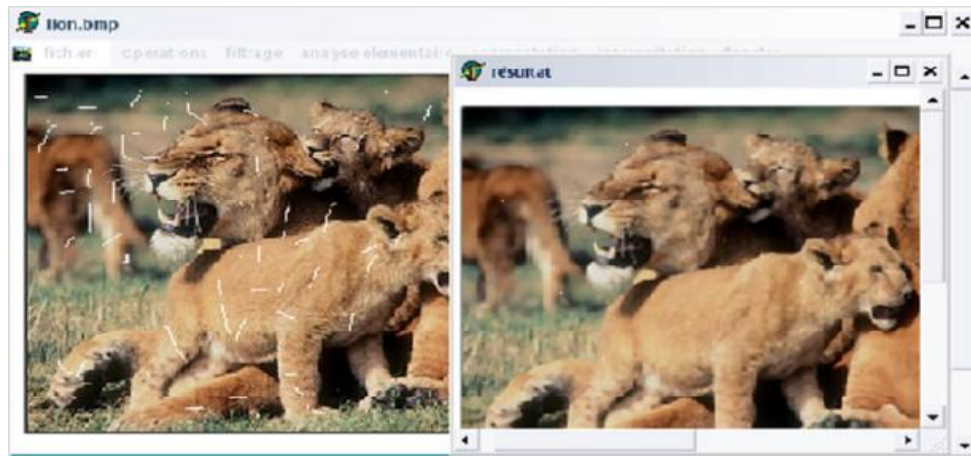


Figure III.7. Principe du filtre médian.

➤ **Filtre maximum**

Même principe mais la valeur choisie est la valeur maximale.



Figure III.8. Principe du filtre maximum.

➤ **Filtre minimum**

Même chose, sauf que la valeur choisie est la valeur minimale.

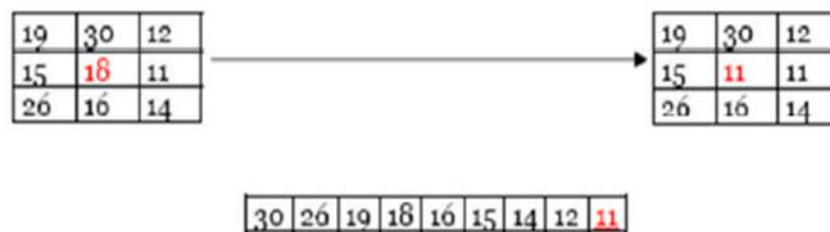


Figure III.9. Principe du filtre minimum.

III.4.4 Segmentation d'une image

III.4.4.1 Définition

La segmentation d'image est une opération de traitement d'images qui a pour but de rassembler des pixels entre eux suivant des critères prédéfinis. Les pixels sont ainsi regroupés en régions, qui constituent un pavage ou une partition de l'image. Il peut s'agir par exemple de séparer les objets du fond. Si le nombre de classes est égal à deux, elle est appelée aussi binarisation.

Si l'homme sait naturellement séparer des objets dans une image c'est grâce à des connaissances de haut niveau (compréhension des objets et de la scène). Mettre au point des algorithmes de segmentation de haut niveau (chaque région est un objet sémantique) est encore un des thèmes de recherche les plus courants en traitement d'images.

III.4.4.2 Méthodes de segmentation

La segmentation est une étape primordiale en traitement d'image. À ce jour, il existe de nombreuses méthodes de segmentation, que l'on peut regrouper en quatre principales classes :

➤ **Segmentation par région**

La segmentation d'image par l'approche région consiste à découper l'image en régions. Les pixels adjacents sont regroupés en régions distinctes selon un critère d'homogénéité ou de similarité donnée. Ce critère peut être, par exemple, le niveau de gris, couleur, texture...etc.

Un processus de groupement est répété jusqu'à ce que tous les pixels dans l'image soient inclus dans des régions. Cette approche vise donc, à segmenter l'image en se basant sur des propriétés intrinsèques des régions.

➤ **Segmentation par contours**

La détermination des limites des objets dans une scène présente un intérêt primordial pour le traitement de l'image. La détection de contours est ainsi un sujet de recherche très important, et les contours ont été et sont toujours beaucoup utilisés comme primitives en analyse d'images et en vision par ordinateur.

Tout d'abord le contour peut être pris comme le bord ou la frontière de deux régions (objets), et la détection des contours est équivalente à détecter les discontinuités à la frontière de deux régions.

Un contour est un ensemble de pixels formant une frontière entre deux ou plusieurs régions voisines, l'épaisseur d'un contour est d'un ou plusieurs pixels et il est défini par une variation "rapide" de caractéristique.

➤ **Segmentation par coopération région-contour**

La segmentation par coopération région contour peut être exprimé comme une entre-aide entre ces deux concepts afin d'améliorer le résultat final. C'est une approche qui peut être une solution pour pallier aux inconvénients des deux approches.

➤ **Segmentation par classification**

Dans cette section nous considérons les pixels de l'image comme étant les objets à classer. Les techniques de la segmentation par classification fournissent une partition de l'image en regroupant des pixels ayant des niveaux de gris similaires dans une même classe de pixels. Les classes sont définies par les ensembles maximaux de pixels connexes appartenant à la même classe.

III.5 Conclusion

Dans ce chapitre on a défini l'image et le traitement d'image. Le traitement d'image permet d'analyser et de comprendre l'image dans le but d'améliorer la qualité d'image et de tirer les informations.

Nous avons cité les différentes opérations qu'on peut appliquer sur une image pour arriver à notre but.

Le chapitre suivant est une application informatique auquel on va choisir une image et appliquer les différentes opérations de traitement d'images pour géolocaliser un objet que nous voulons.



CHAPITRE IV

DESCRIPTION DE L'APPLICATION

IV.1 Introduction

Nous avons utilisées pour notre application des images numérique satellitaire sous forme matricielle, Cette forme numérique va nous permettre d'appliquer un grand nombre des techniques de traitements et d'analyse d'images. Le résultat de ces traitements sera une nouvelle matrice représentant une image optimisée (améliorée).

L'utilisation de l'outil informatique pour faire le traitement d'images est devenu une nécessité à cause du rôle qu'il présente.

L'objectif de notre travail est de développer une application informatique, sous logiciel MATLAB dans laquelle on va appliquer les différentes techniques des traitements d'images (filtrage, segmentation et détection de contour...) sur des images satellitaires de la zone que nous avons choisi « BARRAGE HAMMAM BOUGHRARA ». Enfin, nous allons présenter les résultats que nous avons obtenus pour chaque opération.

IV.2 Présentation du logiciel MATLAB

MATLAB (MATRIX LABORATORY) est un logiciel de calcul scientifique permettant de développer des solutions à des problèmes techniques. Il permet aussi de réaliser du calcul numérique et de tracer des graphiques pour visualiser et analyser les données. Il dispose d'un langage et d'un environnement de programmation interactif ainsi que des outils pour concevoir des interfaces utilisateurs graphiques. MATLAB est associé à des boîtes à outils appelé TOOLBOX permettant d'accéder à des fonctions spécifiques à un domaine d'application comme le traitement d'images par exemple [12].

IV.2.1 Les commandes utilisées sur MATLAB pour réaliser cette application

Nous avons utilisées quelques commandes pour arriver à réaliser notre objectif ; ces commandes sont :

Guide : afficher une fenêtre qui nous permet d'ouvrir ou d'ajouter une nouvelle interface.

Bouton : ajouter un bouton.

Close interface : pour fermer l'interface.

Run interface 1 : pour lancer l'interface 1.

Winopen ('le nom de fichier avec son chemin') : pour ouvrir le fichier.

Imread : lire l'image.

imshow : voir l'image.

figure () : afficher dans une nouvelle fenêtre.

plot : afficher une image.

subplot : afficher plusieurs images dans une seule fenêtre.

imhist() : afficher l'histogramme d'une image.

rgb2gray : transformer l'image en niveau de gris.

sum(sum(cont)) : la fonction qui permet de calculer la surface du contour.

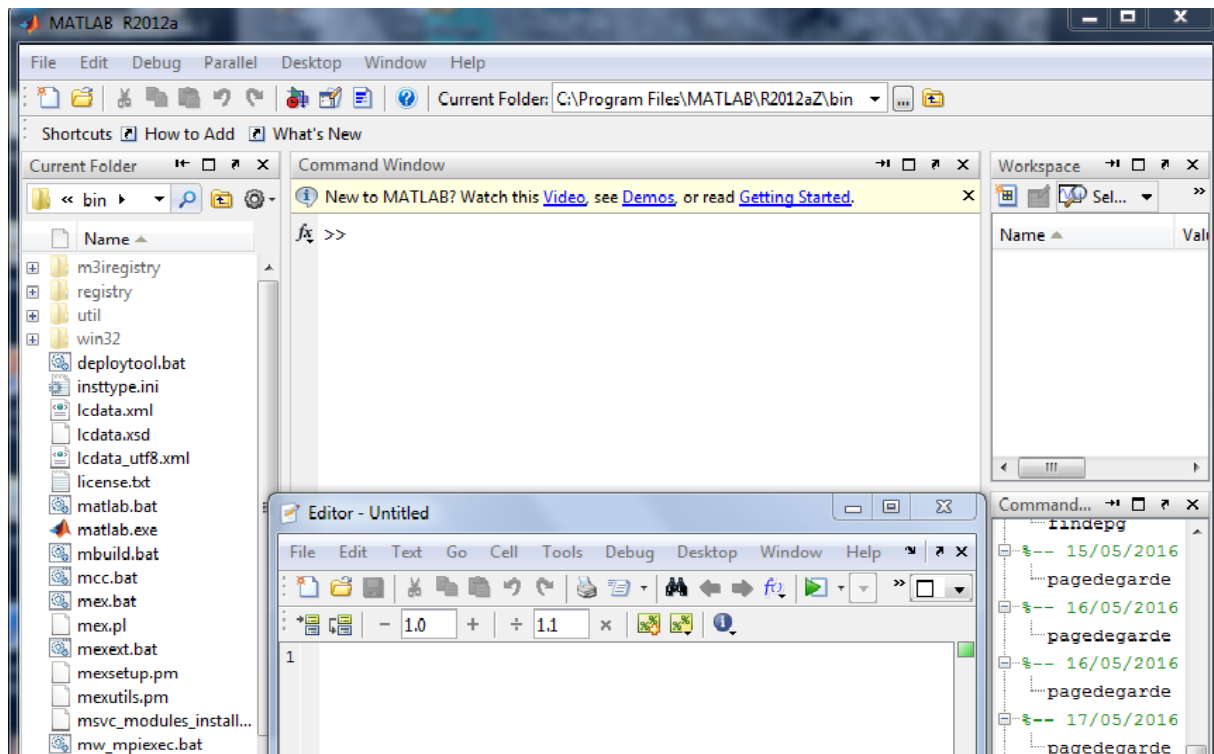


Figure IV.1. Le logiciel de calcul scientifique MATLAB.

IV.2.2 Les domaines d'application de MATLAB

MATLAB est utilisé dans plusieurs domaines nous citons :

- Mathématique, statistique et optimisation.
- Conception d'un système de contrôle et d'analyse.
- Traitement du signal et des communications.
- Traitement de l'image et de vision par ordinateur.
- Test et mesure.
- Computational Finance.

- Computational Biologie.
- génération de code.

IV.3 LaZone suivie [13]

Le barrage de Hammam Boughrara est situé sur la confluence d'oued Mouillah avec oued Tafna à 10 km en aval de la ville de Maghnia et à l'extrême Nord-Ouest de la Wilaya de Tlemcen (Nord-Ouest Algérien). Il appartient au bassin versant de l'oued Mouillah, dont la superficie (largement partagée avec le Maroc) est de 2000 km². Ce bassin est limité par un périmètre de 241 km et composé dans sa majorité par les plaines d'Angad (située à Oujda en territoire Marocain) et celle de Zrigua (située à Maghnia en territoire Algérien).

Ce réservoir est destiné à satisfaire les besoins en eau potable des villes d'Oran (33hm³) et de Maghnia (17 Hm³). Par ailleurs, 09 Hm³ sont prévus pour l'irrigation de la moyenne Tafna. Il est caractérisé par une capacité totale de 177 Hm³, un volume régularisé de 59 Hm³ et une superficie de 984 ha.

La région étudiée est caractérisée par un climat semi-aride à hiver tempéré où les précipitations moyennes annuelles sont de 325 mm, la température moyenne annuelle est de 18,5°C et l'évapotranspiration potentielle de 1167 mm. Les apports liquides mensuels sont très variables avec des moyennes maximales enregistrées au mois de novembre (6,29 Hm³) et minimales au mois de juin (0,8 Hm³). Du point de vue géologique, le bassin versant de l'oued Mouillah est constituée de terrains allant du Paléozoïque au Quaternaire. Les terrains anciens, où prédominent les formations carbonatées (Calcaires et dolomies très fissurés et karst fiés) et gréseuses, constituent les reliefs. Les dépressions sont recouvertes de terrains meubles représentés par des argiles, des marnes à gypse, des alluvions et des limons.



Figure IV.2. Barrage Boughrara.

IV.4 Les opérations du traitement d'images sur les images satellitaires de barrage BOUGHRARA.

IV.4.1 Analyse de l'image

Tout d'abord, comme une première étape de travail, nous avons fait l'analyse de nos images acquises par le logiciel googleearth des années : 2004 -2007 - 2012-2016.

Nous avons utilisé le même zoom (altitude 1720 km) et la même échelle pour toutes les images, qui sont caractérisées par une extension « .PNG ».

IV.4.1.1 Les images en niveau de gris

Les images que nous avons téléchargé en utilisant Google-Earth, sont en couleur. Nous les avons d'abord converti en niveau de gris. Le but de cette opération est de faciliter l'opération suivante (segmentation).

Exemple :

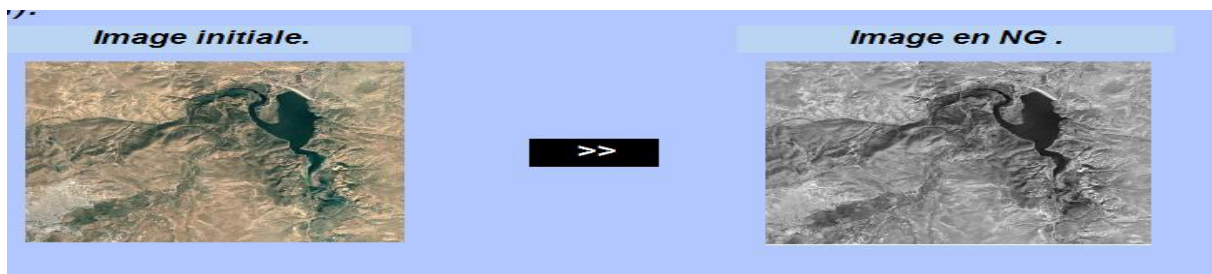


Figure IV .3. Transformation de l'image initiale en niveau de gris.

IV.4.1.2 L'Histogramme

Nous avons essayé manuellement plusieurs seuils de segmentation (le seuil représente le niveau de gris de l'objet qui nous intéresse) pour localiser l'eau à l'intérieur des images. Après simulation, nous avons opté pour les seuils suivants selon les différentes années :

- S=55 pour l'image de l'année 2004.
- S=55 pour l'image de l'année 2007
- S=88 pour l'image de l'année 2012
- S=86 pour l'image de l'année 2016

Exemple :

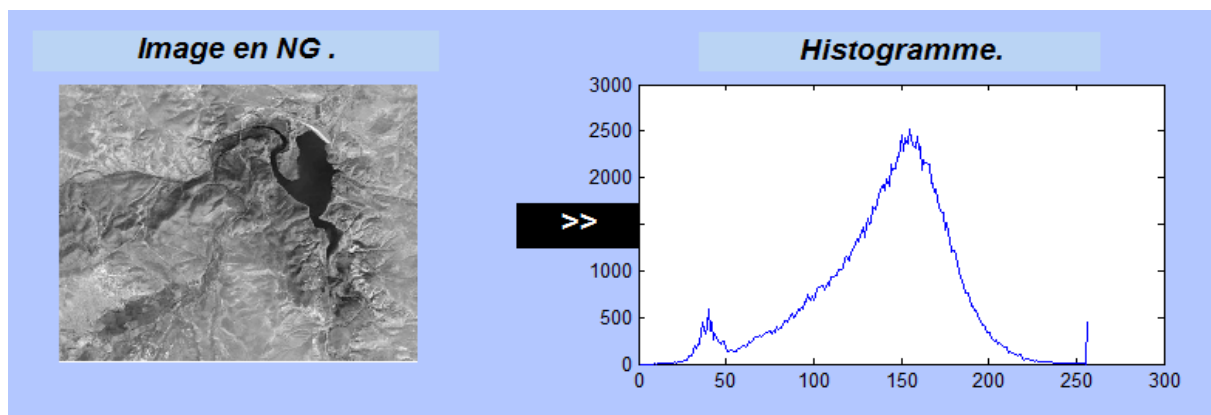


Figure IV.4. Histogramme d'image.

IV.4.2 Amélioration d'image

Dans cette étape, nous allons nous intéresser à améliorer nos images en appliquant les opérations suivantes :

IV.4.2.1 Egalisation

Il s'agit de faire une transformation des niveaux de gris qui rend l'histogramme aussi large que possible.

Exemple :

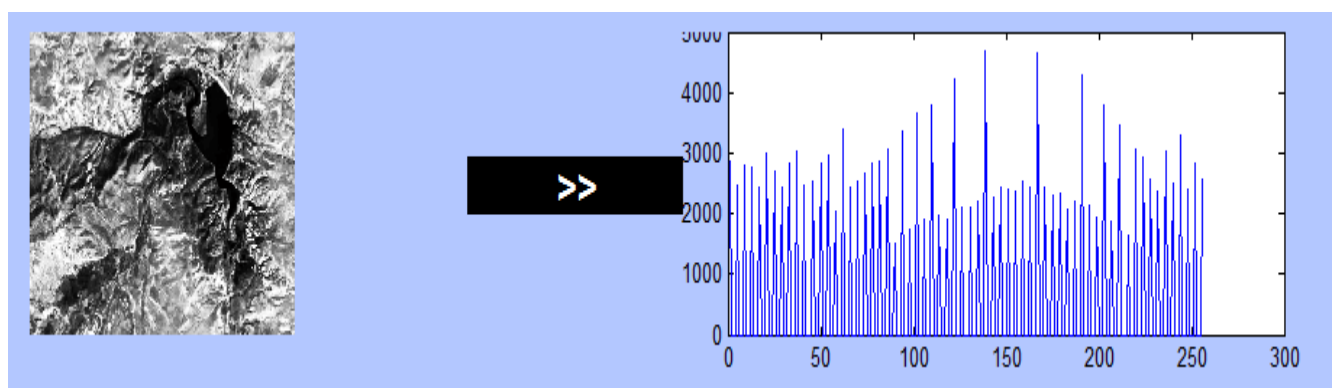


Figure IV.5. Egalisation de l'image.

Dans l'interface que nous avons créée les résultats sont affichés comme suit :

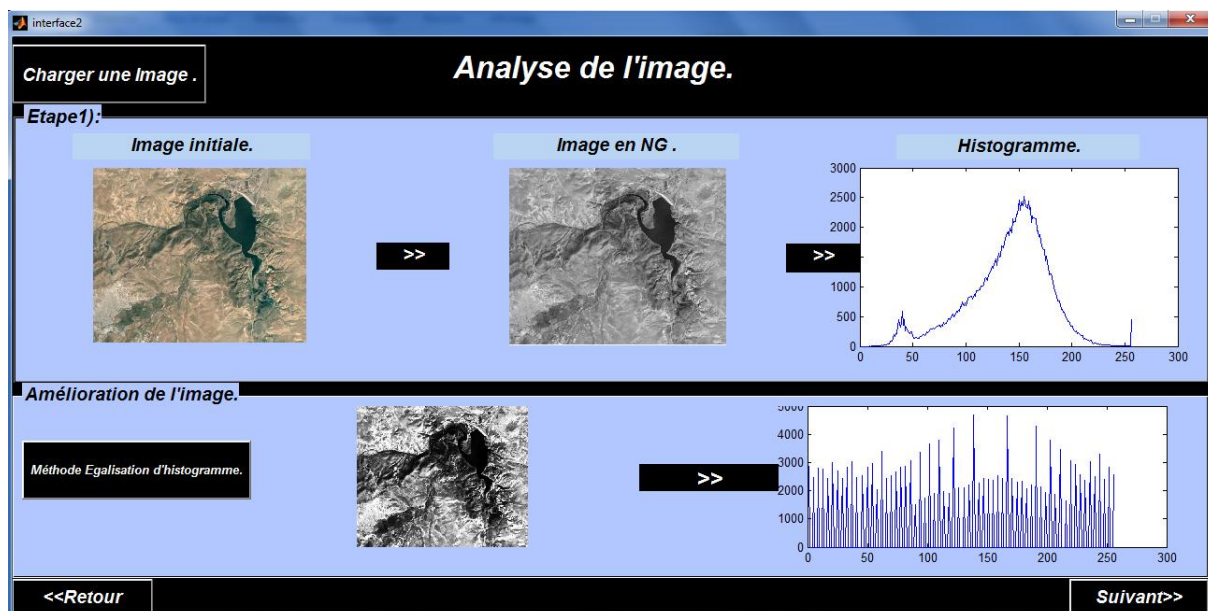


Figure IV.6. L'interface auquel les résultats sont affichés

IV.4.2.2 Filtrage

Cette opération consiste à éliminer les bruits qui contaminent l'image. On a essayé plusieurs filtres ; finalement, on a choisis un filtre de type Sobel parce que c'est le seule qui nous a servir pour avoir des meilleur résultats.



Figure IV.7. Filtrage d'image

IV.4.2.3 Segmentation

Dans cette étape nous avons convertis l'image en binaire (noir et blanc), pour bien visualiser l'objet le résultat que nous avons eu c'est l'objet en blanc (le fond en noir) avec présence des pixels isolés comme indiquent la figure ci-dessous.

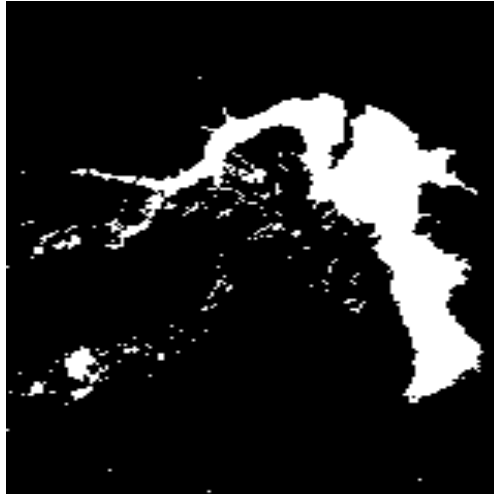


Figure IV.8. Segmentation d'image avec présence des pixels isolés.

Dans la partie suivante nous avons éliminé les pixels isolés par la méthode de remplissage des trous (en utilisant un filtre selon la surface) comme indique la figure ci-dessous.



Figure IV.9. Segmentation d'image avec élimination des pixels isolés.

Dans ce qui suit, nous avons tracé le contour de l'objet d'intérêt (détection de contour) comme la suivante figure montre :

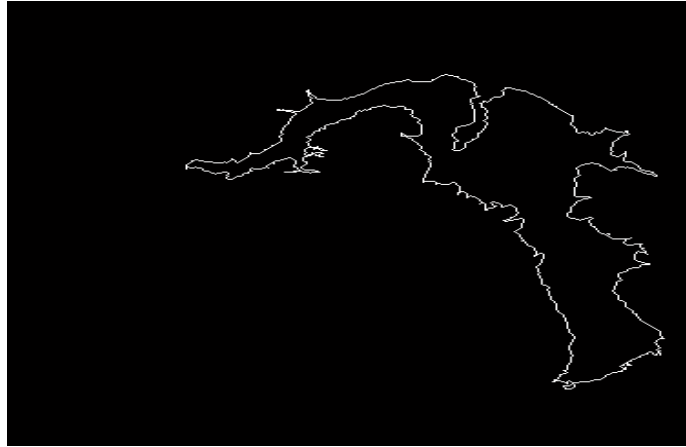


Figure IV.10. La détection du contour de l'image.

La dernière étape nous avons fait la superposition du contour sur l'objet que nous avons voulu dans l'image initiale pour but de confirmer que le contour détecté est confondu sur la frontière du barrage donc la surface obtenu est juste.

Exemple :



Figure IV .11.La superposition du contour de l'image.

Après la détection de contour il ne nous reste que de calculer la surface de l'objet (le nombre des pixels qui constitue l'objet) pour cela nous avons utilisé la fonction de calcul de surface et nous avons trouvé :

L'année	Le nombre des pixels total	Dimension	La surface du barrage (pixels)
2004	172208	458x376	8076
2007	148590	390x381	8868
2012	200644	487x412	18084
2016	183864	376x489	18828

Tableau .VI.1. Les surfaces en pixels.

L'échelle 1 pixel = 127 m²

l'année	La surface en m ²
2004	1025652
2007	1126236
2012	2296668
2016	2391156

Tableau .VI.2. Les surfaces en mètre carrée

D'après les résultats que nous avons obtenus on remarque que la surface de BARRAGE BOUGHRARA est en croissance durant des années.

Les causes de cette augmentation peuvent être les suivantes :

- La nature de la terre.
- Les sources d'eau qui sont au fond.
- La quantité de la pluie tombée.
- Les travaux au niveau de frontière du barrage.

Remarque :

Le traitement d'image donne des résultats avec des incertitudes.

IV.5 Conclusion

Dans cette application nous avons implémenté quelques techniques et méthodes de traitement d'images pour traiter les images numériques de la zone d'étude que nous avons choisi «BARRAGE HAMMAM BOUGHRARA ». D'abord, nous avons utilisé plusieurs techniques (Filtrage, Egalisation, Histogramme et Segmentation). Enfin, nous avons calculé la surface dans le but de suivre progressivement l'évolution de « Barrage Boughrara » durant des différentes années.

Conclusion générale

Conclusion générale

La géolocalisation basée sur l'utilisation des systèmes d'informations géographiques :

Pour simplifier la compréhension de ce thème, nous l'avons décomposé en quatre chapitres :

Dans le premier, nous avons commencé par une introduction du domaine de la géolocalisation suivi par une définition de la géolocalisation, ses origines et un historique. Puis on a cité la technique de la géolocalisation actuels, leurs avantages et leurs inconvénients et les services que peut présenter la géolocalisation.

Le deuxième chapitre c'est le système d'information géographique. D'abord, on a donné une définition en suite nous avons cité les composants de SIG et on a présenté les types et les modes des données SIG. En effet, on a attribué les fonctionnalités d'un SIG nous avons données des exemples d'applications des SIG et ses avantages.

Le troisième chapitre est titré le traitement d'image, dans ce chapitre nous avons défini ce domaine et l'image on a cité les types d'images, nous avons attribué les formats des images et les caractéristiques en suite nous avons parlé sur le système de traitement d'image puis les différentes opérations qu'on peut appliquer.

Le quatrième chapitre on a réalisé une application informatique pour visualiser la géolocalisation pour cela on a utilisé le logiciel MATLAB et on a choisi deux images satellitaires de sebkha d'Oran une ancienne et l'autre nouvelle et on a comparé les résultats obtenus.

La géolocalisation nous a aidés à suivre l'évolution d'une région, et ça c'est l'objectif de ce mémoire « on a suivi BARRAGE BOUGHRARA durant des années ».

Les technologies utilisées se développent de façon continue pour avoir une meilleure précision et en temps réel.

Nous avons rencontré des problèmes pendant la réalisation de notre travail ; d'abord, le problème d'acquisition des images satellitaires : nous avons pris les images à partir de Google Earth ; les images des années précédentes ne sont pas nettes donc impossible d'appliquer les opérations des traitements des images si on veut avoir un bon résultat. Ensuite, on a trouvé des difficultés pendant l'utilisation de logiciel MATLAB parce que nous ne maîtrisons pas bien les fonctions de traitement d'image. Enfin, le manque des ressources pour étudier la zone que nous avons choisie.

Conclusion générale

Pour éviter quelques problèmes il faut commencer d'abord par l'obtention des images avant de passer à autres étapes ; aussi si c'est possible de les avoir avec une qualité HD.

Enfin, utiliser un langage évolué orienté objet comme Delphi.

BIBLIOGRAPHIE

BIBLIOGRAPHIE

[1] CLAUDE GUEDAT, OLIVIER BRETTE, « systèmes de géolocalisation », cours, janvier 2010.

[2] <http://www.hostip.info/index.html>, consulté le 23 mai 2016.

[3] GUILLAUME SCHREINER, « Introduction à la géolocalisation 802.11 » Université Louis Pasteur LSIT

[4] ANDERSNOREN, « géolocalisation en France », publication, 2016

[5] RENAL Paul TATSO, « Intégration d'un observatoire urbain sur Google Maps », mémoire de master, novembre 2011, UNIVERSITE DE DOUALA – CAMEROUN.

[6] WILLY BEAUDEMOULIN, MAGALI CHESNE « système d'information géographique et politique foncière », conférence à Beauvais -France ,20 et 21 mars 2014.

[7] SANA EL JANYANI « introduction au SIG système d'information géographique », projet, 2014-2015

[8] BURROUGH et MCDONNEL «Extraction des unités fonctionnelles du modèle », annexe, 1998.

[9] RENAUD LAIRE « les enjeux de la société de l'information pour le développement des territoires rhônalpins », publication, 2005

[10] Mr. KERFA DJOUDI, « Estimation de mouvement d'objets dans une séquence d'images », thèse de doctorat, université de Tlemcen 2015

[11] BAROUDI ABDERRREZEQ-LARABI NOUR ISLEM, « Comparaison entre les différents filtres d'images », mémoire de Licence, université de Tlemcen ,2012

BIBLIOGRAPHIE

[12]VANDENBROUCKE NICOLAS « traitement d'image sous matlab »,
cours, 2009-2010

[13]EL HAOUATI H, GUECHAOUI M, ARAB A « Effet de l'eutrophisation
sur les milieux aquatiques : Barrage de HammamBouhrara, Tlemcen »
mémoire master, université de Tlemcen, 2013.

Annexe

Filtre Sobel

Le **filtre de Sobel** est un opérateur utilisé en traitement d'image pour la détection de contours. Il s'agit d'un des opérateurs les plus simples qui donne toutefois des résultats corrects.

Le calcul de gradient est mené par l'intermédiaire de deux masques, le premier effectuant un gradient horizontal, le second un gradient vertical. Là encore, le deuxième masque se déduit du premier par une rotation de $\pi/2$. Les masques sont donnés ci-dessous pour les contours horizontaux puis verticaux.

1	c	1
0	0	0
-1	-c	-1

-1	0	1
-c	0	c
-1	0	1

Chaque pixel des masques est normalisé par :

$$\frac{1}{c+2}$$

Lorsque $c=2$, il s'agit des opérateurs de Sobel. Ces masques ont l'avantage de produire deux effets. Outre le calcul du gradient dans une direction, ces masques effectuent un lissage dans la direction orthogonale. Ce lissage rend ces masques un peu moins sensibles au bruit que les précédents.

L'origine de ces masques est toujours le pixel central. La réponse impulsionnelle $h(m,n)$ du filtre de Sobel pour la mise en évidence des contours horizontaux est telle que :

$$h(-1,1) = h(1,1) = 1$$

$$h(-1,-1) = h(1,-1) = -1$$

$$h(0,1) = c$$

$$h(0,-1) = -c$$

La sortie obtenue après filtrage est :

$$\begin{aligned} \sum_{m=-1}^1 \sum_{n=-1}^1 h(m,n) f(m+i, n+j) &= \frac{1}{c+2} (h(-1,1) f(i-1, j+1) + h(0,1) f(i, j+1)) \\ &+ \frac{1}{c+2} (h(1,1) f(i+1, j+1) + h(-1,-1) f(i-1, j-1)) \\ &+ \frac{1}{c+2} (h(-1,0) f(i-1, j) + h(-1,1) f(i-1, j+1)) \end{aligned}$$

$$\begin{aligned} \sum_{m=-1}^1 \sum_{n=-1}^1 h(m,n) f(m+i, n+j) &= \frac{1}{c+2} (f(i-1, j+1) + c f(i, j+1) + f(i+1, j+1)) \\ &- \frac{1}{c+2} (f(i-1, j-1) + c f(i-1, j) + f(i-1, j+1)) \end{aligned}$$

Annexe

L'équation laisse apparaître la double action avec un moyennage horizontal sur trois pixels sur les lignes au-dessus et au-dessous du pixel central et un calcul de gradient vertical entre les deux lignes.

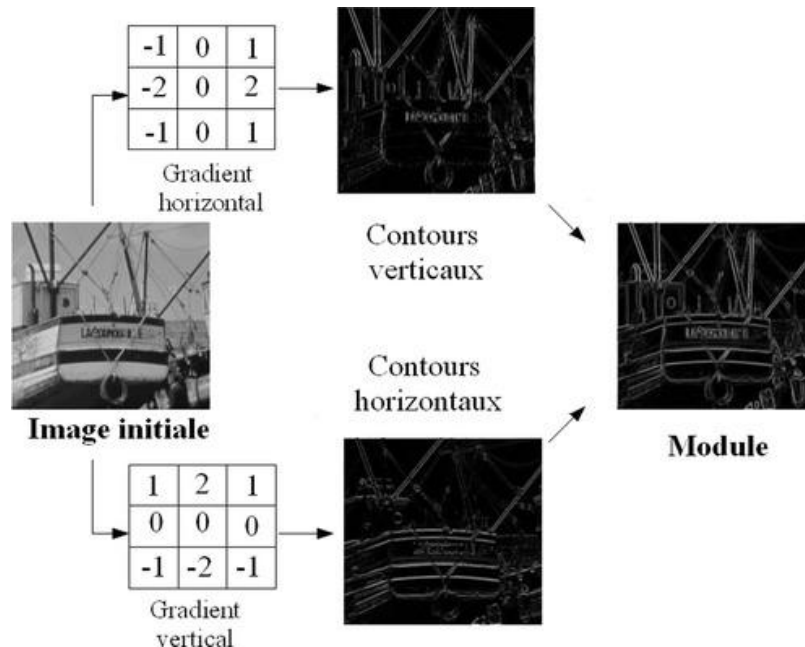
Pour la mise en évidence des contours verticaux, c'est l'autre masque qui est utilisé. La sortie obtenue après filtrage peut se mettre sous la forme :

$$\sum_{m=-1}^1 \sum_{n=-1}^1 h(m,n) f(m+i,n+j) = \frac{1}{c+2} (f(i+1,j+1) - f(i-1,j+1)) \\ + \frac{1}{c+2} (c f(i+1,j) - c f(i-1,j)) \\ + \frac{1}{c+2} (f(i+1,j-1) - f(i-1,j-1))$$

Cette écriture permet elle aussi de mettre en évidence la double action : un gradient horizontal est en effet calculé sur trois lignes puis un lissage vertical est opéré.

Les deux écritures employées montrent bien que les deux actions, lissage et dérivation, du filtrage de Sobel sont séparables.

La figure suivante propose une illustration de l'application des masques de Sobel.



Exemple d'application de l'opérateur de Sobel

Le principal intérêt de ces masques est leur facilité de mise en œuvre ainsi que la rapidité de leur traitement. Leur inconvénient est leur grande sensibilité au bruit. De plus les contours obtenus sont souvent assez larges. Le filtre de Sobel est le plus utilisé dans les applications industrielles nécessitant des contraintes temps-réel.

Annexe

Le programme complet que nous avons créé pour réaliser l'application c'est le suivant :

La page de garde :

```
function varargout = pagedegarde(varargin)
%PAGEDEGARDE M-file for pagedegarde.fig
% PAGEDEGARDE, by itself, creates a new PAGEDEGARDE or raises the existing
% singleton*.
%
% H = PAGEDEGARDE returns the handle to a new PAGEDEGARDE or the handle to
% the existing singleton*.
%
% PAGEDEGARDE('Property','Value',...) creates a new PAGEDEGARDE using the
% given property value pairs. Unrecognized properties are passed via
% varargin to pagedegarde_OpeningFcn. This calling syntax produces a
% warning when there is an existing singleton*.
%
% PAGEDEGARDE('CALLBACK') and PAGEDEGARDE('CALLBACK',hObject,...) call the
% local function named CALLBACK in PAGEDEGARDE.M with the given input
% arguments.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the response to help pagedegarde
% Last Modified by GUIDE v2.5 12-May-2016 23:33:21
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton',  gui_Singleton, ...
    'gui_OpeningFcn', @pagedegarde_OpeningFcn, ...
    'gui_OutputFcn',  @pagedegarde_OutputFcn, ...
    'gui_LayoutFcn',  [], ...
    'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
```

Annexe

```
gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before pagedegarde is made visible.
function pagedegarde_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   unrecognized PropertyName/PropertyValue pairs from the
%           command line (see VARARGIN)
% Choose default command line output for pagedegarde
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
% UIWAIT makes pagedegarde wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% --- Outputs from this function are returned to the command line.
function varargout = pagedegarde_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;
% --- Executes during object creation, after setting all properties.
function axes2_CreateFcn(hObject, eventdata, handles)
a=imread('aa.jpg');
imshow(a);
% hObject    handle to axes2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: place code in OpeningFcn to populate axes2
% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
barre = waitbar(0,'Patiente...');
for k=1:10
pause(0.1);
% pour les besoins de l'exemple, on
```

Annexe

```
% fait ici une pause en lieu et
% place d'autres instructions...
waitbar(k/10);
end
close(barre);
run interface1
% hObject handle to pushbutton6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% --- Executes during object creation, after setting all properties.
function axes3_CreateFcn(hObject, eventdata, handles)
f=imread('aa.jpg');
imshow(f);
% hObject handle to axes3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
% Hint: place code in OpeningFcn to populate axes3
% --- Executes during object creation, after setting all properties.
function axes4_CreateFcn(hObject, eventdata, handles)
f=imread('aa.jpg');
imshow(f);
% hObject handle to axes4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
% Hint: place code in OpeningFcn to populate axes4
```

Interface 1:

```
function varargout = interface1(varargin)
% INTERFACE1 MATLAB code for interface1.fig
% INTERFACE1, by itself, creates a new INTERFACE1 or raises the existing
% singleton*.
%
% H = INTERFACE1 returns the handle to a new INTERFACE1 or the handle to
% the existing singleton*.
%
% INTERFACE1('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in INTERFACE1.M with the given input arguments.
%
% INTERFACE1('Property','Value',...) creates a new INTERFACE1 or raises the
```

Annexe

```
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before interface1_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to interface1_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the response to help interface1
% Last Modified by GUIDE v2.5 13-May-2016 19:16:37
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @interface1_OpeningFcn, ...
    'gui_OutputFcn', @interface1_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);
ifnargin&&ischar(varargin{1})
gui_State.gui_Callback = str2func(varargin{1});
end
ifnargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before interface1 is made visible.
function interface1_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to interface1 (see VARARGIN)
% Choose default command line output for interface1
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
```


Annexe

```
% UIWAIT makes interface1 wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% --- Outputs from this function are returned to the command line.
function varargout = interface1_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;
% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)
m = imread('vv.jpg');
imshow(m);
% hObject handle to axes1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
% Hint: place code in OpeningFcn to populate axes1
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
close interface1
runpagedegarde
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% --- Executes on button press in checkbox2.
function checkbox2_Callback(hObject, eventdata, handles)
winopen('.pptx');
% hObject handle to checkbox2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of checkbox2
% --- Executes on button press in checkbox1.
function checkbox1_Callback(hObject, eventdata, handles)
winopen('.docx');
% hObject handle to checkbox1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
```

Annexe

```
% handles structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of checkbox1
% --- Executes on button press in checkbox3.
function checkbox3_Callback(hObject, eventdata, handles)
run interface2
% hObject handle to checkbox3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of checkbox3
```

Interface 2:

```
function varargout = interface2(varargin)
% INTERFACE2 MATLAB code for interface2.fig
%   INTERFACE2, by itself, creates a new INTERFACE2 or raises the existing
%   singleton*.
%
%   H = INTERFACE2 returns the handle to a new INTERFACE2 or the handle to
%   the existing singleton*.
%
%   INTERFACE2('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in INTERFACE2.M with the given input arguments.
%
%   INTERFACE2('Property','Value',...) creates a new INTERFACE2 or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before interface2_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to interface2_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help interface2
% Last Modified by GUIDE v2.5 13-May-2016 14:11:48
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
```

Annexe

```
'gui_OpeningFcn', @interface2_OpeningFcn, ...
'gui_OutputFcn', @interface2_OutputFcn, ...
'gui_LayoutFcn', [], ...
'gui_Callback', []);
ifnargin&&ischar(varargin{1})
gui_State.gui_Callback = str2func(varargin{1});
end
ifnargout
[varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before interface2 is made visible.
function interface2_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to interface2 (see VARARGIN)
% Choose default command line output for interface2
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
% UIWAIT makes interface2 wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% --- Outputs from this function are returned to the command line.
functionvarargout = interface2_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;
% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)
r=imread('ff.jpg');
imshow(r);
% hObject    handle to axes1 (see GCBO)
```

Annexe

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
% Hint: place code in OpeningFcn to populate axes1
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
nom=uigetfile('*.*)
I=imread(nom);
k = imshow(I,'Parent',handles.axes5)
I1=rgb2gray(I);
l=imshow(I1,'paren',handles.axes6)
h=imhist(I1);
plot(handles.axes3,h);
handles. I1=I1;
guidata(hObject,handles);
% hObject handle to pushbutton3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
I1=handles.I1;
u=histeq(I1);
h=imshow(u,'parent',handles.axes8);
j=imhist(u);
plot(handles.axes9,j);
% hObject handle to pushbutton5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
close interface2
run interface1
% hObject handle to pushbutton6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
close interface2
run interface3
% hObject handle to pushbutton7 (see GCBO)
```

Annexe

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

Interface 3:

```
function varargout = interface3(varargin)
```

% INTERFACE3 MATLAB code for interface3.fig

% INTERFACE3, by itself, creates a new INTERFACE3 or raises the existing
% singleton*.

%

% H = INTERFACE3 returns the handle to a new INTERFACE3 or the handle to
% the existing singleton*.

%

% INTERFACE3('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in INTERFACE3.M with the given input arguments.

%

% INTERFACE3('Property','Value',...) creates a new INTERFACE3 or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before interface3_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to interface3_OpeningFcn via varargin.

%

% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".

%

% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help interface3

% Last Modified by GUIDE v2.5 13-May-2016 14:49:13

% Begin initialization code - DO NOT EDIT

```
gui_Singleton = 1;
```

```
gui_State = struct('gui_Name',    mfilename, ...  
    'gui_Singleton', gui_Singleton, ...  
    'gui_OpeningFcn', @interface3_OpeningFcn, ...  
    'gui_OutputFcn', @interface3_OutputFcn, ...  
    'gui_LayoutFcn', [] , ...  
    'gui_Callback', []);
```

```
if nargin && ischar(varargin{1})
```

```
    gui_State.gui_Callback = str2func(varargin{1});
```

```
end
```

```
if nargin
```

Annexe

```
[varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before interface3 is made visible.
function interface3_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to interface3 (see VARARGIN)
% Choose default command line output for interface3
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
% UIWAIT makes interface3 wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% --- Outputs from this function are returned to the command line.
function varargout = interface3_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
close interface3
run interface2
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
close interface3
run interface4
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

Annexe

```
% handles  structure with handles and user data (see GUIDATA)
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
nom=uigetfile('*.*)
I=imread(nom);
k = imshow(I,'Parent',handles.axes1)
handles.I=I;
guidata(hObject,handles);
I=handles.I;
I1=rgb2gray(I);
I=imshow(I1,'paren',handles.axes2)
handles.I1=I1;
guidata(hObject,handles);
% hObject  handle to pushbutton2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles  structure with handles and user data (see GUIDATA)
% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
I=handles.I;
I1=handles.I1;
%normal_edges = edge(I1,'canny');%%%%%%%%type de filtre canny
%figure(2),imshow(I1);
%edgeFun = @(block_struct) edge(I1,'canny');%%%%%%%%la taille de masque
%block_size = [50 50];
%block_edges = blockproc(I1,block_size,edgeFun);
%figure(3),imshow(I1);
%BW = edge(I1);
BW = edge(I1,'sobel');%%%%%%%%type de filtresobel
figure(4),imshow(I1);
% hObject  handle to pushbutton5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles  structure with handles and user data (see GUIDATA)
```

Interface 4:

```
function varargout = interface4(varargin)
% INTERFACE4 MATLAB code for interface4.fig
%   INTERFACE4, by itself, creates a new INTERFACE4 or raises the existing
%   singleton*.
%
```

Annexe

```
% H = INTERFACE4 returns the handle to a new INTERFACE4 or the handle to
% the existing singleton*.
%
% INTERFACE4('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in INTERFACE4.M with the given input arguments.
%
% INTERFACE4('Property','Value',...) creates a new INTERFACE4 or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before interface4_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to interface4_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the response to help interface4
% Last Modified by GUIDE v2.5 13-May-2016 15:37:37
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @interface4_OpeningFcn, ...
    'gui_OutputFcn', @interface4_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);
ifnargin&&ischar(varargin{1})
gui_State.gui_Callback = str2func(varargin{1});
end

ifnargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before interface4 is made visible.
function interface4_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
```


Annexe

```
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to interface4 (see VARARGIN)
% Choose default command line output for interface4
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
% UIWAIT makes interface4 wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% --- Outputs from this function are returned to the command line.
function varargout = interface4_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
nom=uigetfile('*.*.*)
I=imread(nom);
k = imshow(I,'Parent',handles.axes1)
handles.I=I;
guidata(hObject,handles);
I=handles.I;
I1=rgb2gray(I);
l=imshow(I1,'paren',handles.axes2)
h=imhist(I1);
plot(handles.axes3,h);
handles.I1=I1;
guidata(hObject,handles);
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
close interface4
run interface3
```

Annexe

```
% hObject handle to pushbutton2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
close interface4
run conclusion
% hObject handle to pushbutton3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
I=handles.I;
I1=handles.I1;
h=88;
g=I1<h;
figure,imshow(g);
g1=imfill(g,'holes');
figure,imshow(g1);
g2=bwareaopen(g1,4000);
figure,imshow(g2);
g3=1-g2;
figure,imshow(g3);
cont=bwperim(g2);
figure,imshow(cont);
c=I;
c(cont)=255;
figure,imshow(c);
s=sum(sum(cont))%calculer la surface
% hObject handle to pushbutton4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

Conclusion générale:

```
function varargout = conclusion(varargin)
% CONCLUSION MATLAB code for conclusion.fig
% CONCLUSION, by itself, creates a new CONCLUSION or raises the existing
% singleton*.
%
```

Annexe

```
% H = CONCLUSION returns the handle to a new CONCLUSION or the handle to
% the existing singleton*.
%
% CONCLUSION('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in CONCLUSION.M with the given input arguments.
%
% CONCLUSION('Property','Value',...) creates a new CONCLUSION or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before conclusion_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to conclusion_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the response to help conclusion
% Last Modified by GUIDE v2.5 13-May-2016 16:33:29
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @conclusion_OpeningFcn, ...
    'gui_OutputFcn', @conclusion_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before conclusion is made visible.
function conclusion_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
```

Annexe

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to conclusion (see VARARGIN)
% Choose default command line output for conclusion
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
% UIWAIT makes conclusion wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% --- Outputs from this function are returned to the command line.
function varargout = conclusion_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;
% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)
a=imread('ff.jpg');
imshow(a);
% hObject handle to axes1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
% Hint: place code in OpeningFcn to populate axes1
% --- Executes during object creation, after setting all properties.
function axes2_CreateFcn(hObject, eventdata, handles)
a=imread('ff.jpg');
imshow(a);
% hObject handle to axes2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
% Hint: place code in OpeningFcn to populate axes2
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
close conclusion
run interface4
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
```

Annexe

```
% handles structure with handles and user data (see GUIDATA)
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
close conclusion
runfindepg
% hObject handle to pushbutton2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

La dernière interface:

```
function varargout = findepg(varargin)
% FINDEPG MATLAB code for findepg.fig
% FINDEPG, by itself, creates a new FINDEPG or raises the existing
% singleton*.
%
% H = FINDEPG returns the handle to a new FINDEPG or the handle to
% the existing singleton*.
%
% FINDEPG('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in FINDEPG.M with the given input arguments.
%
% FINDEPG('Property','Value',...) creates a new FINDEPG or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before findepg_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to findepg_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the response to help findepg
% Last Modified by GUIDE v2.5 13-May-2016 16:46:27
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @findepg_OpeningFcn, ...
    'gui_OutputFcn', @findepg_OutputFcn, ...
```

Annexe

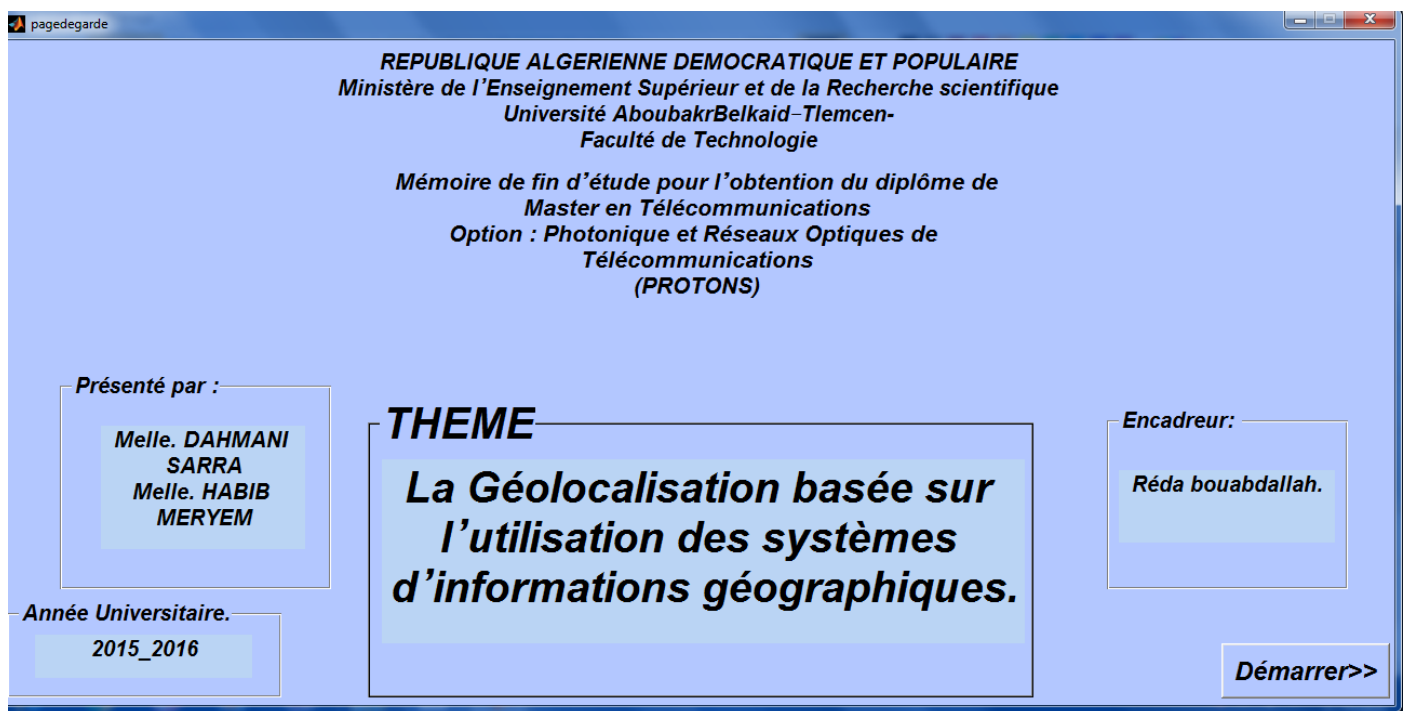
```
'gui_LayoutFcn', [], ...
'gui_Callback', []);
if nargin && ischar(varargin{1})
gui_State.gui_Callback = str2func(varargin{1});
end
if nargin
[varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before findpep is made visible.
function findpep_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to findpep (see VARARGIN)
% Choose default command line output for findpep
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
% UIWAIT makes findpep wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% --- Outputs from this function are returned to the command line.
function varargout = findpep_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;
% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)
z = imread('ff.jpg');
imshow(z);
% hObject handle to axes1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
```

Annexe

```
% Hint: place code in OpeningFcn to populate axes1
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
closefindpg
run conclusion
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

Les interfaces sont:

La page de garde



Bouton démarrer pour passer à l'interface suivant.

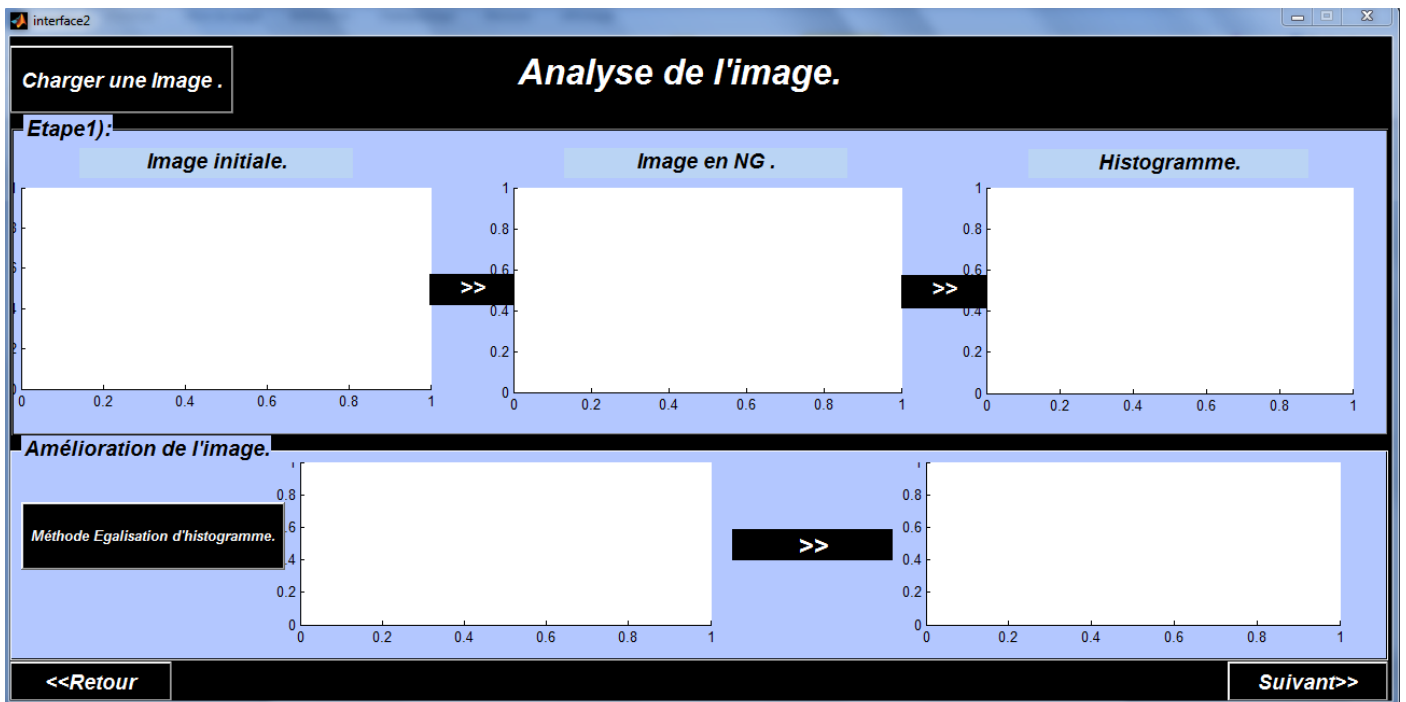
Annexe

L'interface 1



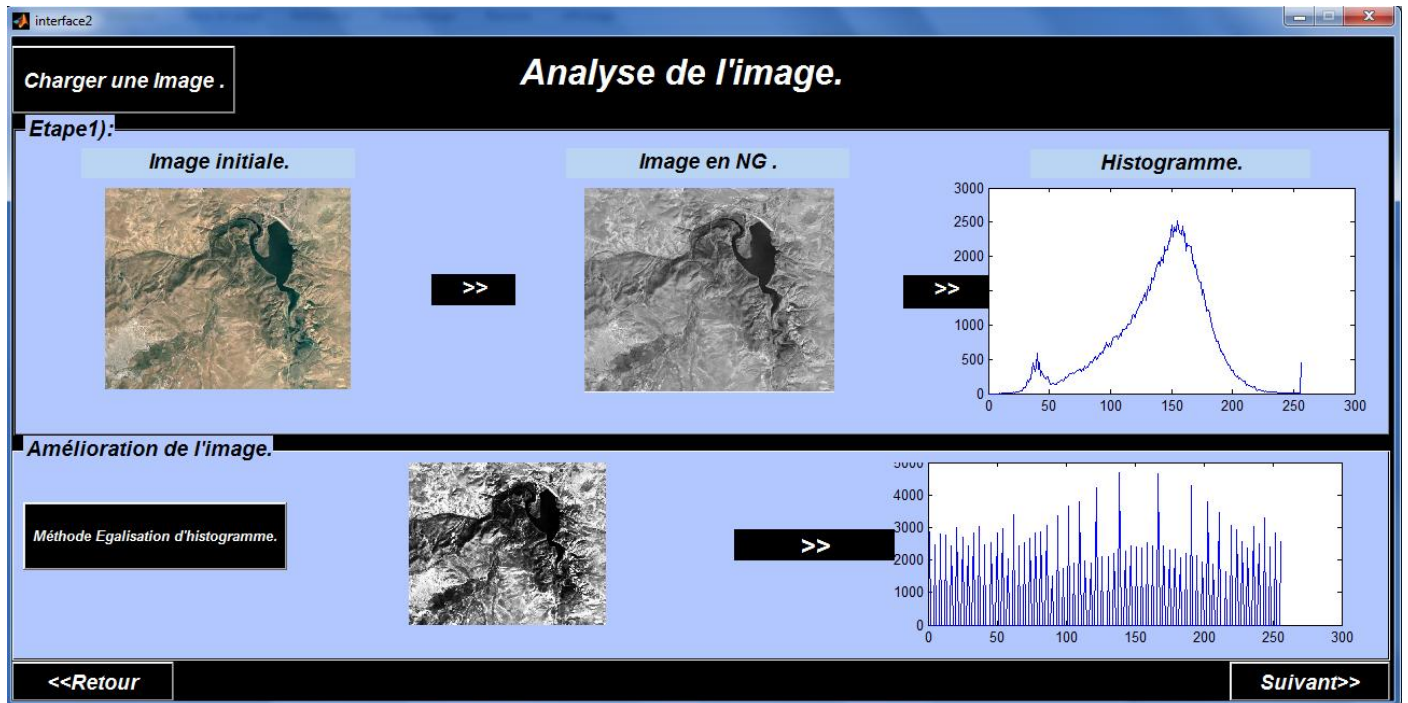
Dans cette interface on choisit la partie qu'on veut ouvrir
Retour pour revenir à la page de garde

L'interface 2

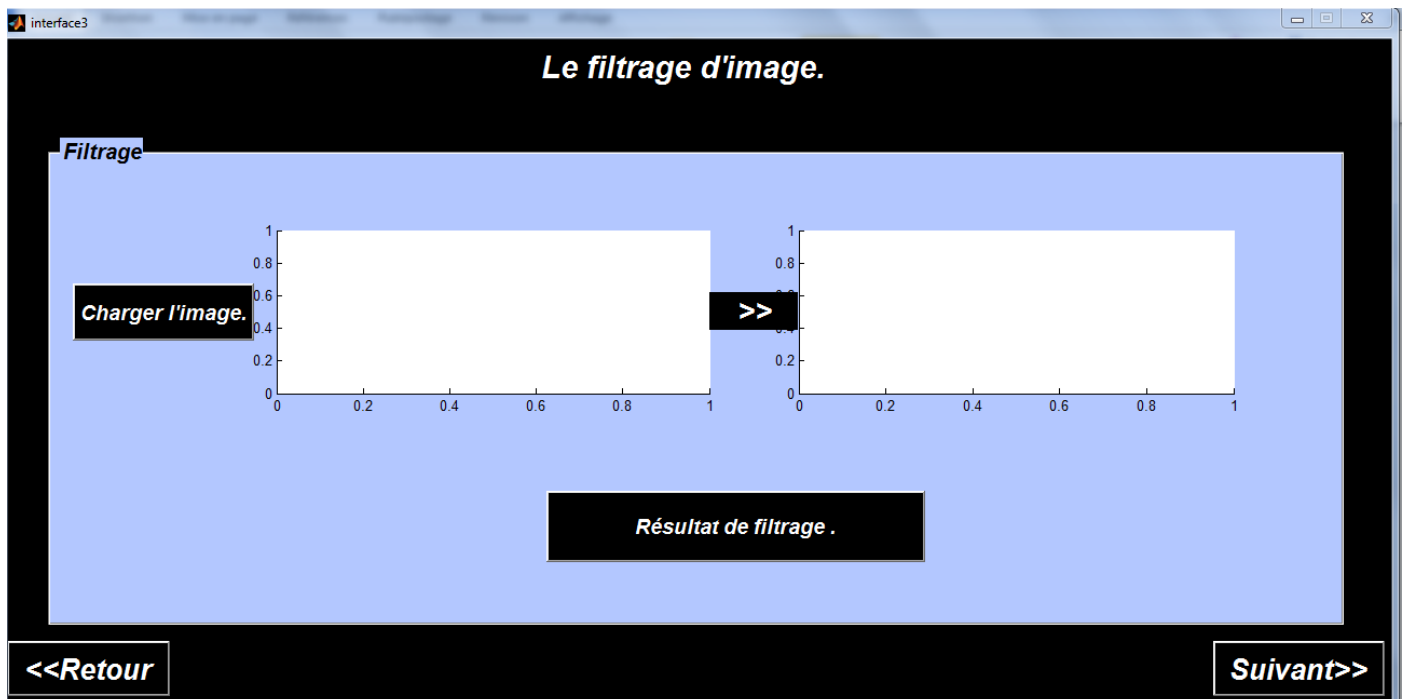


Ici on clique sur le bouton charger image nous choisissons une image ; les résultats s'affichent.
Par exemple :

Annexe



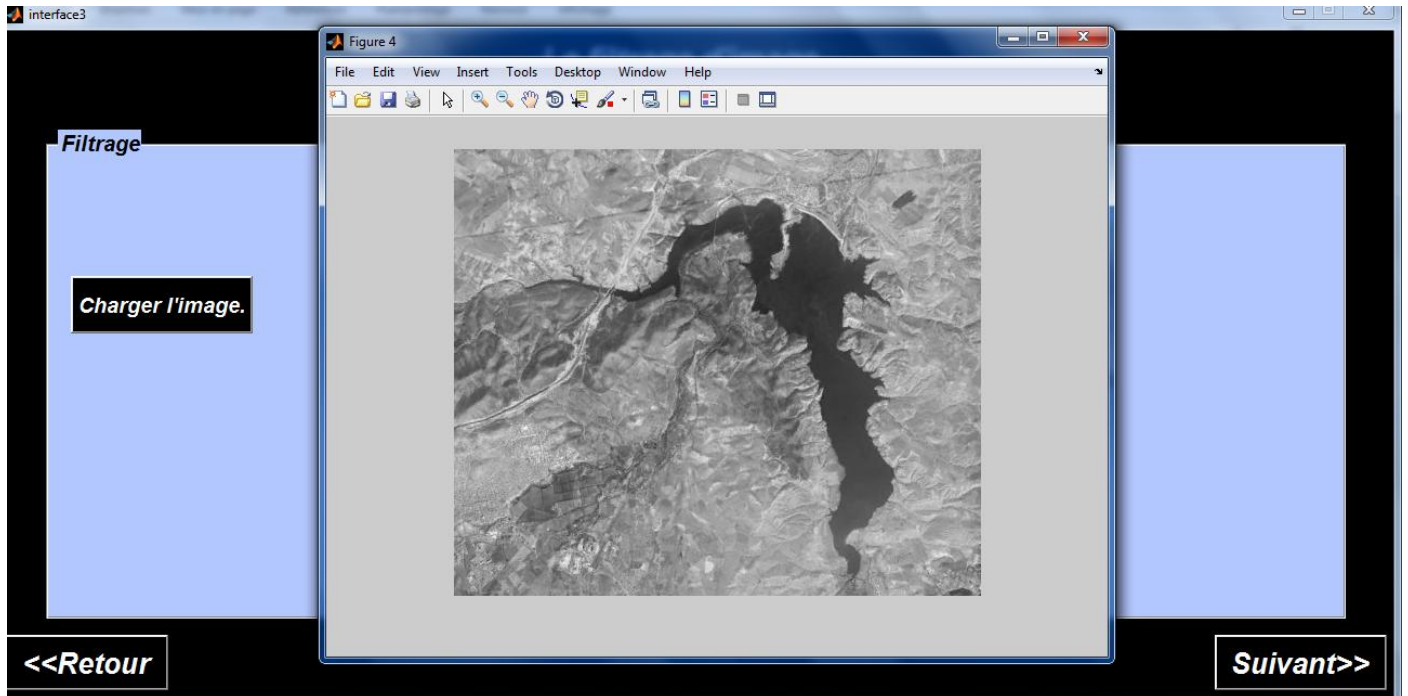
L'interface 3



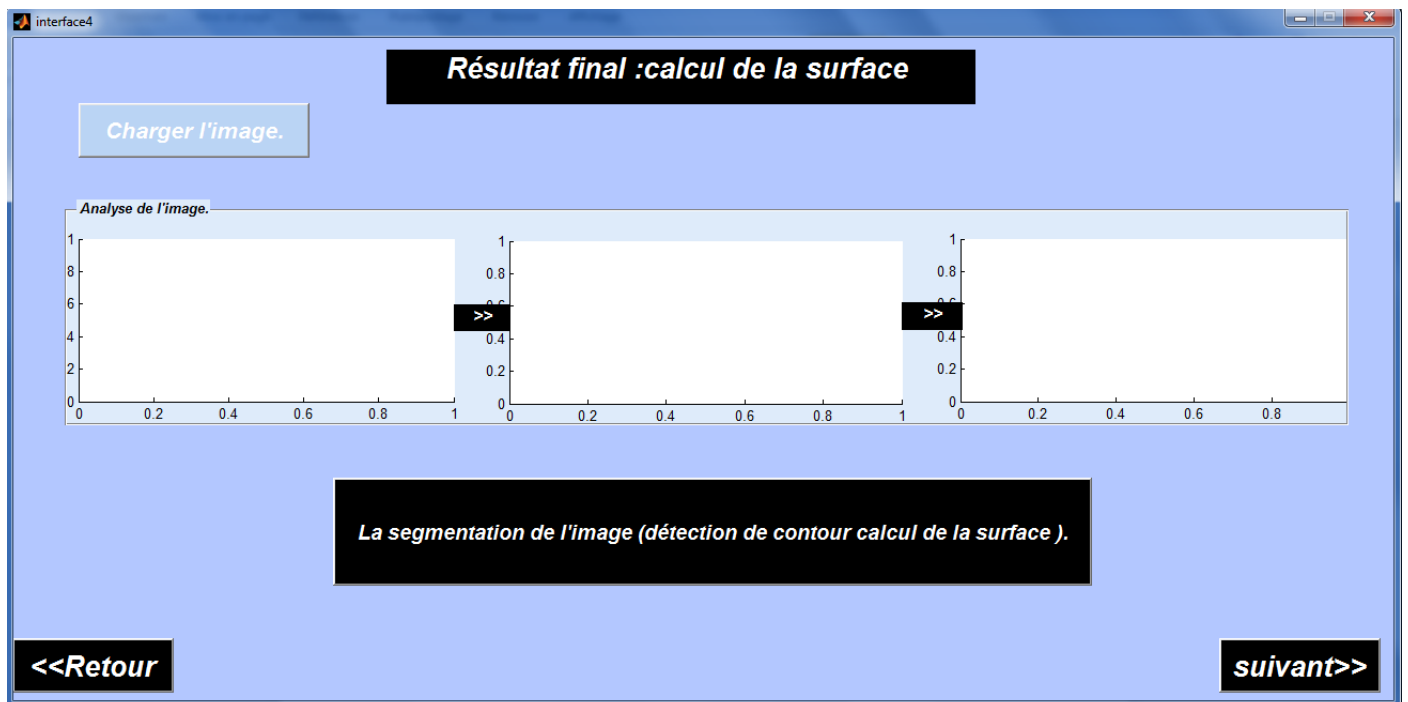
Le bouton résultat de filtrage pour afficher le résultat

Par exemple :

Annexe

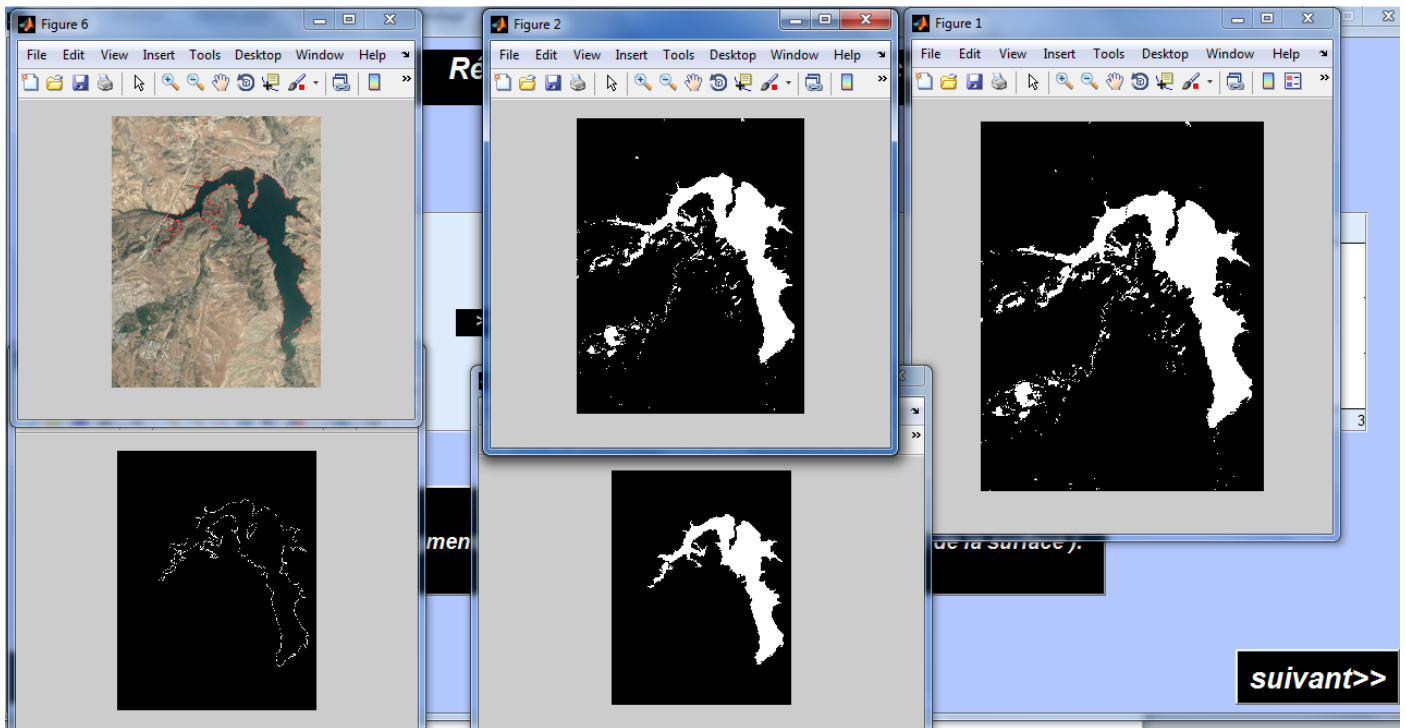


L'interface 4

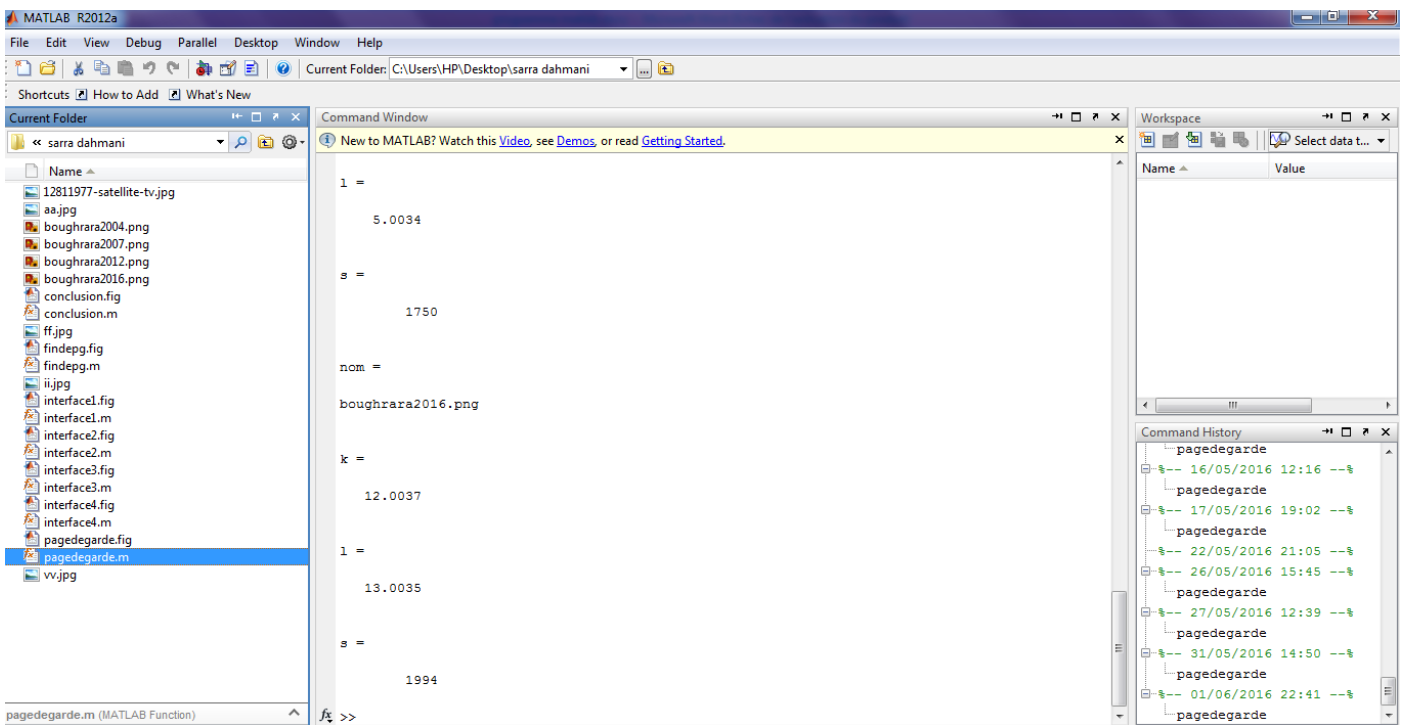


Pour afficher les résultats de segmentation
par exemple :

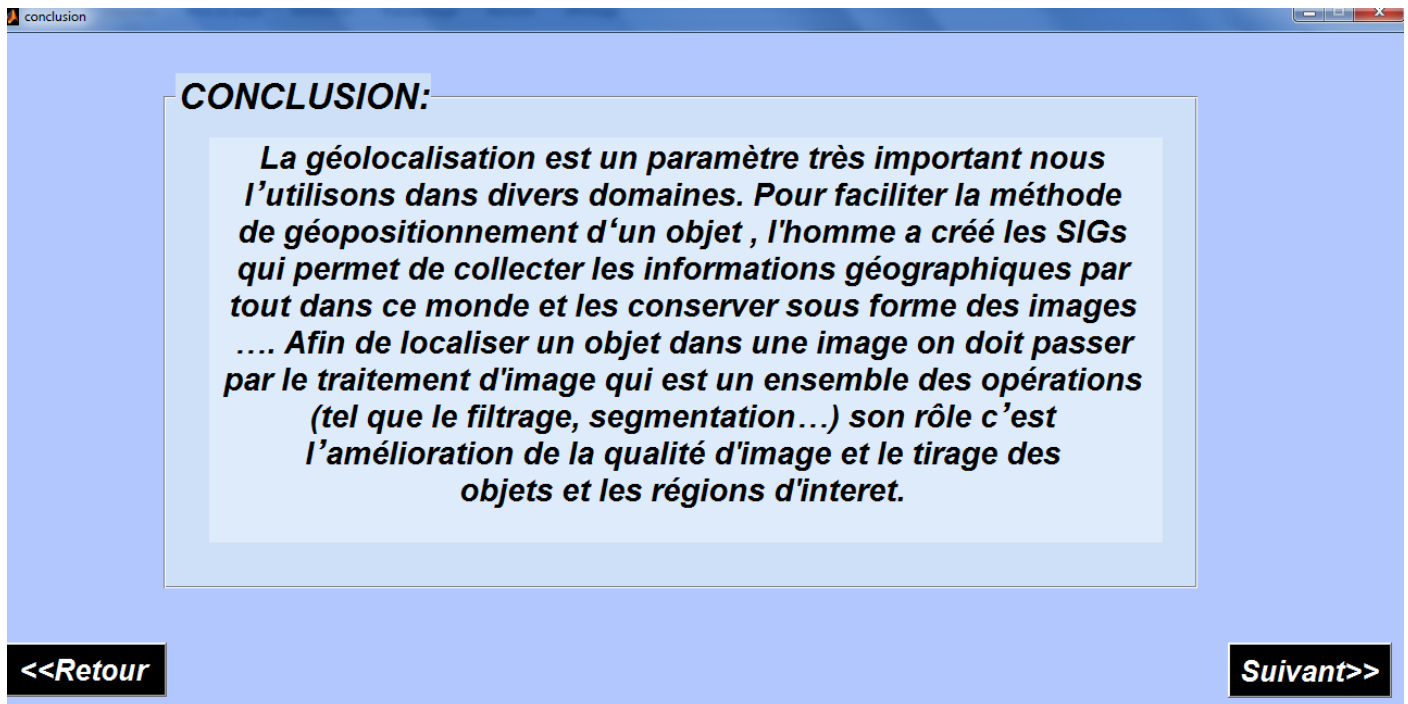
Annexe



Le résultat de la surface il s'affiche dans la fenêtre suivante :



Interface conclusion générale



conclusion

CONCLUSION:

La géolocalisation est un paramètre très important nous l'utilisons dans divers domaines. Pour faciliter la méthode de géopositionnement d'un objet, l'homme a créé les SIGs qui permet de collecter les informations géographiques par tout dans ce monde et les conserver sous forme des images Afin de localiser un objet dans une image on doit passer par le traitement d'image qui est un ensemble des opérations (tel que le filtrage, segmentation...) son rôle c'est l'amélioration de la qualité d'image et le tirage des objets et les régions d'interet.

<<Retour **Suivant>>**

Résumé

L'objectif du travail présenté dans ce mémoire est de suivre l'évolution du Barrage de Hammam Boughrara de la wilaya de TLEMEN pendant des années différentes.

Pour cela nous avons utilisé les techniques de traitement d'images (méthode de segmentation) pour détecter les frontières qui suivent la région étudiée du reste de l'image afin de mesurer sa surface.

Mots clés : SIG, Segmentation, Géolocalisation, Traitement d'image.

المخلص

الهدف من هذا العمل المقدم في هذه المذكرة هو تتبع التطور لحمام بوغرارة لولاية تلمسان خلال سنوات مختلفة . ولهذا استعملنا تقنيات معالجة الصور (طريقة الفصل) من اجل تحديد حدود المنطقة المدروسة من بقية الصورة ثم تحديد مساحتها.

مفاتيح الكلمات : نظام المعلومات الجغرافية - تحديد الموقع الجغرافي-الفصل- معالجة الصور

ABSTRACT

The objective of this work presented in this work is to follow the evolution of the steam HAMMAM BOUGHRARA in the wilaya of TLEMEN for different years.

For this we used image processing techniques (the segmentation methods) to detect the border that followed the study from the rest of the image to measure its area .

Key words: SIG – segmentation – geolocation – treatment of pictures.