



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



Université Amar Thelidji- Laghouat

FACULTE: DE TECHNOLOGIE
DEPARTEMENT : D'ELECTRONIQUE

MEMOIRE DE MASTER

Réalisé par : Chettouh Meriem
Damene Mebarka

DOMAINE : Technologie

FILIERE : Technologie

OPTION : Electronique des systèmes embarqués

Thème

Implémentation d'un système de détection des contours dans les images sur FPGA

Jury de soutenance :

Nom et Prénom	Grade	Qualité
Merah Lahcene	MCA	Encadreur
Seghier Abdelkrim	MAA	Président
Reguigue Mourad	MCB	Examineur

Promotion : 2019/2020

Remerciement

*En premier lieu, nous remercions ALLAH,
le tout-Puissant pour ses faveurs et ses grâces, de nous avoir
donné le courage et la patience durant ce travail*

*Nos remerciements et nos profondes gratitudes vont à
notre encadreur Dr. MERAH LAHCEN son soutien
scientifique et psychique était de valeur et essentiel tout au
long de ce mémoire.*

*Nous remercions les honorables membres du jury pour leur
précieux temps accordé à l'étude de notre mémoire.*

*Adressons également nos remerciements, à tous nos
enseignants, qui nous ont donné les bases de la sci
et l'aides inestimables*



Dédicace

*Je dédie mon diplôme et ma réussite
A mon père, que Dieu ait pitié de lui, qui a toujours voulu me
voir ce jour-là*

*A ma chère mère ma force dans la vie qu'elle ma pousse et elle
ma encourager toujours d'avancer pour le mieux , que Dieu la
protège .*

*À mes chère frères mon soutien dans la vie et mon bras
droit Lhadj et Mohamed*

*A ma chère Hanene et son mari et sa fille Maram . À ma belle-
sœur et à ses deux filles, Miral et Reem.*



*Merci à tous ceux qui ont contribué à mes encouragements, de
près ou de loin.*

Mebarka



Dédicace

Dédie mon diplôme et ma réussite

À mon cher père

Aucune dédicace ne saurait être assez éloquente pour exprimer ce qu'il mérite pour ses sacrifices qu'il n'a jamais cessé de me donner depuis ma naissance

À ma chère mère

mon bonheur permanent la plus belle créature que dieu a créé sur terre, À cette source de tendresse, de patience et de générosité.

À mes chers frères Said et Abdelkader

À ceux qui m'ont soutenu tout au long de ma vie, ma sœur Malika

*À le baume de l'âme et de la vie, mes chères sœurs Khadija et Fatima Ramlia
Hiba*

*À la joie et aux pousses de la famille , Mohamed, Djad, Attallah et ma
princesse Takwa*

À mes amis et à tous ceux qui me connaissent .

Meriem

TABLE DES MATIÈRES

Liste des figures	
Liste des tableaux.....	
Liste des abréviations.....	
Introduction générale.....	

Chapitre 01 : notion de base sur le traitement d'image numérique

1. Introduction	3
2. L'image numérique.....	3
2.1 La numérisation d'image.....	3
2.1.1 Échantillonnage.....	4
2.1.2 Quantification.....	4
2.1.3 Encodage.....	4
3. Les attributs d'une image.....	5
3.1 Pixel.....	5
3.2 Dimension	6
3.3 Résolution.....	6
3.4 Luminance.....	6
3.5 Contraste.....	7
3.6 Histogramme	7
3.7 Bruit.....	8
3.8 Le codage des couleurs.....	8
3.9 Les types de fichiers.....	8
3.10 Contenu d'une image.....	9

4. Les différents type d'image	9
4.1 Images binaires	9
4.2 Images en niveaux de gris	9
4.3 Images en couleur	10
5. Les formats d'image numérique.....	10
5.1 image matricielle.....	11
5.2 les images vectorielles	11
6. acquisition d'image numérique.....	12
6.1 Les outils d'acquisition	12
7. Traitement d'images.....	13
7.1La chaine de traitement d'image	13
7.1.1. Traitements classification et étapes fondamental.....	13
7.1.2 Traitement d'images en temps réel	14
7.2 Domaine d'application du traitement d'image.....	15
7.2 Logiciels pour faire du traitement d'image.....	15
8. Les différents types de traitement d'image	16
8.1. Filtrage d'une image	16
8.2. Application de filtres et introduction à la convolution.....	16
8.3 .Les type de filtres	16
8.3.1 Les filtres linéaires	17
8.3.2 Les filtres non linéaires.....	17
8.4. Amélioration d'images.....	18
8.5. Restauration d'images.....	18
8.6. Compression d'image.....	18
8.7. La segmentation d'image	18

9. Seuillage d'image	19
10. Conclusion	20

Chapitre 02 : la détection des contours

1. Introduction.....	21
2. Détection de contours.....	21
2.1. Définition.....	21
2.2 Les objectifs de la détection des contours	21
2.3 Contour d'une image	22
2.4 Les étapes de détection de contour.....	22
2.4.1. Le Lissage d'image	22
2.4.2. Le filtrage.....	23
2.4.3 Localisation	23
3. L'effet mutuel entre détecteurs et contours	23
4. Les approximations de détection de contours	24
4.1. Approximation de gradient.....	25
4.1.1 Définition	25
4.1.2 Principe général.....	25
4.1.3 La direction du gradient.....	25
4.1.4 Incorporation d'informations sur les gradients	26
4.1.5 Les avantages de Gradient	27
4.1.6 calcul de gradient par masques	27
4.2.Approximation laplacien	28
4.2.1. Définition	28
4.2.2. Le principe de laplacien	28
4.2.3. Formule de laplacien.....	29

4.2.4 Les masques de laplacien	29
4.2.5. Laplacien de détection des contours gaussiens(LOG.....	30
4.2.6. Avantages laplacien de gaussien.....	30
4.2.7. Inconvénients laplacien de gaussien.....	31
5.Détection de contour par seuillage à hystérésis	31
5.1. Choix de seuils.....	32
6. Algorithmes de détection de contours	32
6.1. Filtre de Prewitt.....	32
6.1.1. Exemple d'application de Prewitt sous Matlab.....	33
6.2. Filtre de Sobel.....	34
6.2.1. Le principe de filtre de Sobel.....	34
6.2.2. Pseudo-codes pour la méthode de détection des bords Sobel	34
6.2.3. Avantage de filtre Sobel.....	35
6.2.4.Inconvénients de filtre Sobel.....	35
6.2.5. Exemple pour calculer le module de gradient sous Matlab	35
6.3. Les propriétés de ces filtre.....	36
6.4. Le filtre de Canny	37
6.4.1.Les étapes du détecteur de contour Canny	38
6.4.2. Avantages de filtre Canny.....	38
6.4.3. Inconvénients de filtre Canny.....	38
6.4.4. Exemple d'application de Canny sous Matlab.....	38
6.5. Filtre de Robert	39
6.5.1.Avantages de filtre Robert	40
6.5.2.Inconvénients de filtre Robert	40
6.5.3. Exemple d'application de Roberts sous Matlab.....	40

7. Mesure de Précision	41
8. Choix des masque	41
9. Les seuils de chaque méthode de détection de contour.....	41
10. Choix d'un détecteur de contours.....	42
11. Filtres numériques à réponse impulsionnelle finie RIF.....	42
11.1. Définition	42
11.2. Structure d'un filtre RIF.....	42
11.3. Conception de filtre.....	43
11.4. Caractéristiques des filtres numériques RIF.....	43
11.5. Avantages des filtres RIF.....	44
11.6. Applications du monde réel de filtres RIF.....	44
12. Conclusion.....	45

Chapitre 3 Implémentation du filtre de sobel sur un FPGA

1. Introduction.....	46
2. Pourquoi le FPGA.....	46
2.1. La carte de développement ATLYS	47
2.2. L'outil Xilinx System Generator (XSG).....	48
2.3. Xilinx Integrated Synthesis Environment (ISE).....	48
2.4. Le filtre de Sobel avec Matlab.....	49
3. Conception de filtre de Sobel sous XSG.....	50
3.1. Evaluation du système par simulation.....	55
4. Implémentation sur FPGA	56
5. Conclusion	61
Conclusion général	62
Bibliographie.....	

Liste des figures

Chapitre 1 : notion de base sur le traitement d'image numérique

Fig .1. L'image numérique et notion de pixel.....	4
Fig.2. Conversion d'une image analogique en une image numérique.....	4
Fig.3. Représentation des pixels et niveaux de gris	5
Fig 4. les voisins du pixel de coordonnées (x,y).....	5
Fig.5. Représentation de résolution.....	6
Fig.6. Exemple d'histogramme.....	7
Fig.7. Image avec et sans bruit.....	8
Fig.8. Contenu d'une image.....	9
Fig.9. Représentation d'image binaire.....	10
Fig.10. Représentation d'image en niveau de gris	10
Fig.11. Représentation d'image couleur.....	11
Fig.12. Image numérique matricielle.....	11
Fig.13. Image vectorielle et image bitmap.....	12
Fig.14. L'acquisition des images.....	12
Fig.15. Systèmes à base de traitement d'image	15
Fig.16. Exemple de traitements d'image.....	16
Fig .17. Schéma explicatif de la convolution	16
Fig .18. Exemple d'image filtrée.....	17
Fig.19. Exemple d'amélioration d'image.....	18
Fig.20. Exemple de restauration d'image	18
Fig.21. Segmentation d'une image couleur.....	19
Fig.22. La fonction seuillage	19

Chapitre 2 La détection des contours d'image

Fig.1. Les types de contours.....	22
Fig.2. Principe de la détection de contour par gradient ou laplacien.....	24
Fig.3. Seuillage du gradient.....	26

Fig .4. Le principe de gradient pour détection de contour.....	26
Fig.5. les masques de gradient.....	28
Fig .6. Illustration de l'application de masque vertical et horizonta.....	28
Fig.7. Masque de laplacien de 4-connexités	29
Fig.8. Masque de laplacien de 8-connexités.....	30
Fig.9. Comparaison entre Laplacien 4 et 8 connexités.....	30
Fig.10.Masques de convolution laplacien de gaussien.....	30
Fig.11. Laplacien d'une gaussienne "chapeau mexicain».....	31
Fig.12. Représentation du seuillage par hystérésis.....	32
Fig .13. Masques de convolution d'opérateur de Prewitt (c=1).....	33
Fig .14. Application du filtre de Prewitt.....	33
Fig. 15. Noyaux de convolution Sobel.....	34
Fig.16. Exemple d'application de l'opérateur de Sobel.....	35
Fig .17. Module du gradient de filtre Sobel.....	36
Fig .18. Obtention du module du gradient par filtrage de Canny	37
Fig.18. Application de filtre Canny.....	39
Fig.20. Application du filtre de Roberts.....	40
Fig.21. Structure d'un filtre RIF.....	43

Chapitre 3 Implémentation du filtre de sobel sur un FPGA

Fig.1. La carte de développement ATLYS.....	47
Fig.2. Les périphériques d'entrées/sorties de la carte ATLYS.....	48
Fig.3. Interface principale de ISE.....	49
Fig.4. Résultats de simulation de l'application de filtre de Sobel sur une image en Matlab.....	50
Fig.5. Résultats de simulation de l'application de filtre de <i>Sobel</i> sur une image en Matlab,(a) l'image.....	51
Fig.6. Principe de filtre de <i>Sobel</i>	51
Fig.7. Retardement de signal vidéo.....	52

Fig.8. L'interface graphique du mémoire utilisée.....	53
Fig.9. Montage des mémoires en série et l'assemblage de tous en un seul bloc (gauche).....	53
Fig.10. Structure du premier filtre FIR avec XSG.....	54
Fig.11. Le système complet pour détecter les contours d'une image avec XSG.....	54
Fig.12. Les résultats de simulation obtenus.....	56
Fig.13. Schéma bloc du système de détection de contours complet.....	57
Fig.14. Génération du code compilé à partir de blocs XSG.....	57
Fig.15. L'ajout des fichiers compilés par XSG au projet créé dans ISE.....	58
Fig.16. Evaluation du système réalisé en temps réel.....	60

Liste des tableaux

Chapitre 1 notion de base sur le traitement d'image numérique

Tableau. 1. Codage des couleurs.....	8
Tableau. 2. Quelques exemples d'algorithmes de traitement d'images.....	14

Chapitre 2 la détection des contours

Tableaux .2. les caractéristiques de filtre Roberts.....	39
.Tableaux .1. l'échelle de précision pour les algorithmes canny et sobel.....	41

Chapitre 3 Implémentation du filtre de sobel sur un FPGA

Tableau.1. le rapport des ressources consommés par le système implémenté...	58
---	----

INTRODUCTION GENERALE

Aujourd'hui l'image numérique fait partie de notre vie quotidienne, la télévision, les téléphones mobiles, l'imagerie médicale, contrôle de trafic, robotique, etc. sont tous des domaines où l'image numérique joue un rôle primordial. Les informations visuelles sont essentielles pour rendre les réseaux sociaux si populaires. Des appareils photo numériques aux téléphones portables, prendre des photos est devenu une expérience quotidienne pour la plupart d'entre nous.

Le traitement d'image numérique consiste à utiliser un ordinateur numérique pour traiter des images numériques à l'aide d'un algorithme. En tant que sous-catégorie ou domaine du traitement de signal numérique, le traitement d'image numérique présente de nombreux avantages par rapport au traitement d'image analogique. Cela permet d'appliquer une gamme beaucoup plus large d'algorithmes aux données d'entrée et peut éviter des problèmes tels que l'accumulation de bruit et de distorsion pendant le traitement. La génération et le développement du traitement d'image numérique est principalement affectés par le développement des ordinateurs.

L'un des algorithmes de traitement d'images numériques les plus connus est la détection de contours. La détection des contours comprend une variété de méthodes mathématiques qui visent à identifier des points dans une image numérique auxquels la luminosité de l'image change brusquement ou, plus formellement, présente des discontinuités. Le but de la détection de changements brusques de la luminosité de l'image est de capturer des événements importants et des changements dans les propriétés du monde. On peut montrer que sous des hypothèses assez générales pour un modèle de formation d'image, les discontinuités dans la luminosité de l'image sont susceptibles de correspondre à discontinuités en profondeur, discontinuités dans l'orientation de la surface, modifications des propriétés des matériaux et variations de l'éclairage de la scène.

L'un des plateformes ordinateurs les plus utilisées aujourd'hui est le FPGA (Field-Programmable Gate Array). Aujourd'hui, les circuits FPGA devenus véritablement des dispositifs révolutionnaires qui combinent les avantages du matériel et logiciel. Un FPGA est un circuit intégré conçu pour être configuré par un concepteur après la fabrication - d'où le terme « programmable sur site (Field) ». La configuration FPGA est généralement spécifiée à l'aide d'un langage de description matérielle (HDL), similaire à celui utilisé pour un circuit

intégré spécifique à une application (ASIC). Les FPGAs offrent une performance considérable due à la structure parallèle de ses ressources logiques programmables. Cette avantage permet aux FPGAs de devenir le bon choix pour implémenter des algorithmes qui nécessitent une performance élevée, à titre d'exemple les algorithmes de traitement d'images. Le traitement numérique des images permet l'utilisation d'algorithmes beaucoup plus complexes, et nécessitent donc des performances plus sophistiquées pour des tâches simples.

L'objectif de ce travail est d'implémenter un algorithme (un filtre) pour la détection de contours dans un FPGA en temps réel. L'exemple de filtre de Sobel est prise comme un exemple dans ce travail. Nous avons exploiter l'outil de conception XSG (Xilinx System Generator) pour achever la conception de système désiré. Cet outil permet de réaliser des systèmes numériques assez compliqués mais sans connaissance des langages de description matériel comme le VHDL. L'outil ajoute des blocs dans la bibliothèque de Simulink/Matlab, ces blocs sont utilisés pour concevoir le filtre de Sobel, de simuler le bon fonctionnement de ce dernier et en fin de générer le code VHDL automatiquement. Le code VHDL généré est ensuite sera exporté vers l'outil ISE ((Integrated Synthesis Environment) pour compléter les étapes restants pour l'implémentation en temps réel.

Ce mémoire est organisé comme suite ; le premier chapitre sera consacré à la définition de l'image numérique avec ses différentes variantes ainsi que son traitement numérique. L'objectif de deuxième chapitre est de présenter les bases principaux de la détection de contour et les algorithmes existants (filtres) pour réaliser cette tache notamment le filtre de Sobel que nous devons réaliser sur le circuit FPGA. Le 3ème chapitre sera consacré à la présentation les étapes d'implémentation du filtre de Sobel. Le choix de ce dernier est fait en raison de sa simplicité. Ce chapitre commencera par présenter la carte de développement à base de FPGA utilisée pour l'implémentation, les étapes de réalisation du projet, les résultats obtenus par simulation avec comparaison de ceux obtenus par Matlab et finalement la réalisation pratique sur le circuit FPGA en temps réel.

Le mémoire sera clôturé par une conclusion sur les résultats obtenus ainsi que les perspectives sur les futures travaux dans ce domaine.

CHAPITRE 1 : NOTIONS DE BASE SUR L'IMAGE NUMERIQUE

1. Introduction

Aujourd'hui l'image numérique fait partie de notre vie quotidienne, la télévision, les téléphones mobiles, l'imagerie médicale, contrôle de trafic, robotique, etc. sont tous des domaines où l'image numérique joue un rôle primordial. Le traitement d'image numérique est un domaine de recherche et de développement très intéressant. Le traitement numérique de l'image est utilisé pour manipuler les images en utilisant des algorithmes, ce terme fait généralement référence au traitement d'une image bidimensionnelle par un ordinateur numérique. Le traitement numérique d'image se concentre sur deux tâches principales : Amélioration des informations picturales pour l'interprétation humaine et traitement d'image pour le stockage, la transmission et la représentation pour une perception autonome de la machine. Ce chapitre sera consacré à la définition de l'image numérique et la présentation de ces attributs ainsi que le domaine d'application et des exemples au traitement d'image .

2. L'image numérique

La visibilité des objets se fait par la réflexion de la lumière sur ces objets lorsqu'un rayon de lumière rencontre un objet, cet objet en absorbe une partie correspondant à sa couleur, et disperse le reste sous forme des rayons. rayons peuvent éventuellement être captés par un œil ou tous appareil sensible à la lumière, ces rayons constituent l'image visible. Une image numérique est une représentation bidimensionnelle sous la forme d'un ensemble fini de valeurs numériques, appelées éléments d'image ou pixels

1.2 La numérisation d'image

On peut définir l'image analogique comme une distribution continue d'intensités lumineuses dans un plan (xoy) , la numérisation d'image est la conversion d'image analogique vers l'image numérique qu'elle représentée par une matrice bidimensionnelle de valeurs numériques $X(n,m)$. où: n,m sont les coordonnées cartésiennes d'un point de l'image et $X(n,m)$ le niveau de gris ou de couleur en ce point.

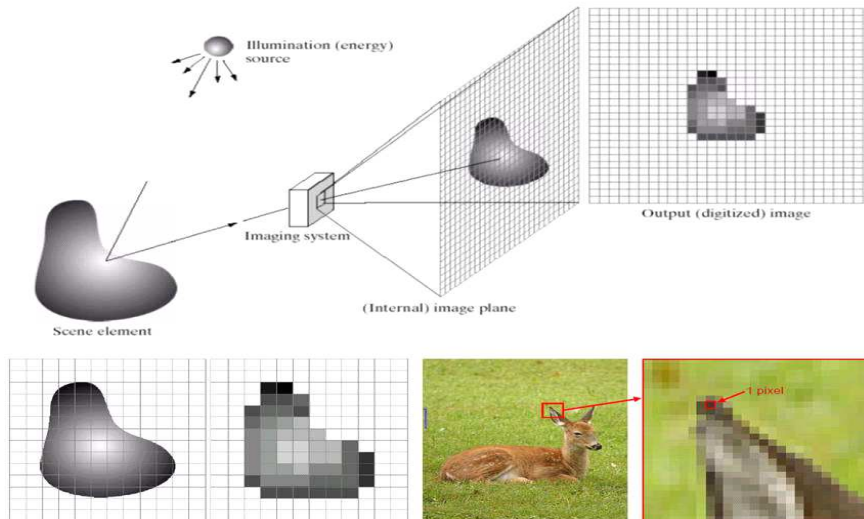


Fig.1. L'image numérique et notion de pixel [1].

L'acquisition d'une image numérique est réalisée en trois étapes:

- 2.1.1. **Échantillonnage** : la lecture d'intensité à des emplacements régulièrement espacés dans les directions x et y en plaçant une grille régulièrement espacée sur l'image analogique, les lectures obtenus à ces endroits sont les intensités définissant la taille du plus petit élément de l'image de pixels.
- 2.1.2. **Quantification** : Les valeurs d'intensité échantillonnées sont quantifiées arrivant à un signal qui est discret dans les deux positions et amplitudes. Ce signal représente l'image.
- 2.1.3. **Encodage** : La conversion des données en une forme binaire[1].

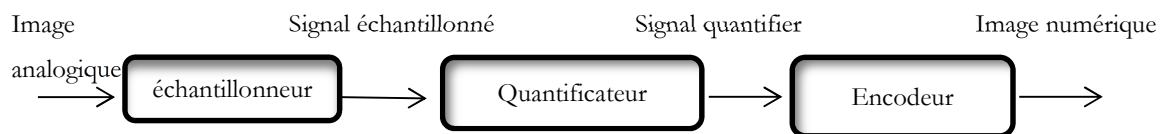


Fig.2. Conversion d'une image analogique en une image numérique.

3. Les attributs d'une image

3.1 Pixel

Contraction de l'expression anglaise " Picture Elements " éléments d'image, une image numérique est composée d'une grille de pixels , ces pixels sont autant de petits carrés porteurs

d'une information de couleur élémentaire, c'est une entité calculable qui peut recevoir une structure et une quantification. Le pixel est le plus petit élément que peuvent manipuler les matériels et logiciels d'affichage ou d'impression. Le nombre de pixels constituant l'image c'est le nombre de colonnes de l'image que multiplie son nombre de lignes , la quantité d'information que véhicule chaque pixel donne des nuances entre images monochromes et images couleurs [2].

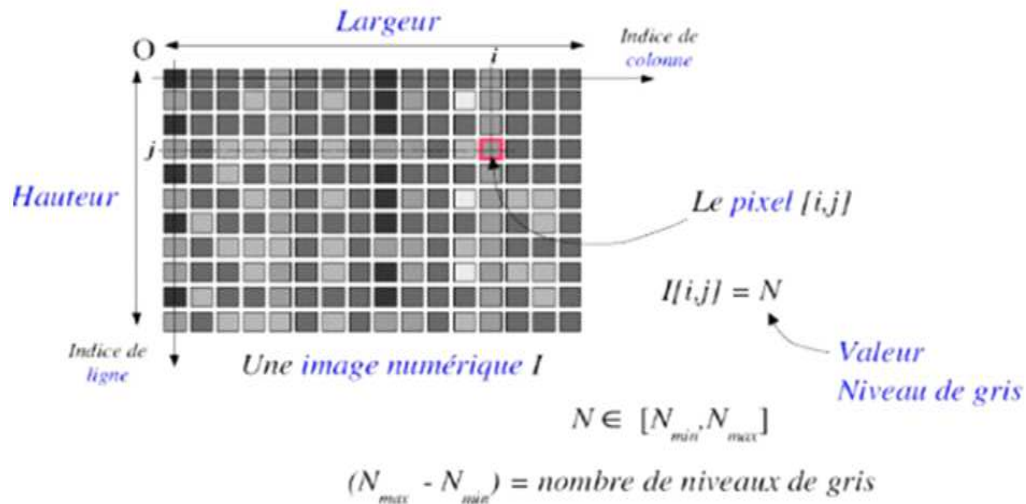


Fig.3. Représentation des pixels et niveaux de gris [3].

- **Relations entre les pixels :**

Les pixels en gris sont les voisins du pixel(m;n).

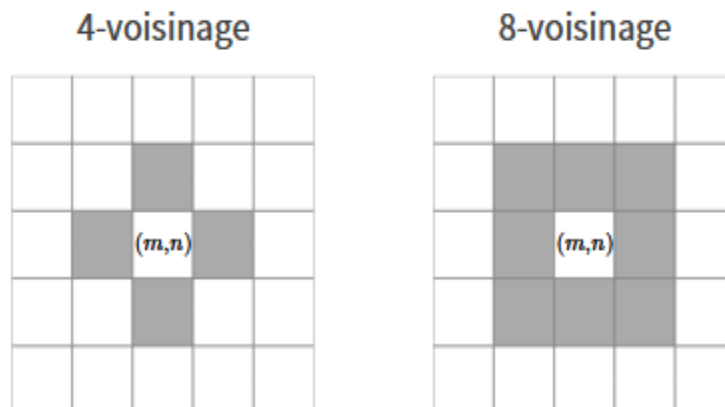


Fig.4. les voisins du pixel de coordonnées (m;n) [4] .

3.2 Dimension

C'est la taille de l'image, se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Pour connaître la taille d'une image,

il est nécessaire de compter le nombre de pixels que contient l'image, cela revient à calculer le nombre des cases du tableau, soit la hauteur de celui-ci que multiplie sa largeur.

Exemple : pour une image de 240 X 420 en vraie couleur :

Nombre de pixels : $240 \times 420 = 100800$

Taille de chaque pixel : $24 \text{ bits} / 8 = 3 \text{ octets}[2]$.

3.3 Résolution

La résolution est définie par un nombre de pixels par unité de longueur de l'image à numériser exprimé en points par pouce DPI pour Dots Per Inch , plus le nombre de pixels est élevé par unité de longueur de l'image à numériser, plus la quantité d'information qui décrit l'image est importante et plus la résolution est élevée[3].

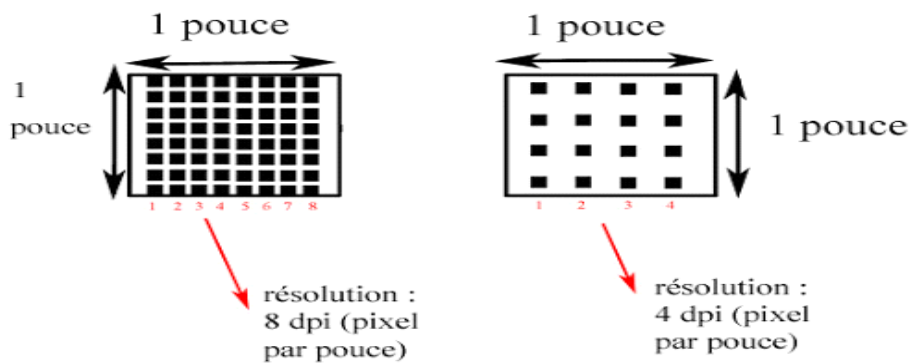


Fig.5. représentation de résolution [3].

Un pouce mesure 2.54 cm, La résolution permet ainsi d'établir le rapport entre la définition en pixels d'une image et la dimension réelle de sa représentation sur un support physique (affichage écran, impression papier...)

3.4 Luminance

Il s'agit du degré de luminosité des points d'image, également connu sous le nom de produit de l'intensité lumineuse d'une surface de la zone visible de cette surface , un bon luminosité se caractérise par les éléments suivants:

- Photos lumineuses (brillantes).
- Bon contraste: évitez les photos dont la plage de contraste tend vers le blanc ou le noir, ces images provoquent la perte de détails dans les zones sombres ou claires.

- Absence de parasites[1].

3.5 Contraste

En fonction des luminances entre deux zones d'images ,c'est l'opposition entre les régions sombres et les régions claires de l'image .Le contraste C est défini par le rapport :

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

Avec L1 et L2 sont les degrés de luminosité respectivement de deux zones voisines A1 et A2 d'une image [2].

3.6 Histogramme

Histogramme est un outil fondamental du traitement d'images et représentation visuelle des zones de lumière d'une image .En imagerie numérique, cette fonction nous donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image .Il existe nombreuses applications d'utilisation d'histogramme les plus connu pour la retouche d'images et la photographie. Le traitement d'histogramme permet d'améliorer la qualité d'une image en intervenant sur la distribution de ses pixels[5].

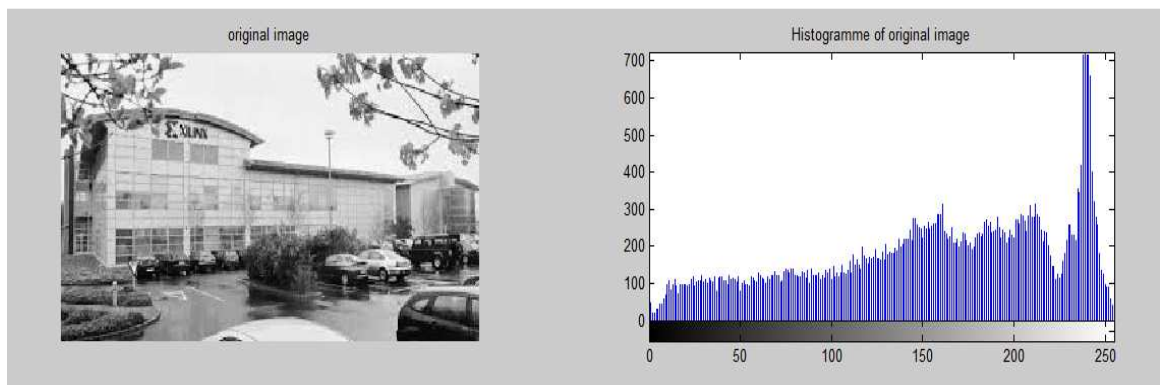


Fig.6. Exemple d'histogramme.

3.7 Bruit

Le bruit définit les interférences d'un signal, ou les parties du signal sont les parasites y sont ajoutés de façon aléatoire aux détails de l'image original , Ainsi le bruit d'une image indique

les pixels de l'image dont l'intensité est très distincte de celles des pixels voisins. Le bruit peut provenir de différentes causes :

- Qualité de l'échantillonnage.
- Capacité des capteurs ou une mauvaise utilisation de ces derniers.
- Environnement lors de l'acquisition [6] .



Fig.7. Image avec et sans bruit [6].

3.8 Le codage des couleurs

Les images peuvent être en noir et blanc ou en couleur. Le nombre de ces couleurs dépend de deux facteurs l'image elle-même ou bien son encodage, à savoir le nombre de bits par pixel utilisés. Le codage de couleur est réalisé sur trois octets, chaque octet représentant la valeur d'une composante couleur par un entier de 0 à 255. Ces trois valeurs codent généralement la couleur dans l'espace RVB [1].

R	V	B	Couleur
0	0	0	noir
0	0	1	nuance de noir
255	0	0	rouge
0	255	0	vert
0	0	255	bleu
128	128	128	gris
255	255	255	blanc

Tableau. 1. Codage des couleurs [1].

3.9 Les types de fichiers

Le fichier de l'image contient toutes les informations de l'image : nom de fichier, taille de l'image, format des pixels, valeurs des pixels...etc. Pour sauvegarder ces informations dans un fichier informatique il existe une multitude de formats :

- **Les formats simples** : où l'on trouve dans cette catégorie des fichiers d'en-tête ASCII ou binaires contenant les dimensions de l'image et le format des pixels, puis la liste des valeurs des pixels.
- **Les formats compressés** : L'information est codée de manière à réduire la taille du fichier image [7].

3.10 Contenu d'une image

- **Texture** : répartition des intensités dans l'image de façon statistique ou géométrique
- **Contour** : représentent la frontière entre les objets de l'image ,ou la limite entre deux (ou un groupe)de pixels dont la différence de niveaux de gris (ou de couleurs).
- **Région** : un contour fermé définit une région , groupe de pixels présentant des caractéristiques similaires (intensité, mouvement).
- **Objet** :région ou bien un groupe de régions entièrement délimitée par un contour , possédant une indépendance dans l'image description sémantique [5] .



Fig.8 Contenu d'une image [5] .

4. Les différents type d'image :

4.1 Images binaires

Le pixel représenter par deux états logique noir (0) et blanc (1). C'est typiquement le type d'image que l'on utilise pour scanner du texte quand celui-ci est compose d'une seule couleur[3].

4.2 Images en niveaux de gris :

Généralement les images en niveaux de gris contiennent 256 couleurs de gris. L'image contient 256 couleurs, chacune de ces 256 couleurs est définie dans la gamme de tons de gris. En règle générale, la valeur zéro est noire (intensité lumineuse nulle) et 255 blanches (intensité

lumineuse maximale)[3].



Fig.9 . Représentation d'image binaire .



Fig.10. Représentation d'image en niveau de gris

4.3 Images en couleur

S'il existe plusieurs modes de représentation des couleurs, le traitement d'image numérique le plus couramment utilisé est la zone rouge, verte et bleu (R, G, B) sur la base de la synthèse de couleur supplémentaire [3]. Pour passer d'une image couleur à une image niveau de gris, on réalise :

$$I(y,x) = \frac{R(y,x) + V(y,x) + B(y,x)}{3} \quad (2)$$



Fig.11. Représentation d'image couleur

5. Les formats d'image numérique:

5.1 image matricielle

On appelle image matricielle ou « carte de points » sont appelées en anglais « bitmap » ou « raster » un ensemble fini de points colorés appelés pixels (contraction de Picture élément) organisés en matrice (tableau à 2 dimensions), cette matrice contraint l'image à des dimensions fixes une largeur et une hauteur le tout exprimé en pixels , une troisième dimension appelée profondeur permet de coder la couleur de chaque pixel plusieurs pixels s'affiche dans une grille .Lorsqu'on agrandi trop l'image, on perd de la qualité (« pixelisation ») exemples de formats : jpeg, png, gif [8].

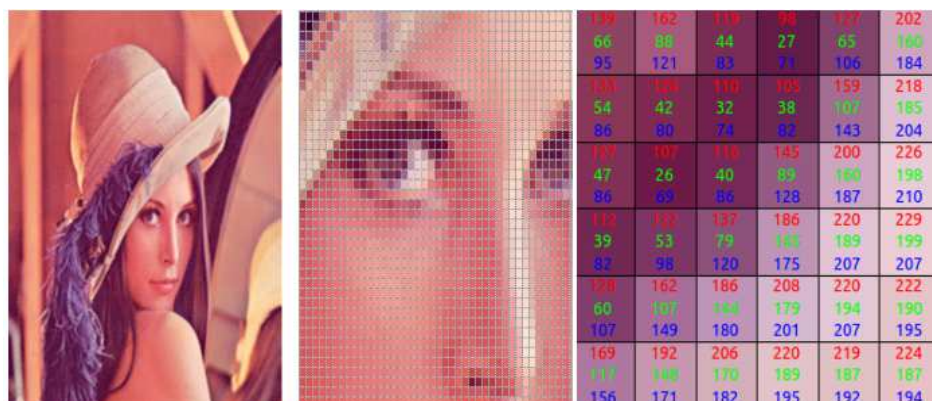


Fig.12. Image numérique matricielle [9].

5.2 les images vectorielles

Les images vectorielles sont constituées de formes géométriques et sont prend la forme d'une suite de formules mathématiques décrivant les formes élémentaires constituant l'image (carrés, rectangles, ellipses, cercles, courbes, etc.). Chaque forme élémentaire constitue un objet et se voit assigné un certain nombre d'attributs tels que la couleur, la transparence, l'épaisseur du trait, le type de trait (pointillé, etc.). Donc le stockage d'une image de type vectoriel consiste en la mémorisation de la représentation des coordonnées des points caractéristiques des formes qui constituent l'image. Il s'agit dès lors d'une représentation relative que l'on peut qualifier de "symbolique", exemple des formats *DXF*, *EPS* [10].

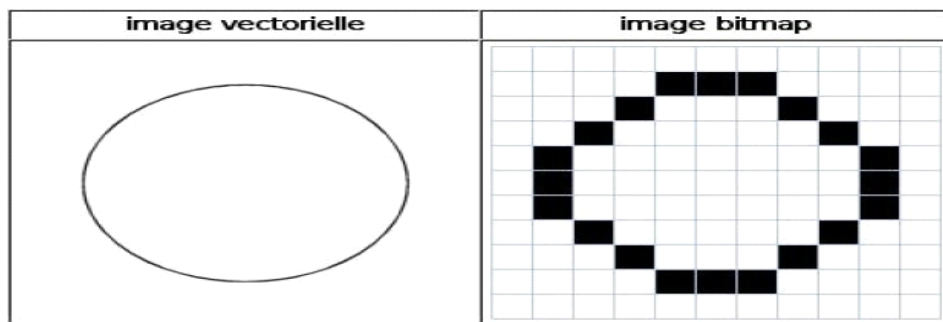


Fig.13. Image vectorielle et image bitmap [10].

6. Acquisition d'image numérique

L'acquisition d'images constitue un des maillons essentiels de toute chaîne de conception et de production d'images. Pour pouvoir manipuler une image sur un système informatique, il est avant tout nécessaire de lui faire subir une transformation qui la rendra lisible et manipulable par ce système. Le passage de cet objet externe (l'image d'origine) à sa représentation interne (dans l'unité de traitement) se fait grâce à une procédure de numérisation. Les outils d'acquisition d'image sont les scanners et les appareil photo numérique [11].

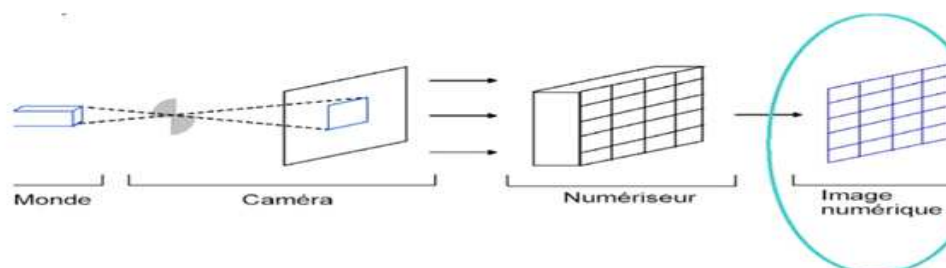


Fig.14. L'acquisition des images [11].

6.1 Les outils d'acquisition :

- ***Le scanner***

Le scanner est certainement l'appareil le plus pratique et le plus répandu pour obtenir des photos. C'est le périmètre du mini-ordinateur . La numérisation est autorisée et communique généralement avec l'ordinateur via la carte SCSI propriétaire, souvent via un port USB.

- **Appareil photo numérique**

Les caméras numériques sont spécialement conçues pour effectuer toutes les opérations de vision numérique populaires . Des appareils photo numériques pratiques, rapides, légers et compacts rendront les vidéos plus réalistes [11].

7. Traitement d'images

On désigne par traitement d'images numériques l'ensemble des techniques consiste à appliquer des transformations mathématiques sur les images dans le but de l'améliorer ou d'en extraire des informations . On peut citer par exemple la création d'effets spéciaux au cinéma, l'amélioration de la qualité des photos en corrigeant les conditions de prise de vue, la détection de lettres ou de visages, l'identification de zones cancéreuses en imagerie médicale, la compression JPEG [4].

7.1 Chaîne de traitement d'images

7.1.1 Traitements classification et étapes fondamental

- ✓ Traitement bas niveau
 - Entrée : image
 - Sortie : image
- ✓ Traitement moyen niveau
 - Entrée : image ou représentation préalable
 - Sortie : nouvelle représentation
- ✓ Traitement haut niveau
 - Entrée : image ou représentation préalable
 - Sortie : sémantique de l'image (abstraction) [12].

Traitement d'images	Algorithmes
Bas niveau : Prétraitement	Histogramme, lissage, filtres passe-bas filtres médians, filtres morphologiques, filtres de contours, transformée de Fourier, transformée en Ondelettes, DCT, transformée de Hough ...
Moyen niveau : Extraction des caractéristiques et passage d'une image en une liste de paramètres	Couleur, texture, distorsion, sémantique, mouvement, profondeur, compression, étiquetage de région, fusion ...
Haut niveau : Système décisionnel avec trois espaces de décision (1D : Temporel, 2D : Spatial et 3D : Spatio-temporel)	Classification, algorithme d'optimisation, qualité de représentation, similarité, homogénéité ...

Tableau .2 .Quelques exemples d'algorithmes de traitement d'images[12].

7.1.2 Traitement d'images en temps réel

La notion de traitement en temps réel peut être définie comme suit : les informations doivent être traitées au même rythme que leur acquisition. Dans le domaine de la vision artificielle, le traitement d'images en temps réel peut être appliqué à plusieurs niveaux :

Au niveau pixel c'est le cas de la segmentation au sens région ou contour .

Au niveau de l'image, une décision prise globalement pour toute l'image, classe celle-ci dans une catégorie donnée .

A des niveaux plus élevés encore, comme dans une séquence d'images. Afin de satisfaire aux contraintes temporelles, des implantations matérielles d'algorithmes sont souvent nécessaires, soit sur une machine standard, soit sur une machine spécifique [12].

7.1 Domaine d'application du traitement d'image

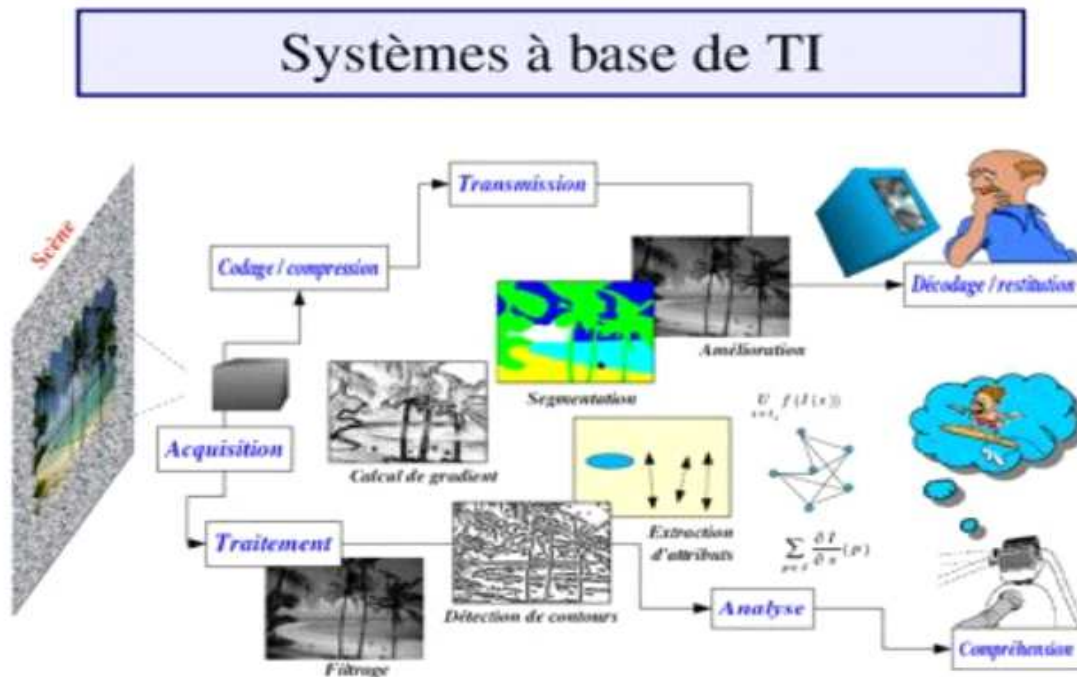


Fig.15. systèmes à base de traitement d'image [12].

Il existe nombreux domaines d'application du traitement d'image, citons par exemple

La télédétection : télédétection à partir d'images satellite , météo, cartographie, analyse des ressources terrestres, astronomie , l'aménagement du territoire .

Le domaine médical : tomographie ,radiographie, l'imagerie par résonance magnétique(IRM) , échographie, scanner.

Le domaine militaire : guidage de missiles en tous genres, reconnaissance aérienne, sous-marine, détection de mouvement , de croisière , la robotique mobile et autonome militaire les différents domaines de, la physique , géologie , La photographie.

7.2 Logiciels pour faire du traitement d'image

Les logiciels plus importants que vous traitez les images sont :

- ✓ MATLAB
- ✓ Des bibliothèques à partir du C et du C++ dont OpenCV
- ✓ Python

8.3 Les différents types de traitement d'image

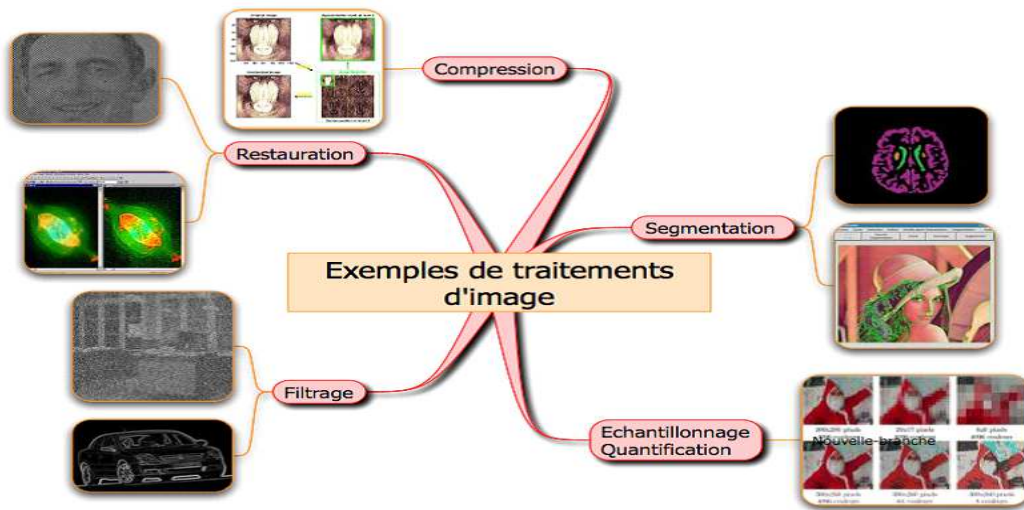


Fig.16. Exemple de traitements d'image [4].

8.3.1 Filtrage d'une image :

8.3.1.1 Application de filtres et introduction à la convolution

Le filtrage consiste à balayer l'image par une fenêtre d'analyse de taille finie ou kernel (noyau). Le calcul du nouveau niveau de gris du pixel considéré ne prend en compte que les plus proches voisins de celui-ci. Bien des traitements d'images sont basés sur ce que l'on appelle l'opérateur de convolution. Ce dernier correspond à une multiplication de matrices. En l'occurrence ici de la matrice image avec un filtre donné [13].

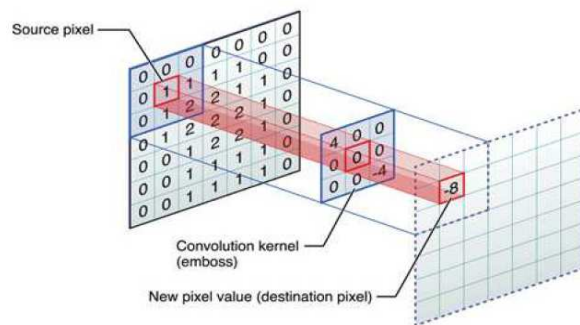


Fig.17. Schéma explicatif de la convolution [13].

8.3.1.1 Les type de filtres

Il existe plusieurs filtres utilisés en traitement d'image :

8.3.1 . Les filtres linéaires

Les filtres linéaires convertissent un ensemble de données d'entrée en un ensemble de

données de sortie avec une méthode appelée la convolution , relation entre les valeurs des pixels de sortie et les valeurs d'entrée est décrite par une table numérique, appelée matrice de convolution

8.3.2. Filtre passe-bas

Ce type de filtre permet de laisser passer les basse fréquence et atténuer les haute fréquence souvent utilisé pour réduire le bruit de l'image on parle dans ce cas de lissage.

- **Filtre moyen** : est un filtre passe-bas ,il permet de faire la moyenne des valeurs des pixels avoisinants. Le résultat de ce filtre est une image plus floue.
- **Filtre gaussien** :Un filtre isotrope spécial avec des propriétés mathématiques bien précises le propriétés de diminution de bruit des filtres Gaussien peuvent être utilisée en combinaisons avec d'autres filtres qui au contraire engendrent du bruit, comme les filtres Laplacien.

A.2 Filtre passe-haut

Permet de laisse passé les haute fréquence et atténuent les composantes de basse fréquence de l'image et permettent d'appuyer les détails et le contraste



Fig .18. Exemple d'image filtrée .

B. Les filtres non linéaires

Permet d' intervenir les pixels voisins suivant une loi non linéaire, opèrent sur un voisinage donné, le filtre le plus utilisé est le filtre médian

- **Filtre médian** : le principe de ce filtre est atténuer des pixels isolés, d'une valeur très différente de leur voisinage. ce filtre n'introduit pas de flou sur les contours des particules contrairement aux lissages[6].

8.3.2 Amélioration d'images

Pour le but d'améliore la qualité d'image en traitement d'image numérique on peut faire des modifications sur les retouche d'image avec un ordinateur a l'aide des logiciel



Fig.19 . Exemple d'amélioration d'image [9].

8.3.3 Restauration d'images

La restauration d'image est une technique a le but de la reconstitution d'une image dégradée à partir de l'image original , la méthode de restauration se fait par le scanner qui il un bon résolution pour importer l'image dans l'ordinateur ,puis la correction ce fait par les logiciels de retouche d'image .



Fig.20. Exemple de restauration d'image [9].

8.3.4 Compression d'image

L'objectif de la compression la réduction du volume de l'image est de représenter l'information contenue dans l'image avec moins de bits que l'image originale. Il existe deux type de compression :

- **Compression sans perte** exemple d'algorithme Codage de Huffman , RLE (*Run Length Encoding*)

- **Compression avec perte** exemple d'algorithme JPEG [4].

8.3.5 La segmentation d'image

Est une opération en traitement d'image a le but de rassembler les pixel entre eux suivant des critères prédéfinis . Les pixels sont ainsi regroupés en régions, qui composent une partition de l'image. La segmentation consiste à diviser l'image en plusieurs régions homogènes L'homogénéité d'une région est basée sur la couleur, la texture, les contours.

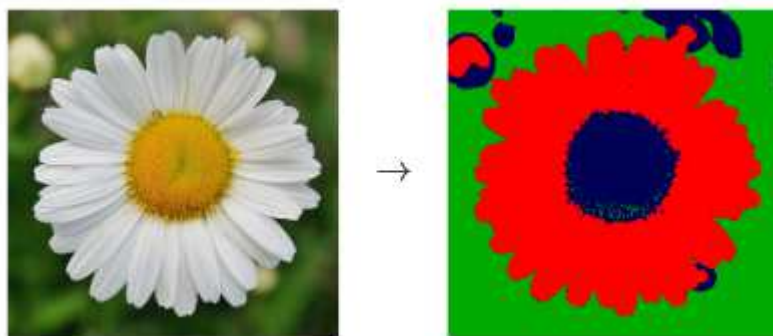


Fig.21. Segmentation d'une image couleur [4].

On peut regrouper les méthodes de segmentation comme suite :

La segmentation fondée sur les régions.

La segmentation fondée sur les contours.

La segmentation fondée sur la classification ou le seuillage des pixels en fonction de leur intensité [4].

- **Seuillage**

Le seuillage consiste à affecter le niveau 255 aux pixels dont la valeur est supérieure à un seuil S et 0 le niveau aux autres. Le graphe de la transformation correspondante est le suivant [3].

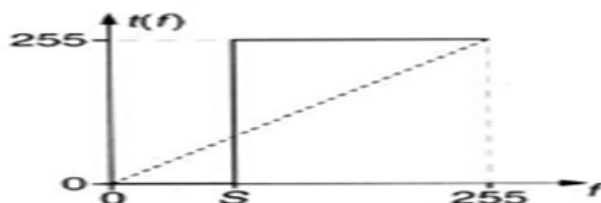


Fig.22. La fonction seuillage [3].

9. Conclusion

Dans ce chapitre nous avons abordé les notions de bases de l'image numérique définition de l'image numérique par les différentes méthodes utilisées pour traiter les images les caractéristiques liées à l'image en plus la différence entre les types d'image. Enfin, nous donnons des exemples d'applications dans lesquelles l'image est traitée.

CHAPITRE 2 : DETECTION DE CONTOURS

1. Introduction

L'une des tâches fondamentales du traitement d'image est la détection des contours. Le traitement d'image de haut niveau, comme la reconnaissance de détection d'objet, la segmentation, le codage d'image et la vision du robot, dépend de la précision de la détection des contours. Les algorithmes de détection des bords sont utilisés pour détecter les bords des objets dans l'image. Les bords se produisent généralement à la frontière entre deux différentes régions de l'image. Il existe plusieurs algorithmes pour cela, mais ceux-ci peuvent être classés comme dérivés basé où l'algorithme prend la dérivée première ou seconde sur chaque pixel, Plusieurs méthodes peuvent être utilisées pour détecter ces contours. L'objective de ce chapitre est de présenter les bases principaux de la détection de contour et les algorithmes existants (filtres) pour réaliser cette tache notamment le filtre de Sobel que nous devons réaliser sur le circuit FPGA.

2. Détection de contours

2.1. Définition

La détection de contours est une technique de réduction d'information dans les images, qui consiste à transformer l'image en un ensemble de courbes, pas forcément fermées, formant les frontières significatives de l'image. Si les structures extraites sont simples à manipuler (courbes fines, régulières, stables...), elles peuvent être utiles pour la mise en correspondance d'images (robotique, indexation)[14].

2.2. Les objectifs de la détection des contours

Produire un dessin au trait d'une scène à partir d'une image de cette scène, des caractéristiques importantes peuvent être extraites des bords d'une image (par exemple, coins, lignes et courbe) et ces fonctionnalités sont utilisées par des algorithmes de vision par ordinateur de niveau supérieur (par exemple, divers événements physiques provoquent des changements d'intensité, des événements géométriques Limite de l'objet où la discontinuité dans la profondeur et / ou la couleur et la texture de la surface et la surface frontière où discontinuité dans l'orientation de la surface et / ou la couleur et la texture de la surface)[15].

2.3. Contour d'une image

Les contours physiques sont représentés dans l'image par l'ensemble des pixels connectés qui forme une frontière entre deux régions disjointes localisés à des positions de changement brusque dans la fonction d'intensité lumineuse. Étant donné que l'intensité de l'image est souvent proportionnelle à l'éclat de la scène. Un contour se matérialise par une rupture d'intensité dans l'image suivant une direction donnée. Parmi ces applications est la reconnaissance des formes, d'objet, classifications des scènes.

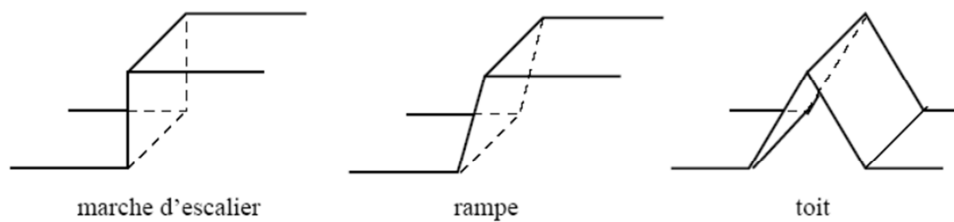


Fig.1. les types de contours [5] .

- **Marche** : d'escalier : si le contour est net
- **Rampe** : si le contour est plus flou
- **Toit** : s'il s'agit d'une ligne sur un fond uniforme

Modèle le plus courant c'est : Marche d'escalier.

2.4. Les étapes de détection de contour

En général, la détection des contours comporte trois étapes nécessaires : lissage, filtrage, localisation.

2.4.1. Le Lissage d'image

L'image n'apparaît pas parfaitement là où les parasites y sont ajoutés de façon aléatoire aux détails de l'image original, ceci est connu sous le nom de bruit et est principalement dû à la physique des capteurs et des composants électroniques. L'objectif de lissage d'image est de réduire le bruit afin d'obtenir une meilleure restitution de l'image pour l'analyse et supprimer autant de bruit que possible sans détruire les vrais bords.

2.4.2. Le filtrage

Certains grands détecteurs de bord conventionnels fonctionnent bien avec des images de haute qualité, mais ils ne sont souvent pas assez bons les images sont bruyantes car elles ne permettent pas de distinguer des bords de sens différents. L'objectif principal des techniques d'optimisation est de produire une image meilleure et plus pertinente que l'image originale d'une application spécifique. Des filtres linéaires ont été utilisés pour résoudre plusieurs problèmes d'amélioration d'image.

Les critères pour de bons filtres de bord:

- **Aucune réponse aux régions plates** : La somme des valeurs de masque est nulle.
- **Isotropie**: La réponse doit être indépendante de l'orientation des bords.
- **Bonne détection**: Minimiser les probabilités de détecter les bords parasites causés par le bruit et bords réels manquants.
- **Bonne localisation**: Les bords détectés doivent être aussi proches que possible des vrais bords.
- **Réponse unique**: Minimiser le nombre de faux maxima locaux autour du vrai bord.

2.4.3. Localisation

Déterminer l'emplacement exact d'un bord (résolution sous-pixel peut être nécessaire pour certaines applications, c'est-à-dire estimer l'emplacement d'un bord pour mieux que l'espacement entre les pixels). L'amincissement des bords et la liaison sont généralement requis [15, 16].

3. L'effet mutuel entre détecteurs et contours

La relation d'influence mutuelle entre bords et détecteurs permet de préciser le contexte dans lequel il est significatif par exemple, le choix d'un détecteur de bord spécifique pour une image donnée nécessite une connaissance de ses performances. Ainsi, cette relation d'influence mutuelle conduit à une meilleure utilisation de détecteurs de contour et résultant en une meilleure performance, l'augmentation continue du nombre de détecteurs de bord rend difficile l'inventaire des algorithmes disponibles. Sur le bord le plus moderne, les détecteurs sont autonomes et comprennent des étages

Les trois principaux sont : lissage, différenciation et étiquetage. Ils diffèrent par les filtres de

lissage, les différentiels, le processus de marquage, les objectifs, la complexité de calcul et les modèles mathématiques utilisés pour les dériver. Les capteurs contextuels sont de loin les plus conçus. Leur objectif est différent, tout comme les connaissances utilisées pour extraire les bords[18].

4. Les approximations de détection de contours

De nombreux algorithmes de détection des contours sont utilisés pour détecter les bords des objets dans l'image, mais ceux-ci peuvent être classés en tant que dérivé où l'algorithme prend la première ou la deuxième dérivée sur chaque pixel, ou en fonction du gradient où un gradient de pixels consécutifs est pris dans les directions x et y. Une opération appelée opération du noyau est généralement effectuée.

Il existe plusieurs facteurs qui peuvent corrompre la performance de la détection de contours dans les images réelles. Ceux-ci incluent le bruit, l'interférence entre les contours adjacents et les erreurs de précision dues au caractère discret de la grille de définition de l'image. De plus, un contour peut se retrouver dans une zone nette, ou noyé dans une zone bruitée ou floue. Pour le détecter, il faudra donc utiliser un détecteur approprié à la caractéristique de la zone [17].

Les approches de détection de contour sont les plus utilisées :

- Approximation de gradient.
- Approximation du Laplacien.

Un contour correspond à une transition. Cette transition en couleur :

- ça correspond à un extremum dans la dérivée première.
- et à un passage par zéro dans la dérivée seconde.

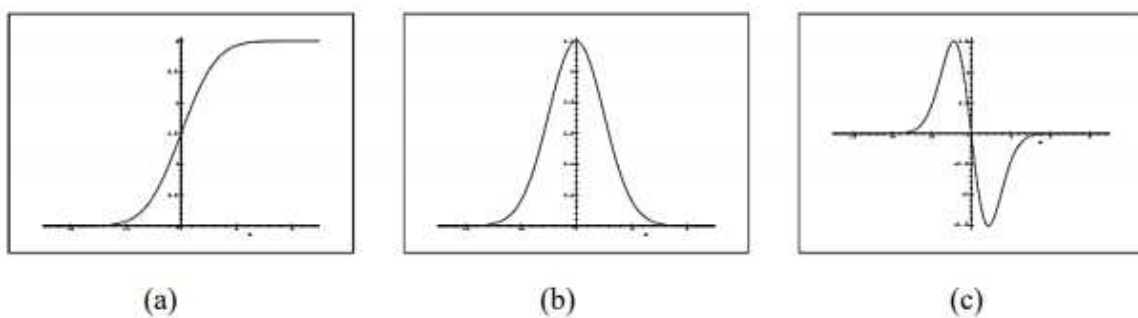


Fig.2. Principe de la détection de contour par gradient ou Laplacien, (a) Contour de type marche. (b) Dérivée première. (c) Dérivée seconde [7].

4.1. Approximation de gradient

4.1.1. Définition

Le gradient en un point est un vecteur dont les deux composantes ($\partial I/\partial x$ et $\partial I/\partial y$) mesurent à quelle vitesse la valeur des pixels évolue dans chacune des directions x et y , autour du point d'intérêt.

L'approche basée sur le gradient est également appelée masque en numérique images. Approximations différentielles dans le sens horizontal ou vertical la direction de l'image est calculée avec le masque numérique

Le Module du gradient est liée à la quantité de variation locale de l'intensité : plus le module est grand, plus le contour est fort [7].

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (1)$$

4.1.2. Principe général

- Calculer d'abord le gradient de l'image dans les deux directions orthogonales (x et y).
- Calculer le module du gradient.
- Il s'agira ensuite d'effectuer une sélection de contours.
- Il s'agira ensuite d'effectuer une sélection de contours les plus marqués et pertinents (les point de plus fort contraste par un seuillage) [18].

4.1.3. La direction du gradient

La présence d'une discontinuité est assurée si le gradient au point considéré est suffisamment fort. Alors, on doit sélectionner les pixels à l'aide d'un seuil fixé a priori pour la norme du gradient $G \geq \text{seuil}$.

Les directions des contours étant orthogonales à la direction du gradient déterminée en chaque pixel de l'image. Le gradient permet d'obtenir la dérivée dans n'importe quelle direction la direction de gradient :

$$\vec{g} = \frac{\vec{\nabla} f}{|\vec{\nabla} f|}$$



Fig.3. Seuillage du gradient.

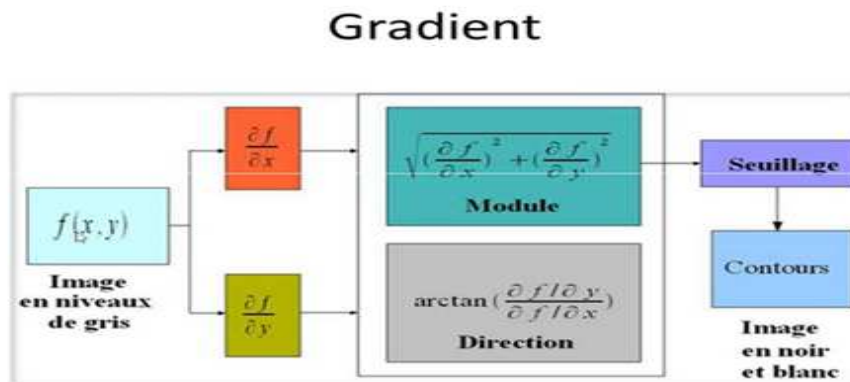


Fig.4. Le principe de gradient pour détection de contour[18].

4.1.4. Incorporation d'informations sur les gradients

Les emplacements d'image avec un fort gradient sont supposés indiquer une transition des tissus, qui sont des lieux de grande valeur informationnelle. Le gradient est calculé à une certaine échelle spatiale. Nous avons étendu les mesures d'information mutuelle (à la fois standard et normalisé). Pour inclure des informations spatiales en multipliant les informations mutuelles par un terme de gradient. Le terme de gradient est basé non seulement sur l'amplitude des gradients, mais aussi sur l'orientation des gradients [19].

4.1.5. Les avantages de Gradient

- Directionnel.
- Moins sensible au bruit.
- Possibilité de mise en œuvre par opérateurs 1D.
- Détection par la magnitude du gradient.
- Opérateurs de Sobel : lissage partiel.
- Possibilité de lissage préalable supplémentaire.
- Possibilité de détection directionnelle [20].

4.1.6. Calcul de gradient par masques

Dans le cas des algorithmes basés sur le gradient (ou opérateurs du premier ordre), les pixels contours sont localisés aux positions où le gradient exhibe des extrema locaux. L'opérateur gradient agit comme un filtre passe-haut d'où la forte sensibilité au bruit de ces algorithmes

Pour chaque opérateur deux masques sont utilisés de façon à déterminer le gradient de l'image dans deux directions orthogonales, le masque le plus intuitif mettre en œuvre pour une approximation de la dérivée discrète est un masque à deux éléments [17] :



Fig.5. les masques de gradient.

La figure 6 présente une illustration de l'application de ces masques.

4.2. Approximation laplacien

4.2.1. définition

C'est un filtre passe-haut dédié à la détection de contours sa particularité est de détecter des lignes horizontales, verticales et diagonales. Elle recherche les passages à zéro dans la dérivée seconde de l'image pour trouver les bords. Une arête a la forme unidimensionnelle d'une rampe et

calculer la dérivée de l'image peut mettre en évidence son emplacement [15].

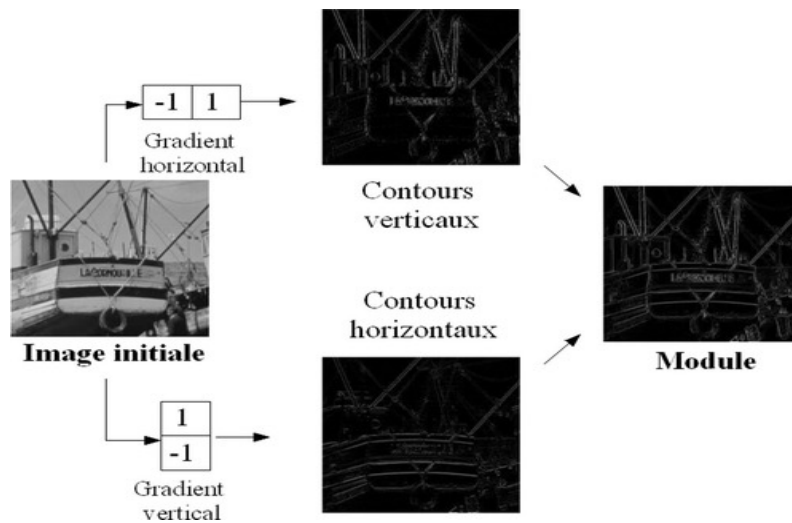


Fig.6. illustration de l'application de masque vertical et horizontal [21].

4.2.2. Le principe de Laplacien

Le détecteur de passage par zéro recherche les endroits dans le Laplacien d'une image où la valeur du Laplacien passe par zéro, c'est-à-dire les points où le Laplacien change de signe.

Le Laplacien d'une image met en évidence les régions de changement d'intensité rapide et est donc souvent utilisée pour les bords détection des détecteurs de front de franchissement zéro). Le Laplacien est souvent appliqué à une image qui a d'abord été lissé avec quelque chose se rapprochant d'un filtre de lissage gaussien afin de réduire sa sensibilité au bruit. L'opérateur prend normalement une seule image de niveau de gris en entrée et produit une autre image binaire en sortie [15].

4.2.3. Formule de laplacien

Le Laplacien est un opérateur scalaire et isotrope (invariant par rotation). Pour simplifier l'évaluation de la concavité locale de la fonction image f , exprimée par sa dérivée seconde, on fait appel à l'opérateur Laplacien, défini par la formule suivante[22] :

$$\Delta^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (3)$$

Pour la direction de x

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y) \quad (4)$$

Pour la direction de y

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y) \quad (5)$$

En substituant les équations (3) et (4) à (2), nous obtenons l'équation (5) :

$$\Delta^2 f(x, y) = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y) \quad (5)$$

4.2.4. Les masques de laplacien

✓ Masque de 4-connexités :

Gx

0	-1	0
-1	4	-1
0	-1	0

Gy

0	1	0
1	-4	1
0	1	0

Fig.7. Masque de laplacien de 4-connexités [22].

✓ Masque de 8-connexités :

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Fig.8. Masque de laplacien de 8-connexités [7].



Fig.9. Comparaison entre Laplacien 4 et 8 connexités[7].

4.2.5. Laplacien de détection des contours gaussiens(LOG)

Il est utilisé pour filtrer le bruit avant la détection des contours. Cette méthode relie le filtrage gaussien au Laplacien et constitue un moyen important de détecter les bords de l'image. Lorsqu'elle est utilisée le seuil avec la méthode Laplacienne ou gaussienne, cette commande est utilisée pour supprimer toutes les contours instables de seuil[23].

0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1
Gx			Gy		

Fig. 10.Masques de convolution Laplacien de gaussien [23].

4.2.6. Avantages Laplacien de gaussien:

- Il est très utile pour détecter les bords abrupts.
- Trouver les bons emplacements des contours.

4.2.7. Inconvénients Laplacien de gaussien:

- Dysfonctionnement dans les coins.
- Les courbes et où la fonction d'intensité du niveau de gris varie[23].

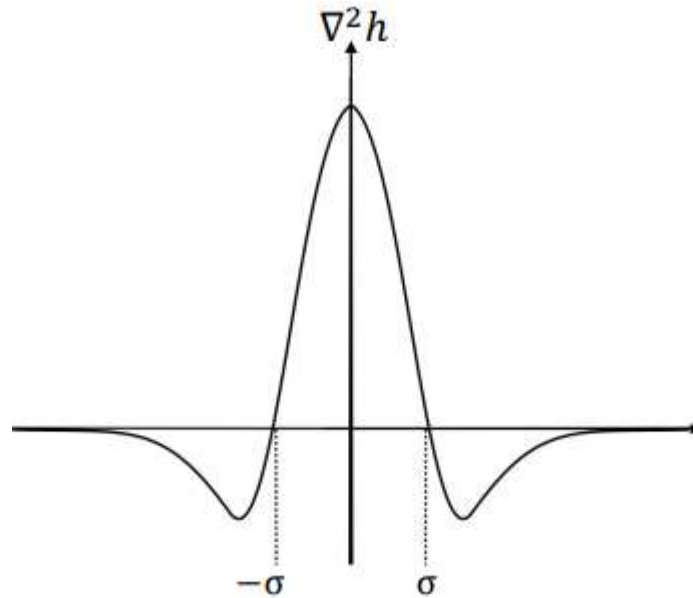


Fig.11. Laplacien d'une gaussienne "chapeau mexicain».

5. Détection de contour par seuillage à hystérésis

L'utilisation d'un seul seuil donne généralement, des contours incomplets ou des faux contours .Il est souvent très difficile, voire impossible, de trouver un seuil adapté à toute l'image. Pour pallier à ces problèmes, on introduit deux seuils : un seuil haut et un seuil bas, tels que : Une comparaison entre la norme du gradient et les seuils choisis est alors effectuée :

- Si $\text{norme} > \text{seuil haut}$, le pixel correspondant appartient au contour.
- Si $\text{norme} < \text{seuil bas}$, le pixel correspondant n'appartient pas au contour.
- Si $\text{seuil bas} < \text{norme} < \text{seuil haut}$ alors le pixel correspondant appartient au contour à condition qu'il soit connexe, au moins, à un pixel pour lequel la norme est supérieur au seuil haut [24] .

5.1. Choix de seuils

Les valeurs des seuils haut et bas choisies sont données sur la figure :

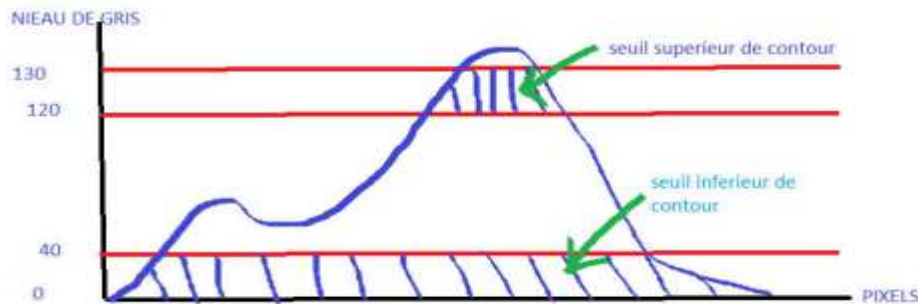


Fig.12. Représentation du seuillage par hystérésis utilisé [24] .

6. Algorithmes de détection de contours

Plusieurs méthodes permettent de déterminer le Gradient ou le Laplacien d'une image. Il en est de même des techniques de seuillage. Ces deux étapes sont indépendantes, il existe donc un grand nombre de combinaisons calcul de Gradient-opération de seuillage conduisant à la mise en évidence des contours. Le rôle de l'ingénieur est souvent de choisir les opérateurs les mieux à même de satisfaire les exigences associées à l'image à traiter .La difficulté réside dans la présence de bruit dans les images

il existe plusieurs algorithmes de détection de contours qui peuvent être classés dans les catégories suivantes :

Les algorithmes basés sur le gradient :

6.1. Filtre de Prewitt

Le filtre de Prewitt est utilisé en traitement d'image pour la détection de contours, il tient son nom de Judith M. S. Prewitt. (1970). Ce filtre utilise deux matrices de dimensions 3, conçu pour la première afin de définir un contour horizontale grâce à un changement important de luminosité et la seconde afin de définir un contour verticale .

La convolution de l'image par deux masques correspond aux calculs des dérivées de l'image pré-filtrée [25] :

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

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

Fig.13. Masques de convolution d'opérateur de Prewitt (c=1)[26].

6.1.1. Exemple d'application de prewitt sous Matlab

Le code de ce filtre :

```
a =imread ('images.jpg');
b =rgb2gray (a);
imshow (b);
c = edge (b,'prewitt');
imshow(c);
```

On visualise le résultat sur la figure suivante :

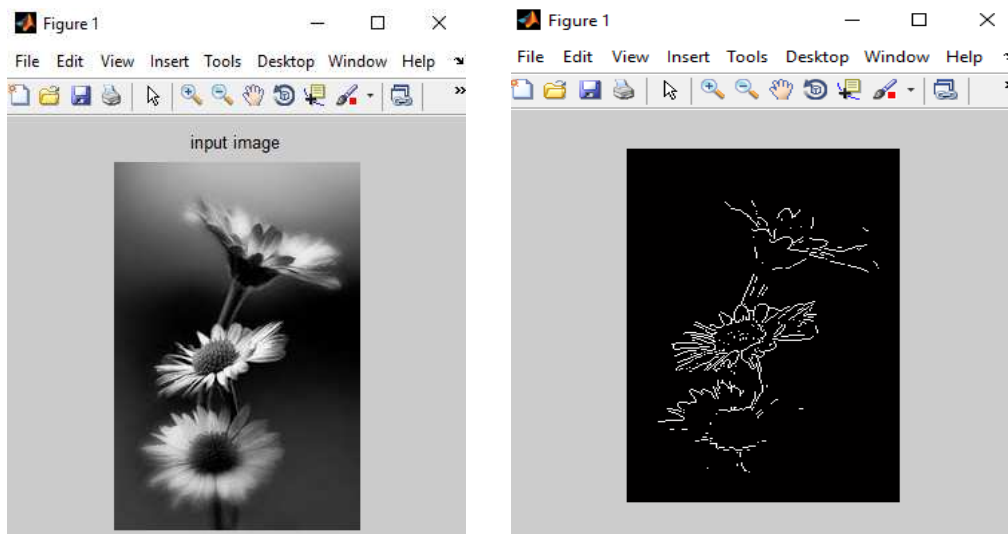


Fig.14. Application du filtre de Prewitt

6.2. Filtre de Sobel

C'est un algorithme largement utilisé pour détecter les contours dans le traitement d'image (opérateur de différenciation). Il est nommé d'après [Irwin Sobel](#) et Gary Feldman(1968). Lorsqu'elle

est utilisée avec la méthode Sobel, cette commande vous permet de supprimer toutes les contours qui ne sont pas fixés comme seuil)[27]

6.2.1. Le principe de filtre Sobel

Le masque est glissé sur l'image principale, traitant d'un carré de pixels à la fois. L'algorithme calcule l'intensité de l'image en chaque point, puis donne la direction pour augmenter l'intensité de l'image en chaque point de la lumière à l'obscurité. Les régions de bord représentent de forts contrastes dans des densités plus sombres ou plus lumineuses.

Une variation existera si le gradient est localement maximum .Ceci indique la direction de la plus forte variation du clair au sombre, ainsi que le taux de changement dans cette direction. On connaît alors les points de changement soudain de luminosité, correspondant probablement à des bords, ainsi que l'orientation de ces bords.

L'opérateur utilise des matrices de convolution. La matrice (généralement de taille 3×3) subit une convolution avec l'image pour calculer des approximations des dérivées dont l'un estime le gradient dans la direction x et l'autre estime le gradient dans la direction y [28].

$$\begin{array}{c}
 \begin{array}{|c|c|c|}
 \hline
 -1 & 0 & 1 \\
 \hline
 -2 & 0 & 2 \\
 \hline
 -1 & 0 & 1 \\
 \hline
 \end{array} \\
 + \\
 \begin{array}{|c|c|c|}
 \hline
 -1 & -2 & -1 \\
 \hline
 0 & 0 & 0 \\
 \hline
 1 & 2 & 1 \\
 \hline
 \end{array} \\
 \begin{array}{cc}
 \text{Direction X} & \text{Direction Y}
 \end{array}
 \end{array}$$

Fig.15. Noyaux de convolution Sobel

6.2.2. Pseudo-codes pour la méthode de détection des bords Sobel

Entrée: un exemple d'image

Sortie: arêtes détectées

Étape 1: Acceptez l'image d'entrée

Étape 2: Appliquer mask Gx, Gy à l'image d'entrée

Étape 3: appliquer l'algorithme de détection des contours Sobel et le gradient

Étape 4: Masque la manipulation de Gx , Gy séparément sur l'image d'entrée

Étape 5: Résultats combinés pour trouver la magnitude absolue du gradient

$$|G| = \sqrt{Gx^2 + Gy^2} \quad (1)$$

Étape 6: la grandeur absolue correspond aux bords de sortie [29].

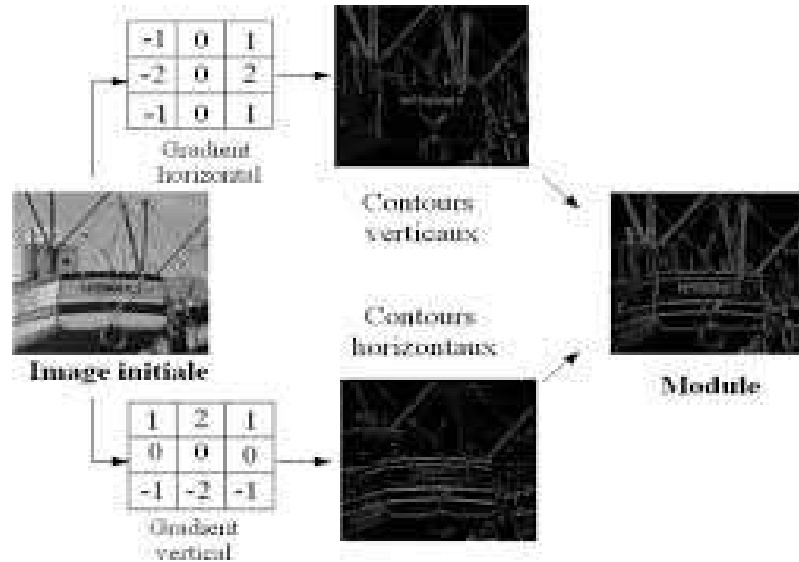


Fig.16. Exemple d'application de l'opérateur de Sobel.

6.2.3. Avantage de filtre Sobel

Le principal avantage de ces masques est leur facilité de mise en œuvre ainsi que la rapidité de leur traitement sur ordinateur. De plus les contours obtenus sont souvent assez larges. L'opérateur de Sobel est très utilisé en vision industrielle nécessitant des contraintes temps-réel.

6.2.4. Inconvénients de filtre Sobel:

- sensibilité au bruit.
- L'opérateur croisé donne une réponse à certains contours[23].

6.2.5. Exemple pour calculer le module de gradient sous Matlab

Le code suivant permet de réaliser le module du gradient de filtre Sobel :

```
rgb = imread('fleur.jpg');
I = rgb2gray(rgb);
hy = fspecial('sobel');
```

```

hx = hy';
Iy = imfilter(double(I), hy, 'replicate');
Ix = imfilter(double(I), hx, 'replicate');
gradmag = sqrt(Ix.^2 + Iy.^2);
figure, imshow(gradmag,[]), title('Module du gradient')

```

On obtient alors la figure suivante:

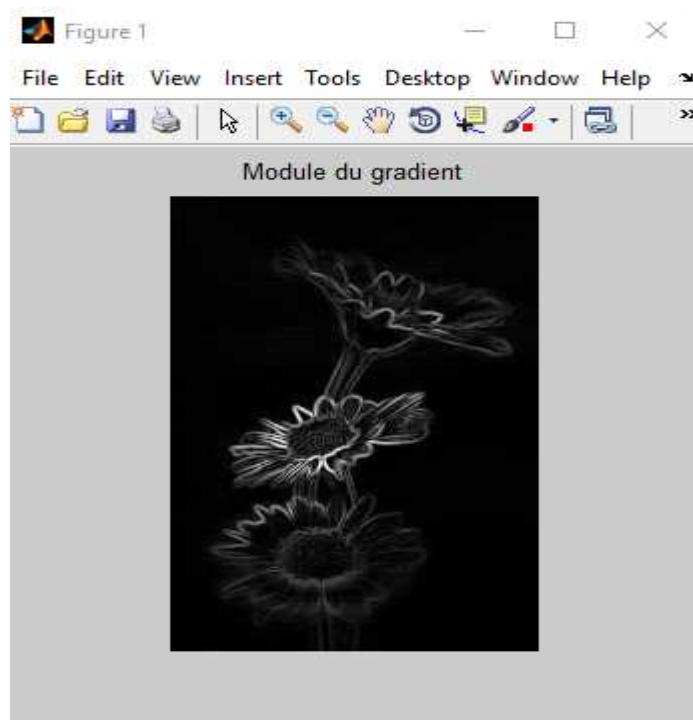


Fig.17. Module du gradient de filtre Sobel .

6.3. Les propriétés de ces filtres

Ces filtres sont très sensibles au bruit vu qu'ils l'accroissent par dérivation. Dans le cas d'un contour de type "rampe", ces filtres donneront un contour épais. Les filtres de Prewitt et Sobel sont moins sensibles au bruit contrairement à ceux de Roberts, le fait d'introduire un moyennage local sur le domaine couvert par le masque provoque une diminution de leur sensibilité. Le filtre de Sobel donne une meilleure évaluation que celui de Prewitt car la série (1 2 1) est approximativement une gaussienne. Ces méthodes ont par ailleurs une grande rapidité et un faible coût.

6.4. Le filtre de Canny

Canny, (Canny, 1986) c'est la Meilleure technique parmi l'algorithme de détection des contours. Est l'un des outils de traitement d'image les plus couramment utilisés, détectant les contours de manière très énergique. Le détecteur de bord Canny est largement considéré comme la méthode de détection de bord standard industrie.

Les deux masques de convolution un de dimension 3×1 et l'autre 1×3 de gradient qui retourne l'intensité des contours suivant les directions X et Y :

$$G_x = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \quad ; \quad G_y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

La valeur du gradient en un point est approximée par la formule :

$$|G| = \sqrt{G_x^2 + G_y^2}$$

Pour la direction l'orientation des contours sa donnée par :

$$\theta = \pm \arctan\left(\frac{G_y}{G_x}\right) \quad (6)$$

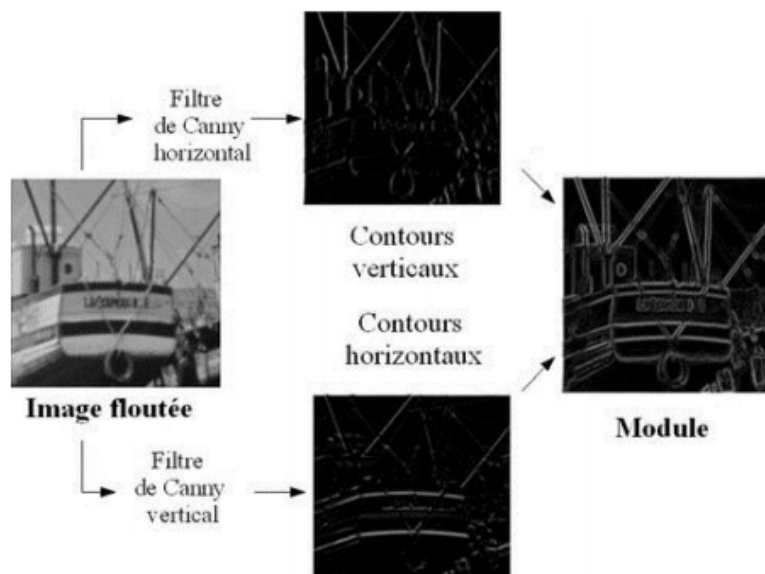


Fig.18. Obtention du module du gradient par filtrage de Canny.

6.4.1. Les étapes du détecteur de contour Canny

- . Lissage: flou de l'image pour supprimer le bruit.
- . Recherche de gradient : les contours doivent être marqués là où les gradients de l'image ont de grandes magnitudes.
- Suppression non maximale: seuls les maxima locaux doivent être marqués comme des contours.
- Double seuillage: les fronts potentiels sont déterminés par seuillage[30] .

6.4.2. Avantages de filtre Canny

- Utilisation de la probabilité pour trouver le taux d'erreur.
- Meilleure détection en particulier dans des conditions de bruit .

6.4.3. . Inconvénients de filtre Canny

- Calculs complexes.
- Faux problème de passage à zéro et prend du temps [23].

6.4.4. Exemple d'application de Canny sous Matlab :

Le code qui réalise ce filtre :

```
a = imread ('images.jpg');  
b = rgb2gray (a);  
imshow (b);  
c = edge (b,'canny');  
imshow (c);
```

La figure 4 montre le résultat suivant :

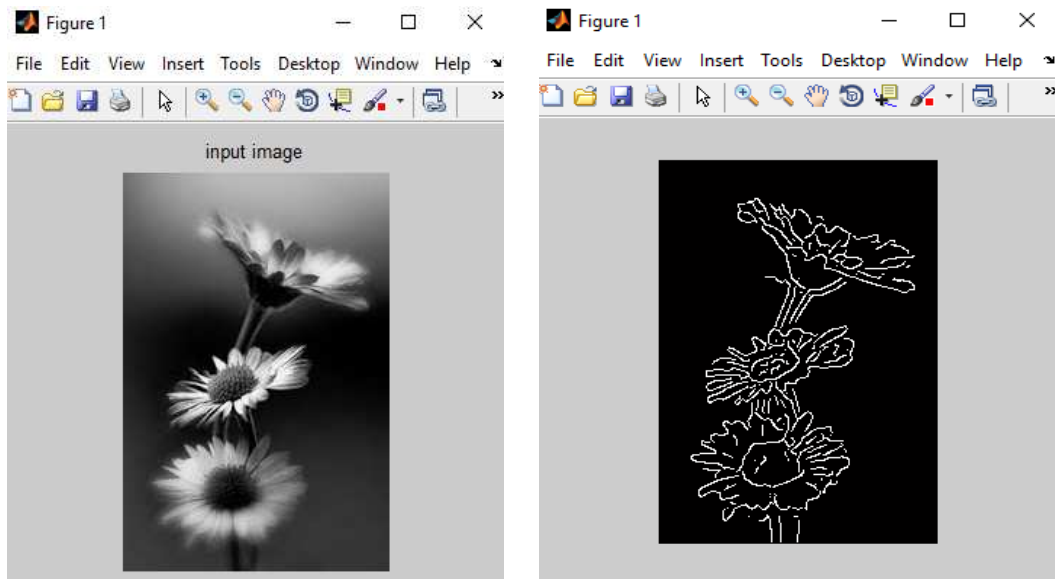


Fig.19. Application de filtre Canny

6.5.Filtre de Roberts

L'opérateur de Roberts (Roberts, 1965) apportent des améliorations à l'estimation discrète du gradient (dérivée) et à la prise en compte de l'orientation du contour. Le filtre de Roberts permet de calculer le gradient d'une image en suivant les diagonales grâce aux deux matrices, or le bruit peut aussi être une brusque variation locale des niveaux de gris. Ces filtres sont donc très sensibles au bruit car ils accentuent par dérivation, le bruit présent dans l'image. De plus. Ces filtres donneront un contour épais si celui-ci est un contour de type "rampe" , si le contour est franc ("marche"), les filtres placeront le contour un pixel a gauche ou au-dessus mais son épaisseur sera respectée [31] .

TYPES DE MASQUE	GRADIENTS PARTIELS	AMPLITUDE	DIRECTION								
<p><i>Masques de Roberts</i></p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>-1</td><td>0</td><td>0</td><td>-1</td></tr> <tr> <td>0</td><td>1</td><td>1</td><td>0</td></tr> </table>	-1	0	0	-1	0	1	1	0	<p>G_1, G_2</p> <p>Substitution du pixel supérieur gauche</p>	$A = \sqrt{G_1^2 + G_2^2}$	$\theta = \frac{\pi}{4} + \arctan\left(\frac{G_2}{G_1}\right)$
-1	0	0	-1								
0	1	1	0								

Tableaux.1. les caractéristiques de filtre Roberts [32].

6.5.1. Avantages de filtre Robert:

- Rapide à calculer.
- Facile .

6.5.2. Inconvénients de filtre Roberts:

- Peu de pixels sont utilisés pour approximer le dégradé.
- Souvent inexact [23].

6.5.3. Exemple d'application de Roberts sous Matlab :

Le code utilisé est le suivant :

```
a =imread('images.jpg');  
b =rgb2gray (a);  
imshow (b);  
c = edge (b,'roberts');  
imshow (c);
```

La figure 20 montre les résultats de détection de contours avec ce filtre :

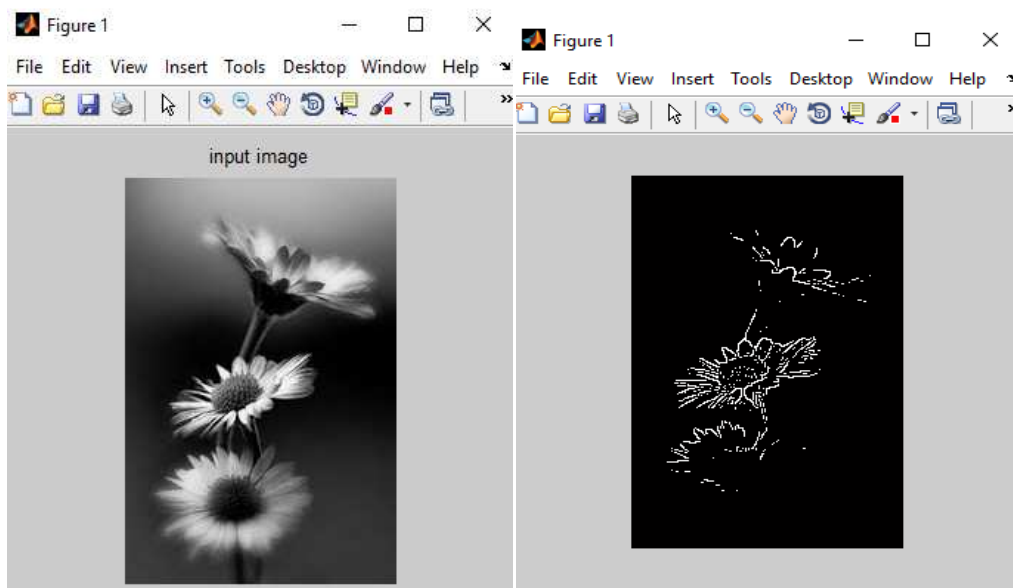


Fig.20. Application du filtre de Roberts

7. Mesure de Précision

Une table de confusion est Un tableau avec deux lignes et deux colonnes affichant le nombre de faux positifs et de faux négatifs et de vrais et réels positifs négatif

Le tableau ci-dessous montre l'échelle de précision pour les algorithmes les plus couramment utilisé pour la détection des contours :

Algorithms	Confusion matrix	
	Accuracy (in %)	Matrix
Sobel	75%	4 0 2 2
Canny	87.5%	6 1 0 1

Tableaux.2. l'échelle de précision pour les algorithmes Canny et Sobel [30].

8. Choix des masques

Nous avons présenté les masques les plus utilisés. Néanmoins, ils ne sont qu'un cas particulier. Afin de savoir la taille à choisir il faut suivre les règles suivantes :

- Plus le masque est grand, moins le gradient est sensible au bruit.
- Plus le masque, est grand, plus le temps de calcul est élevé.
- Plus le masque est grand, moins bonnes est la localisation des contour [6].

9. Les seuils de chaque méthode de détection de contour

Lors de l'utilisation du seuil avec la détection de bord méthodes, on note que:

- Les méthodes de détection de bord Canny ont donné de bons résultats lorsque le seuil est égal à 0,1.
- La méthode de détection des bords Robert a donné de bons résultats lorsque le seuil est égal à 0,03.
- Méthodes de détection des bords Sobel et de détection des bords Prewitt a donné des résultats acceptables lorsque la valeur seuil était égale à 0,1 ou valeur seuil égale à 0,03.
- La méthode de détection des bords du système flou a donné de bons résultats lorsque le seuil est égal à 1 ou le seuil égal à 3 ou seuil égal à 5 [23].

10. Choix d'un détecteur de contours

Les résultats d'expérimentation ont montré que certains détecteurs sont plus efficaces quel que soit le cas ceci va en partie simplifier le choix du détecteur par la suite :

l'algorithme de détection de bords Canny fonctionne mieux que les algorithmes de détection de bords Sobel pour détecter le visage , bien que les filtres de Sobel et Prewitt soient basés sur le même principe (lissage et dérivation), nous avons remarqué que les résultats de détection de contours sont souvent meilleurs pour le filtre de Sobel que Roberts, Prewitt ou Kirsch dans le cas des images nettes avec des niveaux de détail dans le cas des images seulement nettes Prewitt est toujours meilleur que Robert, Sobel ou dans le cas des images bruitées ou floues [17] .

11. Filtres numériques à réponse impulsionnelle finie RIF

11.4. Définition

Le filtre RIF est l'une des moyennes mobiles ou tous les filtres à zéro qui acceptent le présent Entrée et passé des échantillons d'entrée et produit les résultats de sortie. Dans la pratique, la conception du filtre RIF permet de contrôler facilement les multiples applications du DSP. Le filtre est complètement capable de fournir une phase linéaire en raison de son comportement de linéaire et arbitraire peut approximer la réponse en fréquence. Il est de nature non récursive et la conception utilise les éléments de retard et les coefficients de pondération.

L'équation différentielle est donnée ci-dessous qui peut être utilisée pour la structure directe et transposée du filtre RIF [35].

$$y[n] = b_0 x [n] + b_1 x [n-1] + \dots + b_{N-1} x [n-N+1] \quad (7)$$

11.5. Structure d'un filtre RIF

La structure directe d'un filtre FIR est donnée par la figure suivante. Cette forme est habituellement la plus utilisée dans l'implémentation en raison ses meilleurs paramètres.

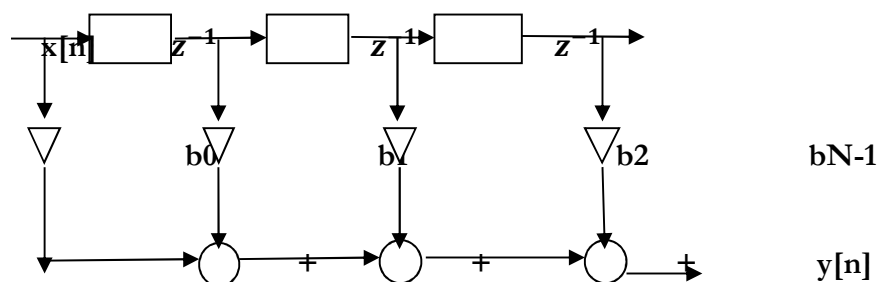


Fig.21. Structure d'un filtre RIF [32].

11.6. Conception de filtre

La conception du filtre RIF peut être comprise à l'aide de l'exemple suivant de deux séquences Considérations de conception. La sortie de la sortie du filtre RIF peut être déterminée

en utilisant l'équation suivante :

$$y[n] = \sum_{k=0}^{N-1} h[k] \cdot x[n - k] \quad (8)$$

Où :

$x [k]$ = Présente les échantillons d'entrée du filtre RIF.

$h [k]$ = coefficients de pondération du filtre RIF sous réponse en fréquence.

$y [n]$ = échantillons de sortie du filtre RIF[32].

11.7. Caractéristiques des filtres numériques RIF

Les termes suivants sont utilisés pour caractériser les performances et le comportement des filtres RIF :

- **Coefficients du filtre:** c'est un ensemble de constantes, appelé aussi poids, utilisé pour multiplier ou pondérer les valeurs retardés de l'échantillon. Pour un filtre RIF, les coefficients de filtre sont, par définition, la réponse impulsionnelle du filtre.
- **Réponse impulsionnelle:** Temps de réponse de la séquence du filtre lorsque l'entrée est une impulsion. Une impulsion est un échantillon unique de valeur égale à l'unité précédée et suivie par des valeurs nulles (zéro échantillons évalués). Pour un filtre RIF sa réponse impulsionnelle est l'ensemble des coefficients filtre.
- **Tap :** Le nombre d'étage N du filtre RIF, généralement N, nous informe sur le type du RIF et surtout, sur quantité de mémoire nécessaire, et le nombre d'étapes de calculs.
- **Mac :** Dans un contexte de filtrage utilisant les RIFs, un "MAC" est l'opération de multiplication d'un coefficient par l'échantillon de données retardé correspondant et en accumulant le résultat. Il ya généralement un MAC par étage [33].

11.8. Avantages des filtres RIF

- Simples à concevoir.
- Muni d'entrée-sortie bornée stable.
- Filtre FIR peut être assuré d'avoir à phase linéaire. Ceci est une propriété souhaitable pour de nombreuses applications telles que la musique et le traitement vidéo .
- Les Filtres FIR ont également une faible sensibilité pour filtrer les erreurs.
- Coefficient de quantification. C'est une propriété importante pour avoir la mise en œuvre d'un filtre sur un processeur DSP ou sur un circuit intégré[33].

11.9. Applications du monde réel de filtres RIF

Quelques applications populaires pour les filtres RIF sont énumérées ci-dessous:

- annulation d'écho.
- télécommunications.
- Les communications sans fil.
- HDTV.
- DTV.
- le traitement de la vidéo .
- La synthèse vocale.
- modems haut débit.
- ADSL[33].

12. Conclusion

À travers ce chapitre , nous avons défini les contours des images, les méthodes pour détecter ces contours suivie d'une application sur des images en niveaux de gris sou Matlab .nous avons aussi présenté le principe de chaque méthode et chaque algorithmes(Sobel) utiliser par l'approche de gradient.

CHAPITRE 3 : IMPLEMENTATION DU FILTRE DE SOBEL SUR FPGA

1. Introduction

L'objectif de ce chapitre est présenter les étapes d'implémentation d'un détecteur de contour à base du filtre de Sobel. Le choix de ce dernier est fait en raison de sa simplicité, les étapes d'implémentation resteront valables pour implémenter n'importe quel autre filtre. Ce chapitre commencera par présenter la carte de développement à base de FPGA utilisée pour l'implémentation, les étapes de réalisation du projet, les résultats obtenus par simulation avec comparaison de ceux obtenus par Matlab et finalement la réalisation pratique sur le circuit FPGA en temps réel.

1. Pourquoi le FPGA

Les FPGAs (Field Programmable Gates Arrays ou "réseaux logiques programmables) sont des composants entièrement reconfigurables ce qui permet de les reprogrammer à volonté afin d'accélérer notablement certaines phases de calculs. L'avantage de ce genre de circuit est sa grande souplesse qui permet de les réutiliser à volonté dans des algorithmes différents. En un temps très court. Le progrès de ces technologies permet de faire des composants. Toujours plus rapides et à plus haute intégration, ce qui permet de programmer des applications importantes. Cette technologie permet d'implanter un grand nombre d'applications et offre une solution d'implantation matérielle à faible coût pour des compagnies de taille modeste pour qui, le coût de développement d'un circuit intégré spécifique implique un trop lourd investissement [34].

Les FPGA sont utilisés dans de nombreuses applications, on en cite dessous quelques-unes :

- Prototypage de nouveaux circuits.
- Fabrication de composants spéciaux en petite série.
- Adaptation aux besoins rencontrés lors de l'utilisation.
- Systèmes de commande à temps réel.
- DSP (Digital Signal Processor).
- Imagerie médicale.

2.1. La carte de développement ATLYS

Dans cette partie on va présenter la carte d'évaluation ATLYS qui a été utilisée dans notre

projet, cette carte est une plate-forme complète de développement de circuits numériques prête à l'emploi basée sur un FPGA Spartan-6 LX45 Xilinx, elle dispose des ressources pertinentes suivantes :

- Mémoire DDR2 Micron MT47H64M16-25E de 1 Gbit ou 64 millions de cases de 16 bits compatible avec des taux de transfert jusqu'à 800 MHz : cette mémoire est compatible avec les MCB (Memory Controller Block) de Xilinx, elle permet de proposer un système avec une plus grande capacité de stockage que la simple mémoire interne d'un FPGA.
- Horloge de base à 100 MHz : cette horloge permet de générer les différentes horloges du design, pour la mémoire externe ou pour le cœur du système, à partir d'une PLL. L'horloge principale du système est de 156.25 MHz et l'horloge de la mémoire est de 625 MHz. La section 5.3.4, page 110, donne plus de détails à propos des horloges utilisés dans l'outil.
- Port d'extension de 8 entrées/sorties : ce port permet de faire des tests simples, avec un nombre d'entrées/sorties limitées.
- Port d'extension haute vitesse de 40 entrées/sorties : ce port permet de faire des tests plus complexes [35].

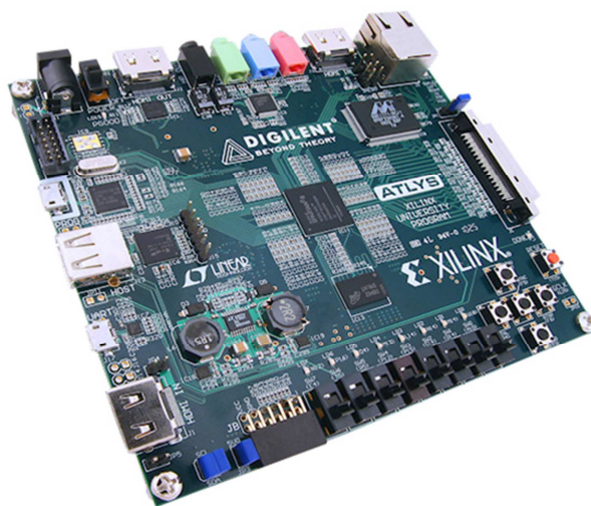


Fig.1. La carte de développement ATLYS.

D'autres extensions contenues dans la carte ATLYS peuvent être résumés sur la figure suivante :

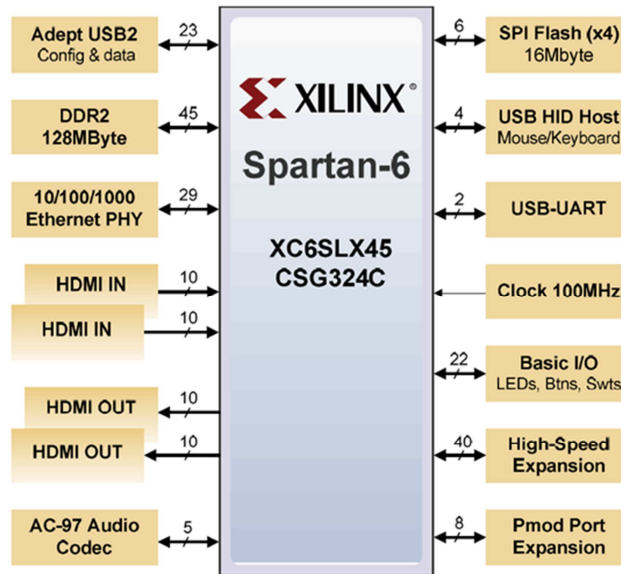


Fig.2. Les périphériques d'entrées/sorties de la carte ATLYS.

2.2. L'outil Xilinx System Generator (XSG)

Généralement les circuits FPGAs peuvent être configurés à l'aide d'un langage de description matériel tel que le Verilog ou le VHDL, mais avec l'évolution dramatique des outils de conception, la réalisation des circuits assez compliqués ne nécessite pas une grande expérience sur les langage de description matériel comme le cas de l'outil Xilinx System Generator (XSG).

XSG est un outil de modélisation au niveau du système qui facilite la conception matérielle FPGA. Il étend Simulink de nombreuses manières pour fournir un environnement de modélisation bien adapté à la conception matérielle. L'outil fournit des abstractions de haut niveau qui sont automatiquement compilées dans un FPGA en appuyant simplement sur un bouton. L'outil permet également d'accéder aux ressources FPGA sous-jacentes via des abstractions de bas niveau, permettant la construction de conceptions FPGA très efficaces.

2.3. Xilinx Integrated Synthesis Environment (ISE)

Xilinx ISE (Integrated Synthesis Environment) est un outil logiciel produit par Xilinx pour la synthèse et l'analyse des conceptions HDL, permettant au développeur de synthétiser («compiler») leurs conceptions, d'effectuer une analyse temporelle, d'examiner des diagrammes RTL (Register Transfer Level), de simuler réaction d'une conception à différents stimuli, et configurer le circuit cible avec le programmeur. On peut même importer le code généré par

des blocs de XSG est les compiler via ISE.

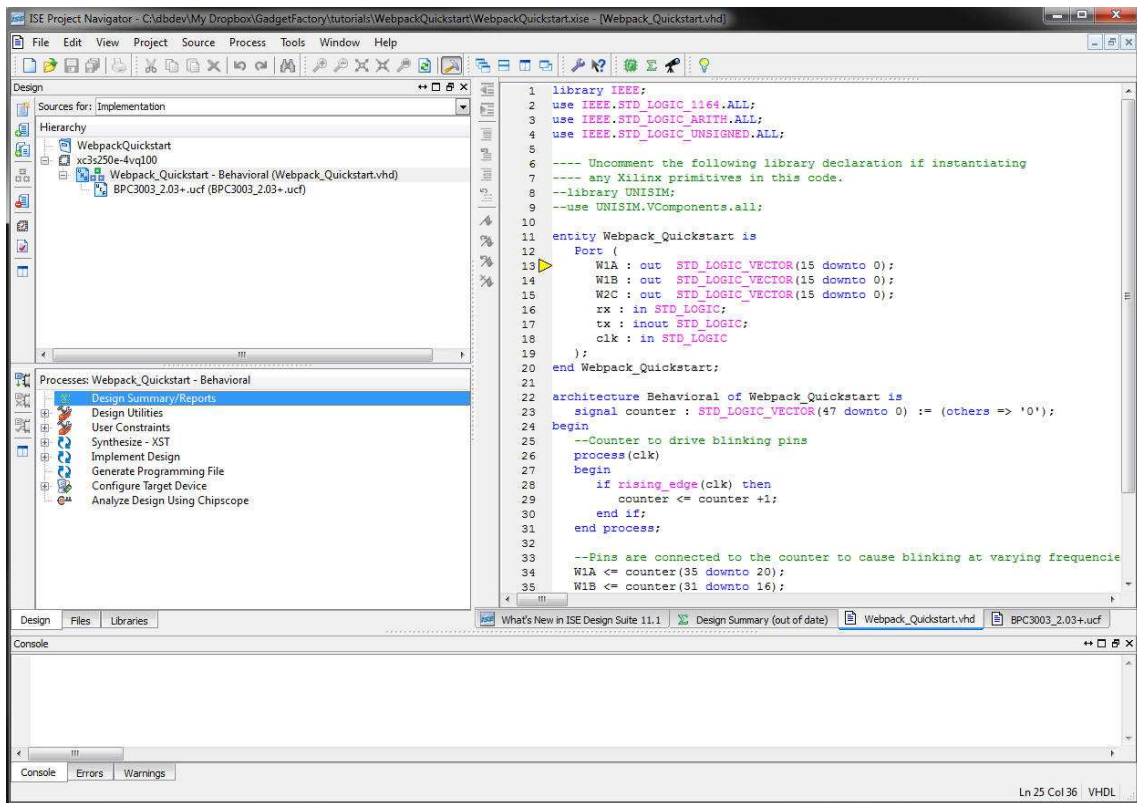


Fig.3. Interface principale de ISE.

2. Le filtre de Sobel avec Matlab

La simulation d'un détecteur de contour d'une image sous Matlab est assez simple grâce à la fonction *edge* dédiée pour ça, il suffit de configurer cette fonction en choisissant le filtre désiré, le seuil pour éliminer tous les contours inférieurs, la direction des contours (horizontal, vertical ou bien les deux) comme le cas de filtre de Sobel. Le code suivant importe une image, la charge à l'espace de travail de Matlab, ensuite détecte les contours de cette image pour différents seuils choisis :

```

I = imread('CITY.jpg'); % importation de l'image
Ig = rgb2gray(I); % conversion de l'image en niveau gris
C = edge(Ig, 'sobel', 0.1); % calcul des contours dans l'image
imshow(C); % affichage de résultat

```

La figure suivante présente les résultats obtenus pour différentes valeurs de seuil :

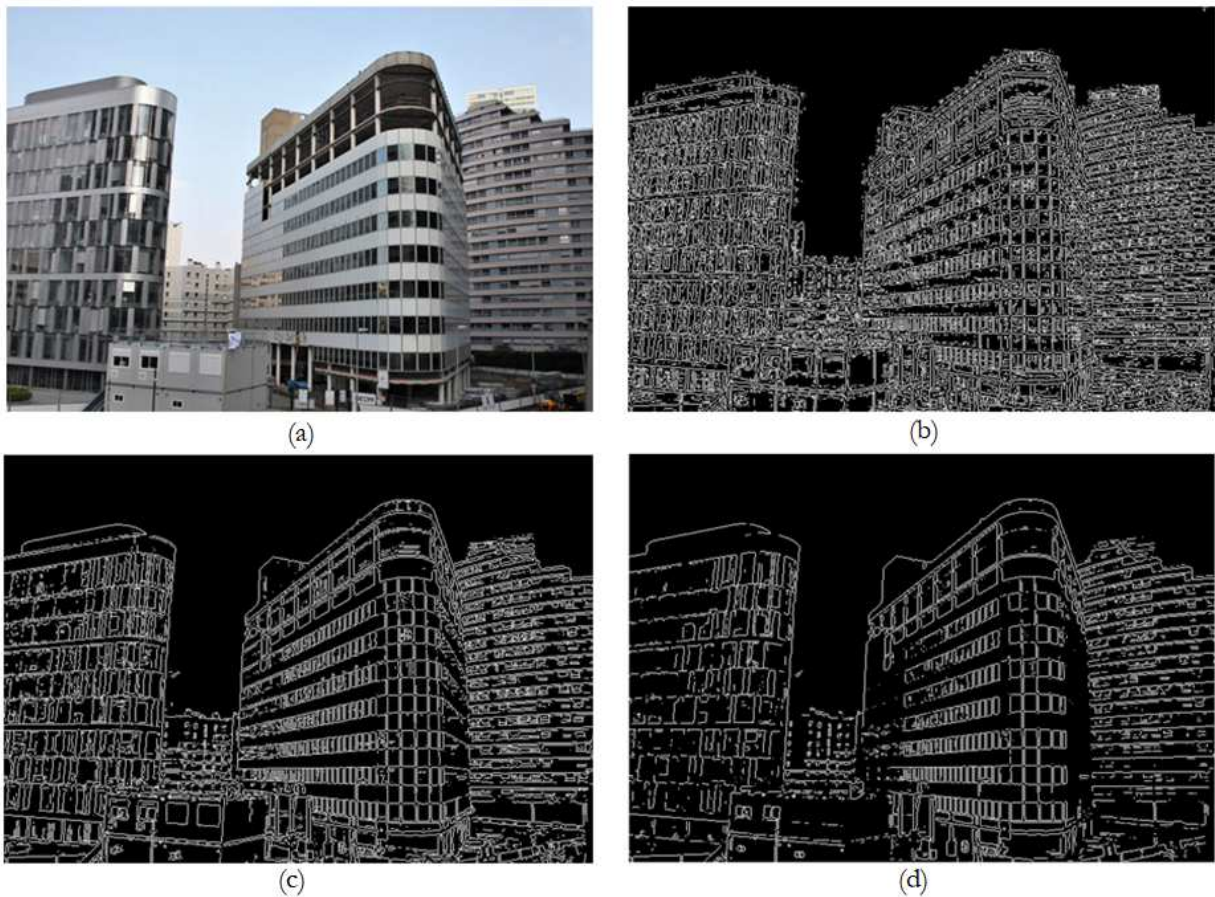


Fig.4. Résultats de simulation de l'application de filtre de *Sobel* sur une image en Matlab, (a) l'image originale, (b) résultat obtenu par un seuil = 0.01, (c) résultat obtenu par un seuil = 0.05, (d) résultat obtenu par un seuil = 0.1.

Pour les résultats obtenus ci-dessus, le filtre de *Sobel* est par défaut configuré pour les deux directions (horizontale et verticale), si on veut une seule direction, on doit configurer le filtre comme suite :

```
C = edge(Ig, 'sobel', 0.1, 'horizontal'); % calcul des contours dans l'image pour la direction horizontale.
```

La figure suivante présente les résultats si on spécifie la direction de filtre :



Fig.4. Résultats de simulation de l'application de filtre de *Sobel* sur une image en Matlab, (a) l'image

3. Conception de filtre de Sobel sous XSG

Comme nous avons mentionné auparavant, XSG fourni une bibliothèque riches de blocs de calcul, de mémoires, blocs DSP, etc. On va exploiter cette bibliothèque pour la conception de filtre de *Sobel*. En effet, la vue de coté de hardware est un peu différente de celle de logiciel (programmation en Matlab). Le Matlab peut traiter les images comme des matrice il n y a pas de problème, mais pour un FPGA et une application en temps réel ; les données d'une images sont traitées comme un seul vecteur (streaming). Rappelons le principe de filtre de *Sobel* :

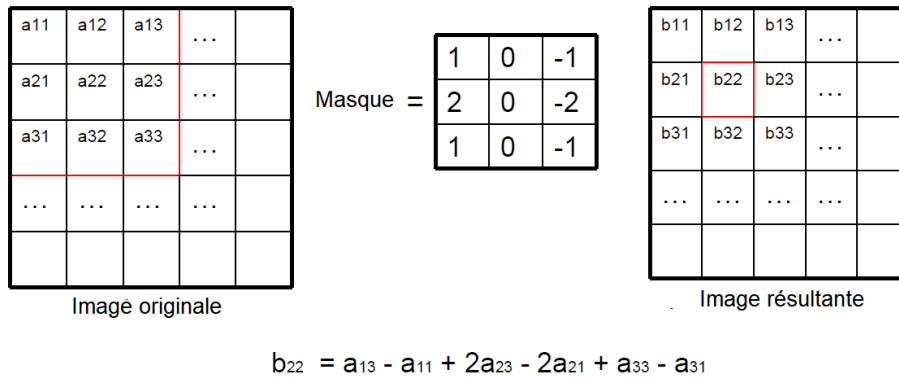


Fig.5. Principe de filtre de *Sobel*.

Dans un FPGA, une séquence vidéo peut être traitée pixel par pixel, jusqu'à la fin de nombre de pixel par chaque image. D'après la figure 5, chaque ligne est une filtre FIR de 3 coefficients, alors on ait besoin de trois filtres FIR (le masque) qui travail en parallèle pour un seul pixel de l'image résultante. Pour compléter une ligne de pixels dans l'image résultante, il faut déplacer le masque à la droite à chaque front montant de l'horloge de balayage de la séquence vidéo.

Pour que les 3 filtres travail en parallèle ; on ait besoin de retarder le signal vidéo pour chaque filtre par L pixels, où L représente la longueur de la ligne, à titre d'exemple ; si on travaille avec une définition HD-1080, la longueur de la ligne dans ce cas est 1920 pixels. Pour retarder le signal comme montre la figure 6, XSG fourni dans sa sous-bibliothèque *Memory*; le bloc *Single Port Ram*. Ce bloc nous permet de retarder le signal vidéo. Le nombre de cases mémoires de chaque mémoire utilisée est L.

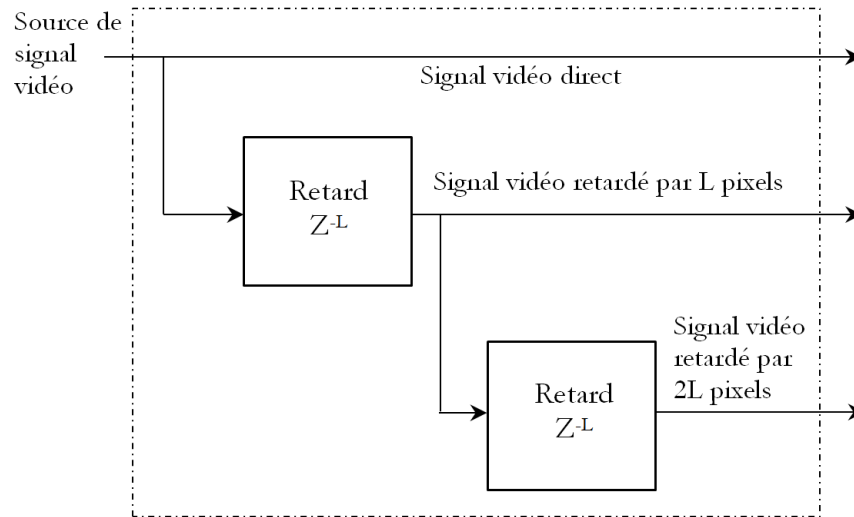


Fig.6. Retardement de signal vidéo.

L'interface graphique du mémoire utilisée avec sa fenêtre de configuration sont présentés dans la figure 7. Comme montre cette figure ; le rôle de compteur est d'incrémenter l'adressage des cases mémoire. La mémoire est configurée de tel sort que la lecture est effectuée avant l'écriture, autrement-dit ; à chaque adresse mémoire, le contenu de la case mémoire est lu puis écrasé par la nouvelle donnée.

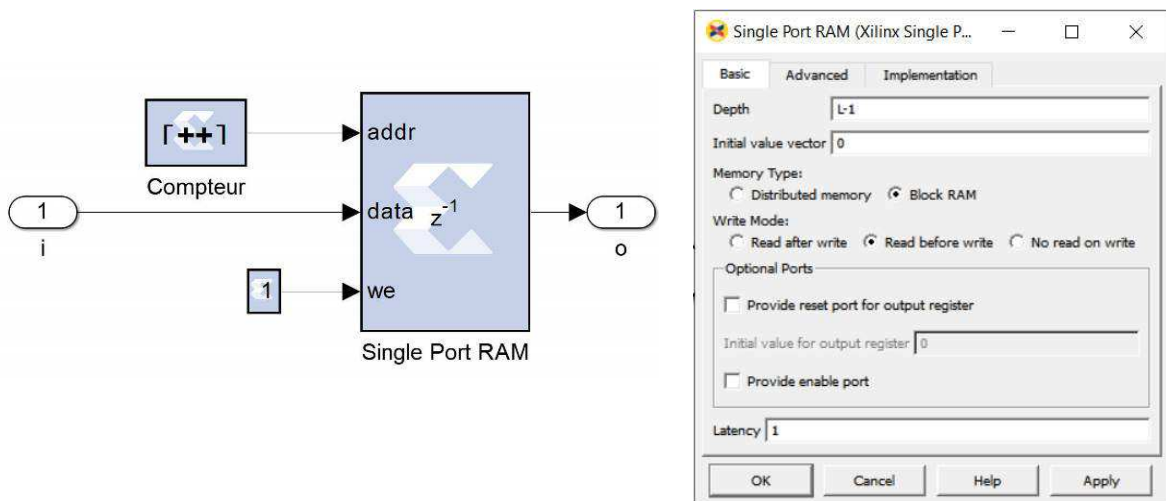


Fig.7. L'interface graphique du mémoire utilisée

La figure suivante montre le montage des mémoires en série pour notre tâche où *Dir* est la sortie direct, *RL* le signal retardé par *L* pixels, et *R2L* le signal retardé par *2L* pixels.

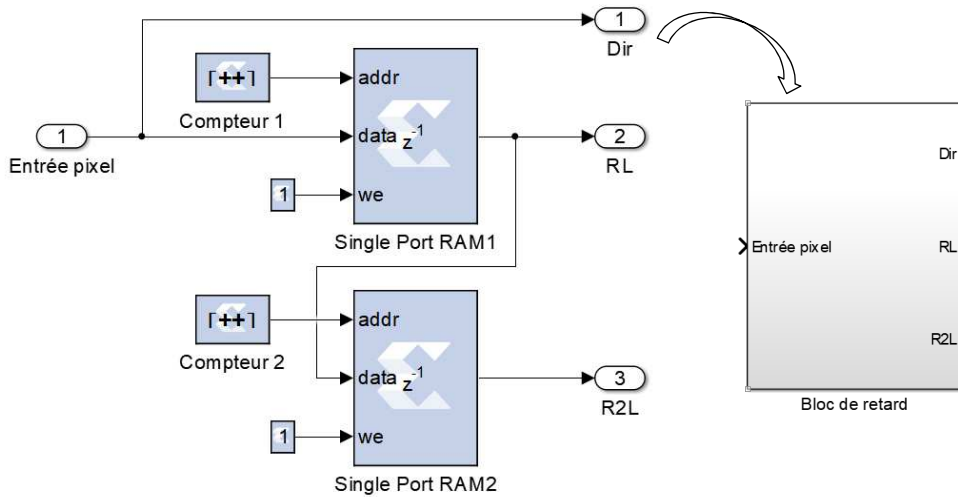


Fig.8. Montage des mémoires en série et l'assemblage de tous en un seul bloc (gauche).

L'objectif de l'étape suivante est de réaliser les filtres FIR, comme mentionné auparavant ; chaque ligne nécessite un filtre FIR (Fig.8) à 3 coefficients. La figure suivante présente le premier filtre FIR :

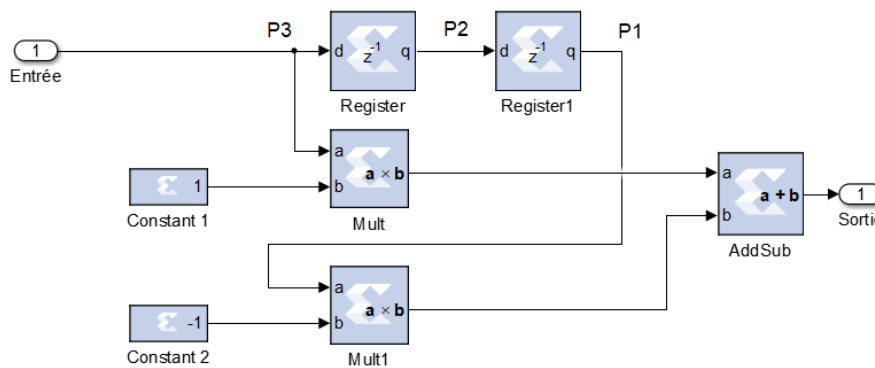


Fig.9. Structure du premier filtre FIR avec XSG.

Selon les paramètres de filtre de Sobel ; nous avons l'entrée P3 (3ème pixel dans le signal) est multipliée par le constant 1, le deuxième pixel (P2) normalement est multiplié par 0, mais on le néglige tant qu'il ne change rien dans le résultat. Le premier pixel P1 est multiplié par -1. Pour le 2ème filtre, il est similaire au premier filtre de la figure 8 sauf les constants sont changés le 1 par 2 et le -1 par -2. Pour le 3ème filtre il est identique au premier filtre.

Les résultats des 3 filtres sont collectés par un additionneur, ensuite la valeur obtenue est passée à travers le bloc *Absolute* (Fig.10) pour obtenir la valeur absolue du résultat, s'il y aura un dépassement (valeur obtenue > 255 (8 bits)) ; le bloc *Cast* va limiter la valeur à 255.

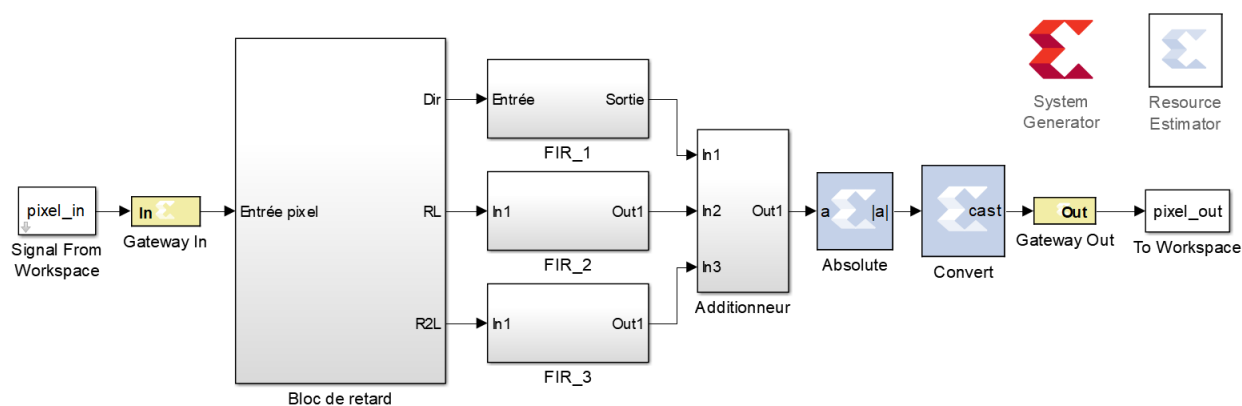


Fig.10. Le système complet pour détecter les contours d'une image avec XSG.

3.1. Evaluation du système par simulation

Dans cette partie on va évaluer le bon fonctionnement de notre système réalisé à base de XSG pour détecter les contours d'une image par simulation, cette étape est importante avant de passer à l'étape d'implémentation et vérification en temps réel. Une image en niveau gris et envoyée au model Simulink du système créé à partir de l'espace de travail de Matlab on utilisant le code suivant, ensuite le résultat obtenu est affiché sur l'écran comme montre la figure 11.

```
I=imread('piece.png'); % importer l'image RGB
im = rgb2gray(I); % convertir l'image vers le niveau gris
sz =size(im); % la taille de l'image
M = sz(:,1); % Hauteur de l'image
N = sz(:,2); % Larguer de l'image
L = N*M; % L = N*H : la longueur de vecteur (de simulation)
pixel_in = im;
pixel_in =pixel_in(:); % pixel d'entrée comme vecteur de longueur L
sim('test_stud') % Lancement du model Simulnik
c=reshape(pixel_out, M, N); % récupération de l'image résultante
imshow(uint8(c)) % Affichage de l'image.
```

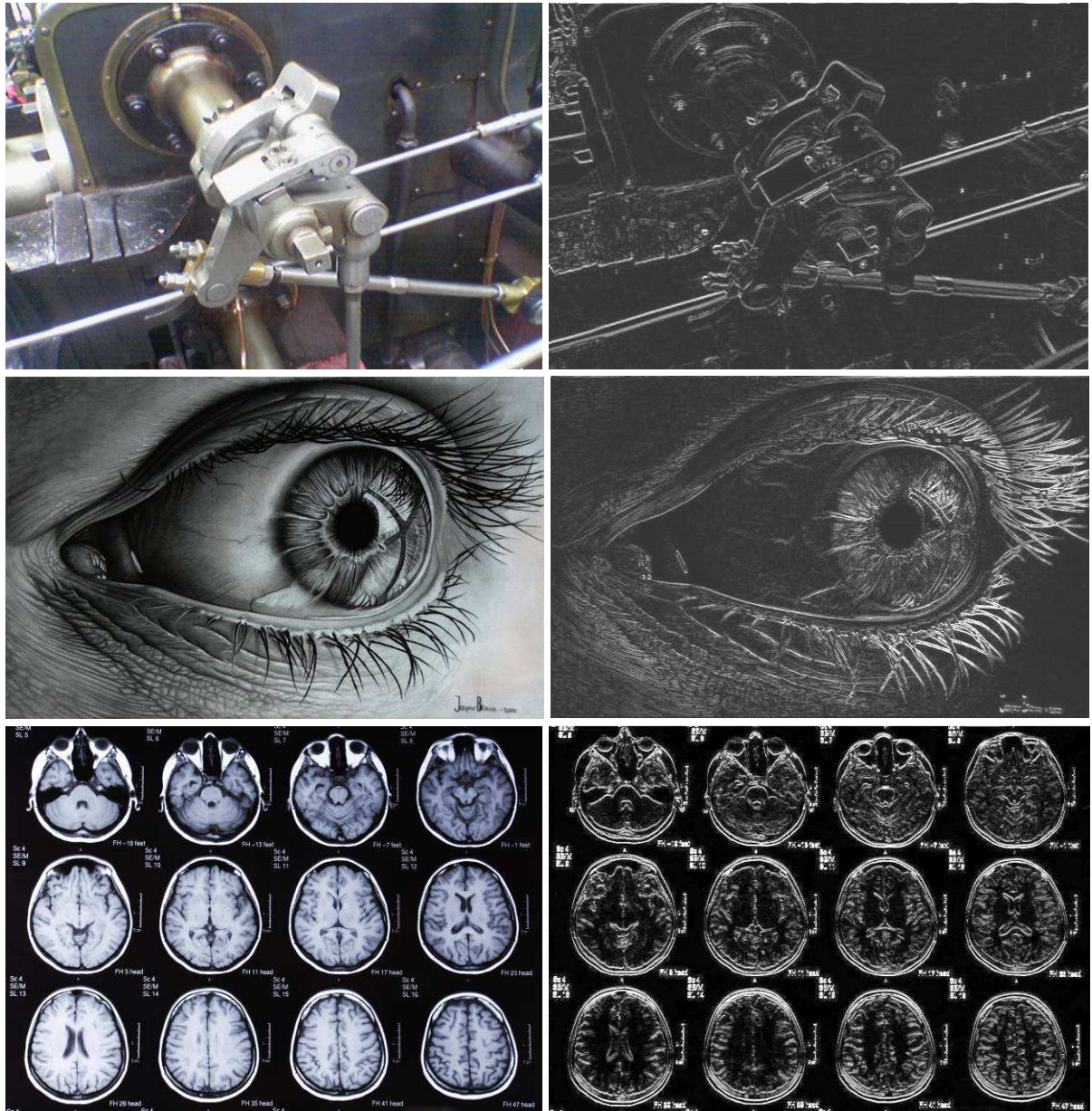


Fig.11. Les résultats de simulation obtenus.

4. Implémentation sur FPGA

Après la confirmation de la bonne fonctionnement du notre système réalisé à l'aide de XSG, nous passons vers l'implémentation sur la carte FPGA. Cette section présente les différents étapes pour achever ce point.

Il faut mettre en considération que la source de l'image ainsi que l'affichage de l'image obtenu en temps réel à partir de la carte FPGA est assurés par les interfaces HDMI. Le pilotage de ces interfaces par le VHDL est une tâche compliquée et il est hors de considération

ce travail. Nous envoyons les fichiers VHDL générés par XSG vers un projet déjà créé sous ISE et qui contient les fichiers nécessaire pour assurer la communication avec les ports HDMI. La figure 12 résume tout ça.

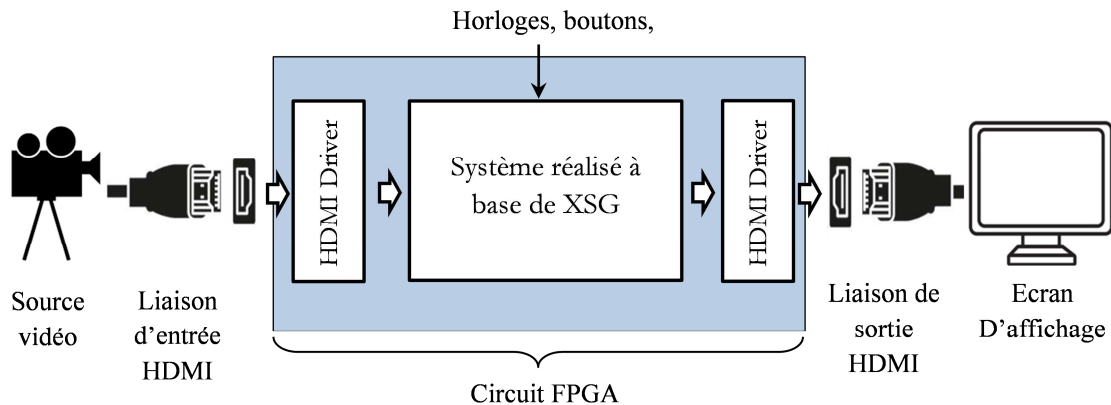



Fig.12. Schéma bloc du système de détection de contours complet.

La première des choses donc c'est la génération du code VHDL équivalent de notre système à partir de symbole de XSG () comme montre la figure 13. Double clic sur ce symbole ; une fenêtre de configuration s'ouvre, cette fenêtre permet de sélectionner la série de circuit FPGA utilisée, l'endroit pour sauvegarder les fichiers compilés, le type de HDL ; VHDL ou Verilog, et d'autres paramètres. Après la configuration on clique sur *generate*.

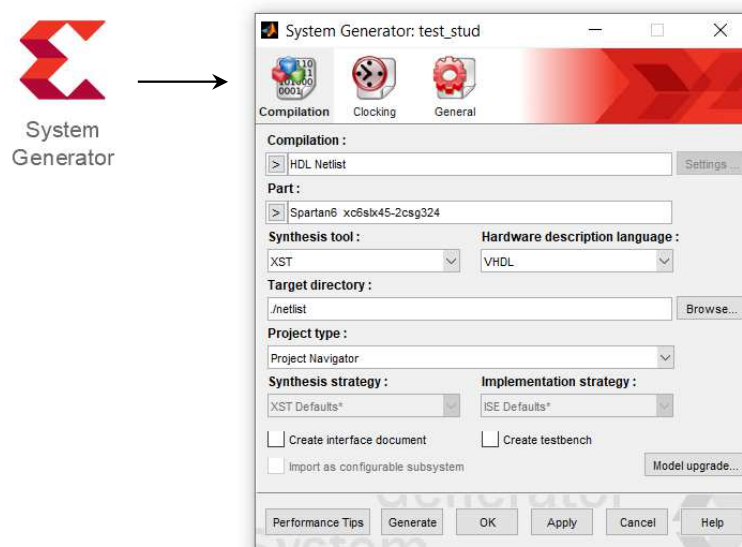


Fig.13. Génération du code compilé à partir de blocs XSG.

La figure suivante présente l'ajout des fichiers compilés par XSG au projet déjà créé pour poursuivre les étapes nécessaires. Après certaines étapes le fichiers de programmation de la carte FPGA est généré.

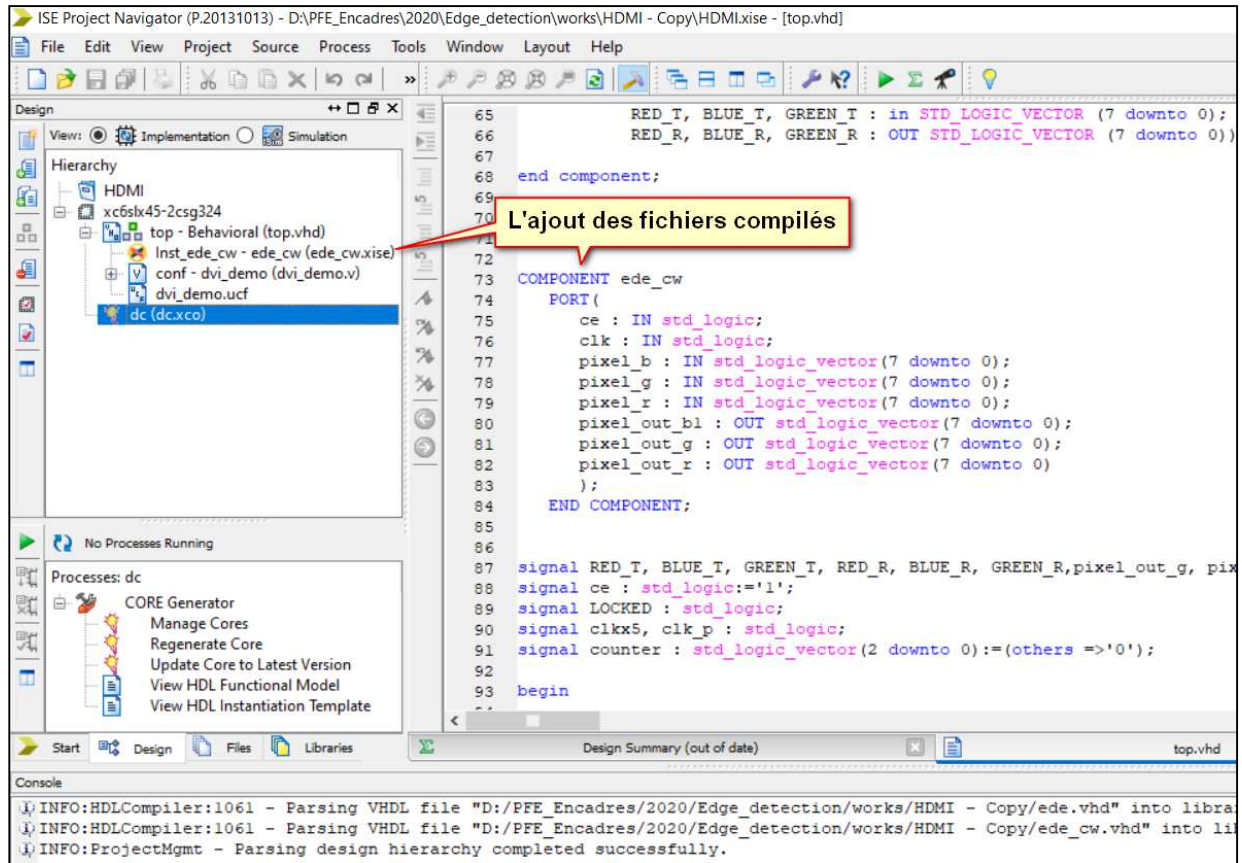


Fig.14. L'ajout des fichiers compilés par XSG au projet créé dans ISE.

Le tableau suivant présente le rapport des ressources consommés par le système implémenté y compris les autres composants tels que les drivers des portes HDMI :

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	2217	54576	4%
Number of Slice LUTs	2488	27288	9%
Number of fully used LUT-FF pairs	1262	3443	36%
Number of bonded IOBs	46	218	21%
Number of Block RAM/FIFO	8	116	6%
Number of DSP48A1s	15	58	25%

Tableau.1. Ressources utilisés par le système implémenté.

La fréquence achevable est 132.15 MHz selon le rapport suivant :

```

Timing summary:
-----

Timing errors: 1 Score: 52 (Setup/Max: 0, Hold: 0, Component Switching Limit: 52)

Constraints cover 12163 paths, 0 nets, and 3255 connections

Design statistics:
  Minimum period: 7.567ns{1} (Maximum frequency: 132.153MHz)
  Maximum path delay from/to any node: 5.197ns
    
```

Après la configuration de la carte FPGA, on peut facilement évaluer le système implémenté en temps réel comme montre la figure suivante :

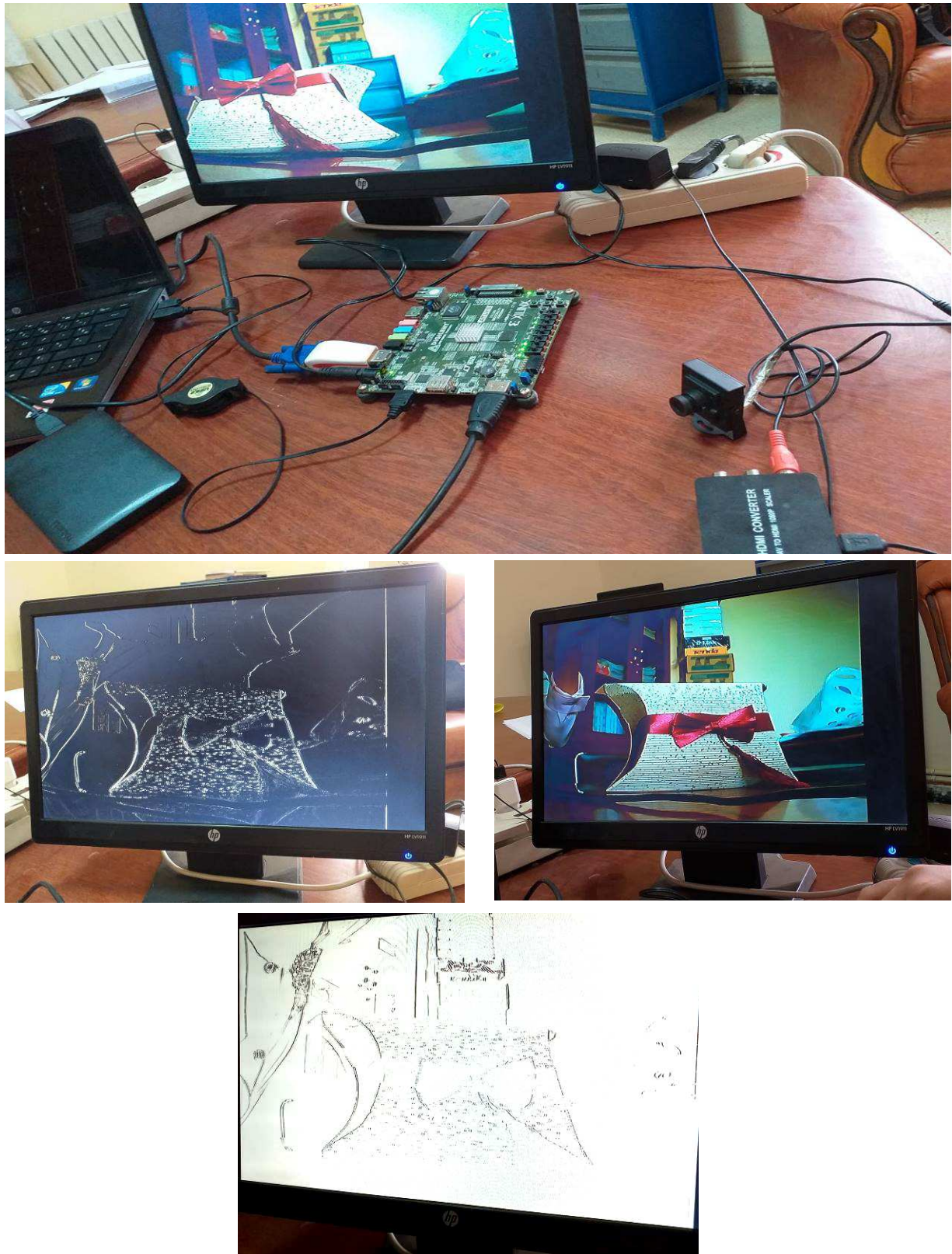


Fig.14. Evaluation du système réalisé en temps réel.

5. Conclusion

Le long de ce chapitre, nous avons vu les étapes de conception et d'implémentation d'un filtre de Sobel en un FPGA pour détecter les contours d'une image. Nous avons commencé par la simulation d'un filtre de Sobel sous Matlab, ensuite la simulation de même filtre de Sobel on utilisant XSG sous Simulink. XSG nous à offrir la possibilité de concevoir ce filtre on utilisant une bibliothèque riche de bloc d'opération arithmétique, logique, fonctions prédéfinies, mémoires, etc. La conception par XSG ne nécessite pas une expérience approfondie sur la maîtrise des langages de description matériels. Le modèle créé à base de XSG pour la détection de contours était évalué par simulation où nous avons pris des images et les traiter.

CONCLUSION GENERALE

Conclusion générale

Le traitement d'image numérique est devenu un axe de recherche très important, car l'image numérique aujourd'hui est devenue un élément essentiel dans notre vie quotidienne, il se trouve par tous ; la sécurité, la robotique, la médecine, etc.

La détection des contours est l'une des étapes de prétraitement importantes dans de nombreuses applications de traitement d'image. C'est également une opération importante à la fois en vision biologique et en vision par ordinateur. Dans ce mémoire, nous avons présenté les étapes de réalisation d'un système qui détecte les contours d'une image, de premier pas (validation par simulation) au dernier pas (l'implémentation sur FPGA validation en temps réel). Nous avons choisi le filtre de Sobel comme exemple de réalisation à cause de sa simplicité.

Nous avons utilisé l'outil XSG qui nous a permis de réaliser de notre système de façon rapide et simple ; la conception de ce système sera peut-être difficile et prendre du temps si on utilise un langage de description matériel (VHDL, Verilog). XSG nous a permis de faire la simulation sous Matlab ainsi que la génération du code VHDL équivalent du système réalisé de façon automatique (compilation).

Les résultats de simulation obtenus ainsi que celles de temps réel sur FPGA montrent bien le bon fonctionnement de notre système avec une bonne performance. Ce travail ouvre nous l'horizon d'utiliser ce genre d'outil de conception (XSG) pour résoudre d'autre problème de conception ainsi que la maîtrise de la programmation des circuits FPGA, nous espérons de continuer dans cet axe.

Bibliographie

Bibliographie

- [1] <http://dspace.univ-eloued.dz/bitstream/123456789/552/1/621.382-049.pdf>
Transmission sans fil d'image par le système SC-FDM Réalisé par :Bilal DOUDI- Souheila SEROUTI 2017/2018.
- [2] https://www.memoireonline.com/12/09/3040/m_La-liaison-automatique-des-plusieurs-images-percues-sur-un-scanner4.html, La-liaison-automatique-des-plusieurs-images-percues-sur-un-scanner4.htm par Jean-marie MUNGUAKONKWA BIRINGANINE 2008
- [3] <https://cel.archives-ouvertes.fr/cel-00125868v4/document>. Quelques méthodes mathématiques pour le traitement d'image Maïtine Bergounioux on 4 Jan 2009
- [4] <http://miv.u-strasbg.fr/mazet/ofti/segmentation-handout.pdf> . Vincent MAZET (Université de Strasbourg)- Outils fondamentaux en traitement d'images 12/12/2019
- [5] <http://webia.lip6.fr/~thomen/Teaching/BIMA/cours/intro.pdf>. Bases du traitement des images - Introduction et fondements auteur Nicolas Thome, 13 Septembre 2016
- [6] https://www.univ-usto.dz/laboratoires/laar/images/Doctorat/THESE_BENDAOU_D_MOHAMED_HABIB_Partie_1.pdf
, Développement de méthodes d'extraction de contours sur des images à niveaux de gris Présenté par : BENDAOU_D Mohammed Habib 2016 / 2017.
- [7] POLYCOPIE Cours de Traitement d'images et vision) Préparée par : Dr. Abdelnour BOUKAACHE 2016 –2017.
- [8] <http://www.lossendiere.com/2016/08/31/les-types-dimages-numeriques>, Les types d'images numériques par M. Lossendière | Août 31, 2016 | [Images numériques](#).
- [9] https://igm.univ-mlv.fr/~vnozick/teaching/slides/M1_ti/01%20image_numerique.pdf Introduction à l'image numérique - UPEM - Master 1 auteur :Vincent Nozick (01 image_num)
- [10] <https://tecfa.unige.ch/tecfa/teaching/staf13/fiches-mm/bitmapvectoriel.htm>, Les types d'images Made by Daniel Peraya.
- [11] <http://blogs.lyceecfadumene.fr/informatique/files/2010/03/Image.pdf>, Image numérique : acquisition et traitement d'images - Lycée du Mené présenté par CRIPT Bretagne 22 mars 2010.
- [12] <https://tel.archives-ouvertes.fr/tel-00708233/document>. Conception d'un outil de prototypage rapide sur le FPGA pour des applications de traitement d'images 13/10/2011.

- [13] http://files.yoannsculo.fr/documents/UTT/Dossier_AC_TI_Yoann_Sculo.pdf.
Dossier_AC_TI_Yoann_Sculo auteur Yoann 18/06/2009.
- [14] http://www.ensta-paris.fr/~manzaner/Cours/IAD/AM_Introduction.pdf
- [15] https://www.researchgate.net/publication/306394987_edge_detectionapplication_of_first_and_second_order_derivative_in_image_processing
- [16] https://www.researchgate.net/publication/2336485_Edge_Detection_Techniques_-_An_Overview
- [17] https://www.researchgate.net/profile/Rokia_Missaoui/publication/255640977_Detection_efficace_de_contours_par_decomposition_de_l'image_en_regions_homogenes/links/53fb3b660cf27c365cf08753/Detection-efficace-de-contours-par-decomposition-de-limage-en-regions-homogenes.pdf
- [18] Cours IMAGERIE ET VISION INDUSTRIELLE Introduction à la Détection de Contour B.
BOULEBTATECHE Imagerie et Vision Industrielle Master 1 : Auto. Info. Industrielle
- [19] Image Registration by Maximization of Combined Mutual Information and Gradient Information
Josien P. W. Pluim, J. B. Antoine Maintz and Max A. Viergever
- [20] Segmentation d'images et détection de contours (Transparents grace à Yves Goussard) Lecture:
Jennifer Campbell ELE8812 29-31 mars 2016.
- [21] <https://www.youtube.com/watch?v=P7tYmNNIARw>
- [22] TI – Traitement d'Images Semaine 10 : Détection de contours (2) Olivier Losson
- [23] Segmentation of Images by Using Edges Detection Techniques Shahla Hazim Ahmad Kharrofa
2008.
- [24] mémoire Traitement d'images monochromes Détection de contours, Filtrage (Spatial et fréquentiel) et segmentation par Réseaux de Neurones Réalisé par : ZerouguiAissaMme : Boulgamh Sari nabil, Juin 2017.
- [25] Traitement d'image, détection des contours par filtre Par Loïc FEUGA
<https://www.supinfo.com/articles/single/5679-traitement-image-detection-contours-filtre>
- [26] Projet de MTI Eric Lormeau, Mastere IST (98) Localisation et epaisseur des contours des detecteurs par derivation (Sobel, Prewitt, Roberts)ou par Gradient Morphologique.
- [27] Edge Detection of Images Based on Improved Sobel Operator and Genetic Algorithms Zhang Jin-Yu¹ , Chen Yan² , Huang Xian-Xiang³ .
- [28] Comparison of Canny and Sobel Edge Detection in MRI Images.
- [29] A Descriptive Algorithm for Sobel Image Edge Detection O. R. Vincent O. Folorunso 2009.
- [30] Performance Analysis of Canny and Sobel Edge Detection Algorithms in Image Mining
Dr.S.Vijayarani¹ , Mrs.M.Vinupriya October 2013
- [31] https://idpoisson.fr/louchet/teaching/timo/ben_hamadi/rapport_benhamadi.pdf
- [32] Transposed Structure Design of FIR Filter using VHDL Hanny Kumar¹ and Kamal Kumar².
- [33] MEMOIRE Implantation d'application de traitement de signal sur système mono-puce reconfigurable 'SoPc' par Mr. Moncef KHITAS 2014.

[34] Composants logiciels et matériels d'un système embarqué, lien :

<https://mouridsmida.files.wordpress.com/2014/04/chapitre-2-partie1partie2partie31.pdf>

[35] Atlys Board Reference Manual, Digilent INC, December 19, 2011.

ملخص

تعد معالجة الصور الرقمية اليوم أحد مجالات البحث النشطة نظرًا لأهميتها المتزايدة في حياتنا اليومية. المعالجة الرقمية للصور نجد استعمالها اليوم في كل مكان حولنا؛ من أجل الأمن، التصوير السينمائي، علم الأحياء، الصناعة، التطبيقات العسكرية، إلخ. العمل الحالي يعالج هاته المسألة؛ بحيث اخترنا أحد أنواع معالجة الصور المهمة؛ وهي الكشف عن الحواف في الصور. تستخدم هاته التقنية بشكل كبير في العديد من المجالات (علم الأحياء، الصناعة، ...). لقد صممنا ونفذنا على شرائح الـ FPGA نظامًا يعتمد على مرشح Sobel لغرض اكتشاف الحواف في الصور. أين استخدمنا أحد أدوات التصميم وحديثة وهي Xilinx System Generator (XSG). هذا البرنامج يتيح لنا تحقيق تصميمات معقدة بأقل جهد ممكن. أظهرت النتائج التي تم الحصول عليها عبر المحاكاة والتحقق الفعلي في الوقت الأني على شريحة FPGA، مع الأداء الجيد الذي تم الحصول عليه، أن اختيار هذا النوع من منصات الحوسبة كان اختيارًا صائبًا لتنفيذ نظام الكشف عن الحواف في الصور أو تنفيذ أي نظام لمعالجة الصور بصفة عامة.

Abstract

Today digital image processing is one of active research areas due to its increasing importance of in our daily life. Image processing is needed everywhere; for security, cinematography, biology, industry, military, etc. The current work treats this issue; we have selected one of the important kinds of image processing; the edge detection, edge detection is largely used in many areas (biology, industry, ..). We have designed and implemented on FPGA a system based on Sobel filter for the purpose of edge detection in images. One of the important designing tools, namely Xilinx System Generator (XSG) is used for the design, it lets us achieve complicated designs with minimum efforts. The obtained results for both simulation and real time validation on FPGA, with the high performance obtained showed the good choice of this kind of computing platforms for implementing this edge detection system and in general for implementing any other image processing system.

Résumé

Aujourd'hui, le traitement d'images numériques est l'un des domaines de recherche les plus actifs en raison de son importance croissante dans notre vie quotidienne. Le traitement d'image est nécessaire partout; pour la sécurité, la cinématographie, la biologie, l'industrie, les applications militaires, etc. Les travaux actuels traitent cette affaire; nous avons sélectionné l'un des types importants de traitement d'image; la détection des contours, la détection des contours est largement utilisée dans de nombreux domaines (biologie, industrie, ..). Nous avons conçu et implémenté sur FPGA un système basé sur le filtre Sobel à pour la détection des contours dans les images. L'un des outils de conception importants, à savoir Xilinx System Generator (XSG), est utilisé pour la conception, il nous permet de réaliser des conceptions complexes avec un minimum d'efforts. Les résultats obtenus tant pour la simulation que pour la validation en temps réel sur FPGA, avec les hautes performances obtenues; ont montré le bon choix de ce type de plates-formes de calcul pour la mise en œuvre de ce système de détection de contours, et en général pour la mise en œuvre de tout autre système de traitement d'image.

