

Système temps réel de suivi de mouvement respiratoire en IRM

Maroua Garci

▶ **To cite this version:**

Maroua Garci. Système temps réel de suivi de mouvement respiratoire en IRM. Sciences de l'ingénieur [physics]. 2012. hal-01783560

HAL Id: hal-01783560
https://hal.univ-lorraine.fr/hal-01783560

Submitted on 25 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-memoires-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10
http://www.cfcopies.com/V2/leg/leg_droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

Master Systèmes Embarqués et Energie

Faculté des sciences et technologies
BP70239

54506 VANDOEUVRE LES NANCY

Université de Lorraine
Faculté des Sciences et Technologies

Master Systèmes Embarqués et énergie

Spécialité « Electronique embarquée et microsystèmes »

Année universitaire 2011/2012

Système temps réel de suivi de mouvement respiratoire en IRM

Mémoire présenté par « Garci Maroua »

Soutenu le

12 Septembre 2012

Stage effectué au sein du laboratoire IADI - Imagerie Adaptive Diagnostique et Interventionnelle. Nancy

CHU de Nancy Brabois

4ème étage – Tour Drouet

Rue du Morvan

54511 Vandœuvre-lès-Nancy

Tuteur industriel : Jovanovic Slavisa

Tuteur universitaire : Tisserand Etienne

Dedicace

A ma famille pour m'avoir toujours soutenue et encouragée,
A mes amis pour leur présence et disponibilité,
Je dédie ce modeste travail

Remerciement

Ce travail s'inscrit dans le cadre du stage de fin d'études de Master 2 Systèmes embarqués et microsystèmes, parcours électronique embarquées et microélectronique. Ce stage a été effectué au sein du laboratoire Imagerie Adaptative Diagnostique et Interventionnelle (IADI, INSERM U947). Au terme de ce projet je tiens à exprimer ma profonde gratitude à toutes les personnes qui ont contribué directement ou indirectement à l'aboutissement de ce travail.

Je remercie M. Jacques Felblinger, directeur du laboratoire IADI pour m'avoir offert l'opportunité d'effectuer ce projet au sein du laboratoire et pour l'attention qui m'a accordée au cours de ce stage

Je suis très reconnaissante à M. Slavisa Jovanovic, mon maître de stage, pour sa disponibilité et ses observations pertinentes. Ses conseils et son assistance précieuse m'ont offert l'opportunité d'acquérir une expérience enrichissante et pleine d'intérêt durant ce stage.

Je tiens à exprimer ma profonde gratitude à Z'hor Romdane Cherif, pour m'avoir encouragée et soutenue durant ce stage et à toute l'équipe travaillant au sein du laboratoire pour la bonne ambiance.

Enfin, je remercie tous les enseignants du Master2 systèmes embarqués et énergie pour la qualité de la formation qu'ils nous ont offerte.

Table des matières

1 IRM, detection des mouvements en IRM **10**

1.1 Imagerie par résonance magnétique 11

1.1.1 Définition 11

1.1.2 Historique 11

1.1.3 Principe de fonctionnement 11

1.2 Mouvements en IRM : problématique et solutions 14

1.2.1 Problématique du mouvement en IRM 14

1.2.2 Solutions pour éviter les artéfacts liés à la respiration 15

1.2.3 Solutions optiques pour la correction des mouvement en IRM 17

1.2.4 Solution proposée 18

2 Algorithme de suivi de mouvement en Matlab **21**

2.1 Choix de l'algorithme 22

2.2 Validation de l'algorithme sur Matlab 22

2.2.1 Lecture des images 22

2.2.2 Calcul de la distance 23

2.2.3 Seuillage 24

2.2.4 Filtrage morphologique : érosion et dilatation 25

2.2.5 L'étiquetage 27

2.2.6 Calcul des coordonnés des centres de gravité 29

2.2.7 Traçage de la trajectoire des formes au cours du temps 30

3 Algorithme de suivi du mouvement en VHDL **32**

3.1 Traitement d'images en temps réel 33

3.1.1 Système temps réel 33

3.1.2 Structures reconfigurables : FPGA en traitement d'images 33

3.1.3 VHDL : langage de description Matériel 34

3.2 Implantation d'une première architecture 35

3.2.1 Lecture des images et calcul de la distance 35

3.2.2 Filtrage morphologique : érosion et dilatation 36

3.2.3 Étiquetage 37

3.2.4 Calcul des coordonnées des centres de gravité 38

3.3 Synthèse de l'architecture 40

3.3.1 Synthèse de la première architecture 40
3.3.2 Proposition d'une deuxième architecture optimisée 42

A **48**

Table des figures

1.1 Orientation de la magnétisation du proton 12
1.2 Relaxation logitudinale et transversale[24]. 13
1.3 Mouvement respiratoire : inspiration et expiration[9] 14
1.4 Extrait d'une séquence dynamique en coupe sagitale 15
1.5 Courbes de respiration avec des échos-navigateurs 16
1.6 Système optique avec mouthpiece 17
1.7 Système optique avec marqueur codé 18
1.8 Système proposé 19
1.9 Méthode de la disparité binocluaire [32] 20

2.1 Schéma bloc de l'algorithme choisi 22
2.2 Image capturée de la ceinture sur le dispositif 23
2.3 Image réelle 24
2.4 Résultat de la difference 24
2.5 Résultat du seuillage 25
2.6 Schéma de l'érosion 26
2.7 Schéma de la dilatation 26
2.8 Résultat de l'érosion 27
2.9 Résultat de la dilatation 27
2.10 étiquetage d'une image binaire 28
2.11 Schéma fonctionnel de l' étiquetage simplifié 28
2.12 Coordonnées des centres de gravités en Matlab 29
2.13 Trajectoire des formes en fonction du temps 30
2.14 Mouvement des formes 31

3.1 Flot de conception en VHDL [3] 34
3.2 Graphe flot de données du bloc effectuant le calcul de la distance 35
3.3 lecture sur Matlab du résultat de l'algorithme de seuillage en VHDL 35
3.4 Difference entre les résultats pour le seuillage 36
3.5 Difference entre les deux résultats 36
3.6 Graphe flot de données de l'érosion 37
3.7 Graphe flot de données de la dilatation 37
3.8 Graphe flot de données de la fonction étiquetage 38

3.9 Calcul des centres de gravités 39
3.10 Comparaison des coordonnées 40
3.11 Graphe flot de données de l'architecture complète 41
3.12 Description structurelle de la fonction calcul de la distance 42
3.13 Architecture optimisée du filtrage morphologique 43

A.1 Caméra comptabile IRM choisi, fabricant MRC 48
A.2 Carte V707 49
A.3 Ceinture en carton avec des formes rouges 49
A.4 Structure interne d'un FPGA 49

Liste des abréviations

A

ALU : Arithmetic and Logic Unit ou unité arithmétique et logique

F

FPGA : Field Programmable Gate Arrays ou réseaux logiques programmables
FID : Free Induction Decay ou signal de précession libre

I

IADI : Imagerie Adaptative Diagnostique et Interventionnelle
IRM : Imagerie par Résonance Magnétique

R

RMN : Résonance Magnétique Nucléaire
RTL : Registre Transfert Level

V

VHDL : Hardware Description Language.
VHSIC : Very High Speed Integrated Circuit.

Introduction générale

L'imagerie par résonance magnétique IRM est une technique d'imagerie médicale permettant d'obtenir des vues 2D ou 3D de l'intérieur du corps de façon non-invasive avec une résolution relativement élevée. L'IRM est basé sur la résonance magnétique nucléaire qui a pour principe l'observation des modifications de l'aimantation d'un noyau au contact d'un champ magnétique statique et d'un champ électromagnétique tournant[24]. L'IRM nécessite donc un champ magnétique puissant et stable produit par un aimant supraconducteur qui crée une magnétisation des tissus par alignement des moments magnétiques de spin. Des champs magnétiques oscillants plus faibles, dits radiofréquences, sont alors appliqués de façon à légèrement modifier cet alignement et produire un phénomène de précession qui donne lieu à un signal électromagnétique mesurable [4, 22].

Une des contraintes majeures de l'imagerie par résonance magnétique est que le corps à imager doit être parfaitement statique. Si cette condition n'est pas respectée, des artéfacts peuvent apparaitre dans les images. Ces artéfacts peuvent provenir de plusieurs sources telles que les mouvements volontaires ou involontaires du patient et les mouvements physiologiques (respiration, activité cardiaque,etc). Pour éviter ces problèmes, plusieurs techniques de correction peuvent être introduites à différents moments de la production des images en IRM.

L'unité INSERM, IADI (Imagerie Adaptative, Diagnostique et Interventionnelle) dans lequel ce stage s'est déroulé, a pour thématique principale l'imagerie des organes en mouvement. L'objectif est d'intervenir sur l'ensemble de la chaîne d'acquisition afin de corriger et de compenser les mouvements pouvant être gênants pour le diagnostic. Un des mouvements les plus difficiles à contrôler lors de l'acquisition des images est le mouvement respiratoire car il nécessite des apnées qui ne sont pas toujours réalisables, par exemple chez le petit enfant ou les personnes âgées ou dyspnéiques [1][31, 24, 21]. Ainsi, plusieurs travaux de recherche menés au sein du laboratoire se focalisent sur différents types de corrections de ce mouvement[24, 21, 30, 25].

Le but de ce stage est d'appliquer une méthode de suivi du mouvement respiratoire à base d'une caméra associée à des structures de calcul puissantes. Cette méthode permettra d'extraire, à partir des séquences vidéo acquises, le mouvement respiratoire voulu.

1. Personnes ayant une anomalie de respiration

L'objectif de ce stage peut être réparti en quatre phases :

- Le but de la première phase est de faire un état de l'art des solutions de suivi de mouvement existantes, et d'en choisir la plus adaptée pour le mouvement respiratoire.
- Dans la deuxième phase, la méthode de suivi de mouvement choisie, doit être validée dans un premier temps au niveau comportemental sous un environnement de simulation de type Matlab.
- Dans une troisième étape, une architecture de l'algorithme choisi sur l'environnement de simulation doit être proposée et décrite en VHDL puis testée sur une plateforme de calcul FPGA.
- Dans la dernière étape, l'objectif est de tester le système mis en place lors des deux premières phases sur des objets tests (fantômes) hors et dans l'environnement IRM.

Ce manuscrit est organisé de la façon suivante :
Dans le premier chapitre de ce rapport, nous présentons brièvement le principe de fonctionnement de l'IRM, le problème des artéfacts en IRM et nous citons quelques méthodes déjà existantes, développées pour la détection et le suivi du mouvement respiratoire en IRM.

Le deuxième chapitre sera dédié à la description de l'algorithme de détection du mouvement respiratoire que nous avons choisi et à sa validation comportementale sur Matlab.

Le dernier chapitre sera consacré à la conception d'une architecture de l'algorithme, à la description de cette architecture en VHDL et à sa synthèse sur FPGA.

Chapitre 1

IRM, detection des mouvements en IRM

Dans ce chapitre, nous allons décrire très succinctement les bases physiques de l'IRM. Nous citons par la suite le problème d'artéfacts liés à la respiration ainsi que différentes solutions existantes pour le corriger. A la fin, nous présentons notre solution de suivi de mouvement respiratoire en temps réel à base d'une caméra.

1.1 Imagerie par résonance magnétique

1.1.1 Définition

L'IRM est une technique de diagnostique médical puissante qui fournit des images tridimensionnelles et en coupe de grande précision anatomique. L'IRM est une technique radiologique récente, non invasive et sans effets secondaires connus, basée sur le phénomène physique de résonance magnétique nucléaire. Il s'agit simplement d'observer la résonance magnétique nucléaire RMN des protons de l'eau contenus dans l'organisme, c'est à dire la réponse des noyaux soumis à un champ magnétique extérieur et à une excitation électromagnétique.

L'intensité recueillie pour un élément de volume dépend de la concentration de l'eau à l'endroit considéré et on obtient ainsi une image tridimensionnelle de la répartition de l'eau dans le corps d'un patient. Selon la méthode utilisée, le contraste entre deux éléments de volume peut être augmenté si les temps de relaxation des *spins* nucléaires (décrivant le retour à l'équilibre des noyaux après l'excitation) diffèrent dans les deux zones. Il est donc possible d'observer des altérations des tissus grâce aux différences de densité et de relaxation de l'eau.

1.1.2 Historique

En 1930 Isidor Rabi a commencé par étudier les propriétés magnétiques des noyaux atomiques et a développé des méthodes pour les mesurer [20]. Il a reçu le prix Nobel de Physique pour ses travaux en 1944. En 1952, Felix Bloch et Edward Purcell ont reçu le prix Nobel après qu'ils ont décrit et précisé ce qu'est le phénomène de résonance magnétique. En 1969, Raymond Damadian a posé les premières pierres de l'imagerie par résonance magnétique en observant que les propriétés magnétiques des tissus tumoraux sont différentes des tissus normaux. Il avait obtenu des images du thorax en 1977. En 1976, parallèlement, Paul Lauterbrur a travaillé sur le même sujet et a publié ses travaux dans le journal *Nature* en 1973. Peter Mansfield de son côté affina les méthodes permettant d'utiliser les signaux émis par les noyaux atomiques pour former des images.

1.1.3 Principe de fonctionnement

L'IRM est une technique basée sur l'observation de la résonance magnétique nucléaire (RMN) des protons de l'eau[16]. En effet, l'eau constitue environ 70% du corps humain et le proton 1^H est naturellement abondant et très sensible en RMN.
L'intensité du signal observé va donc dépendre de la concentration en eau, mais aussi du temps de relaxation des spins nucléaires. Ainsi on pourra obtenir une image de la répartition en eau dans le corps du patient. Le rôle des agents de contraste va être de diminuer ce temps de relaxation afin d'augmenter l'intensité des signaux.

Magnétisme et atome

Tout noyau porte une charge. Cette charge tourne autour de l'axe nucléaire et engendre un dipôle magnétique qui s'exprime par une grandeur appelée moment magnétique et noté $\vec{\mu}$. En absence de champs, les moments magnétiques ne sont pas orientés.

En présence d'un champ magnétique $\vec{B_0}$ statique, les moments magnétiques prennent 2I+1 orientations soit deux orientations pour 1H (I=1/2). Ces deux orientations correspondent à deux états d'énergie. En présence de ce champ, les moments magnétiques associés aux deux états d'énergie se mettent en mouvement et décrivent deux cônes de précession. Ce mouvement a une vitesse correspondant à une fréquence de précession $\vec{\omega_0}$[16].

Résonance et signal

Pour observer la résonance, il faut fournir une énergie permettant aux noyaux de passer de l'état fondamental à l'état excité. Cette énergie est fournie par un second champ magnétique $\vec{H_1}$ d'intensité 10^6 fois plus faible que $\vec{H_0}$.

Dans le cas d'un spectromètre à transformée de Fourier, $\vec{H_1}$ est envoyé sous forme d'impulsions très brèves RF(de l'ordre de la microseconde) afin d'obtenir la résonance. $\vec{H_1}$ perpendiculaire à $\vec{H_0}$, tourne autour de celui-ci à une vitesse angulaire variable ω. Lorsque la fréquence de rotation de $\vec{H_1}$ est égale à la fréquence de précession de spin, il y a résonance et passage du spin à un niveau d'énergie supérieur[20].

Les protons alignés dans le champ magnétique sont représentés par un vecteur de magnétisation $\vec{M}$, qui a deux composantes, la magnétisation longitudinale M_z et la magnétisation transversale M_{xy}. Si on supprime le champ, le vecteur a tendance à revenir à sa position initiale, c'est à dire colinéaire à l'axe O_z. Sa composante M_y se déphase et tend alors vers 0, de même pour M_x, tandis que M_z croît. Cette décroissance de M_y se fait de façon exponentielle et engendre un courant induit dans une bobine située sur l'axe O_y.

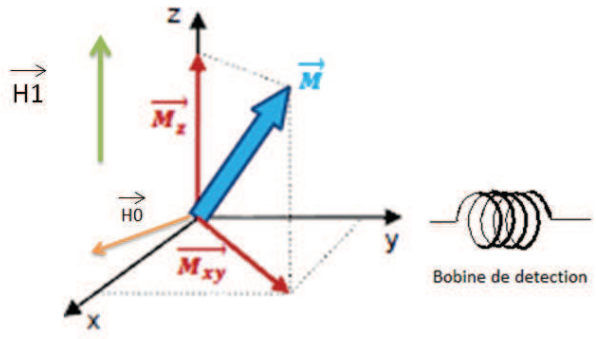


FIGURE 1.1 – Orientation de la magnétisation du proton sous l'effet de champ magnétique.

La relaxation On parle de relaxation quand, après absorption de l'énergie électromagnétique fournie par $\vec{H_1}$, les noyaux tendent à retrouver la distribution de Boltzmann (c'est à dire quand $\vec{M}$ revient à sa position d'équilibre). On peut la décomposer en deux phénomènes, la relaxation longitudinale et la relaxation transversale.

La relaxation longitudinale A l'équilibre $M_z = M_o$. Après le basculement $M_z = 0$.
Le retour de M_z à sa valeur de départ Mo est exponentielle : $M_z(t) = M_0(1 - e^{(-t/T)})$.
Cette relaxation longitudinale, dite relaxation T_1 ou encore relaxation "spin-réseau" correspond au retour à l'équilibre énergétique du système après l'excitation. La constante de temps T_1 est le temps nécessaire pour que les protons atteignent les deux tiers de leur aimantation [24]. Elle dépend en fait de la mobilité des atomes d'hydrogène ou de celle des molécules où ils sont engagés. T_1 sera d'autant plus court que ces hydrogènes seront liés à de grosses molécules.

La relaxation transversale A l'équilibre $M_{xy} = 0$. Après le basculement de 90 degrés, $M_{xy} = M_o$. Le retour de M_{xy} vers 0 est exponentiel $M_{xy}(t) = M_o(e^{(-t/T2)})$.
Cette décroissance de la composante transversale se fait en général plus vite que ne le veut le simple retour à l'équilibre de la composante longitudinale. Elle se caractérise par le temps de relaxation T_2 (encore appelé temps de relaxation "spin-spin"). T_2 est en réalité le temps pendant lequel l'intensité décroît de deux tiers de sa valeur initiale. Ce temps de relaxation T_2 est toujours inférieur au temps de relaxation T_1. Il dépend lui aussi de la mobilité des atomes ou des molécules sur lesquelles ces protons sont engagés.

Ces temps de relaxation vont varier pour un tissu donné selon l'organisation physico-chimique de l'eau dans ce tissu, et c'est sur ces variations que l'on s'appuie pour détecter au sein d'un tissu les modifications liées à la présence d'une lésion.

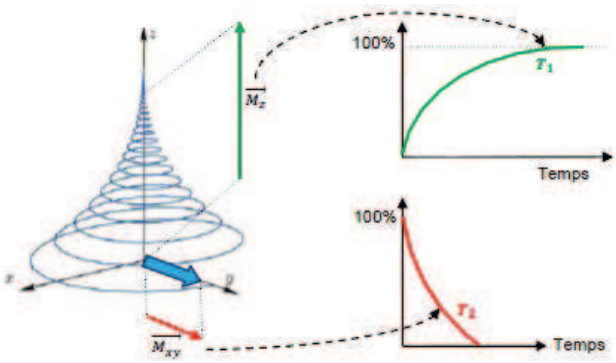


FIGURE 1.2 – Relaxation logitudinale et transversale[24].

Du signal à l'image Le signal réceptionné par les antennes provient de l'aimantation totale du système de spin mais ne contient aucune information spatiale. Il est donc indispensable de localiser spatialement la provenance du signal afin d'obtenir une image. En IRM, l'utilisation de gradients permet la localisation spatiale du signal.
Le signal RMN brut est enregistré après une conversion analogique numérique dans un tableau (une

matrice) appelé espace K. Les données de l'espace K permettront ensuite de constituer l'image 2D grâce à une transformée de Fourier 2D inverse[20, 23].

1.2 Mouvements en IRM : problématique et solutions

1.2.1 Problématique du mouvement en IRM

La problématique du mouvement est un sujet inévitable en IRM car les mouvements volontaires ou involontaires du patient peuvent provoquer des artéfacts qui affectent la qualité de l'image. Ces artéfacts de mouvements sont sources de dégradations qui peuvent masquer des informations importantes pour le diagnostic. Pendant une séquence, l'ensemble des impulsions RF et de gradients de champ magnétique, l'acquisition des données n'est pas instantanée. C'est pour cela que les mouvements perturbent essentiellement l'échantillonnage en phase et que les artéfacts se produisent surtout dans cette direction [20]. De plus, les artéfacts de mouvements sont beaucoup plus prononcés à haut champ et lors de séquences longues.

Les mouvements peuvent être séparés en deux catégories :

- Les mouvements aléatoires qui produisent une image floue avec un bruit,
- Les mouvements périodiques (battements cardiaques, mouvements respiratoires) qui produisent des images fantômes. Les images fantômes seront d'autant plus nettes que la structure en mouvement a un signal élevé.

Les principaux mouvements périodiques qui influencent fortement les images sont les mouvements cardiaques et respiratoires. Pendant ce stage, nous nous sommes intéressés uniquement au mouvement respiratoire.

Le mouvement respiratoire

La respiration est un mouvement périodique, comportant l'inspiration et l'expiration, figure 1.3. Le mouvement de respiration fait le déplacement entre autres des poumons, de la cage thoracique, de certains muscles et centres nerveux ce qui rend l'imagerie IRM très sensible à ce mouvement.

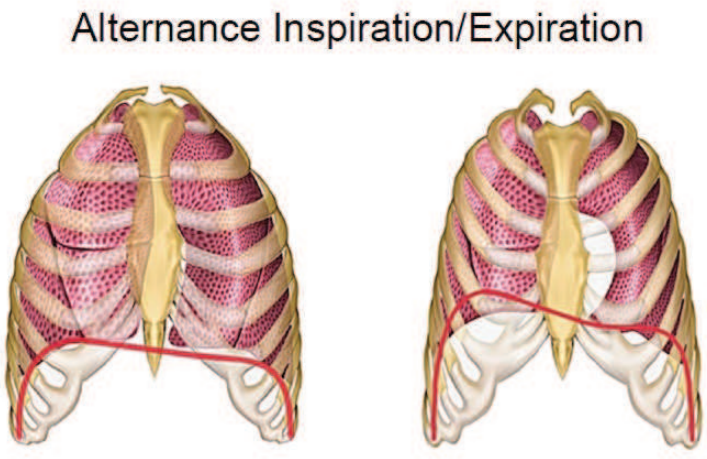


FIGURE 1.3 – Mouvement respiratoire : inspiration et expiration[9]

Déplacements des organes abdominaux dus à la respiration

Dans des études précédentes, Korin et al [13] ont mesuré en IRM et avec la technique de l'imagerie rapide [1] le mouvement des organes abdominaux. Le résultat est que les organes abdominaux subissent principalement un mouvement de translation dans la direction supérieure-inférieure avec une amplitude moyenne de 13 mm en respiration normale et de 39 mm en respiration forcée. Takazakura et al[28] ont mesuré, sur 10 volontaires sains de sexe masculin, le déplacement du diaphragme en utilisant une IRM ouverte à 0,5T dans deux positions : assise et allongée. Durant l'examen, il a été demandé aux volontaires de respirer lentement et profondément pour avoir une plage de mesure entre leur inspiration maximale et leur expiration maximale. Cette étude montre que le déplacement du diaphragme est plus important en position allongée qu'en position assise lorsque les volontaires forcent leur respiration.
Ces études [24, 20, 8] montrent que la respiration influence fortement le mouvement des organes et des muscles de la cage thoracique, figure 1.4 Pour cette raison, la correction du mouvement respiratoire est indispensable. Pour s'affranchir à ce mouvement gênant en IRM, des approches de correction ont été utilisées. Nous allons détailler certaines de ces méthodes dans le paragraphe qui suit.

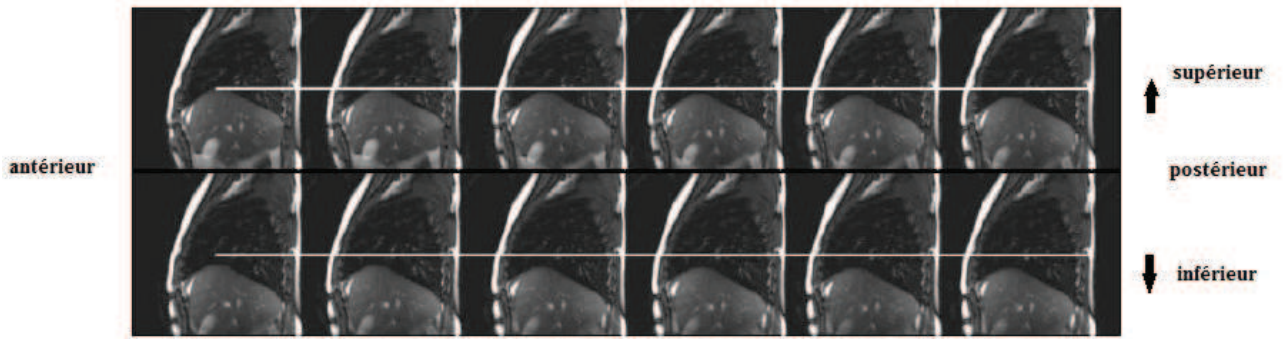


FIGURE 1.4 – Extrait d'une séquence dynamique en coupe sagitale, montrant les modifications de position du diaphragme et de la paroi thoracique au cours d'un cycle (de la fin d'expiration à la fin d'inspiration)[8]

1.2.2 Solutions pour éviter les artéfacts liés à la respiration

Pour réduire la formation d'artéfacts dus à la respiration au cours de l'IRM, plusieurs méthodes de correction sont utilisées, nous pouvons citer les plus connues :

Les apnées

L'apnée est l'arrêt momentané de la respiration (durée entre 20s et 30s). C'est la méthode la plus simple et la plus répandue en clinique pour contrôler le mouvement respiratoire au cours de l'IRM afin d'eviter les artéfacts. Cette technique nécessite une coopération avec le patient, qui est amené à contrôler sa respiration pour que le cycle respiratoire soit reproductible en amplitude et en fréquence. Cette méthode est délicate chez les petits enfants ou les patients souffrant de maladies respiratoires comme les dyspnéiques. En plus, vu que la période de l'apnée est courte, l'acquisition de l'image se fait en plusieurs apnées. Cela nécessite que la position des organes de la zone d'intérêt soit la même

1. Dans les séquences d'écho de spin rapide, le temps libre après le premier écho est utilisé pour recueillir un train d'échos qui va servir à remplir d'autres lignes de l'espace K de la même coupe

entre les différentes apnées ce qui n'est pas facilement réalisable vu la difficulté de maintenir la même position pour le patient.

L'utilisation des capteurs externes pour la correction prospective et rétrospective

Les méthodes de correction de mouvement prospectives en IRM se basent sur la détection et l'estimation du mouvement au cours de l'acquisition alors que les méthodes rétrospectives permettent de corriger les mouvements à postériori et de reconstituer l'image[20, 24]. Ces méthodes de correction ont besoin d'informations précises sur le mouvement en question. Pour ce faire, il faut placer des capteurs externes, compatibles avec l'environnement IRM, capables de détecter le mouvement au cours du temps. Les capteurs les plus utilisés en IRM pour la détection et le suivi du mouvement respiratoire sont :

- Les échos-navigateurs[6],
- Les ceintures pneumatiques[12],
- Les ceintures optiques et les accéléromètres[1, 2, 7].

Les échos-navigateurs Le principe des échos-navigateurs consiste à enregistrer les mouvements du diaphragme pour que les acquisitions se fassent toujours au même moment du cycle respiratoire. Les échos navigateurs permettent de cibler des organes et de suivre leur mouvement interne (voir figure 1.5). Les échos navigateurs ne peuvent pas être situés dans la zone à imager car ils introduisent une modification de l'aimantation susceptible de créer des artéfacts dans l'image. En plus, ils ne sont pas disponibles pour toutes les séquences et ils ne sont utilisables que pour les interfaces à fort contraste (par exemple entre le cœur et les poumons ou entre le diaphragme et les poumons)[24, 23].

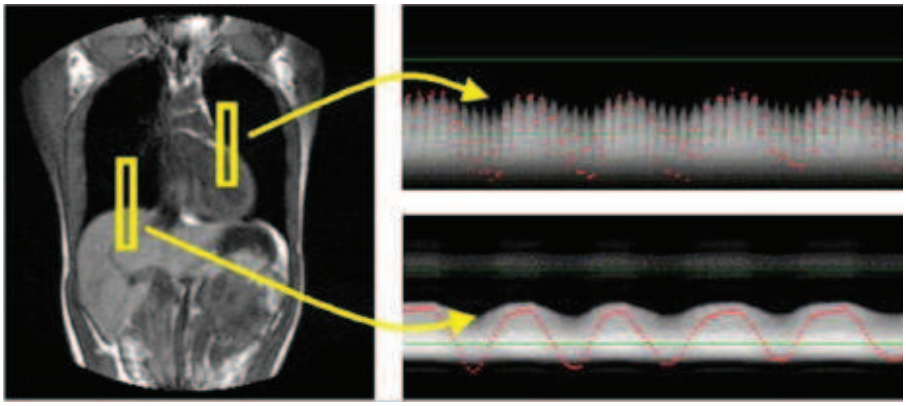


FIGURE 1.5 – Courbes résultantes du placement des échos-navigateurs pour la mesure de la respiration, nehrek et al[10].

Les ceintures pneumatiques

Les ceintures pneumatiques sont des capteurs externes dédiés au suivi du mouvement respiratoire pour la synchronisation respiratoire et le contrôle des apnées. Le principe de ces ceintures est basé sur la mesure des mouvements thoraco-abdominaux provoqués par la respiration [23]. La ceinture pneumatique est le capteur respiratoire le plus couramment utilisé en IRM. La ceinture peut prendre diverses formes, mais le principe reste le même. L'impact du mouvement sur la ceinture entraîne une

variation de volume qui au final se traduit par une variation de pression.
Pour des applications plus complexes telles que la synchronisation ou la détection de mouvements, deux ceintures peuvent devenir nécessaires (une placée sur le thorax [2] et une sur l'abdomen [3]) pour donner le même signal si le patient change de mode de respiration (respiration abdominale ou thoracique), et pour fournir une information supplémentaire sur la direction supérieure-inférieure plus représentative de la respiration.
Avec cette méthode, l'apparition au cours du temps d'une dérive du signal due au changement de position de la ceinture sur le patient pendant l'examen est possible. La variation de la température peut faire varier la pression, les fuites dans le circuit pneumatique peuvent aussi erroner les mesures.

1.2.3 Solutions optiques pour la correction des mouvement en IRM

Des solutions optiques, à base de caméras ont été utilisées dans l'imagerie cérébrale pour faire le suivi du mouvement de la tête au cours de l'IRM. Zaitsev et al [33] en 2006 ont utilisé un système à deux caméras afin de créer une vision "stéréo" [4] couplée à des marqueurs qu'il faudra suivre pour pouvoir estimer le mouvement, (voir figure 1.6). Les marqueurs sont rétro-réflectifs utilisés pour la calibration. Les sphères sont remplis d'eau dopée pour être visibles dans les images IRM. Des "mouthpiece" ont été attachés avec des marqueurs ainsi que des attachement dentaires pour assurer le contact entre la partie supérieure de la bouche et le "mouthpieces".

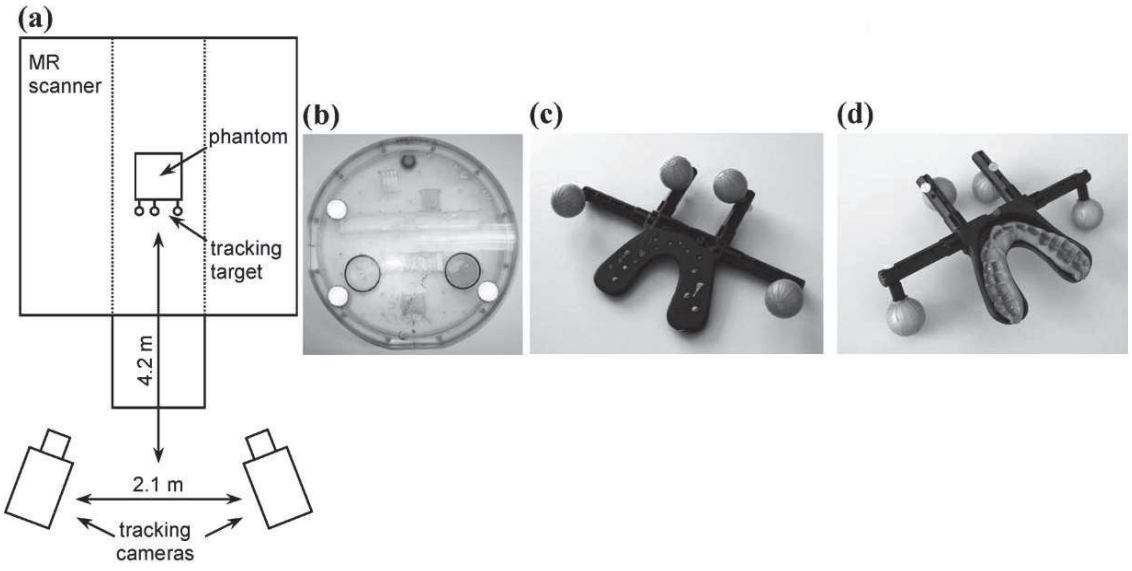


FIGURE 1.6 – a)b)c)-Le système optique installé b)- Fantôme avec des rétro-rectifs pour la calibration, "Mouthpiece" : objets à détecter par le système optique, "Mouthpiece" avec des attachements dentaires pour la fixation dans la bouche du patient[33]

.

2. Région supérieure du tronc, le thorax est limité en haut par la ceinture scapulaire et la première paire de côtes, en arrière par le rachis thoracique et l'arc postérieur des côtes

3. la région du corps située entre le thorax en haut et le petit bassin en bas, contenant la plus grande partie des organes digestifs

4. La stéréoscopie est l'ensemble des techniques mises en œuvre pour reproduire une perception du relief à partir de deux images planes

Cette méthode présente l'inconvénient de tenir le système de marqueurs encombrant et lourd en bouche pendant la séance IRM, ce qui n'est pas évident pour certains patients. En plus, ce système a une taille unique difficilement stable pour tous les patients. Avec cet appareil, pour déterminer la position, il faut au moins détecter trois marqueurs. Donc il faut positionner le patient et les caméras avec une grande précision.

Autres études se sont basées sur l'utilisation d'une seule caméra avec un marqueur de type "échiquier" présentant dans chaque case un motif à détecter. Forman et al. [11] proposent un marqueur avec des motifs codés. Ces motifs donnent une information exacte sur l'emplacement du marqueur sur la tête du patient même si le champ de vision de la caméra ne couvre pas tout le marqueur (voir figure 1.7). La caméra compatible IRM est placée sur l'antenne en face du marqueur. Le signal acquis par la caméra est ensuite envoyé à un ordinateur, dédié au suivi du mouvement, pour faire l'ajustement de la pose en fonction du mouvement. Ces informations sont envoyées à la machine IRM pour faire la compensation prospective du mouvement en temps réel.
Cette méthode permet de détecter le mouvement de la tête mais reste couteuse et difficile à réaliser.

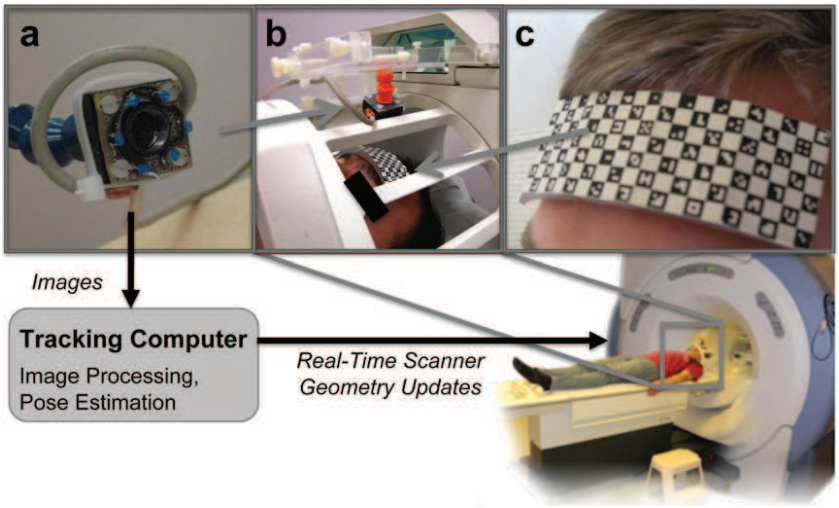


FIGURE 1.7 – Méthode de suivi de mouvement à base d'une seule caméra et d'un marquer codé[11]

1.2.4 Solution optique proposée pour la détection du mouvement respiratoire en temps réel

Nous avons présenté précédemment le problème d'artéfacts lié à la respiration ainsi que les solutions proposées dans la littérature pour corriger ce défaut et améliorer la qualité d'images. Nous avons également cité quelques méthodes utilisées dans l'imagerie cérébrale et qui se basent sur des systèmes optiques et l'utilisation d'une ou plusieurs caméras.
Dans ce qui suit, nous allons détailler notre méthode de suivi du mouvement respiratoire en temps réel qui est basée sur un système optique semblable à ceux utilisés en IRM cérébrale mais avec une plateforme de calcul temps réel. Le système proposé est composé de (voir figure1.8 :

Deux caméras Ce sont des caméras compatibles IRM (cf. annexe) utilisée pour enregistrer la vidéo du déplacement de la cage thoracique lors de la respiration d'un patient en IRM. L'usage de deux caméras permettra d'avoir les déplacements des formes en 3 dimensions à l'aide des techniques

de la stéréoscopie.

Ceinture thoracique C'est la ceinture qui va être placée autour du thorax du patient et qui va se déformer pendant le mouvement respiratoire. Cette ceinture comporte des formes en couleurs qui vont être capturées par les caméras au cours du temps.

Carte V707 (c.f annexe) Cette carte, connectée aux caméras, va recevoir les vidéos de respiration et traiter les images en temps réel à l'aide du FPGA Virtex7 (cf. annexe) qui existe à son bord.

Ordinateur de traitement de données Cet ordinateur reçoit les coordonnées des formes de la ceinture à partir de la vidéo traitée par la carte V707 et en fonction de ces données il adapte le gradient appliqué, les RF et le plan de coupe.

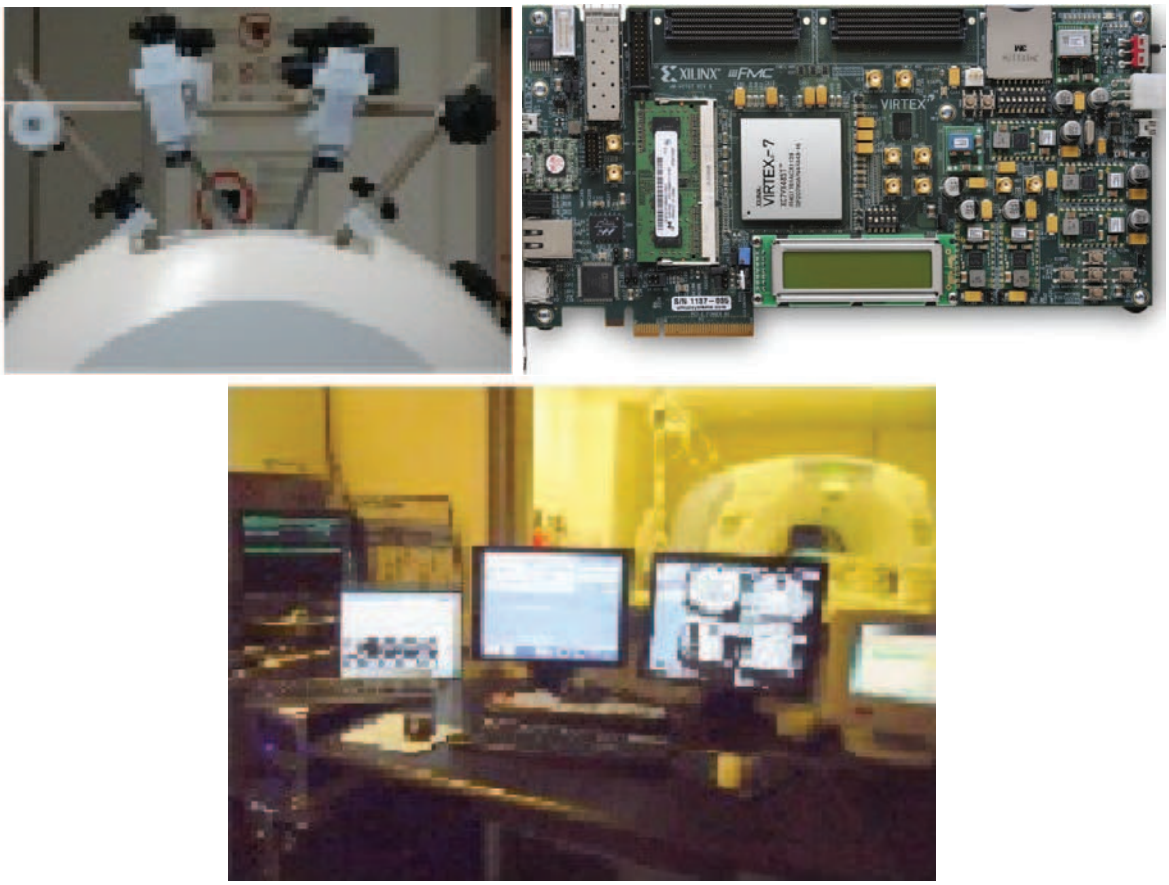


FIGURE 1.8 – Le système proposé pour la détection du mouvement respiratoire en temps réel- à gauche- Les deux caméras placées en face de la ceinture, à droite- La carte V707, au centre- Le PC de traitement de données

Pour pouvoir le reconstruire le mouvement en 3D, il faudra avoir au moins deux images 2D prises depuis deux angles d'observation différents. Il existe de nombreuses méthodes pour une reconstruction 3D à partir des images 2D [32]. Une des méthodes la plus utilisée pour la reconstruction 3D est la méthode de la disparité binoculaire [29]. Cette méthode nécessite la connaissance des paramètres optiques et de la distance entre les caméras (voir figure1.9).

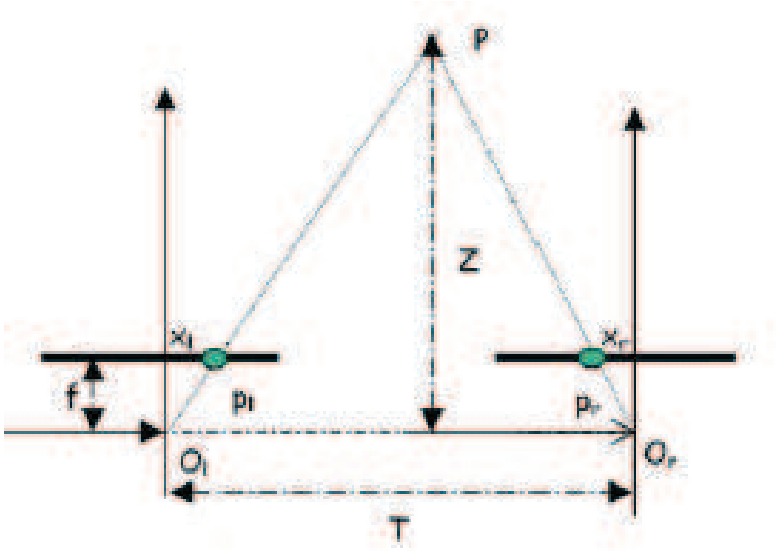


FIGURE 1.9 – Méthode de la disparité binocluaire [32]

Ceci étant, la distance physique d'un point peut être calculée par la formule suivante : $Z = f \cdot \frac{T}{d}$. La difficulté principale de toute méthode de reconstruction 3D à partir des images 2D réside dans la façon de retrouver des objets correspondants dans les images stéreo.
Le système complet de suivi de mouvement comprendra donc deux blocs identiques de détection de région sur des images indépendantes 2D et un bloc de reconstruction 3D. Cette étude se limite uniquement à la détection de region sur des images 2D.

En première étape, nous avons créé une ceinture qui comporte des formes connues en couleurs (cf.annexe). Ensuite, nous avons fait une étude sur le choix d'un l'algorithme de traitement d'images permettant de faire le suivi du mouvement de ces tâches au cours du temps.
Le chapitre suivant détaille l'algorithme choisi et sa validation dans un environnement de simulation de type Matlab.

Chapitre 2

Algorithme de suivi de mouvement : Validation sur Matlab

Dans ce chapitre, nous allons détailler l'algorithme de suivi du mouvement que nous avons choisi puis présenter les résultats obtenus sur Matlab en l'appliquant.

2.1 Choix de l'algorithme

Le choix de l'algorithme pour la détection des mouvements respiratoires s'est porté sur les algorithmes de reconnaissance des visages, des véhicules, dans les vidéos-surveillance [3, 26, 5, 27] d'une part parce qu'ils sont caractérisés par une faible complexité et peuvent être relativement facilement adaptés au suivi temps réel, et d'autre part parce qu'ils montrent de très bons résultats notamment dans le suivi des formes géométriques et couleurs connues, comme dans notre cas.
Le schéma synoptique de l'algorithme choisi est présenté dans la figure 2.1.
L'algorithme prends comme entrée une image avec des marqueurs en couleurs à capturer et donne en sortie les coordonnées des centres de gravité de ces marqueurs suivis.

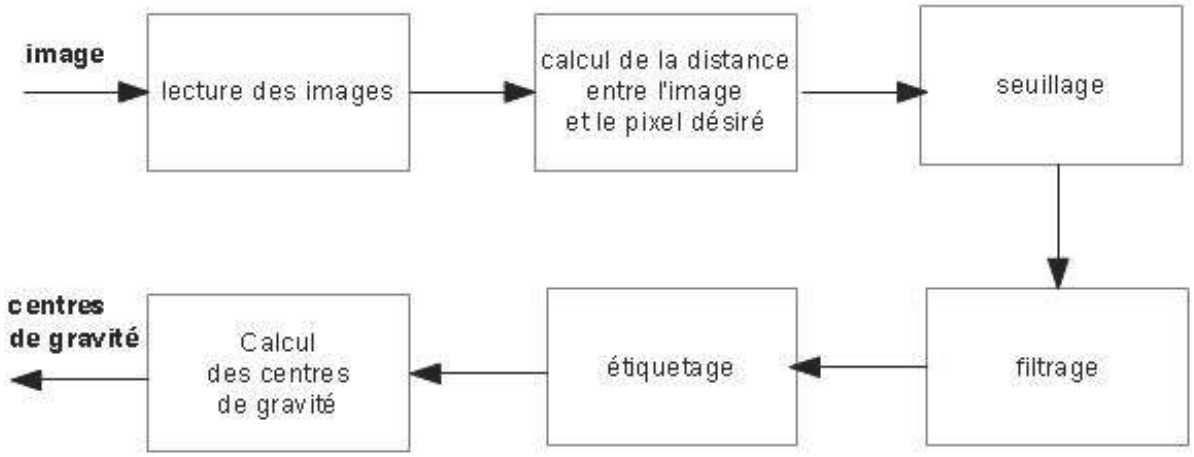


FIGURE 2.1 – Schéma bloc de l'algorithme choisi

2.2 Validation de l'algorithme sur Matlab

Afin de valider l'algorithme choisi dans l'environnement Matlab, nous avons procédé à l'enregistrement d'une séquence vidéo d'un mouvement généré par une plateforme mobile compatible IRM. Le mouvement généré est un mouvement d'une amplitude de 30mm et d'une fréquence de 0,15Hz correspondant à une respiration libre. Pour ce faire, nous avons mis en place une ceinture avec des formes rondes de diffrentes tailles et de couleur rouge disposées au long de la ceinture.

2.2.1 Lecture des images

L'image est capturée à l'aide d'une caméra couleurs qui produit un flux de pixels RGB. La caméra est tenue face à la plateforme mobile, l'arrière plan de l'image est tenu fixe (voir figure2.2). Ce placement de la caméra est choisi en prenant compte son installation finale dans un tunnel IRM
Nous avons enregistré une vidéo au format avi. La taille de l'image vidéo est réglée à 640x480 pixels.

La vidéo obtenue est lue dans l'ordinateur en utilisant l'outil ffmpeg qui traite toute la vidéo et la convertit en trame d'images à raison de 30 images par seconde.

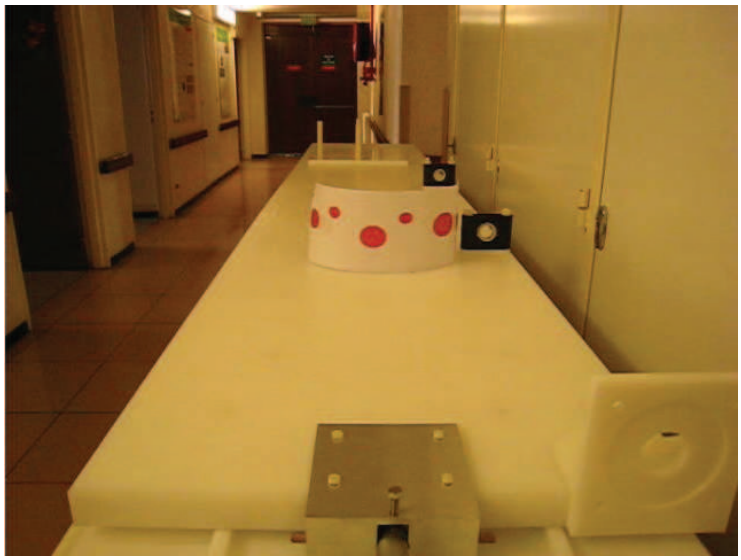


FIGURE 2.2 – Image capturée de la ceinture sur le dispositif

2.2.2 Calcul de la distance

Algorithme

La détection des formes dans notre cas est basée sur la distinction par couleur des formes à suivre. Chaque pixel de l'image (RGB), est constitué de 24 bits, 8 pour chaque couleur. Dans l'éspace des couleurs, un pixel est repéré par ses coordonnés couleurs X, Y et Z qui représentent respectivement R, G et B. Les formes à détecter dans notre ceinture sont de couleur rouge dont le (RGB) varie légèrement en fonction de l'emplacement de la forme par rapport à la lumière, ayant une moyenne de (RGB)=(180,47,8).

Notre objectif est de savoir si un pixel donné de l'image fait partie de la forme à détecter ou pas. Pour ce faire, nous avons eu recours au calcul de la distance séparant ce pixel (dans l'espace des couleurs) des pixels des régions voulues. Cette distance représente le rapprochement en couleurs. Le calcul de la distance est fait à l'aide de cette formule :

$$D = \sqrt{(X_1 - X_0)^2 + (Y_1 - Y_2)^2 + (Z_1 - Z_0)^2}$$

où $(X_1, Y_1, Z_1) = (R_{pixel}, V_{pixel}, B_{pixel})$ et $(X_0, Y_0, Z_0) = (R_{region}, V_{region}, B_{region})$

Simulation sur Matlab

Les régions à détecter dans notre ceinture ont une valeur moyenne de (R,G,B)=(180,47,8). Nous avons utilisé sur Matlab, la fonction *"imabsdiff"* qui effectue la différence entre l'image réelle et l'image de (R,V,B) =(180,47,8) pixel par pixel. Les pixels qui ont une valeur proche de (180,47,8) vont devenir des pixels presque noirs, proches de (0,0,0) (voir figure2.3).

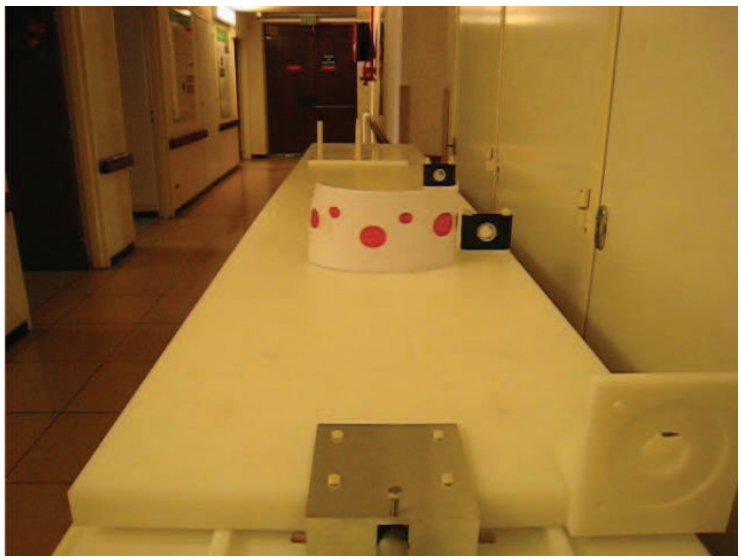


FIGURE 2.3 – Image réelle

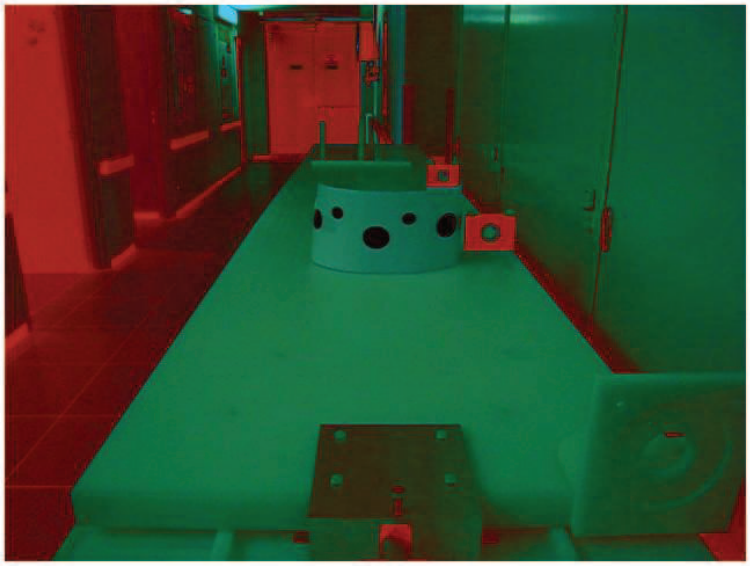


FIGURE 2.4 – Résultat de la différence entre l'image réelle et la couleur des régions suivies

2.2.3 Seuillage

Algorithme

Le pixel le plus proche du pixel à suivre est celui qui a la plus petite valeur de la distance D. D'où, plus D diminue, plus la probabilité que ce pixel est un pixel appartenant à la région à détecter est plus importante. Pour décider si un pixel appartient à la région ou pas, nous avons fixé une valeur de seuil

$$\gamma^2 = 500$$

Pour une valeur de D inférieure à cette valeur, le pixel est considéré appartenant à la région, au delà de cette valeur, le pixel est considéré comme appartenant à arrière plan.

- $D^2 < 500$ pixel devient blanc (R,V,B)=(255,255,255)$\longrightarrow$ appartient à la région.
- $D^2 >= 500$ pixel devient noir (R,V,B)=(0,0,0)$\longrightarrow$ appartient à l'arrière plan.

Simulation sur Matlab

Nous avons simulé ce seuillage sur Matlab $\gamma^2 = 500$ qui a donné une image binaire où le blanc présente les régions détectées et le noir représente l'arrière plan (voir figure2.5).

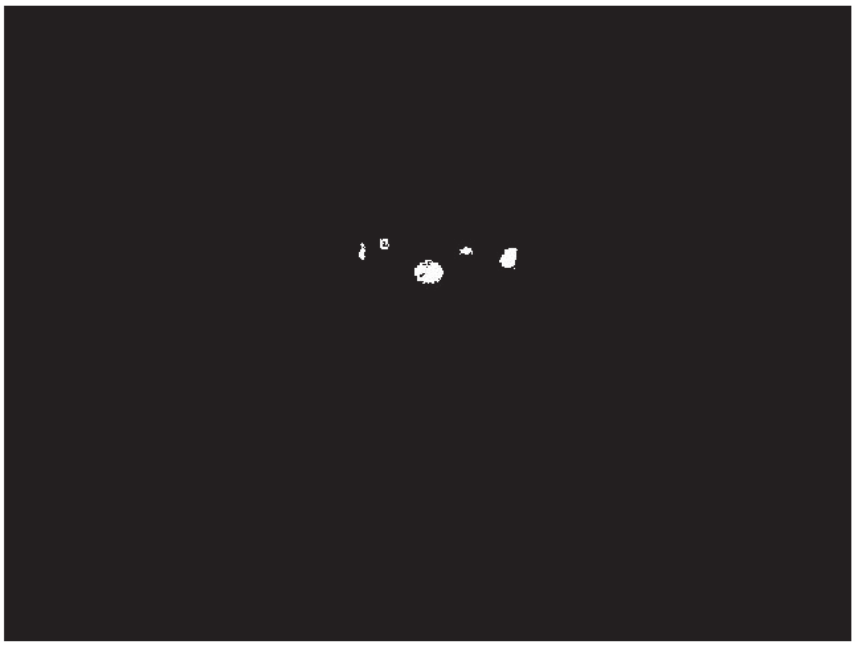


FIGURE 2.5 – Résultat du seuillage

A la fin du seuillage, nous pouvons distinguer nos régions qui représentent les cercles rouges sur la ceinture, mais les formes ne sont pas très compactes et nettes, nous aurons donc recours au filtrage.

2.2.4 Filtrage morphologique : érosion et dilatation

Algorithme

Le filtrage morphologique est un processus de traitement d'image basé sur les formes existantes sur l'image. Les opérations morphologiques appliquent un élément "*structurant*" à une image en entrée pour créer une image de même taille en sortie. Dans le filtrage morphologique, la valeur de chaque pixel de l'image en sortie est déterminée par une comparaison de la valeur de ce pixel en entrée avec son voisinage qui est défini par l'élément "*structurant*" [18, 19, 15].
Filtrage morphologique :

- **Érosion** La valeur en sortie d'un pixel donné est la valeur **minimale** de tous les pixels de son voisinage. Dans une image binaire, si un pixel du voisinage est égale à "0", la valeur du pixel en

sortie, après l'érosion, prend "0". Cette description est traduite par l'équation :

$$(A \oplus B)(x, y) = minA(x - x', y - y'), (x', y') \in D_B$$

où D_B est le domaine de l'élément "*structurant*".

La valeur du pixel traité est celle de la valeur minimale de tout le voisinage qui représente la distance entre le pixel et l'étendue de l'élement "*structurant*".

Ce schéma (figure 2.6) représente le processus d'érosion d'une image binaire avec un élément structurant ligne de trois pixels.

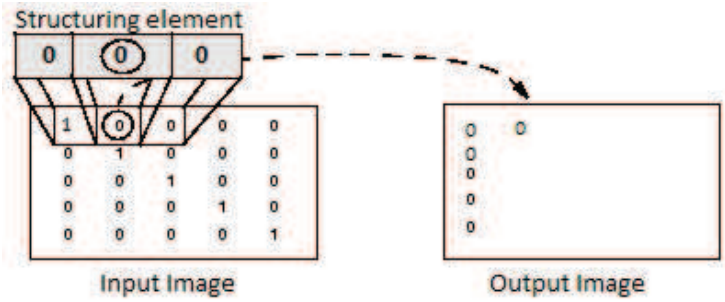


FIGURE 2.6 – Érosion morphologique d'une image binaire avec un élément structurant : ligne de trois pixels

L'érosion permet de :

- éliminer les composants connexes plus petits que l'élément structurant,
- élargir les trous,
- transformer une presque île en une île.

– **Dilatation** La valeur en sortie d'un pixel donné est la valeur **maximale** de tous les pixels de son voisinage. Dans une image binaire, si un pixel du voisinage est égale à "1", la valeur du pixel en sotie, après la dilatation prend "1". Cette description est représentée par l'équation

$$(A \oplus B)(x, y) = maxA(x - x', y - y'), (x', y') \in D_B$$

où D_B est le domaine de l'élément "*structurant*".

La valuer du pixel traité est celle de la valeur maximale de tout le voisinage qui représente la distance entre le pixel et l'étendue de l'élement "*structurant*".

Ce schéma (figure [2.7]) représente le processus de dilatation d'une image binaire avec un élément structurant ligne de trois pixels.

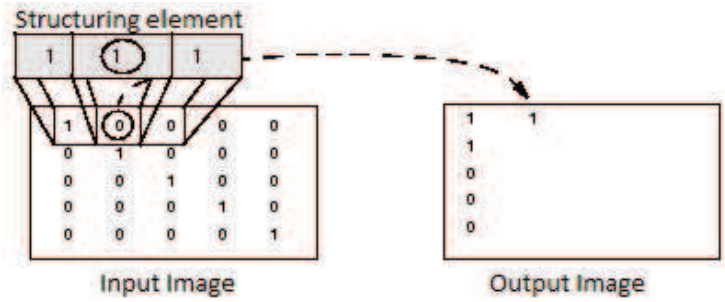


FIGURE 2.7 – Dilatation morphologique d'une image binaire avec un élément structurant : ligne de trois pixels

La dilatation permet de :
- Remplir les trous qui apparaissent dans une région,
- Élargir les formes,
- Souder les formes proches.

Simulation sur Matlab

Nous avons utilisé pour la simulation sur Matlab, un élément "*structurant*" sous forme de disque qui regroupe le pixel de dessus, le pixel de dessous, le pixel de gauche et le pixel de droite.
Les figures suivantes (figure 2.8, et figure2.9) montrent les résultats obtenus sur Matlab de l'érosion et de la dilatation respectivement en utilisant les fonctions "*imerode*" et " *imdilate*".

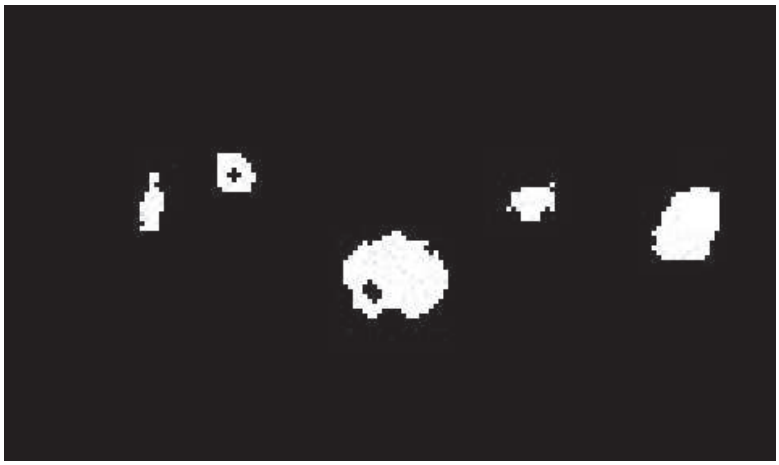


FIGURE 2.8 – Résultat de l'érosion avec l'élément structurant : disk (1,1) des régions à détecter

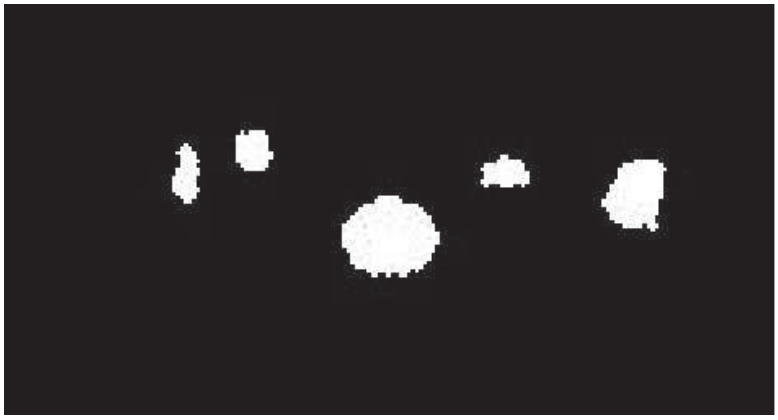


FIGURE 2.9 – Résultat de la dilatation avec l'élément structurant : disk (1,1) des régions à détecter

2.2.5 L'étiquetage

Algorithme

Les éléments connectes sont des regroupements de pixels qui forment une même région dans une image. L'étiquetage est un processus de traitement d'images binaires qui consiste à chercher

les éléments connectés et à attribuer un label pour chaque nouvelle région. Cette opération sert pour distinguer les régions et compter leur nombre dans une image binaire [17].

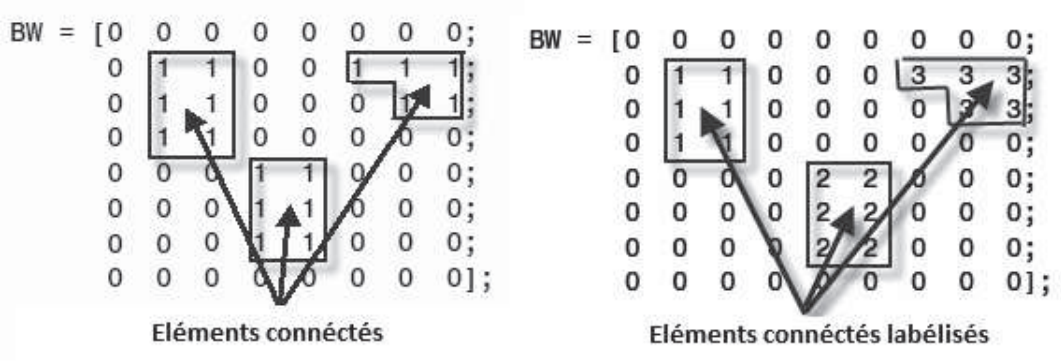


FIGURE 2.10 – L'étiquetage d'une image binaire : Gauche- image binaire non labellisée, Droite- image binaire labellisée[17]

Le processus d'étiquetage est résumé dans le schéma fonctionnel suivant :figure2.11

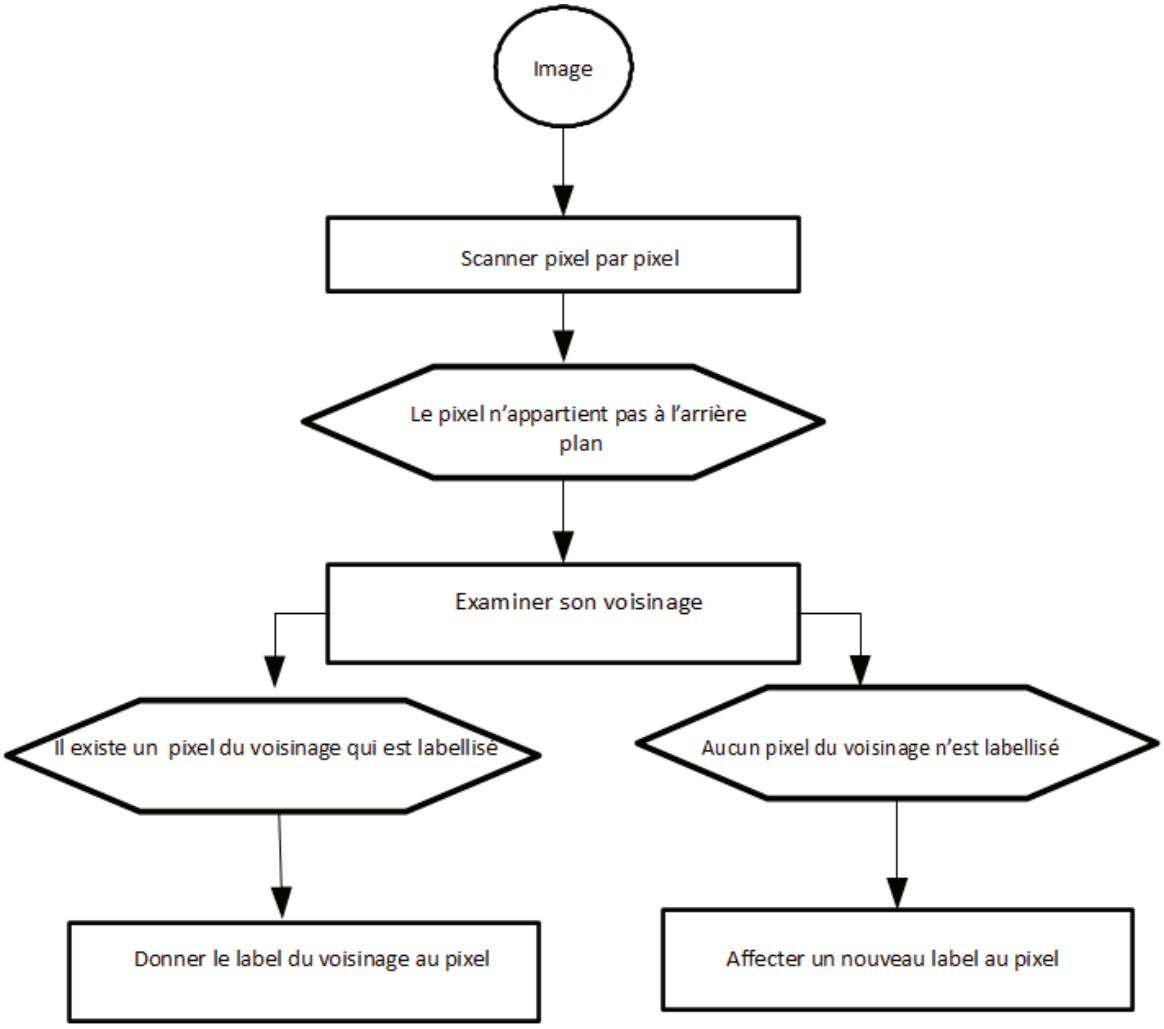


FIGURE 2.11 – Schéma fonctionnel de l' étiquetage simplifié

Simulation sur Matlab

Nous avons simulé sur Matlab la fonction d'étiquetage en utilisant la fonction "*bwlabeln*". Pour visualiser le résultat, nous avons utilisé la fonction "*rgb2label*".

2.2.6 Calcul des coordonnés des centres de gravité

Après avoir étiqueté nos régions, nous avons déterminé les coordonnés des centres de gravité de chaque région en se basant sur le nombre de pixels qu'elle contient. Pour ce faire, nous avons utilisé une formule simplifiée :

$$G(x,y) = (\frac{\sum(x_{lab})}{N_{lab}}; \frac{\sum(y_{lab})}{N_{lab}})$$

où

- x_{lab} et y_{lab} sont les coordonnées des pixels dans une région,
- N_{lab} est le nombre de pixels total de la région.

Simulation sur Matlab

Nous avons utilisé la fonction "*regionprops*" pour tracer les centroïdes de chaque région comme l'indique la figure 2.12.



FIGURE 2.12 – Résultat sur Matlab après la liaison entre les centres de gravité et en superposition avec l'image réelle

Pour la première image de la vidéo, nous avons trouvé les centres de gravité suivants :

Abscisses	270,88	287,148	320,69	350	382,45
Ordonnées	187,91	180,48	202,48	186	191

2.2.7 Traçage de la trajectoire des formes au cours du temps

A l'aide de Matlab, nous avons fait le traitement en boucle de 100 images de la vidéo.
Nous avons tracé en 3D les coordonnés des centres de gravité et leurs déplacements au cours du temps. Superposés avec l'image réelle comme arrière plan, nous avons obtenu le résultat suivant : (figure 2.13)

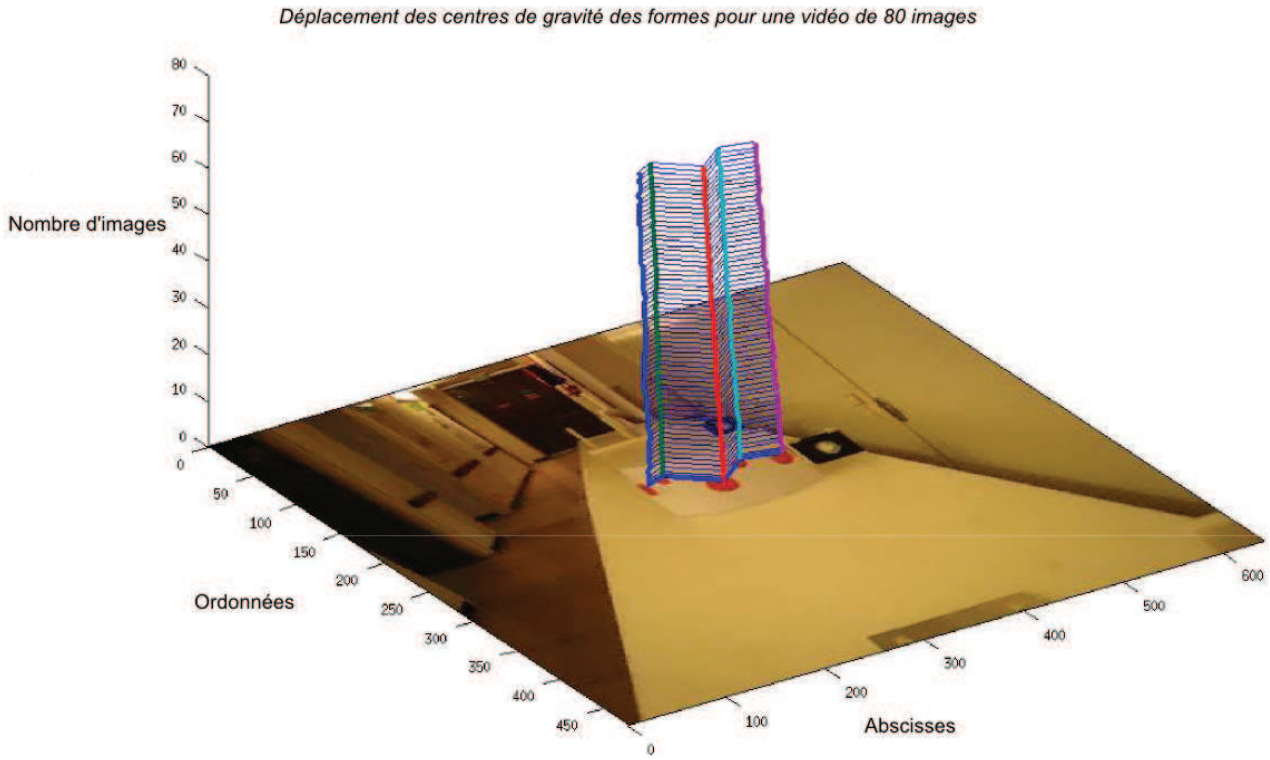


FIGURE 2.13 – Déplacement des centres de gravité en fonction du temps, en superposition avec l'image réelle

Si nous visualisons le mouvement de plus prés, nous obtenons cette figure (voir figure 2.14) où nous pouvons voir les petits déplacements des centres de gravité.
Etant donné que le mouvement généré par la plateforme est un mouvement ayant une dynamique très faible (une fréquence aux alentours de 0.15Hz), on remarque sur l'extrait des centres de gravité calculés (voir figure 2.14), une petite variation des coordonnées des centres de gravité des régions suivies. En une seconde, ce qui correspond à 30 images, le déplacement généré est de l'ordre de quelques millimètres.

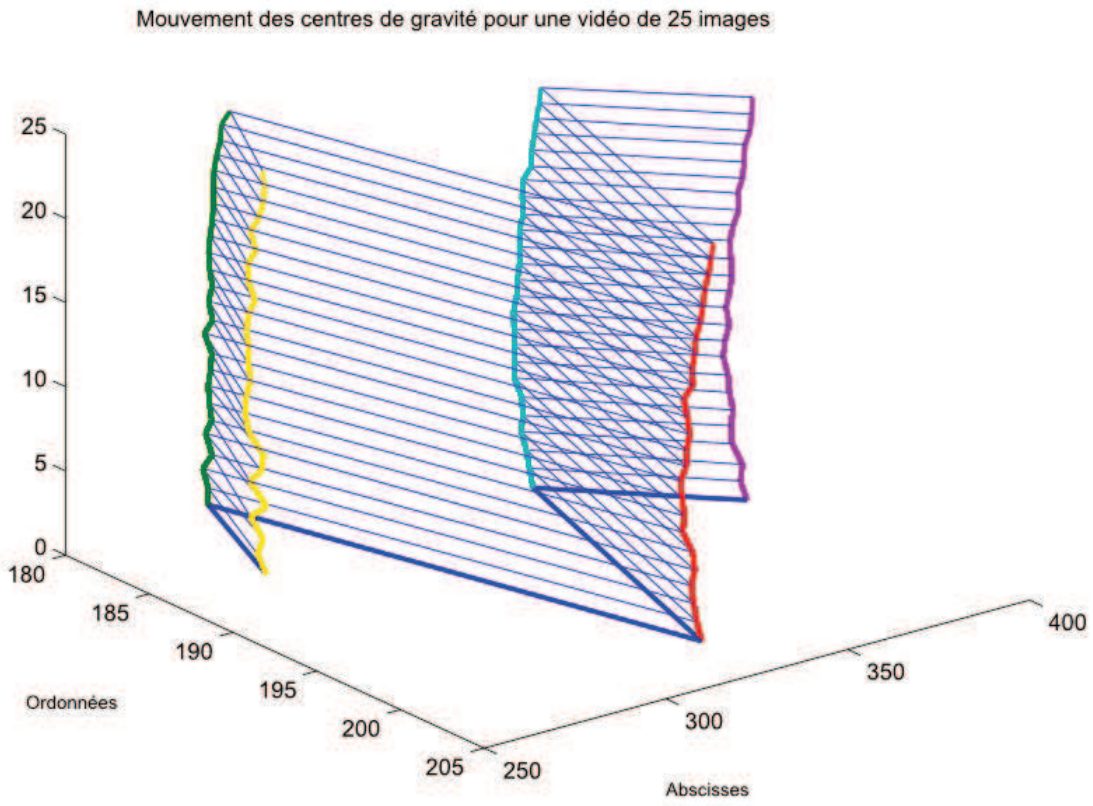


FIGURE 2.14 – Mouvement des centres de gravité en fonction du temps

Chapitre 3

Algorithme de suivi de mouvement en temps réel : Validation et implantation d'une architecture en VDHL sur une plateforme reconfigurable de type FPGA

Notre objectif est d'effectuer le traitement de la vidéo et le calcul des centres de gravité des objets à suivre en temps réel afin de mieux adapter les paramètres d'acquisition des images en IRM en fonction des coordonnées calculées.

3.1 Traitement d'images en temps réel

3.1.1 Système temps réel

Un système temps réel est un système dont la réponse à un événement doit se produire dans un délai déterminé, sinon le système est considéré comme ayant échoué [14].
En traitement d'images, un système d'imagerie en temps réel est celui qui capte des images, les analyse régulièrement pour obtenir certaines données, puis utilise ces données pour contrôler certaines activités tout en tenant compte que l'ensemble du traitement doit se faire dans un temps prédéfini et trop limité. Pour un unique processeur et un unique algorithme, un ordre de grandeur est à gagner en rapidité pour assurer un débit temps réel d'images (25 images par seconde : fréquence supérieure ou égale à la cadence de capture d'image). Les applications qui nécessitent un traitement d'images en temps réel sont abondantes. On peut en citer à titre d'exemples les systèmes robotisés qui utilisent la vision pour la reconnaissance du chemin, ainsi que le monitorage et le contrôle des véhicule à distance. Ces applications exigent une reconnaissance des formes en temps réel afin d'éviter les collisions.
Dans notre cas, l'objectif est de capturer une vidéod'un mouvement respiratoire et d'effectuer le traitement de la vidéo acquise et le calcul des centres de gravité des objets à suivre en temps réel afin de mieux adapter les paramètres d'acquisition des images en IRM en fonction des coordonnées calculées. Etant donné que le mouvement respiratoire a une faible dynamique (à l'ordre d'un dixième de Hz), pour avoir une bonne résolution temporelle de ce mouvement capté par une caméra, nous nous sommes fixés comme contrainte temps réel le traitement d'au moins 25 images par seconde.

3.1.2 Structures reconfigurables : FPGA en traitement d'images

L'idée de base derrière le matériel programmable est d'avoir un circuit générique où la fonctionnalité peut être programmée pour une application particulière. Les ordinateurs classiques sont basés sur cette idée, où l'ALU [1] peut effectuer une de plusieurs opérations basées sur un ensemble de signaux de commande. La limitation de l'ALU est qu'il ne peut effectuer qu'une seule opération à la fois. Par conséquent, une application particulière doit être divisée en une séquence de signaux de commande pour commander la fonction de l'ALU en même temps que la logique nécessaire afin de fournir les données appropriées aux entrées de l'ALU. La séquence de contrôle est fournie par une séquence d'instructions de programme stockée dans la mémoire. D'autre part, la logique programmable, représente la fonctionnalité d'un circuit, où un circuit particulier peut être programmé pour répondre aux exigences d'une application. La principale différence est que la fonctionnalité est implantée comme un système parallèle, plutôt que séquentiel. Toute fonction logique peut être implémentée en utilisant les deux fonctions logiques (OR de ET)[3]. Un FPGA implante la logique requise par une application en construisant un matériel distinct pour chaque fonction. Les FPGA sont intrinsèquement parallèles,

1. L'unité arithmétique et logique, abrégée UAL (ou bien ALU, Arithmetic Logic Unit en anglais), est l'organe de l'ordinateur chargé d'effectuer les calculs. Le plus souvent, l'ALU est incluse dans l'unité centrale ou le microprocesseur.

cette conception matérielle leur donne une grande vitesse de traitement de données tout en conservant la flexibilité reprogrammable logicielle à un coût relativement faible. Cela rend les FPGA bien adaptés au traitement d'images (cf. annexe).

3.1.3 VHDL : langage de description Matériel

Le Vhic Hardware Description Language (VHIC = Very High Speed Integrated Circuit) a été normalisé en 1987 par l'IEEE, sous la norme IEEE 1076-87. Une importante évolution est parue en 1993, sous la norme IEEE 1076-93. C'est cette version du langage qui est majoritairement supportée par les outils du marché. D'autres révisions de la norme sont parues depuis, mais ce sont des évolutions « mineures », et pas forcement supportés par les logiciels. Le VHDL est un langage de description de matériel, ce n'est pas absolument un langage "software" comme le C ou le Java [3].
A partir de ce langage, on peut définir un système par une structure hiérarchique de fonctions, par une structure matérielle, ou encore par une modélisation temporelle (même si elle n'est pas utilisable pour faire du code synthétisable). Ce langage permet d'aller d'un niveau d'abstraction très élevée, par une description algorithmique, jusqu'à un niveau proche du matériel, où l'on décrit le système par un ensemble de porte logique et d'interconnexion (*gate level*). Entre les deux, se trouve le niveau RTL (*Register Transfer Level*). C'est le niveau RTL que l'on utilise le plus quand on fait de la synthèse. En effet, le niveau algorithmique n'est pas forcement synthétisable, il est plutôt utilisé pour la validation comportementale et la simulation.

Flot de conception Le code à synthétiser est généralement écrit dans un niveau RTL. Parallèlement, un banc de test doit être écrit. Ce banc de test, écrit sous la forme d'une modélisation de l'environnement, servira à appliquer les stimuli d'entrée et à extraire les résultats, pour ensuite permettre la validation du système. Une fois le code validé, on le synthétise grâce aux librairies du fabriquant, afin de générer la *netlist* et le code de type *porte*. Ensuite, on fait le Placement/routage du circuit. Une fois le circuit routé, on en extrait les temps de propagation et l'on re-simule le système, pour vérifier que le placement routage est suffisamment optimisé pour respecter les contraintes de temps et de l'espace.

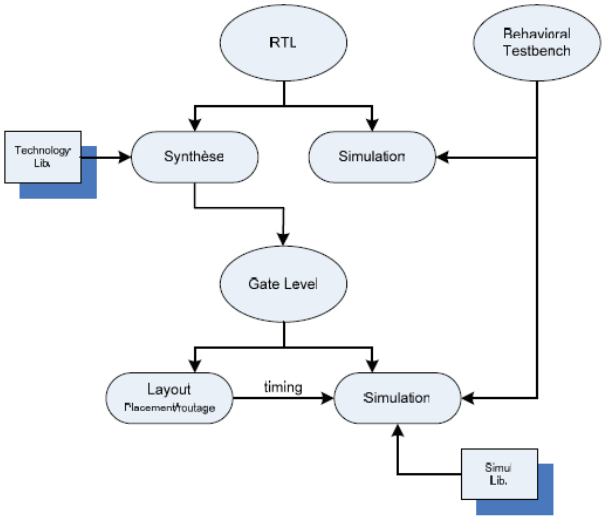


FIGURE 3.1 – Flot de conception en VHDL [3]

3.2 Implantation d'une première architecture

3.2.1 Lecture des images et calcul de la distance

Archirecture en VHDL

Dans cette partie de l'algorithme que nous avons détaillé précédemment dans le 2.1, nous avons fait la soustraction de chaque composante R, G et B de chaque pixel de l'image avec les octets R, G et B désirés des formes à détecter. Pour ce faire, nous avons utilisé 3 opérateurs de soustraction et 3 opérateurs de multiplication avec un additionneur. La valeur calculée de la distance est comparée ensuite avec une valeur seuil ($\gamma = 500$). Le résultat obtenu après le seuillage est un pixel noir ou blanc.

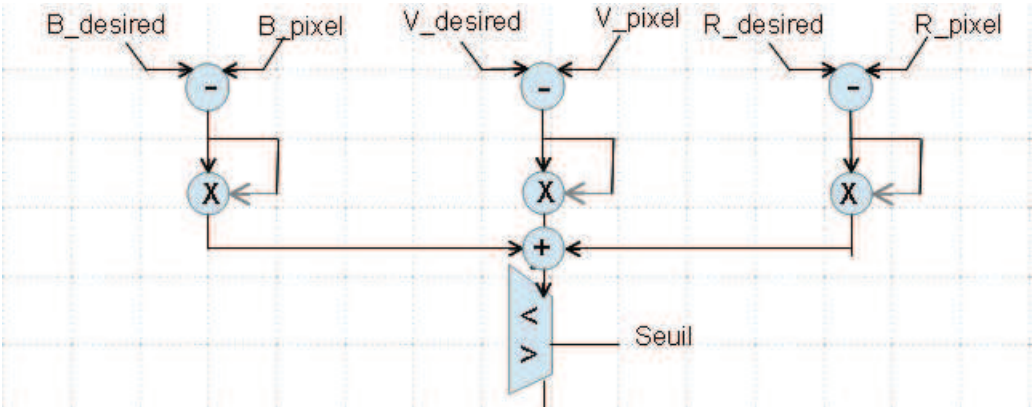


FIGURE 3.2 – Graphe flot de données du bloc effectuant le calcul de la distance

Résultats de la simulation

Afin de voir concrètement en image le résultat de l'architecture de calcul de distance que nous avons implémenté en VHDL, nous avons récupéré dans un fichier binaire la valeur de chaque pixel après le seuillage (noir ou blanc) puis nous avons reconstruit l'image à partir de ce fichier à l'aide de Matlab.

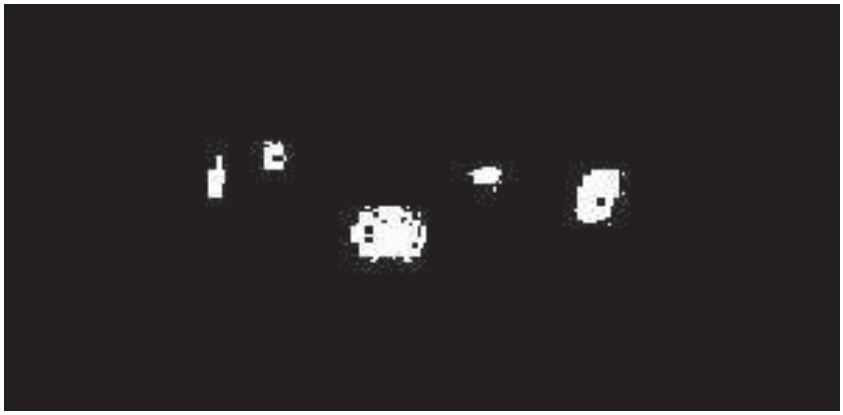


FIGURE 3.3 – lecture sur Matlab du résultat de l'algorithme de seuillage en VHDL

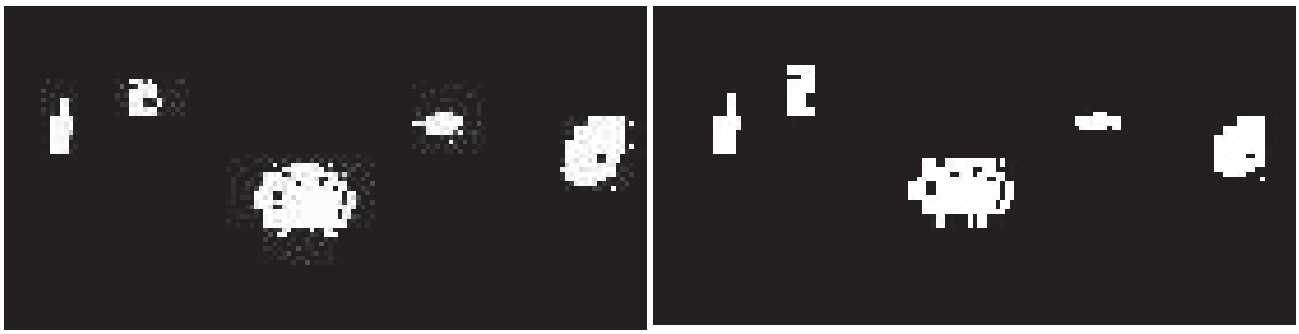


FIGURE 3.4 – (à gauche) : Résultat sur Matlab de l'algorithme de seuillage, (à droite) : Résultat de la simulation de l'algorithme de seuillage sous l'environnement MATLAB

Comparaison entre les résultats de l'algorithme en Matlab et l'algorithme en VHDL Pour mieux voir les différences entre les résultats obtenus à l'aide de l'algorithme en Matlab et en VHDL, nous avons fait la soustraction en valeur absolue des deux images.



FIGURE 3.5 – Différence entre les résultats obtenus avec l'algorithme en Matlab et en VHDL

Cette légère différence revient à la perte de précision à cause de la formule simplifiée que nous avons utilisée pour le calcul de la distance en VHDL par rapport à celle utilisée derrière la fonction "*imabsdiff*" de matlab et qui fait intervenir plus de paramètres tel que l'intensité lumineuse des pixels.

3.2.2 Filtrage morphologique : érosion et dilatation

Architecture en VHDL

Erosion L'algorithme d'érosion a comme principe d'examiner le voisinage d'un pixel selon un élément structurant. Si le pixel traité ainsi que tous les pixels du voisinage sont blancs, le pixel devient blanc, sinon il devient noir, comme nous l'avons déjà expliqué précédemment (voir figure 2.6. Pour implémenter cet algorithme en VHDL, nous avons utilisé un tableau de 3 octets, de taille 640*3, ce qui permet de stocker jusqu'à trois lignes de l'image. Nous avons utilisé aussi deux opérateurs logiques AND à trois entrées.

Après le seuillage, les pixels sont stockés un par un dans le tableau à l'aide des registres à décalage. Une fois le tableau est rempli (nombre de pixels entrants égale à 1920 pixels), on peut commencer à tester le voisinage du pixel 1280 comme l'indique (voir figure 3.6). Le pixel et son voisinage constitué de 4 pixels sont testés à l'aide des portes logiques AND. Si tous les pixels sont blancs, le pixel de sortie sera blanc sinon s'il existe un ou plusieurs pixels noirs, le pixel de sortie va être noir.

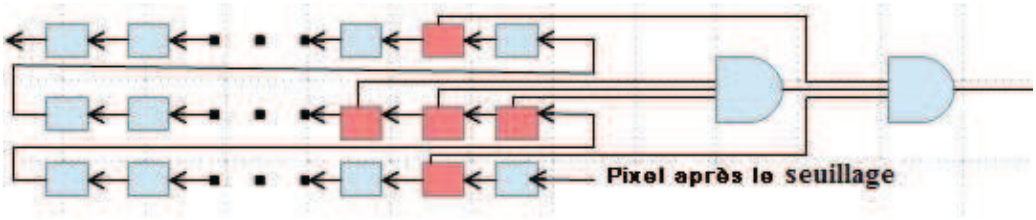


FIGURE 3.6 – Graphe flot de données de l'érosion

Dilatation L'algorithme de la dilatation a le même principe que celui de l'érosion comme nous avons déjà expliqué dans le chapitre précédent (3.7). La différence est que si le pixel traité ou l'un des pixels de son voisinage est blanc, le pixel en sortie sera blanc, sinon il sera noir. En analogie avec l'érosion, nous avons utilisé pour implanter cet algorithme un tableau de 3 octets de taille 640*3. Une fois rempli à l'aide des registres à décalage avec les pixels qui résultent de l'érosion, nous commençons par traiter le pixel 1280 en examinant son voisinage à l'aide des deux portes OR. La sortie est un pixel noir ou blanc qui va passer à l'étape prochaine : l'étiquetage.

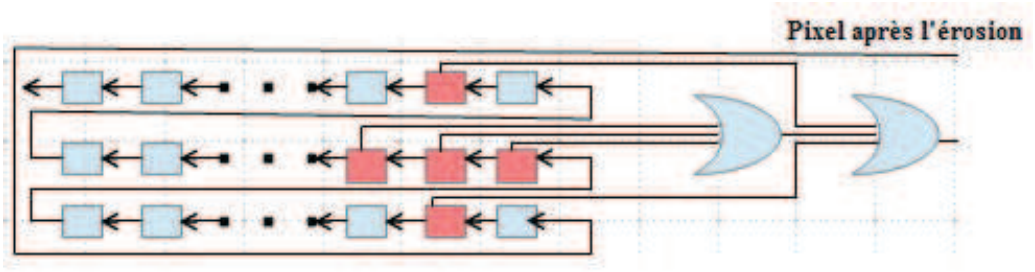


FIGURE 3.7 – Graphe flot de données de la dilatation

Résultats de la simulation

3.2.3 Étiquetage

Architecture en VHDL

Comme nous avons déjà expliqué dans le 2.11, l'étiquetage sert à attribuer une nouvelle étiquette pour chaque nouvelle région d'éléments connectés détectée dans l'image. Pour chaque pixel traité, nous examinons son voisinage constitué de 4 pixels : le pixel de côté et les trois pixels de dessus. Si le pixel traité est blanc et qu'il existe déjà un pixel du voisinage avec lequel il est connecté qui est aussi blanc, le pixel prend la même étiquette que lui, sinon il prend une nouvelle étiquette.
Nous avons implémenté cet algorithme à l'aide de l'architecture suivante (voir figure 3.8) où nous avons utilisé principalement un tableau de 641 pixels, 2 multiplexeurs et un démultiplexeur.
Pour chaque pixel arrivant après le filtrage morphologique (dilatation), nous effectuons le traitement suivant :

- Si le pixel est noir : nous n'affectons pas une nouvelle étiquette et nous effectuons pas le reste du traitement.
- Si le pixel est blanc :

1. Si le pixel de côté est blanc : le pixel prend la même étiquette que lui, cette étiquette est stockée dans la case 641 du tableau et le nombre de pixel de la même région s'incrémente.
2. Si l'un des trois pixels de dessus est blanc : le pixel prend la même étiquette que lui, cette étiquette est stockée dans la case 641 du tableau et le nombre de pixel de la même région s'incrémente.
3. Si aucun des pixels du voisinage n'est blanc, c'est à dire nous avons détecté une nouvelle région, le pixel prend une nouvelle étiquette qui sera stockée dans la case 641 du tableau et global label (le nombre total des régions labellisées) s'incrémente.

A la sortie de cette architecture, nous récupérons le nombre des régions (global label), le nombre de pixels dans chaque région (voir figure 3.8).

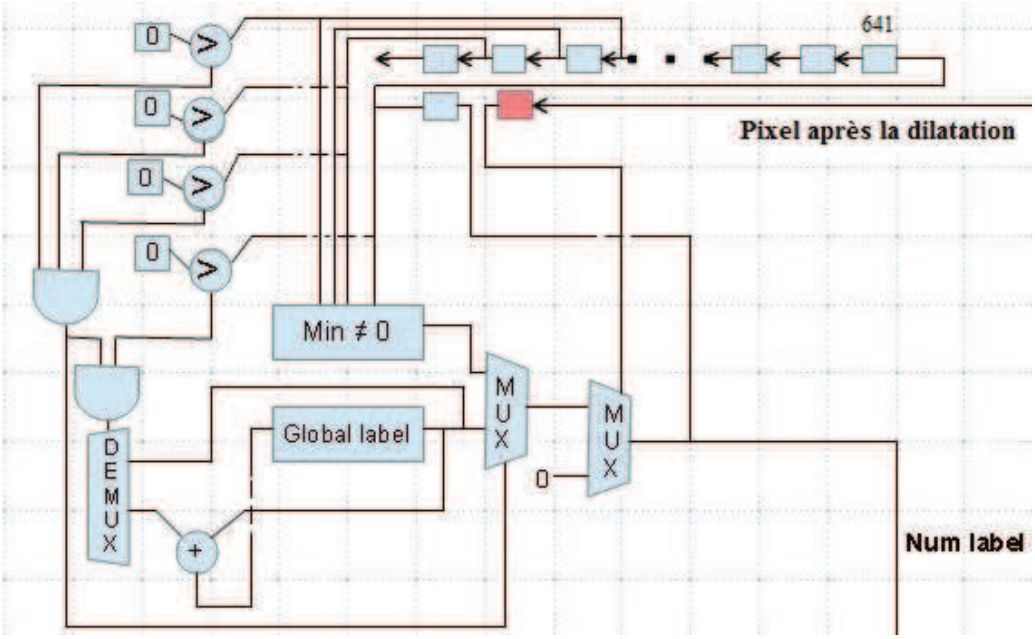


FIGURE 3.8 – Graphe flot de données de la fonction étiquetage

Résultats de la simulation

Après la simulation sur Modelsim, nous avons retrouvé les 5 régions étiquetées qui comportent entre 50 pixels pour la région la plus petite et 250 pixels pour la région la plus grande. Ces résultats sont similaires à ceux donnés avec l'algorithme en Matlab.

3.2.4 Calcul des coordonnées des centres de gravité

Architecture en VHDL

En parallèle avec l'étiquetage, un processus incrémente le nombre de lignes et de colonnes et stocke, pour chaque région, les coordonnées (X,Y) des pixels qui lui appartiennent (pixels ayant la même étiquette, voir figure3.9). En fin de traitement, après que nous parcourons toute l'image et que le nombre de colonnes atteint 640 et le nombre de lignes atteint 480, nous effectuons une division de la somme des abscisses des pixels par région par le nombre de pixels étiquetés de cette région pour

obtenir l'abscisse du centre de gravité, de même pour les ordonnées. Ce processus est répété pour chaque région (voir figure3.9).

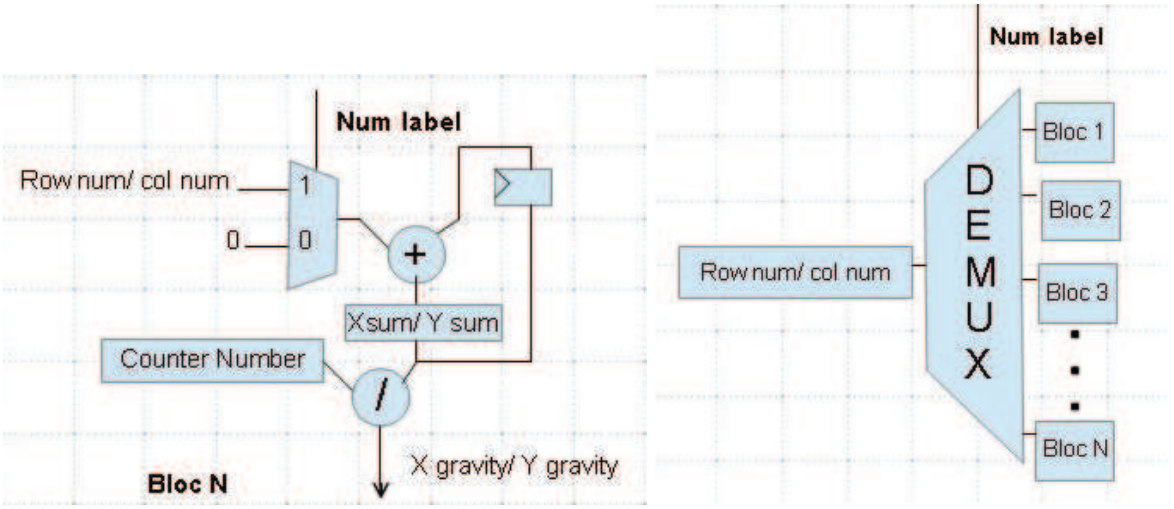


FIGURE 3.9 – A)-Bloc de calcul d'un centre de gravité d'une région B)-Calcul des coordonnées des centres de gravité pour chaque région

Résultats de la simulation

Pour la première image de la vidéo, nous avons obtenu avec cette architecture implémentée en VHDL les centres de gravité suivants :

Abscisses	271	288	321	351	383
Ordonnées	188	181	203	187	192

Comparaison entre les résultats de l'algorithme en Matlab et l'algorithme en VHDL

La figure 3.10 montre la différence entre les coordonnées des centres de gravité des formes détectées sur la ceinture avec l'algorithme implémenté en Matlab, comme nous l'avons détaillé dans le chapitre 2.1, et l'algorithme implémenté en VHDL avec l'architecture précédente. La différence entre les deux coordonnées ne dépasse pas un pixel en abscisse et un pixel en ordonné pour une image de 640*480 pixels. Cette différence est expliquée par :

- Les simplifications dans la formule de calcul de distance en VHDL (formule qui ne tiens pas compte de certains paramètres comme la luminance) par rapport à celle utilisée par l'environnement de simulation Matlab.
- Les simplifications dans l'algorithme de l'étiquetage en VHDL (test sur un voisinage limité) par rapport à l'algorithme utilisé par l'environnement de simulation Matlab.
- La division que nous avons utilisé en VHDL : division entière qui ne tient pas compte des chiffres après la virgule.

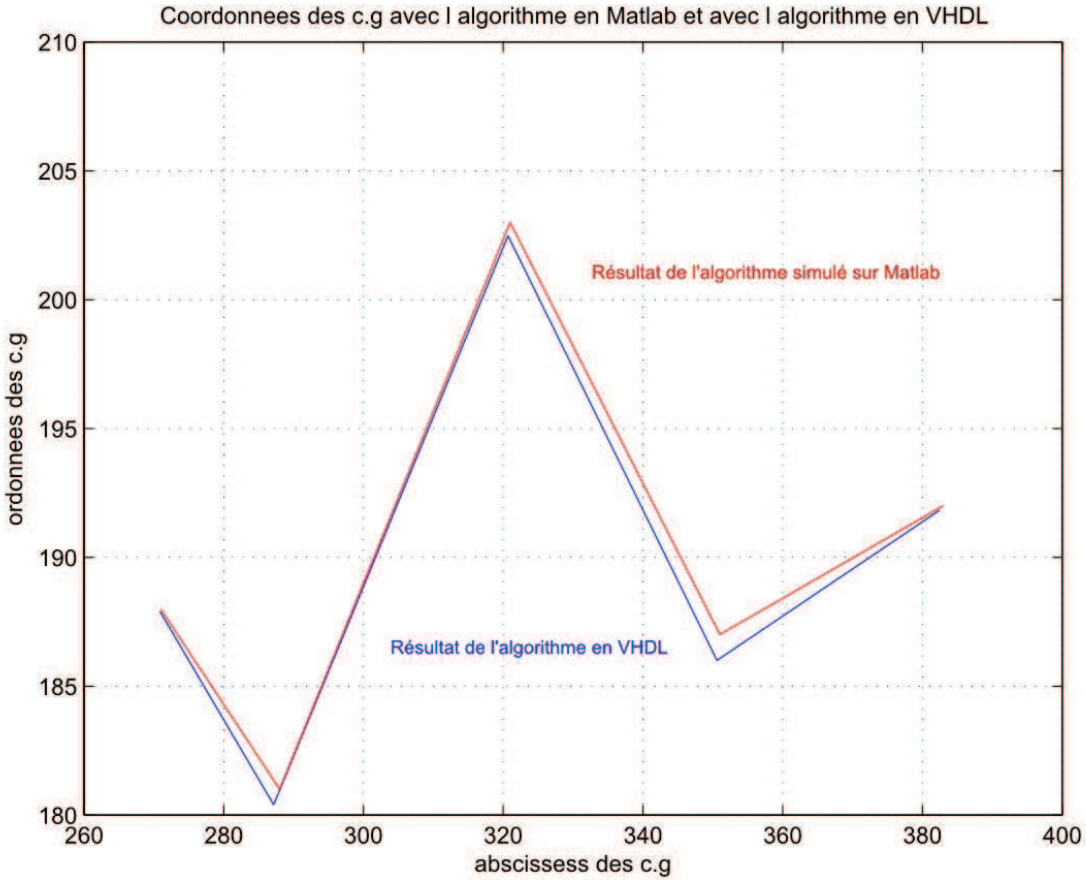


FIGURE 3.10 – comparaison des coordonnées des centres de gravité obtenus avec l'algorithme en Matlab et avec l'algorithme en VHDL

3.3 Synthèse de l'architecture

3.3.1 Synthèse de la première architecture

La première architecture complète est représentée sur la figure 3.11. Nous avons synthétisé notre première architecture à l'aide de l'outil de synthèse XST/ISE [2]de Xilinx [3]. Nous avons obtenu le résultat suivant :

Contraintes temporelles :

T_{clk}=7,77ns donc f=128MHz.

Cette fréquence est adéquate à notre contrainte temporelle fixée tout au départ car en simulation sur Modelsim nous obtenons aux alentours de 20ms/10ns de cycles d'horloge nécessaires pour une seule image. A une fréquence de 128MHz, nous aurons besoin de 15,54 ms pour traiter une image, ce qui correspond à un traitement de 64 images par seconde, largement supérieur ànotre cahier des charges initial.

2. Le logiciel ISE ® intègre la technologie de synthèse de Xilinx (XST), qui synthétise VHDL, Verilog, ou autres conceptions linguistiques mixtes pour créer des fichiers netlist Xilinx ® spécifiques

3. Xilinx est une entreprise américaine de semi-conducteurs. Inventeur du FPGA, Xilinx fait partie des plus grandes entreprises spécialisées dans le développement et la commercialisation de composants logiques programmables et des services associés

Occupation en surface sur FPGA :

Ressources	Occupation en pour cent
Slice register	0
Slice LUT	21
Flip Flop	91
Number of BUFG	9

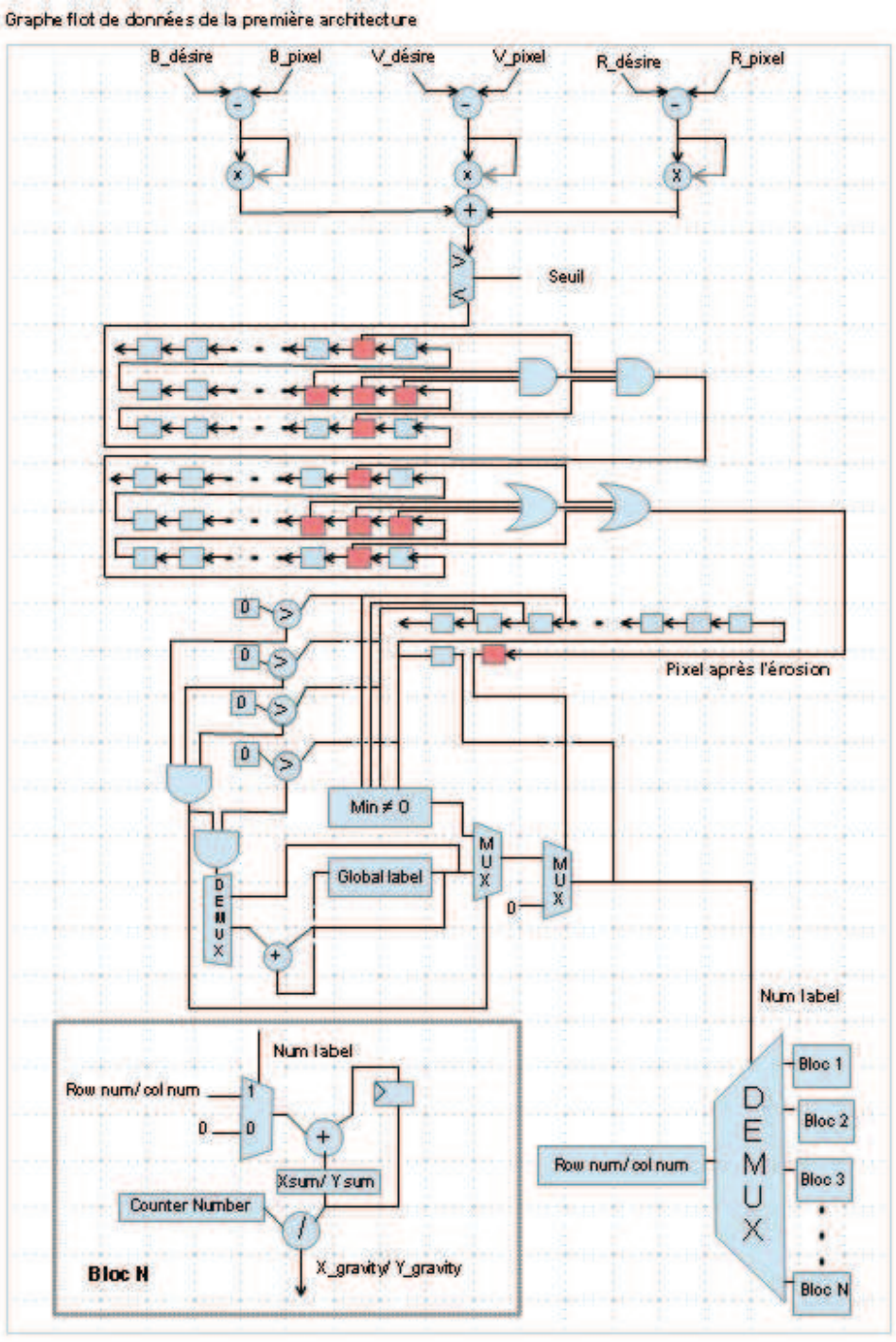


FIGURE 3.11 – Graphe flot de données de l'architecture complète

3.3.2 Proposition d'une deuxième architecture optimisée

La première architecture (voir figure 3.11) remplit largement le cahier des charges fixé au niveau temporel mais pas au niveau spatial. Afin de valider le système de détection de mouvement réspiratoire en 3D, il faudra implanter non seulement 2 fois l'architecture proposée, mais y ajouter un nouveau bloc effectuant la reconstruction en 3D des images 2D. Pour ce faire, nous proposons une nouvelle architecture se basant sur un nombre d'opérateurs inférieur à celui de l'architecture initiale. Cette architecture est présentée dans la figure 3.12

Pour cette raison, nous proposons ces optimisations :

Pour la première partie : la fonction de calcul de distance peut être décrite d'une manière séquentielle comme le montre (figure 3.12) pour réduire l'occupation en éléments logiques sur FPGA et pour réduire le temps de traitement. Nous avons utilisé dans cette description seulement un multiplieur, un additionneur/soustracteur et 2 multiplexeurs et une unité de commande. Le temps du chemin critique est ainsi donc réduit à la période d'un seul multiplieur et un multiplexeur.

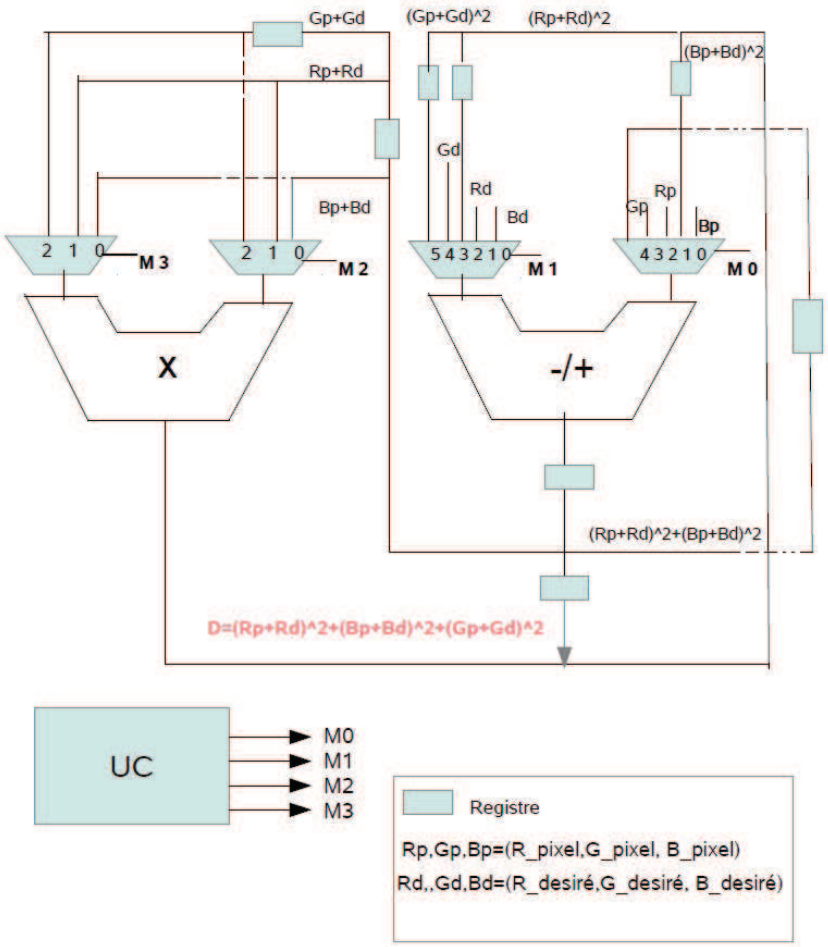


FIGURE 3.12 – Description structurelle de la fonction calcul de la distance

Pour le filtrage morphologique : nous pouvons utiliser un seul tableau pour effectuer l'érosion puis la dilatation au lieu de deux et cela en utilisant une machine d'état qui gère l'utilisation des opérateurs AND et OR. Une fois le tableau est rempli, le voisinage du pixel est testé avec l'opérateur AND pour faire l'érosion, l'entrée du pixel sortant au même tableau pour la dilatation

est contrôlée par la machine d'état à l'aide du signal M0 (voir figure3.13) .

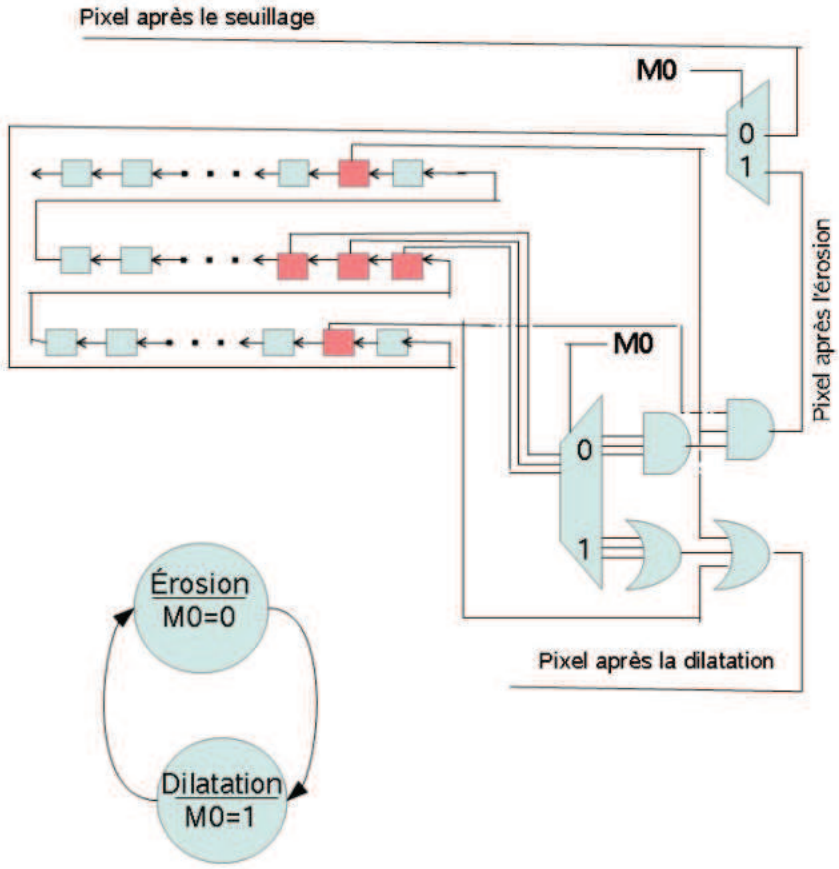


FIGURE 3.13 – Architecture optimisée du filtrage morphologique

De plus, une lecture d'image par bloc s'impose. De cette façon, au lieu d'utiliser des grands tableaux de taille d'une ligne, on serait amené à lire une image bloc par bloc et faire des traitements morphologiques sur les blocs choisis. Bien évidement, cette approche rend la logique de calcul un peu plus complexe, mais d'autre part, nous permet un gain conséquent en surface.

Conclusion générale

Le but de ce travail était de faire le suivi du mouvement respiratoire en IRM en temps réel via une plateforme de calcul reconfigurable de type FPGA.

Nous avons tout d'abord introduit le cadre de ce projet par la présentation de l'IRM et des problèmes d'artéfacts dus au mouvement de respiration.

Nous avons ensuite effectué une recherche dans la littérature sur les algorithmes de suivi du mouvement existants pour en choisir l'algorithme qui correspond le plus à notre application et qui est basé sur la reconnaissance des couleurs pour le traçage des trajectoires des centres de gravité des formes suivies au cours du temps.

Dans le deuxième chapitre de ce manuscrit, nous sommes passés à la validation comportementale de l'algorithme choisi dans un environnement de simulation de type Matlab.

La dernière partie de ce chapitre a été consacrée à la validation et l'impléméntation d'une architecture en VHDL sur une plateforme reconfigurable de type FPGA. Les résultats obtenus à l'aide d'une première architecture coïncident avec ceux donnés par l'environnement de simulation Matlab. La synthèse de cette architecture sur un FPGA Virtex7 a donné une fréquence élevée de l'ordre de 120MHz, permettant de traiter une vidéo d'au moins 50 images par seconde, ce qui répond aux exigences de notre cahier des charges initial.
Cependant, pour le traitement d'une vidéo nous avons besoin d'une occupation minimale de l'architecture sur FPGA. Pour ce faire, des réflexions sur des optimisations de la première architecture ont été introduites en fin du dernier chapitre.

Pour mettre en œuvre notre système de suivi de mouvement respiratoire en IRM, il reste à faire une étude de l'emplacement des caméras compatibles IRM ainsi que le développement d'un module, basé sur la stéréoscopie, permettant la visualisation du mouvement en 3D à partir de notre travail.

Bibliographie

[1] Development of a single 3-axis accelerometer sensor based wearable gesture recognition band. *Ubiquitous Intelligence and Computing*, 2007.

[2] Moca : A low-power, low-cost motion capture system based on integrated accelerometers. *Advances in Multimedia*, 2007.

[3] Donald G. Bailey Christopher T. Johnston, Kim T Gribbon. *FPGA based remote object tracking for real time control*. Wiley-Liss, Inc, A Wiley Company, 2005.

[4] R. V. Pound E. M. Purcell, H. C. Torrey. Resonance absorption by nuclear magnetic moments in a solid. *Physical Review*, 1946.

[5] Adam Postula Brian C. Lovell Ehsan Norouznezhad, Abbas Bigdeli. Object tracking on fpga-based smart cameras using local oriented energy and phase features. *University of Queensland, School of ITEE*, 2010.

[6] C. Jahnke et al. Coronary mr imaging : Breath-hold capability and patterns, coronary artery rest periods, and beta-blocker use. *Radiology*, 2006.

[7] T. Itoh et al. Development of new instrument for evaluating leg motions using acceleration sensors. *Environmental Health and Preventive Medicine*, 2007.

[8] Brice FERNANDEZ. *Techniques Adaptatives pour l'Imagerie par Résonance Magnétique des Organes en Mouvement*. PhD thesis, Nancy université, 2010.

[9] Benedice FILLER. *Techniques de respiration*. Psycho Textes, 2010.

[10] R. M. Botnar K. Nehrke P. Boesiger W. J. Manning et D. C. Peters Fischer, R. W. Analysis of residual coronary artery motion for breath hold and navigator approaches using real-time coronary mri. *Magnetic Resonance in Medicine 55, no. 3*, 2007.

[11] Christoph Forman, Murat Aksoy, Joachim Hornegger, and Roland Bammer. Self-encoded marker for optical prospective head motion correction in MRI. *Medical Image Analysis*, 15(5) :708–719, October 2011.

[12] C. Boesch J. Felblinger. Amplitude demodulation of the electrocardiogram signal (ecg) for respiration monitoring and compensation during mr examinations. *Magnetic Resonance in Medicine*, 1997.

[13] Hope W. Korin, Richard L. Ehman, Stephen J. Riederer, Joel P. Felmlee, and Roger C. Grimm. Respiratory kinematics of the upper abdominal organs : A quantitative study. *Magnetic Resonance in Medicine*, 23(1) :172–178, 1992.

[14] Philippe LAPLANTE. Exploratory testing for mission critical real-time and embedded systems. *IEEE Reliability Society Technical Operations Annual Technology Report for 2010*, 2010.

[15] Hugues TALBOT Laurent NAJMAN. *Morphologie Mathématique 1 : approches déterministes*. 2008.

[16] M. H. Levitt. *Spin dynamics : Basics of nuclear magnetic resonance*. 2008.

[17] library of congress cataloging, editor. *Computer and robot vision*. Addision Wisly Publishing company, 1998.

[18] Georges MATHERON. *Eléments pour une théorie des milieux poreux*. 1967.

[19] Juliette MATTIOLLI Michel SCHMITT. *Morphologie Mathématique*. 1993.

[20] Fredy ODILE. *Imagerie Adaptative en IRM : Utilisation des Informations de Mouvements Physiologiques pour l'Optimisation des Processus d'Acquisition et de Reconstruction*. PhD thesis, Nancy université, 2007.

[21] B. Lohezic M. Mandry D. Vuissoz P. A. Pietquin O.Felblinger J. Oster, J. Fernandez. Adaptive heart rate prediction for black-blood systolic imaging. In *17th Scientific Meeting, International Society for Magnetic Resonance in Medicine*.

[22] A. Maudsley P. Mansfield. Planar spin imaging by nmr. *Journal of Physics C : Solid State Physics*, 2010.

[23] Cedric PASQUIER. *Capteurs de mouvements dédiés à l'imagerie adaptative en IRM : Developpement technique et méthodologique*. PhD thesis, Nancy université, 2007.

[24] Laure ROUSSELET. *Integration des mpuvement respiratoire en imagerie adaptative*. PhD thesis, Nancy Université, 2011.

[25] Jovanovic.S Brosser. G Pasquier. C Felblinger. J Rousselet.L, Filipovic.M. 18th scientific meeting, international society for magnetic resonance in medicine. *MR compatible spirometer*, 2010.

[26] Rajan SEGHAL. *Video Image Enhancement and Object Tracking*. PhD thesis, Deemed University, 2007.

[27] Darko Zikica Ben Glockera Tobias Waggershauserb (MD) Marcus Pfisterc Selen Atasoya, Martin Grohera and Nassir Navaba. Real-time respiratory motion tracking : Roadmap correction for hepatic artery catheterization. *Computer Aided Medical Procedures (CAMP)*, 2008.

[28] Ryutaro Takazakura, Masashi Takahashi, Norihisa Nitta, and Kiyoshi Murata. Diaphragmatic motion in the sitting and supine positions : Healthy subject study using a vertically open magnetic resonance system. *Journal of magnetic resonance imaging : JMRI*, 19(5) :605–609, May 2004. PMID : 15112310.

[29] E. Verri Trucco. *Introductory Techniques for 3-D Computer Vision", Chapter 7*. 1998.

[30] Fernandez.B Lohezic. M Benhadid. A Mandry. D Felblinger. J Vuissoz P. A, Odille.F. High frame rate cardio-respiratory imaging using model-based reconstruction. *17th Scientific Meeting, International Society for Magnetic Resonance in Medicine*, 2009.

[31] O. L. Wade. Movements of the thoracic cage and diaphragm in respiration. *Journal of Physiology*, 1945.

[32] Qingqing WEI. Converting 2d to 3d : A survey. Technical report, Information and Communication Theory Group (ICT) Faculty of Electrical Engineering, Mathematics and Computer Science Delft University of Technology, the Netherlands, 2005.

[33] M Zaitsev, C Dold, G Sakas, J Hennig, and O Speck. Magnetic resonance imaging of freely moving objects : prospective real-time motion correction using an external optical motion tracking system. *NeuroImage*, 31(3) :1038–1050, July 2006. PMID : 16600642.

Annexe A

Caméra compatible IRM

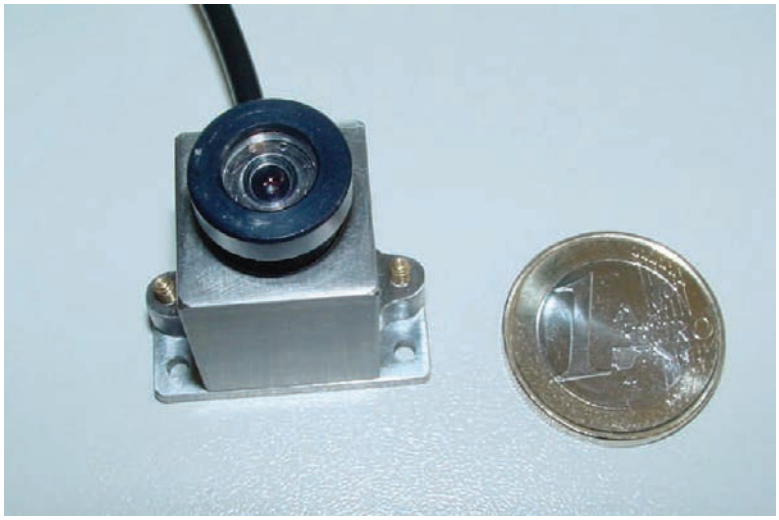


FIGURE A.1 – Caméra comptabile IRM choisi, fabricant MRC

Caractéristiques du FPGA Virtex7

Logic Cells	1,955K
Block RAM	68 Mb
DSP Slices	3,600
Peak DSP Performance	5,335 GMAC/s
Transceivers	96
Peak Transceiver Speed	5,335 GMAC/s
Peak Serial Bandwidth (Full Duplex)	28.05 Gb/s
PCIe Interface	2,784 Gb/s
Memory Interface	x8 Gen3
I/O Pins	1,200
I/O Voltage	1.2V, 1.35V, 1.5V, 1.8V, 2.5V, 3.3V

Carte V707

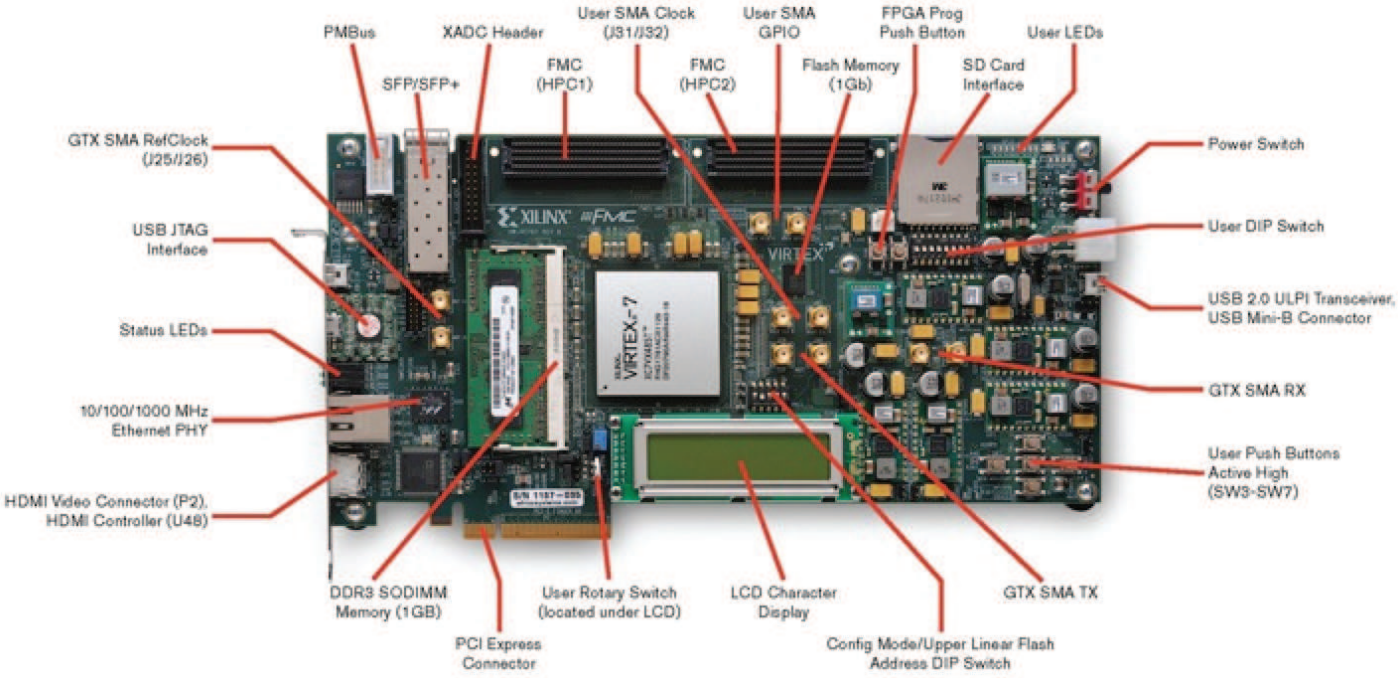


FIGURE A.2 – Carte V707

Ceinture en carton utilisée pour les vidéos test sur la plateforme mobile

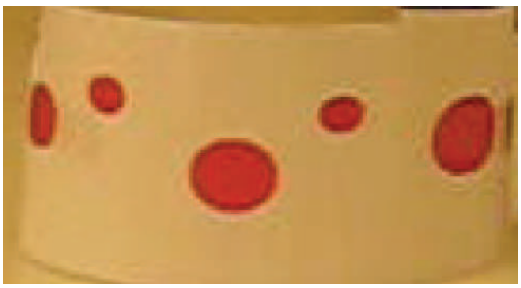


FIGURE A.3 – Ceinture en carton avec des formes rouges

Exemple de structure de FPGA :

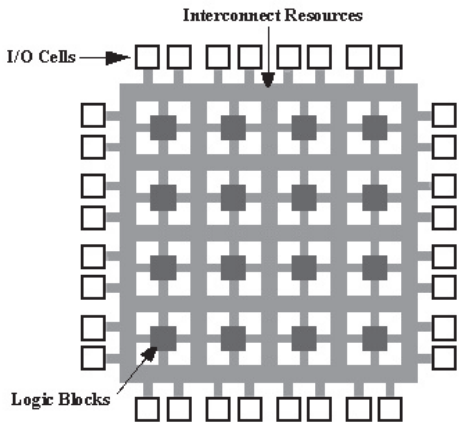


FIGURE A.4 – Structure interne d'un FPGA

```
% Ce programme permet de faire le suivi des tâches d'une même
couleur
% Etapes:lecture des images,
% soustraction de l'image et de la couleur à suivre,
% calcul de la distance pour chaque pixel, seuillage(threshold),
%filtrage(dilatation et erosion),
%labelisation des regions
% traçage du Mvt des centroides des régions au cours du temps (2D et
3D)


clear all
close all

sequence_path='/home/sjov/maroua_stage/maroua_essais/mesfonctions/vi
deobmp/';
extention='bmp'
[sequence_path '*.' extention]
files = dir([sequence_path '*.' extention]);
files = sort({files.name});

% Pour trier les images
 imageStrings = regexp([files{:}],'match');
 imageNumbers = str2double(imageStrings);
 disp(imageNames(1:10));

 imageStrings = regexp([files{:}],'(\d*)','match');
 imageNumbers = str2double(imageStrings);

 [~,sortedIndices] = sort(imageNumbers);
 sortedImageNames = imageNames(sortedIndices);

disp(sortedImageNames(1:10));

%Initialisation
sizes =size(files,2);
nbr_taches=5;
sizes=30;
 U=zeros((nbr_taches-1));
 V=zeros((nbr_taches-1));
 T=zeros((nbr_taches-1));
 X= zeros((nbr_taches-1),sizes);
 Y= zeros((nbr_taches-1),sizes);
 Z= zeros((nbr_taches-1),sizes);

 %Lecture de la video
    fprintf(1, 'Reading file %d: %s\r', count, files{count});
    respiration = imread([sequence_path files{count}]);
    figure('Name','Initial
image','NumberTitle','off'),imshow(respiration);
```

```
    %Initialisation
    black = zeros(size(respiration),'uint8');
    desired_image = black;
    heigth =size(respiration,1);
    width =size(respiration,2);

   %Introduction du (RGB) de la couleur à detecter
    for i= 1:heigth
        for j = 1:width
            desired_image(i,j,1)=180;
            desired_image(i,j,2)=47;
            desired_image(i,j,3)=8;
        end
    end

    figure('Name','Desired
image','NumberTitle','off'),imshow(desired_image);
    pause (0.1);

    %Soustraction de l'image desirée par l'image initiale
     diff_resp=imabsdiff(respiration,desired_image);
     figure('Name','Difference between initial and red
image','NumberTitle','off'),imshow(diff_resp);


    %calcul de la distance pour chaque pixel de l'image
     distance = zeros(heigth,width);
    for i= 1:heigth
        for j = 1:width

distance(i,j)=(uint32(diff_resp(i,j,1)))^2+(uint32(diff_resp(i,j,2))
)^2+(uint32(diff_resp(i,j,3)))^2;
        end
    end


    %application du seuillage pour chaque pixel
    for i= 1:heigth
        for j = 1:width
                if distance(i,j) > 800;
                   distance(i,j)=0;
                else
                   distance(i,j)=1;
                end
        end
    end

 %Recherche des pixels de l'image qui ont une valeur non nulle

    [I,J]=find(distance);
    size_index=size(I,1);
    final_image= zeros(size(respiration),'uint8'
```

```
%distance
    for i= 1:size_index
        final_image(I(i),J(i),1) = 255;
        final_image(I(i),J(i),2) = 255;
        final_image(I(i),J(i),3) = 255;
    end
figure('Name','Final image','NumberTitle','off'),imshow(final_image)
     %Element structurel de forme disque (1,1)
     se = strel('disk',1);
%Filtrage morphologique : Erosion puis dilatation
I2 = imerode(final_image,se);
I2 = imdilate(I2,se);
figure('Name','Final image after morphological
filtering','NumberTitle','off'),imshow(I2), title('Filtered image');
%Black image
region_image=black;
for i= 1:heigth
    for j = 1:width
        region_image(i,j,1)=0;
        region_image(i,j,2)=0;
        region_image(i,j,3)=0;
    end
end
%Etiquetage
    bw = bwlabeln(I2, 8);
%Traçage des centroîdes des régions
  for i=1:5
     [r,c]=find(bw==i);
            for j=1:size(r,1)
                    region_image(r(j),c(j),1) = 255;
                    region_image(r(j),c(j),2) = 255;
                    region_image(r(j),c(j),3) = 255;
           end
      stats = regionprops(bw, 'Centroid');
      bc = [stats(i).Centroid];
       Z(count,i)=count;
       X(count,i)=bc(1);
       Y(count,i)=bc(2);
       % Tracing 2D
       T(i)=Z(count,i);
       U(i)=X(count,i);
       V(i)=Y(count,i);

  end
%Traçage du mouvement en trois dimensions
  plot3(U,V,T);%axis ([0 640 0 480]);
hold on
   plot3(X,Y,Z,'lineWidth',2,'MarkerFaceColor','r'),axis ij;

   pause(.5)
end
```

Résumé

L'unité INSERM, IADI dans lequel ce stage s'est déroulé, a pour thématique principale l'imagerie des organes en mouvement.
Un des mouvements les plus difficiles à contrôler, lors de l'acquisition des images, est le mouvement respiratoire car il nécessite souvent des apnées qui ne sont pas toujours réalisables. Le but de ce stage est d'appliquer une méthode de suivi de mouvements respiratoires à base d'une caméra associée à des structures de calcul puissantes permettant d'extraire, à partir des séquences vidéo acquises, le mouvement respiratoire voulu.
Le travail qui a été réalisé pendant ce stage consistait à choisir l'algorithme de suivi de mouvement, le valider au niveau comportemental sous un environnement de simulation de type MATLAB puis l'implanter sous une plateforme reconfigurable de type FPGA.

Mots-clefs
IRM, mouvement respiratoire, suivi des formes, traitement temps réel, Matlab, VHDL et FPGA.

Abstract

The main field of the laboratory IADI in which the training was carried out, is imaging of moving organs. One of the most difficult movements to control, during image acquisition, is the respiratory motion because it requires breath hold method which are not always feasible.
The purpose of this study was to apply an object tracking method for respiratory motion detection in MRI environment. The used method is based on system composed of camera connected to a reconfigurable FPGA plateform whose main purpose is extracted of respiratory motion from acquired video frames.
The main objectives of this study were to choose a motion tracking algorithme adapted for respiratory motion detection in MR environment validate it in a behavioral evironment such as Matlab and to finaly implement it on a real-time reconfigurable FPGA plateform.

Key words
IRM, respiratory object, object tracking, real-time processing , VHDL and FPGA.