

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

UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE  
DEPARTEMENT D'AUTOMATIQUE

**Mémoire de Fin d'Etudes  
De MASTER ACADEMIQUE**  
Domaine : **Sciences et Technologies**  
Filière : **Génie électrique**  
Spécialité : **Commande des systèmes**

*Présenté par*  
**ABADA Amar**  
**OUACHEM Rabah**

Thème

**Segmentation d'images de la  
Kinect**

**M<sup>ed</sup> Ouardia CHILLALI**  
M.A.A, UMMTO, Présidente

**M Kamal HAMMOUCHE**  
Professeur, UMMTO, Encadreur

**M<sup>ed</sup> Ouiza ADJMOUT**  
M.A.A, UMMTO, Co-Encadreur

**M Takfarinasse CHELLI**  
M.A.A, UMMTO, Examineur

**M<sup>ed</sup> Zouhra HALICHE**  
M.A.A, UMMTO, Examineur

2015 /2016

# ***Remerciement***

*En premier lieu, nous tenons à remercier « Allah » qui nous a procuré ce succès. Nous tenons à exprimer nos vifs remerciements et nos sincères reconnaissances à*

***Mr HAMMOUCHE Kamal*** professeur à l'UMMTO pour sa disponibilité, ses conseils judicieux, ses directives et ses orientations concernant notre projet de fin d'étude, tout en nous accordant sa confiance et en nous faisant profiter de sa large expérience tout au long de la réalisation de ce projet de fin d'étude.

Nous ne manquerons pas de remercier Madame « ADJEMOUT Ouiza », MAA à l'UMMTO *pour ses aides, ses encouragements, et ses critiques constructifs qui nous avons beaucoup aidé à apprécier ce travail et à mieux éclairer nos perspectives, et particulièrement pour la confiance qu'elle nous a fait.*

Que les membres du jury trouvent ici nos vifs remerciements pour avoir accepté de juger notre travail.

*On tient à remercier également nos familles et nos ami(e)s, ce travail n'aurait certainement jamais vu le jour sans leurs aides, et leurs soutiens, je tiens vivement à les remercier.*

# ***DÉDICACES***

*À MES CHERS PARENTS POUR LEURS SOUTIENS MORAL ET FINANCIER DURANT MES ÉTUDES, À MES FRÈRES, SŒURS ET LEURS ENFANTS, ET À LA MÉMOIRE DE MES GRANDS PARENTS.*

*AINSI À TOUS MES FIDÈLES AMIS(ES), À TOUS CEUX QUE J'AIME, ET À TOUS CEUX QUI ME SONT CHERS.*

**A.AMAR**

# ***DÉDICACES***

*À MES CHERS PARENTS POUR LEURS SOUTIENS MORAL ET FINANCIER DURANT TOUS MES ÉTUDES, À MON FRÈRES, SA FEMME ET SA FILLE ET SUR TOUT MA PETITE SŒURS, ET À LA MEMOIRE DE MES GRANDS PARENTS.*

*AINSI À TOUS MES FIDÈLES AMIS(ES), À TOUS CEUX QUE J'AIME, ET À TOUS CEUX QUI ME SONT CHERS.*

**O.RABAH**

Introduction générale.....	01
----------------------------	----

## Chapitre I : La caméra Kinect

I.1 Introduction.....	03
I.2 Présentation de la Kinect.....	04
I.2.1 Système audio.....	05
I.2.2 Capteur couleur RGB.....	05
I.2.3 Capteur Infrarouge associé à un émetteur détectant la profondeur.....	05
I.2.4 LED.....	06
I.2.5 Champ de vision.....	06
I.3 La mesure de profondeur (Depth).....	07
I.4 Principe de fonctionnement.....	09
I.4.1 Caméra de profondeur.....	09
I.5 Les pilotes de la caméra Kinect.....	11
I.5.1 CLNUI.....	11
I.5.2 Open Ni.....	11
I.6 Applications de la kinect en vision par ordinateur.....	12
I.7 Conclusion.....	13

## Chapitre II : Segmentation d'image

II.1 Introduction.....	14
II.2 Définition de la segmentation.....	15

II.3 Méthodes de la segmentation d'image.....	15
II.3.1 Approche région.....	16
II.3.1.1 Classification par les Réseaux de neurones.....	17
II.3.1.2 Classification par K-Means(K-Moyens).....	18
II.3.1.3 C-moyennes floues.....	20
II.3.1.4 Croissance de régions.....	21
II.3.1.5 Champs Aléatoires de Markov.....	21
II.3.2 Approche contour.....	22
II.3.2.1 Méthodes dérivatifs.....	23
II.3.2.2 Espace –échelle.....	23
II.3.3 Approches Structurelles.....	24
II.3.3.1 Opérateurs Morphologique.....	24
II.3.3.2 Ligne de Partage des Eaux.....	25
II.3.4 Approche forme.....	25
II.3.4.1 Contours Actifs.....	26
II.3.4.2 Modèle actif de forme.....	27
II.3.4.3 Modèle actif d'apparence (AAM).....	28
II.3.5 Théorie des Graphes.....	30
I.3.5.1 Graph Cuts.....	30
II.3.5.1.1 Définition d'un Graph Cuts.....	30
II.3.5.1.2 Poids minimale.....	31

II.4 Conclusion.....	32
----------------------	----

## Chapitre III : Segmentation d'images de la Kinect

III.1 Introduction .....	33
III.1 Description la méthode de segmentation .....	34
III.1.1 Extraction des attributs.....	34
III.1.1.1 Matrice de cooccurrence pour des images couleur.....	35
III.1.1.1.1 L'énergie ou moment angulaire d'ordre deux est donné par la relation.....	36
III.1.1.1.2 Le contraste ou (l'inertie) est défini par la relation.....	36
III.1.1.1.3 La corrélation est donnée par la relation.....	36
III.1.1.1.4 L'homogénéité donnée par la relation.....	36
III.1.2 L'algorithme de classification.....	37
III.2 Testes et résultats et interprétations .....	39
III.3 Conclusion.....	42
Conclusion générale .....	43

Figure I.1 :	Dispositif Kinect.....	04
Figure I.2 :	Composante de la Kinect.....	06
Figure I.3:	La plage de visualisation de la Kinect.....	07
Figure I.4:	Principe de fonctionnement de la Kinect.....	08
Figure I.5:	Image profondeur et image couleur fournie par la Kinect.....	08
Figure I.6:	Le diagramme de fonctionnement de la Kinect.....	10
Figure I.7:	Mire émise par la Kinect telle que perçue par la caméra infrarouge.....	10
Figure I.8:	Modules Open NI.....	11
Figure I.9:	Image de profondeur et couleur.....	12
Figure I.10:	Articulations détectées par NITE. Affichées sur la carte de profondeur et sur couleurs.....	12
Figure II.1:	Classification des différentes méthodes de segmentation.....	16
Figure II.2:	Structure d'un neurone artificiel.....	17
Figure II.3:	Le processus de l'algorithme K-means.....	18
Figure II.4:	Quelques modèles de contours.....	22
Figure II.5:	Operateurs dérivatifs.....	23
Figure II.6:	Etiquetage (manuel) des formes par N points.....	27
Figure II.7:	Exemple de Modèle Actif de Forme.....	28
Figure II.8:	La méthode AAM incluant à la fois la forme et la texture.....	29
Figure II.9:	Coupe d'un graphe. La ligne pointillée sépare le graphe en deux ensembles disjointe. ....	31

## Liste des figures

---

Figure II.10	Coupe minimale du graphe. Ici le poids de la coupe minimale est $w=6$ ...	31
Figure III.1	Image de la distance 'd' .....	35
Figure III.2	Exemple de classification par la méthode de K-means.....	41

## Introduction Générale

L'objectif de ce projet est de réaliser une application permettant d'acquérir des données image en utilisant des capteurs intégrés sur la Kinect. Ce dispositif est un périphérique développé par Microsoft, qui est destiné à la console de jeux vidéo Xbox 360. Il contient principalement deux caméras. La première fournit une image couleur et la seconde une image de profondeur. Grâce à ces deux types d'images, plusieurs applications de traitement d'images ont été développées à base de la Kinect.

Dans ce mémoire, nous proposons de segmenter une image à partir des informations fournies par la Kinect à savoir l'information couleur et celle de profondeur.

La segmentation d'images est une étape incontournable dans tout processus d'analyse d'image. C'est un traitement de bas niveau qui précède l'étape de mesure, de compréhension et de décision.

Elle a pour but de déterminer les régions d'une image cohérentes, au sens d'un critère fixé a priori. De nombreux critères de segmentation existent, suivant le domaine d'application et le type d'images traitées, un critère donné prendra en compte le niveau de gris, la texture, la couleur, la profondeur ou le mouvement. Plusieurs approches de segmentation sont apparues depuis quelques années. Certaines d'entre elles cherchent à délimiter les régions homogènes par leurs contours (approche contour) alors que d'autres cherchent à retrouver les régions homogènes (approche région).

Dans ce mémoire, nous avons choisi de classer les pixels de l'image à segmenter par l'intermédiaire de la technique de classification K-means et en tenant compte des attributs couleur et profondeur extraits à partir de la caméra Kinect.

Ce mémoire comprend principalement trois chapitres.

Le premier chapitre est consacré à la présentation de la caméra Kinect, ses caractéristiques techniques et les pilotes de développement nécessaires pour son exploitation.

Le second est dédié à la segmentation d'image en générale, sa définition et les principales méthodes de segmentation sont brièvement présentés.

Le troisième chapitre est consacré à la description de la méthode de segmentation développée. Les résultats obtenus y sont également présentés.

# *Chapitre I*

## *La caméra kinect*

### **I.1.Introduction**

Le domaine de la vision 3D évolue de manière exponentiel, notamment grâce à son côté attractif et l'apparition de nouvelles technologies. Dans la vision tridimensionnelle, l'utilisation des caméras 3D est très appréciée pour des applications dans plusieurs domaines tel que, la robotique, l'architecture, l'industrie, la vision par ordinateur, ...etc.

Dans notre travail, on s'intéresse a un nouveau capteur proposé par Microsoft nommé « Kinect ». C'est un dispositif principalement utilisé pour la détection de mouvement et qui a été initialement développé pour la console de jeu Xbox 360 par Microsoft. Une kinect version Windows est sortie plus tard. La description de la kinect est donnée dans ce chapitre.

## I.2 Présentation de la Kinect

La Kinect, initialement connu sous le nom de code « Project Natal », a été conçue par PrimeSense, une société Israélienne indépendante de Microsoft.

L'ensemble est installé sur un socle motorisé permettant de faire pivoter les capteurs optiques et le projecteur infrarouge afin d'adapter le champ de vision du périphérique en fonction du positionnement des utilisateurs.

Début 2012, Microsoft commercialise une version Kinect pour Windows, le capteur de profondeur ayant été modifié pour pouvoir fonctionner à une distance plus proche. De plus, Microsoft a publié un SDK facilitant la programmation du capteur sous Windows 7 et 8, ceci permettant d'ouvrir ses applications pour un coût modique. Des pilotes libres (non Microsoft) existent aussi, ce qui permet d'utiliser la Kinect pour d'autres applications que celles prévues par Microsoft (traitement d'image, le suivi d'objets,...etc.).

La Kinect présente une évolution ingénieuse dans l'univers des caméras 3D. Ce périphérique de Microsoft se forme principalement d'une barre horizontale constituant l'élément principale de la technologie Kinect et contenant une caméra RGB fournissant des images couleur et un capteur de profondeur pour détecter la profondeur de chaque pixel de l'image. A partir de ces deux images, la reconstruction 3D de la scène est rendu possible (Figure I.1). La Kinect contient aussi un petit moteur pour déplacer l'appareil, un accéléromètre 3 axes afin d'améliorer la précision du moteur et un petit ventilateur.



**Figure I.1 : Dispositif Kinect**

### I.2.1 Système audio

1. Chat vocal Xbox Live et chat vocal dans les jeux vidéo.
2. Suppression de l'écho.
3. Reconnaissance vocale multilingue.

16 bits à 16 KHz grâce aux 4 microphones (cf. figure I.2).

### I.2.2 Capteur couleur RGB

Le capteur de couleur RGB consiste en 3 photodiodes silicium sensibles chacune à une couleur (rouge, verte ou bleue) (figure I.2). Il est utilisé pour les balances de couleurs, pour les corrections de couleurs des afficheurs, le contrôle d'impression, etc.

#### Longueur d'onde:

- Rouge: 630 nm
- Vert: 560 nm
- Bleu: 470 nm.

IL:

- Rouge: 0,35  $\mu$ A
- Vert: 0,05  $\mu$ A
- Bleu: 0,30  $\mu$ A

Vf: 0,5 V

640 x 480 en couleur 32 bits à 30 images par seconde pour le capteur couleur

### I.2.3 Capteur Infrarouge associé à un émetteur détectant la profondeur

Le capteur infrarouge est constitué d'un récepteur qui détecte l'intensité lumineuse dans la gamme des lumières infrarouge et d'un émetteur de lumière infrarouge (figure I.2).

Ce capteur peut être utilisé comme capteur de contact. On fait une mesure avec la LED infrarouge éteinte et une avec la LED infrarouge allumée. S'il n'y a aucun obstacle proche, la valeur lue est la même. Sinon, l'obstacle aura réfléchi la lumière infrarouge et la deuxième mesurée donnera un résultat plus élevé.

320 x 240 en couleur 16 bits à 30 images par seconde pour le capteur infrarouge.

### I.2.4 LED

Un voyant est placé entre la caméra et le projecteur infrarouge. Il est utilisé pour indiquer l'état de l'appareil Kinect. La couleur verte de la LED indique que les pilotes de périphériques Kinect sont chargés correctement. Si vous branchez la Kinect dans un ordinateur, la LED va commencer avec une lumière verte une fois que votre système détecte le périphérique.

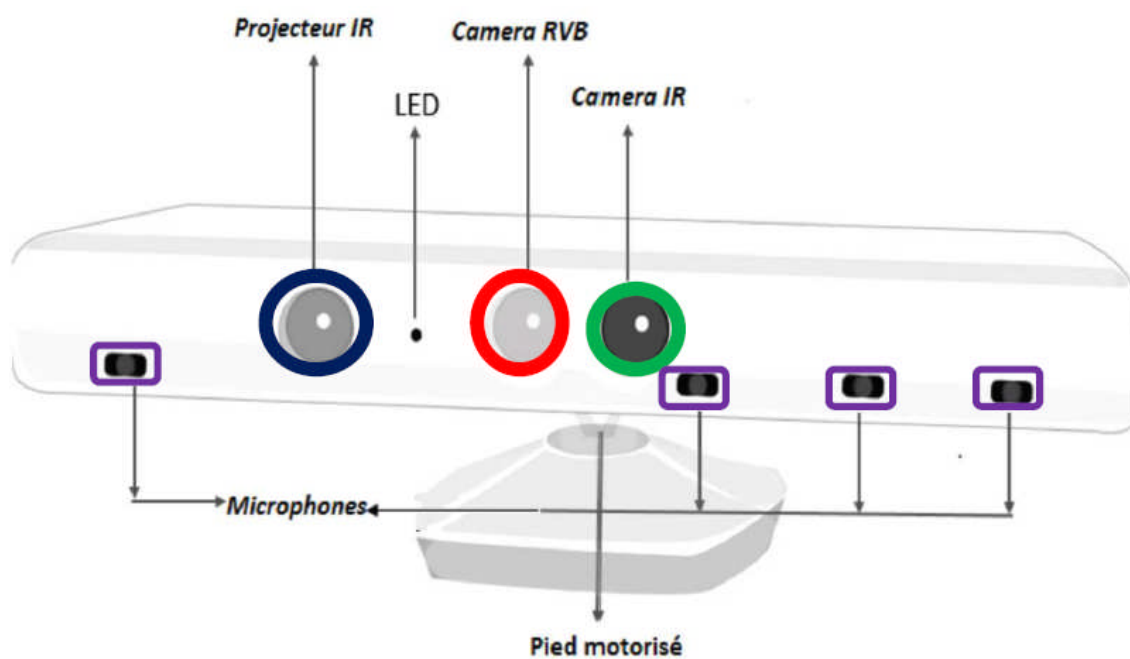


Figure I. 2 : Composante de la Kinect.

### I.2.5 Champ de vision

1. Champ de vision horizontal: 57 degrés.
2. Champ de vision vertical : 43 degrés.
3. Marge de déplacement du capteur (motorisation): +/- 27 degrés grâce au socle (figure I.3).
4. Portée du capteur : 1,2 m – 3,5 m (à partir de 50 cm pour la version Kinect for Windows).

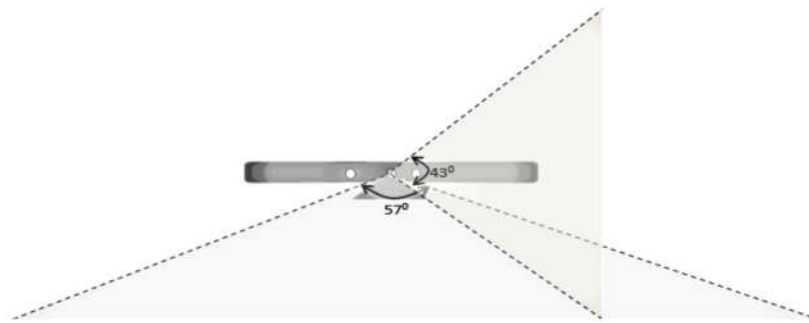


Figure I. 3 : La plage de visualisation de la Kinect.

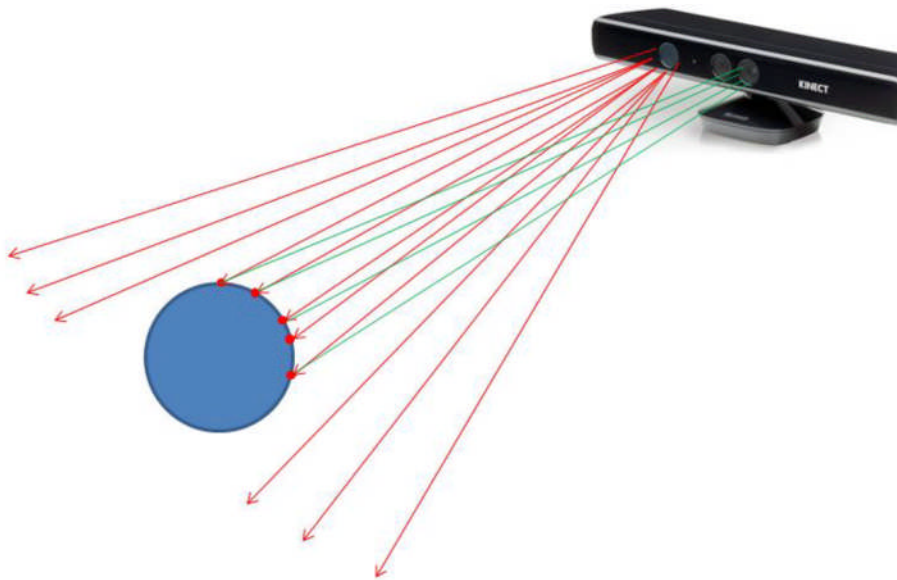
### I.3 La mesure de profondeur (Depth)

La mesure de la profondeur se fait par un processus de triangulation. L'émetteur laser émet un faisceau unique est subdivisé en faisceaux multiples par un réseau de diffraction afin de créer un modèle constant de mouchetures projetées sur la scène (figure I.4). Cette tendance est capturée par la caméra infrarouge et est comparée avec un modèle de référence.

Le motif de référence est obtenu en capturant un plan à une distance connue du capteur, et est stocké dans la mémoire de la Kinect. Cette configuration de la Kinect est réalisée en usine avant la commercialisation. Quand la nuée de points est projetée sur un objet dont la distance au capteur est plus petite ou plus grande que celui du plan de référence alors la nuée de points dans l'image infrarouge est décalée dans la direction de la ligne de base entre le laser projecteur et le centre optique (centre focal, point  $f$ ) de la caméra infrarouge. Ces changements sont mesurés pour toutes les taches par une simple procédure de corrélation, ce qui donne une image de disparité. Pour chaque pixel la distance de la sonde peut alors être extraite de la disparité correspondante.

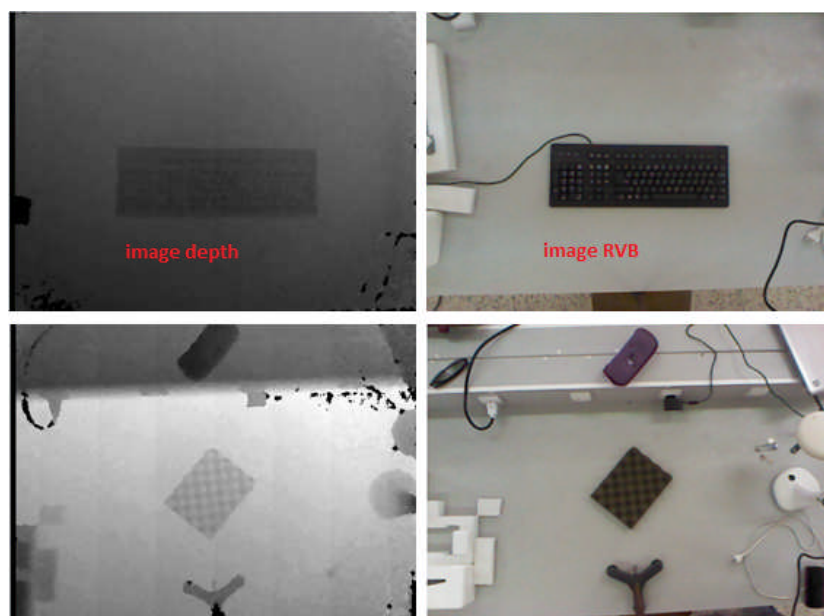
Pour mesurer la profondeur la Kinect utilise la lumière structurée, le principe de lumière structurée permet d'obtenir l'information de profondeur d'objets présents dans une scène ne

projetant un motif lumineux sur les objets et en analysant la déformation de ce motif au contact des objets. Cette lumière structurée se présente sous forme d'un ensemble de rayons calibrés, Ces rayons calibrés évitent le besoin d'une seconde caméra tels qu'illustré dans la (Figure I.4). Exemple de nos acquisition couleur et profondeur sont illustrés dans la (Figure I.5).



**Figure I.4 : Principe de fonctionnement de la Kinect.**

Kinect utilise 11 bits pour représenter les valeurs de profondeur, ce qui limite la profondeur



**Figure I.5 : Image profondeur et image couleur fournie par la Kinect.**

## I.4 Principe de fonctionnement

### I.4.1 Caméra de profondeur

La Kinect est une caméra 3D active de type stéréoscopique, elle comprend deux parties :

- une source de lumière infrarouge « structurée ».
- une caméra infrarouge.

La méthode se base sur les mêmes principes géométriques que ceux utilisés pour la stéréoscopie. En stéréoscopie, on exploite le fait que plus un objet central est proche des deux caméras, plus l'objet est décalé vers la droite dans l'image de gauche et vers la gauche dans l'image de droite. De même, pour un objet distant, le décalage est plus faible. Cependant, grâce à sa propre source lumineuse, la Kinect lève les indéterminations inhérentes à la stéréoscopie dans les zones dépourvues de texture la figure I.6.

En effet, la lumière émise par la Kinect permet de donner une texture facilement identifiable à toute la scène filmée. La figure I.7 montre un exemple de la mire, d'allure aléatoire, projetée en infrarouge (non visible) par la Kinect. Elle dispose également d'une caméra couleur, en plus du dispositif stéréoscopique de vision 3D. Pour chaque pixel de l'image, elle fournit ainsi 4 informations : les 3 composantes de couleur et la profondeur (sauf aux bords de l'image). D'un point de vue technique, la Kinect produit des images ayant une résolution de  $640 \times 480$  pixels.

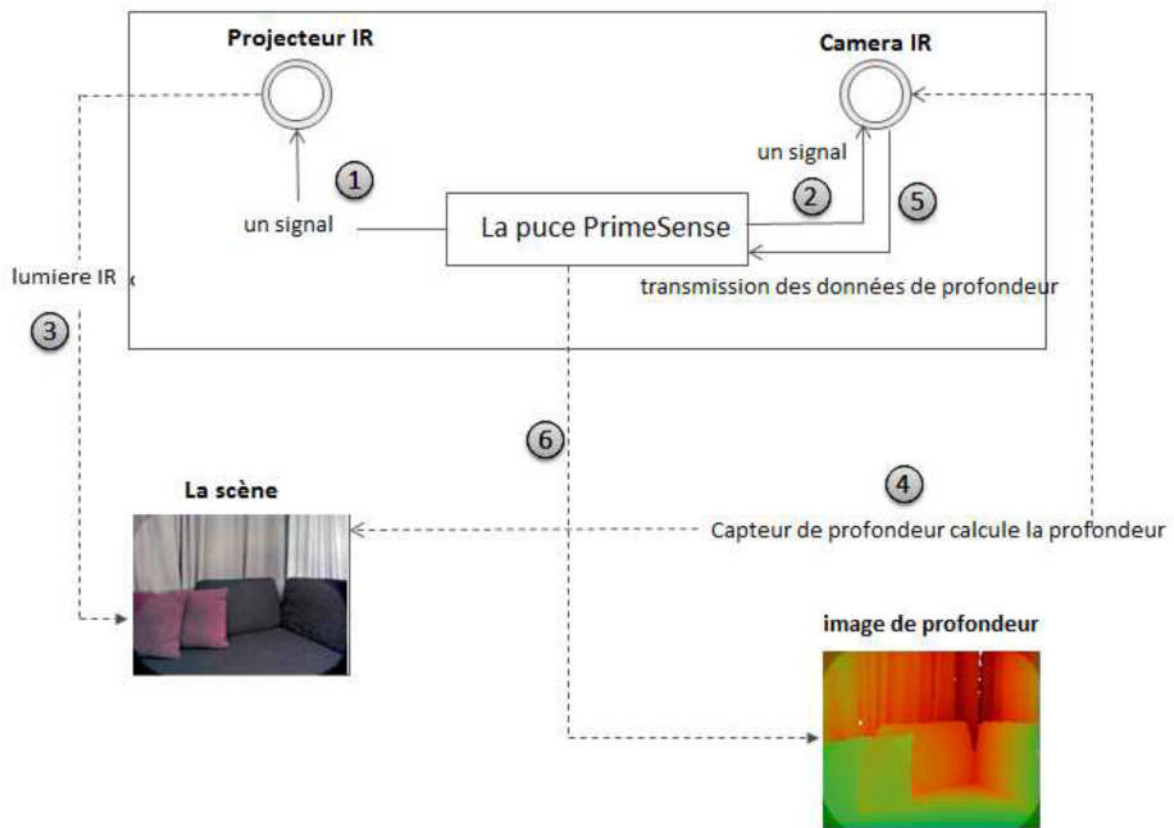


Figure I.6 : Le diagramme de fonctionnement de la Kinect.

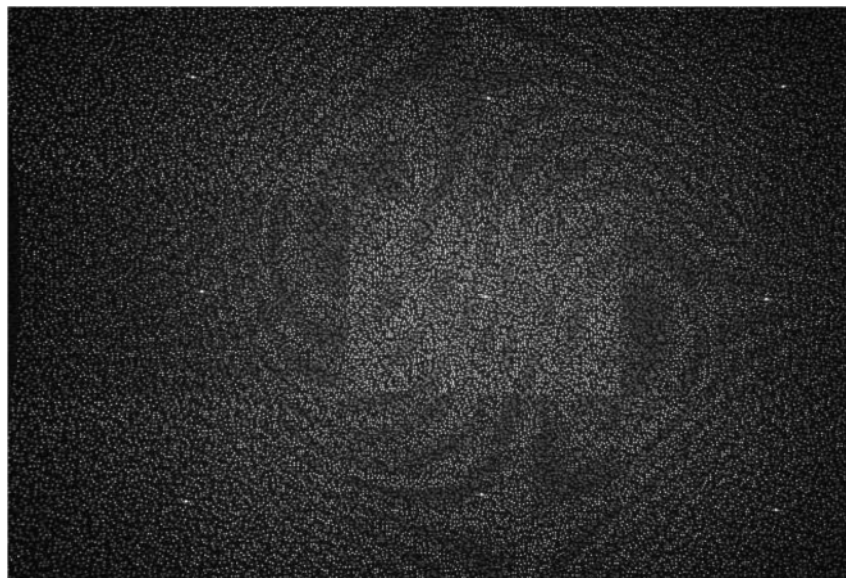


Figure I.7 : Mire émise par la Kinect telle que perçue par la caméra infrarouge.

## I.5 Les pilotes de la caméra Kinect

### I.5.1 CLNUI

CLNUI est une plate-forme fournie par Code Laboratory. Les utilisations possibles sont nombreuses, comme par exemple la robotique, la surveillance, la capture de mouvement, le suivi de personne ou d'objet ou encore la numérisation 3D. CLNUI propose aussi une application pour la caméra KINECT qui affiche l'image couleur et l'image de profondeur, en offrant la possibilité d'incliner la caméra et de faire clignoter sa LED.

### I.5.2 Open Ni

Le Cadre **Open NI** est une couche abstraite qui fournit l'interface pour des périphériques physiques et composants middleware. L'API (*Application Programming Interface*) permet à de multiples composants d'être enregistrés dans le cadre Open NI. Ces composants sont appelés modules, et sont utilisés pour produire et traiter les données sensorielles.

Les modules compatibles avec Open Ni sont les suivants :

1. Capteurs 3D
2. Caméra RGB
3. Caméra infrarouge
4. Matériel audio

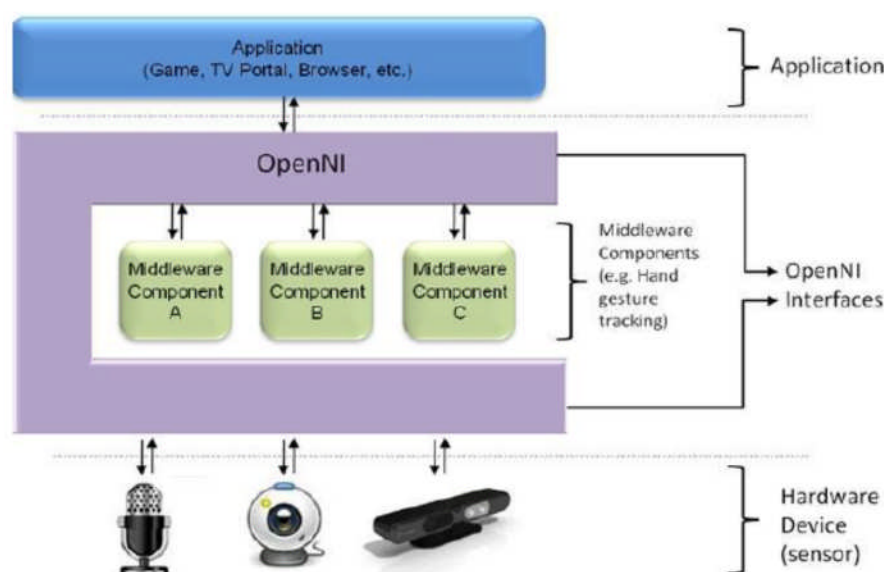


Figure I.8 : Modules Open NI

## I.6 Applications de la kinect en vision par ordinateur

Avec l'invention et le faible coût de la camera Kinect de Microsoft, qui offre une carte de profondeur à haute résolution et une image couleur (RGB), son utilisation est devenu généralisée et cela ouvre un grand nombre de nouveaux travaux de recherche et de développement.

La nature complémentaire de la profondeur et de l'information visuelle fournie par la Kinect ouvre de nouvelles opportunités pour résoudre les problèmes fondamentaux dans le domaine de la vision par ordinateur. De nombreuses applications utilisant la kinect ont été développés dans le domaine de vision par ordinateur, tels que le prétraitement, le suivi et la reconnaissance d'objets, l'analyse de l'activité humaine, l'analyse et la reconnaissance des gestes de la main [I.1] (figure I.9), et la cartographie intérieure 3-D [I.2] (figure I.10)...etc.

Dans la littérature, on trouve un état de l'art complet sur cette problématique, à savoir l'utilisation de la kinect en vision par ordinateur [I.3].

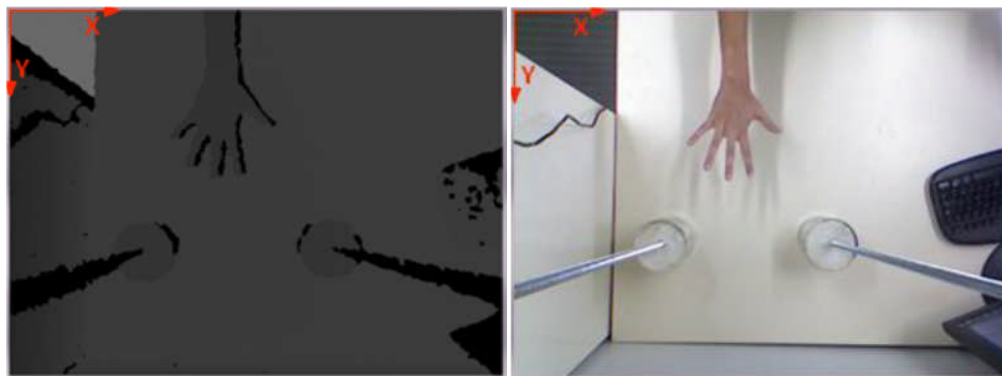


Figure I.9 : Image de profondeur et couleur



Figure I.10 : Articulations détectées par NITE. Affichées sur la carte de profondeur et sur couleurs

## I.7 Conclusion

La Kinect est un appareil très récent qui offre encore de nombreuses opportunités pour les années à venir. Ainsi, dans ce premier chapitre, nous avons présenté une brève étude sur la Kinect. L'étude de ses caractéristiques prouve que c'est un appareil fiable. Son capteur RGB-D trouve un intérêt par rapport à une caméra classique qui se situe dans son image de profondeur, qui reste plus facile à analyser qu'une image couleur classique.

Le potentiel de la Kinect ne s'arrête pas à certains jeux et de nombreux projets utilisent déjà la Kinect, spécifiquement dans le domaine du traitement d'image et vision par ordinateur.

# *Chapitre II*

## *Segmentation d'images*

### **II.1 Introduction**

Dans ce chapitre, nous allons nous efforcer de dessiner un état de l'art de la segmentation d'image. Cette revue des différentes techniques et approches n'a pas pour but d'être exhaustive mais de montrer la grande diversité qui existe au sein de ce vaste domaine.

L'analyse d'images a pour but l'extraction de l'information caractéristique contenue dans une image. Le résultat d'une telle analyse s'appelle très souvent la description structurelle. Celle-ci peut prendre la forme d'une image ou de toute structure de données permettant une description des entités contenues dans l'image. Par opposition avec la phase d'interprétation, l'analyse tente, dans la mesure du possible, de ne pas prendre en compte le contexte (i.e. l'application). Essentiellement, l'analyse de l'image fait appel à la segmentation où l'on va tenter d'associer à chaque pixel de l'image un label en s'appuyant sur l'information portée (niveaux de gris ou couleur), sa distribution spatiale sur le support image, des modèles simples (le plus souvent des modèles géométriques).

Nous allons dans ce chapitre définir la segmentation et nous nous intéresserons aux différentes méthodes que nous avons classé en cinq grands thèmes que sont l'approche basée sur la forme, celle reposant sur un paradigme de régions, l'approche structurelles, celle opérant sur les contours et celles faisant appel à la théorie des graphes.

## II.2 Définition de la segmentation

La segmentation est définie comme la partition d'une image en régions connexes homogènes. Elle constitue un point central de l'analyse d'image. Etape obligée de tout système d'analyse intelligente de scènes (modules d'assistance à la conduite, d'aide au diagnostic médical, de télésurveillance... pour ne citer que quelques exemples), la segmentation est également utilisée dans des domaines a priori moins évidents tels que le codage d'image (codage orienté objet), l'analyse de matériaux... Analyser et comprendre une scène sous-entend d'abord extraire, segmenter et mettre en correspondance les différentes régions de la scène. La question de l'interprétation de cette information est généralement un problème intervenant dans une seconde étape et qui fait appel à des techniques ne relevant plus à proprement parler de l'analyse d'image mais de l'intelligence artificielle. La segmentation d'image est presque essentiellement basée sur une méthode : la Ligne de Partage des Eaux (LPE) calculée sur une image gradient à partir de marqueurs des régions à extraire. Trouver ces marqueurs est toujours un problème délicat. Or, sur ce point précisément, les fonctions d'extinction que nous avons introduites laissent présager un apport important.

## II.3 Méthodes de la segmentation d'image

La segmentation est un domaine vaste de l'imagerie numérique. Ces dernières années, pas mal de thèses l'ont été concernées. Il est tout de même difficile de choisir une méthode parmi les nombreuses qui existent déjà et cela est dû au fait que chaque méthode est fortement liée à un domaine d'application.

Dans ce qui suit nous allons présenter cinq approches qui sont [II.1] :

- La segmentation utilisant les contours comme critère de décision,
- Celles basées sur les régions,
- Celles basées sur la forme,
- Celles préférant une approche structurelle,
- Et enfin celles faisant appel à la théorie des graphes.

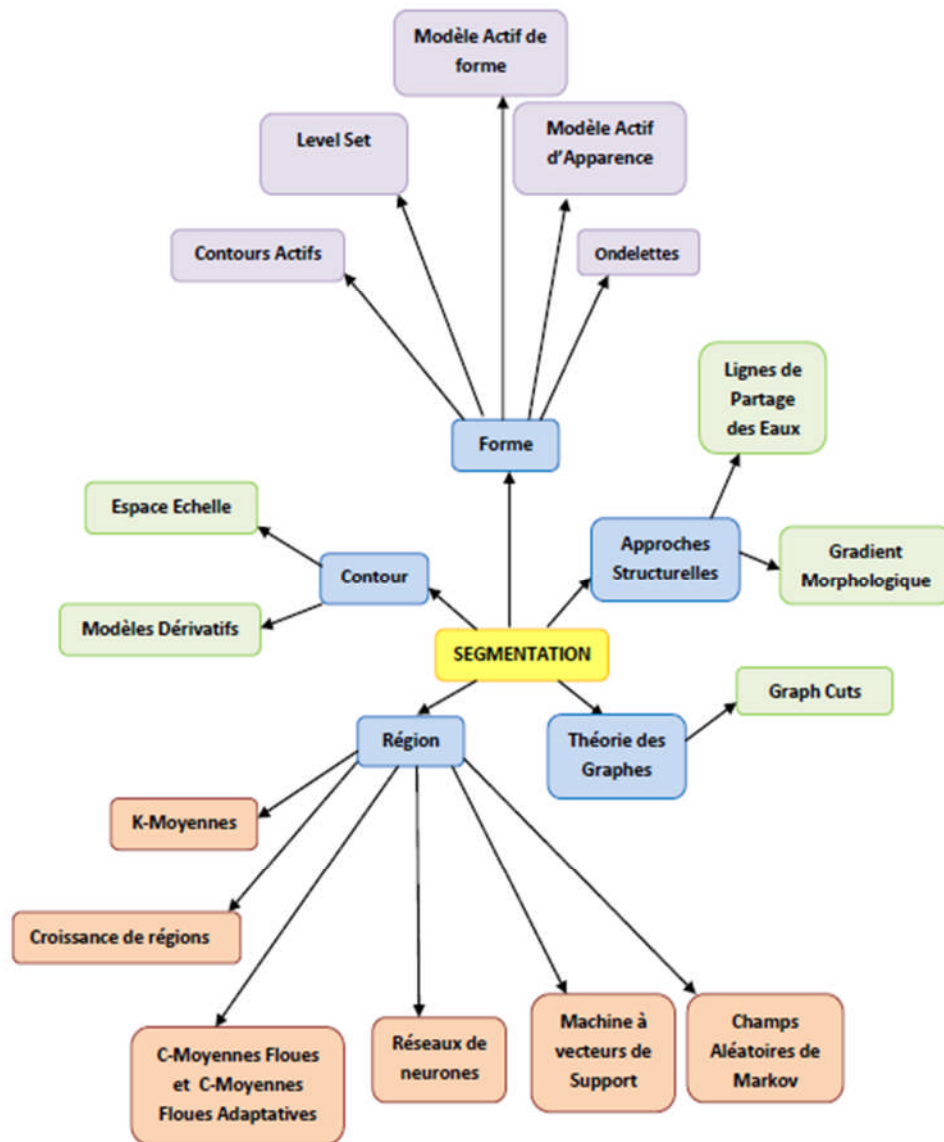


Figure II.1 : Classification des différentes méthodes de segmentation.

### II.3.1 Approche région

L'approche région de la segmentation utilise des techniques d'identification et de localisation d'ensembles connexes de pixels. Elle comprend plusieurs techniques parmi les quelles on peut citer les classes.

Réseaux de neurones, l'algorithme K-means, ainsi que les méthodes basées sur la croissance de régions.

### II.3.1.1 Classification par les Réseaux de neurones

Un réseau de neurone est une succession de couche de neurones dont chacune prend ses entrées à partir des sorties des couches précédentes. Chaque couche  $i$  est composée de  $N_i$  neurones.

Les entrées de chaque neurone sont pondérées par des poids dits synaptiques de sorte que les contributions des neurones des couches précédentes sont chacune multipliées par ces poids synaptique et additionnées pour donner une valeur globale qui sera affectée à un neurone donné.

La fonction de sortie dit fonction d'activation qui introduit une non-linéarité à chaque étape. Ceci montre l'importance du choix judicieux d'une bonne fonction de sortie : un réseau de neurones dont les sorties seraient linéaires, n'aurait aucun intérêt [II.1]. Il est possible de donner au réseau de neurones des boucles en liant les sorties d'une couche avec les entrées de couches précédentes. Cela donnerait au réseau de neurones d'autres propriétés intéressantes mais augmentera sa complexité.

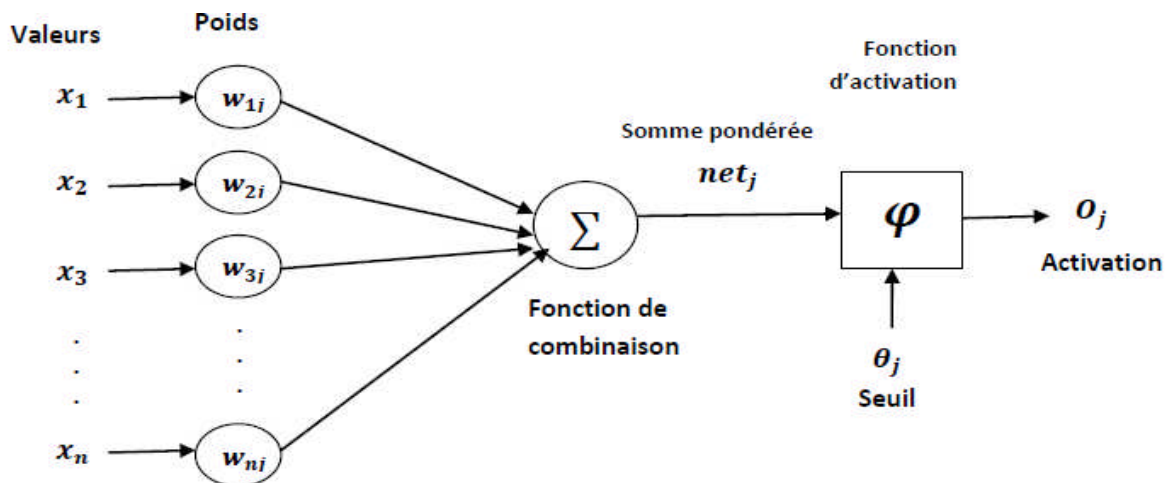


Figure II.2 : Structure d'un neurone artificiel.

Le calcul de la sortie est obtenu à travers la sommation pondérée des entrées et puis l'application d'une fonction d'activation.

En segmentation d'images chaque neurone de la couche d'entrée reçoit la valeur d'un attribut d'un pixel et un neurone de la couche de sortie correspond une classe.

### II.3.1.2 Classification par K-Means (K-Moyens)

L'algorithme des k-means constitue l'un des algorithmes le plus utilisé en segmentation d'images à cause de sa simplicité et sa rapidité pour classer les pixels à partir de leurs caractéristiques (couleur, texture, profondeur,..) en k parties (ou clusters) en supposant que les attributs forment un espace vectoriel. L'objectif est de minimiser la variance intra-cluster  $V$  :

$$V = \sum_{i=1}^k \sum_{y_j \in S_i} \|y_j - \mu_i\|^2 \quad (\text{II.1})$$

Où  $S_i$ ,  $i = 1, 2, \dots, k$  sont les k clusters et  $\mu_i$  est le point moyen des pixels  $y_j \in S_i$

L'algorithme de K-means se résume dans les étapes suivantes et la figure (II.3) :

1. Partitionnement initiale des données en k sous ensembles (non vides)
2. Calcul des centres des groupes de la partition courante
3. Les données sont affectées au groupe dont leur centre est plus proche
4. Retour à l'étape 2
5. Si les groupes sont changeant.



Ensemble des pixels initiaux



Choix aléatoire des pixels initiaux



Association des pixels à des classes



Calculs des nouveaux centres



Groupe stable

Figure II.3 : Illustration de l'algorithme K-means.

L'algorithme K-means est un des algorithmes les plus connus dans la communauté de classification des données. Dans cet algorithme, chaque classe est représentée par la moyenne (mean) ou la moyenne pondérée de ses points de données constituant ainsi son centre. K-means est un algorithme itératif. Il commence avec un ensemble de  $K$  points de référence (centres) choisis de manière aléatoire ou par l'utilisateur. Au début, les points de données sont partitionnés en  $K$  classes : un point appartient à une classe si le point de référence de cette classe est le plus proche de lui. La mise à jour des points de référence et l'affectation des points de données aux classes sont réalisées pendant les itérations successives. [II.4]

Il y a plusieurs versions de K-means [II.5] qui diffèrent dans la mise à jour des classes et/ou dans le critère pour faire cette mise à jour. Les plus connus sont les algorithmes Standard K-means, K-means de Lloyd et Continuous K-means [II.6].

Dans l'algorithme dit Standard K-means, la mise à jour des centres est faite pendant et après une itération. Si un point appartient à une classe et que pour lui, le centre de cette classe est le point de référence le plus proche, alors il n'y aura aucun ajustement. Mais si après avoir affecté un point  $y$  à une classe  $C_i$ , on trouve qu'il y a une autre classe  $C_j$  dont le centre est le point de référence le plus proche de  $y$  que celui de  $C_i$ , alors il faut réaffecter  $y$  à la classe  $C_j$  et recalculer les centres des deux classes. Les points de référence de ces deux classes se déplacent donc vers de nouveaux centres. Dans l'algorithme de Lloyd, les ajustements sont réalisés en calculant le centre pour chaque classe et en utilisant ces centres comme les points de référence dans l'itération suivante pour tous les points de données. La mise à jour des centres n'est faite qu'après une itération. Dans l'algorithme Continuous K-means proposé par McQueen [II.7] les  $K$  centres initiaux sont choisis arbitrairement parmi les points de la population entière. Si l'échantillon est assez gros, alors la distribution des points de référence initiaux pourrait refléter celle des points de la population. Dans cet algorithme, contrairement au standard K-means où tous les points sont séquentiellement examinés, on n'examine qu'un échantillon aléatoire des points. Si le jeu de données est gros et l'échantillon est représentatif du jeu de données, alors l'algorithme peut converger plus vite qu'un algorithme qui doit examiner séquentiellement tous les points [II.6].

Cet algorithme est parmi les algorithmes de segmentations les plus populaires parce qu'il est extrêmement rapide en pratique. En terme de performance, cet algorithme ne garantit pas un optimum global. La qualité de la solution dépend grandement des ensembles initiaux et peut, en pratique, être bien en deçà de l'optimum global. Comme l'algorithme est très rapide, une méthode courante est de le lancer plusieurs fois et de retourner la meilleure partition [II.1] en plus l'intervention d'un expert est généralement nécessaire pour fixer le nombre  $K$  de classes.

L'algorithme de K-means peut être aussi confronté au des classes hétérogènes (non homogènes). Cependant, des méthodes existent pour corriger ce problème. On parle alors de régularisation de l'image.

### II.3.1.3 C-moyennes floues

Dunn [II.2] a introduit l'algorithme c-moyennes floues (fuzzy c-means ou FCM) qui, est un algorithme qui repose sur le même principe que le K-Means mais qui à recours à la logique des ensembles flous. Bezdek s'y est intéressé [II.3] et a développé cet algorithme. Dans le cas des c-moyennes floues, la fonctionnelle  $L$  à minimiser est :

$$L = J_{\omega}(x, v; y) = \sum_{j=1}^n \sum_{i=1}^k (x_{ij})^{\omega} \|y_j - v_i\|_A^2 \quad (\text{II.2})$$

Où  $n$  est le nombre de points à traiter,  $k$  le nombre de classes désirées,  $w \in [1 ; +\infty$  [est le poids de fuzzyfication,  $v = (v_1; \dots; v_k)$  est le vecteur des centres de classes,  $\|\cdot\|_A$  est un produit scalaire où  $A$  est une matrice définie positive et  $x = [x_{ij}]_{\in \mathbb{R}^{k \times n}}$

L'algorithme suivant montre les étapes de FCM :

1. initialiser les centres.
2. fixer le coefficient flou.
3. calcul la partition floue initiale (La matrice d'appartenance)
4. répéter : - Calcul des nouveaux centres  
- Calcul de la nouvelle partition floue

Tant que le critère d'arrêt n'est pas vérifié.

En [II.4] Bensaïd et al. Proposent un algorithme semi-automatique permettant d'introduire la notion d'ensemble d'appartenance. L'apprentissage peut être de taille variable et son influence sur la classification est plus ou moins importante.

Les algorithmes FCM souffrent du manque de prise en compte d'informations contextuelles et fournissent des classifications moins efficaces en présence de bruit. Pour corriger ce défaut, une segmentation floue Markovienne est proposée dans [II.9] et appliquée dans [II.10] à la segmentation des principaux tissus.

Certains auteurs ont combiné l'algorithme FCM avec des systèmes à base de règles, cette coopération a permis l'intégration de connaissances a priori sur les entités recherchées comme Dans [II.11], l'algorithme FCM et un système à base de règles coopèrent pour la détection de glioblastomes multiformes, sur des images multi-échocs, les résultats ont montré une bonne adéquation avec la vérité terrain.

#### **II.3.1.4 Croissance de régions**

Les algorithmes de croissance de régions sont fondés sur l'agrégation progressive des pixels à partir d'un point initial appelé « germe », selon un ou plusieurs critères. Les points voisins du germe sont successivement examinés ; s'ils vérifient les critères d'agrégation, ils sont inclus à la région et deviennent à leur tour un germe ; dans le cas contraire, ils sont rejetés.

La segmentation par croissance de régions est simple à mettre en œuvre. L'avantage principal de cette technique se réside dans la prise en compte des informations contextuelles par le mécanisme de croissance. Alors que la disposition des germes initiaux représente un inconvénient. En effet, celle-ci nécessite quasi-systématiquement l'intervention d'un opérateur.

#### **II.3.1.5 Champs Aléatoires de Markov**

Le formalisme des champs de Markov permet aisément d'exprimer des contraintes de voisinage entre les pixels et ainsi d'améliorer la classification obtenue. Cette technique a aussi comme rôle d'utiliser les résultats d'autres algorithmes de segmentations tel que Le K-means pour corriger certaines imperfections essentiellement liées au bruit engendré par le matériel

d'acquisition et à la nature texturée des images échantillonnées. C'est pour cela qu'on parle de régularisation Markovienne.

Il existe d'autres méthodes qui font partie de l'approche région tel que la segmentation par seuillages qui visent à discriminer les pixels en fonction de leurs niveaux de gris [II.12], La méthode Mean Shift [II.13], machines à vecteurs de support (SVM)[II.14] et l'algorithme ensemble de niveau (level sets)[II.15].

### II.3.2 Approche contour

Contrairement à l'approche région, qui cherche à former des zones homogènes, dans l'approche contours (ou frontières), il s'agit de reconnaître les zones de transition et de localiser au mieux la frontière entre les régions.

On distingue trois types de contours simples :

- Marche d'escalier : le contour est net (contour idéal).
- Rampe : le contour est plus flou.
- Toit : il s'agit d'une ligne sur un fond uniforme.

Parmi les principales techniques qui appartiennent à cette approche, on distingue les modèles dérivatifs et les modèles d'espace échelle.

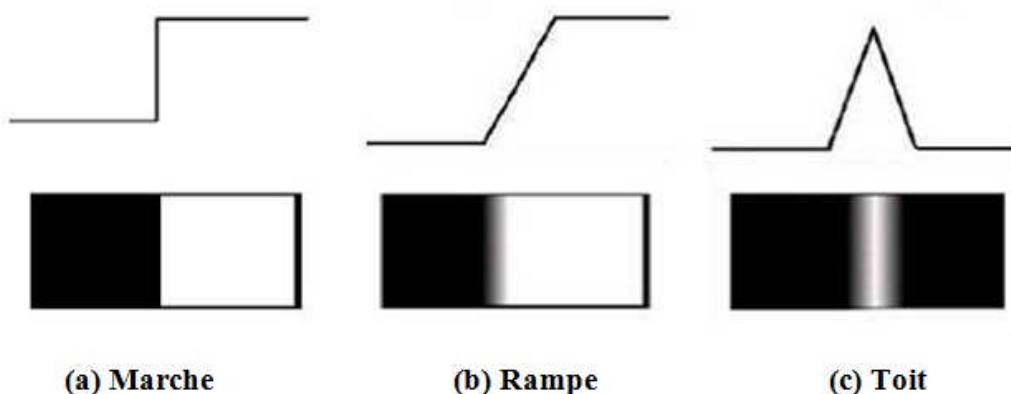


Figure II.4 : Quelques modèles de contours.

### II.3.2.1 Méthodes dérivatifs

Les méthodes dérivatives permettent de trouver les contours dans une image d'une manière simple. On peut détecter le point de contour par la détermination du maximum de la norme du gradient ou bien en étudiant le passage par zéro du Laplacien comme la montre la figure (II.5) suivante :

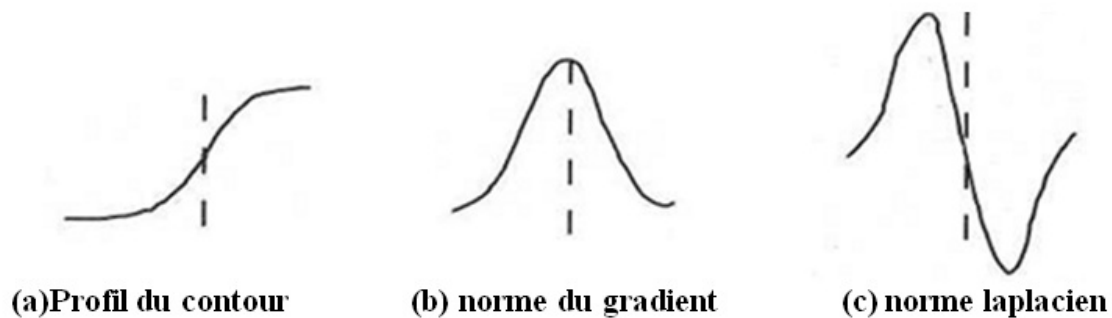


Figure II.5 : Opérateur dérivatifs.

### II.3.2.2 Espace –échelle

L'espace-échelle (scale-space) est une théorie formelle pour la manipulation des structures d'images à des échelles différentes de telle sorte que les structures soient successivement supprimées avec un paramètre « t » d'échelle qui est associé à chaque niveau dans la représentation de l'espace d'échelle.

La représentation en espace-échelle d'une image  $f(x,y)$  est une famille de signaux dérivés  $L(x, y, t)$  qui est défini la convolution de  $f(x, y)$  avec le noyau gaussien  $g(x, y, t)$

tel que :

$$g(x, y, t) = \frac{1}{2\pi t} e^{-(x^2+y^2)/2t} \quad (\text{II.3})$$

$$L(x, y, t) = g(x, y, t) * f(x, y) \quad (\text{II.4})$$

$t = \sigma^2$  est la variance de la gaussienne et avec une condition initiale  $L(x, y, 0) = f(x, y)$ .

Dans [II.16] Lindeberg a essayé de lier les minimums locaux et les points selles à travers les échelles. Il a proposé une représentation de l'image intitulée « scale-space primal sketch » qui explicite les relations entre structures à différentes échelles.

La méthode espace-échelle a montré sa efficacité pour la segmentation des structures anatomiques particulières. Elle a l'avantage, par rapport aux méthodes dérivatives de fournir des contours ou surfaces fermés. Mais l'inconvénient majeur de cette modèle est la génération des faux contours.

L'opérateur DOG (Différence of Gaussian) est aussi utilisé pour augmenter la visibilité des contours ou autres détails présents sur l'image en niveau de gris (NG) traitée par la différence de deux filtres gaussiens [II.17].

### II.3.3 Approche Structurelle

Les méthodes de segmentation d'images les plus célèbres de cette approche sont: le gradient morphologique et la méthode de Ligne de Partage des Eaux.

#### II.3.3.1 Gradient Morphologique

La morphologie mathématique est une branche des mathématiques fondée sur l'étude des ensembles permettant d'analyser les structures géométriques. Ses bases théoriques ont été établies par Matheson [II.18] et Serra [II.19].

La différence symétrique entre l'image dilatée et érodée par le même élément structurant de taille unitaire donne le gradient morphologique qui est un opérateur de détection de contour et qui peut se résumer par l'équation :

$$\text{grad}_B(X) = \delta_B(X) - \varepsilon_B(X) \quad (\text{II.5})$$

### II.3.3.2 Ligne de Partage des Eaux

La ligne de partage des eaux (LPE) constitue une méthode de la segmentation d'une image en proprement dite.

L'algorithme LPE, proposé par Digabel et Lan tuéjoul [II.20], utilise la description des images en termes géographiques. Une image peut en effet être perçue comme un relief si l'on associe le niveau de gris de chaque point à une altitude. Il est alors possible de définir la ligne de partage des eaux comme étant la crête formant la limite entre deux bassins versants. Pour l'obtenir, il faut imaginer l'immersion d'un relief dans de l'eau, en indiquant que l'eau ne peut pénétrer dans les vallées que par ses minima. La LPE est représentée par les points où deux lacs disjoints se rejoignent au cours de l'immersion.

Une des difficultés à la mise en œuvre de cette analogie intuitive est qu'elle laisse beaucoup de liberté quant à sa formalisation. Il existe deux classes principales d'implémentation : l'une est basée sur un algorithme récursif d'immersion et une autre basée sur l'utilisation de fonctions de distances géodésiques. Leur définition algorithmique ainsi que leur analyse critique est donnée dans l'article de Roerdink et Meijster [II.1].

La sur-segmentation représente l'inconvénient majeur de LPE, On peut remédier à ce problème soit par un filtrage morphologique du gradient, ou on associant à la méthode LPE à une méthode de fusion de régions.

### II.3.4 Approche forme

L'approche basée sur la forme tend à rechercher des régions qui dérivent une forme donnée. Trois types de techniques sont généralement utilisées:

Le modèle des contours actifs(Snakes), le modèle ASM et le modèle AAM.

### II.3.4.1 Contours Actifs

Les contours actifs à forme libre appelés « snakes » proposés par Kass en 1987 dans [II.21], permet de modéliser le contour d'une région ou l'objet d'une image par une courbe paramétrée  $C$  telle que:

$C = \{v_s = (x_s, y_s ; s \in [a, b])\}$   $S$  est l'abscisse curviligne

Le contour initial  $v_0$  est initialisé par l'utilisateur près des frontières de l'objet à détecter. La courbe est ensuite déplacée et déformée au cours d'un processus itératif cherchant à minimiser une fonction d'énergie  $E$  :

$$E(v(s)) = \underbrace{\int_{\Omega} \alpha v'(s)^2 ds + \int_{\Omega} \beta v''(s)^2 ds}_{E_{int}} - \lambda \cdot \underbrace{\int_{\Omega} |\nabla I(v(s))|^2 ds}_{E_{ext}} \quad (\text{II.6})$$

Où

$E_{int}$ : Représente l'énergie interne dépend uniquement de la forme du snake, elle utilise des notions comme la courbure du contour ou la régularité d'espacement des points.

$v'(s)$  et  $v''(s)$  sont les dérivées premières et secondaires du contour,  $\alpha$  et  $\beta$  sont des constantes réelles représentant les coefficients d'élasticité et de rigidité de la courbe.

$E_{ext}$  : Représente l'énergie externe qui est définie par l'utilisateur selon les spécificités du problème et associe le contour aux données de l'image (comme l'intensité), Elle dépend du gradient de luminance de l'image  $\nabla I(v(s))$  . La constante réelle  $\lambda$  permet de pondérer son influence.

### II.3.4.2 Modèle actif de forme

Les modèles actifs de forme Active Shapes Method (ASM) ont été initialement proposées par Cootes et Taylor pour localiser des objets déformables dans des images [II.22]. Rapidement, elles ont été étendues à de nombreux domaines. Ce sont des modèles statistiques de la forme d'un objet qui se déforment itérativement pour s'adapter à un exemplaire de cet objet dans une nouvelle image. Pour cela, une base représentative est étiquetée manuellement. Sur chaque image,  $N$  points (quelques dizaines) sont positionnés sur les contours. Ainsi, le contour de l'image  $i$  est représenté par le vecteur suivant :

$$X_i = (X_{i1}, Y_{i1}, X_{i2}, Y_{i2}, \dots, X_{in}, Y_{in})^T \quad (\text{II.7})$$

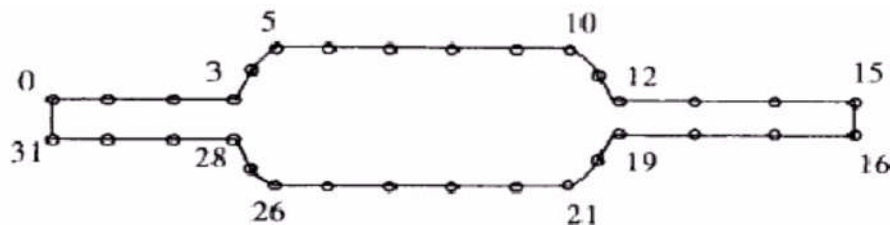


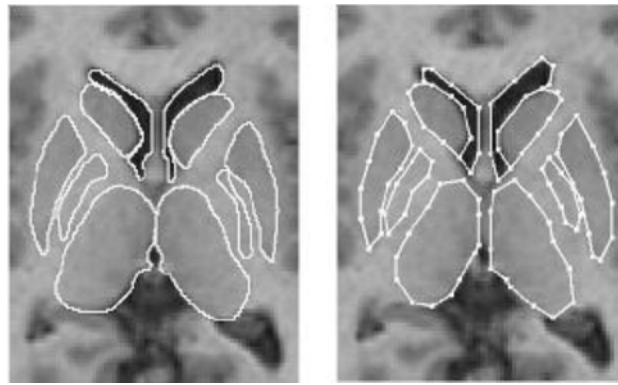
Figure II.6 : Etiquetage (manuel) des formes par  $N$  points.

Puis, un algorithme itératif est utilisé pour traduire, tourner et mettre à l'échelle les ensembles de points  $X_i$  de manière à les aligner. Cet alignement permet de supprimer les variations dues à la pose et de ne garder que celles reflétant des changements de forme.

Une analyse en composantes principales (Principal Components Analysis - ACP) sur l'ensemble d'entraînement aligné donne ensuite la position moyenne des points ainsi que les axes de déformation représentant au mieux les écarts à la moyenne, toute forme peut être approchée par une somme pondérée des modes principaux de l'ACP.

L'ACP permet d'obtenir une forte réduction du nombre de dimensions du problème.

Une fois que l'entraînement a été effectué, le contour de région est localisé par un algorithme itératif [II.23].



(a) Segmentation Manuelle (b) Segmentation par ASM

**Figure II.7 :** Exemple de Modèle Actif de Forme.

### II.3.4.3 Modèle actif d'apparence (AAM)

La modélisation de l'apparence, Extension directe de l'ASM. Introduite également par Cootes [II.24], cette technique connue sous le nom de modèles d'apparence active (Active Appearance Models - AAM) est en fait une généralisation de la ASM. Elle permet de construire un modèle statistique de l'objet à segmenter incluant à la fois la forme et les niveaux de gris (la texture) (Figure II.8).

La technique des AAM nécessite le même type d'apprentissage que les ASM. Un certain nombre de points caractéristiques sont placés manuellement sur des images d'entraînement. La forme moyenne  $\bar{x}$  ainsi que ses principaux modes de variation sont ensuite calculés par ACP. Pour constituer le modèle de texture  $\bar{t}$ , la base d'images d'entraînement sont préalablement déformés par un algorithme de triangulation de manière à mettre en coïncidence les points du modèle de forme avec ceux de la forme moyenne. Puis, les niveaux de gris des pixels situés à l'intérieur du contour sont échantillonnés. Une analyse en composantes principales permet ensuite de calculer les modes principaux de variation de la luminance [II.25].

Enfin, le modèle d'apparence est obtenu en concaténant les paramètres de forme et de texture de chaque image.

➤ Tout forme  $x$  peut approcher par  $x = \bar{x} + Pb$

- Tout texture  $x$  peut approcher par  $t = \bar{t} + p_2 b_2$



**Figure II.8 :** La méthode AAM incluant à la fois la forme et la texture.

L'apprentissage statistique est la grande puissance des AAM, mais c'est aussi l'origine de nombreux inconvénients. Pour une simple question matérielle, il est souvent difficile de constituer les modèles d'entraînement (nombre important des points pour étiqueter un nombre important d'images). Bien sûr, il est possible de réduire le nombre d'images d'entraînement ou le nombre de points du modèle ; mais, dans ce cas, l'analyse statistique devient peu représentative et les modes de déformations extraits ne permettent de couvrir qu'une faible variabilité de formes. Ensuite, le nombre de modes considérés doit permettre de représenter au moins 95% de la variance observée lors de l'apprentissage. Or, le fait d'inclure la texture dans le modèle statistique augmente cette variance de manière très significative et oblige à utiliser beaucoup plus de modes que pour les ASM.

L'inconvénient le plus gênant des AAM est probablement le fait qu'elles synthétisent des apparences nouvelles par combinaison linéaire d'apparences connues. Ainsi, tout changement structurel non représenté dans la base d'entraînement ne pourra être reproduit. Par exemple, les AAM ne pourront pas créer un visage dont l'un des yeux est fermé si aucune image de la base ne contient une telle configuration. Si le système doit pouvoir gérer les clins d'œil, il faut obligatoirement que l'entraînement inclut un nombre suffisant de clins d'œil. Cela rend le travail d'étiquetage lourd, d'autant plus que chaque configuration doit être suffisamment représentée [II.25].

### II.3.5 Approche basée sur la théorie des graphes

Dans cette approche, l'idée de base est de créer un graphe à partir de l'image selon des procédés assez simples et de travailler sur ces graphes. Parmi les méthodes basées sur cette théorie, on peut citer la méthode des coupes de graphe (Graph Cuts en anglais).

#### II.3.5.1 Graph Cuts

Les Graph Cuts(coupes de graphe) sont des méthodes d'optimisation combinatoire basées sur la théorie des graphes. Ils ont été introduits dans le domaine de vision par ordinateur en 1989. Ces méthodes sont utilisées beaucoup dans la segmentation d'images. L'efficacité des Graph Cuts vient du fait qu'ils permettent sous certaines conditions de trouver des minima globaux de fonctions [II.27]. La segmentation par Graph Cuts est une méthode agissant par minimisation d'énergie (comme les Snakes). Mais ce problème de minimisation d'énergie peut se ramené à un problème de coupe minimale dans un graphe.

Nous commençons par définir quelques notions sur les graphes :

Soit  $G = (V, E)$  un graphe orienté, où  $V$  est l'ensemble des sommets du graphe, et  $E$  l'ensemble des arrêtes. On définit aussi la fonction poids  $w(e) > 0$  (appelé aussi capacité). Nous rajoutons aussi deux sommets  $s$  et  $t$ , qui représentent respectivement la source et le puits.

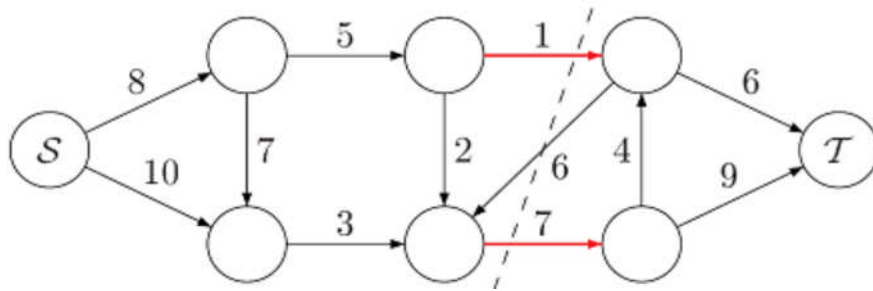
##### II.3.5.1.1 Définition d'un Graph Cuts

Il existe deux définitions [II.28] :

1- On appelle coupe dans un graphe une partition  $(V_1, V_2)$  de l'ensemble des sommets

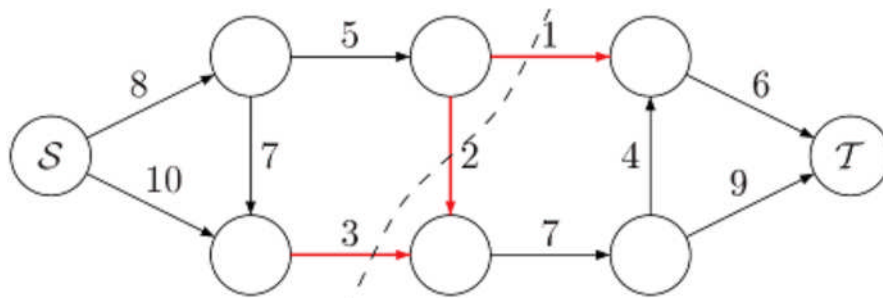
$V$  (tel que  $V_1 \cap V_2 = \emptyset$  et  $V_1 \cup V_2 = V$ ), telle que  $s \in V_1$  et  $t \in V_2$ .

2- On appelle un coupe de graphe un sous-ensemble  $F$  de  $E$  telle que  $G' = (V, E - F)$  ne contient pas de chemin de  $s$  vers  $t$ .



**Figure II.9** : Coupe d'un graphe. La ligne pointillée sépare le graphe en deux ensembles disjointe. Les arêtes en rouge forment le sous-ensemble F de la définition 2.

**II.3.5.1.2 Poids minimale :** Si nous prenons la deuxième définition d'une coupe, son poids serait alors égal à la somme des poids des arrêtes appartenant à F. On appelle coupe minimale pour un graphe G, une coupe de poids minimal. [II.28]. Il existe plusieurs coupes minimales pour un même graphe.



**Figure II.10** : Coupe minimale du graphe. Ici le poids de la coupe minimale est  $w=6$ .

## II.4 Conclusion

Dans ce chapitre, nous avons présenté un certaines méthodes dédiées à la segmentation d'images. La revue qui est faite ici n'est pas exhaustive mais nous avons essayé de mettre en évidence la diversité des techniques abordées par les différents auteurs.

Cependant chacune d'entre elles à ses qualités et ses défauts, et c'est sur cette base que doit se faire le choix de l'emploi de l'une ou l'autre de ces techniques.

La diversité des techniques est représentative de la complexité du problème de segmentation d'images, Ces difficultés sont liées, d'une part, aux objets traités.

Dans le prochain chapitre, nous allons présenter une méthode de segmentation d'image de la Kinect en utilisant l'algorithme de K-means.

# *Chapitre III*

## *Segmentation d'images de la kinect*

### **III.1 Introduction**

Dans ce chapitre, nous présentons la méthode de segmentation des images acquises avec la caméra Kinect, chaque pixel est généralement caractérisé par plusieurs paramètres en suite ils sont soumis à un classifieur qui réalisera la segmentation.

Nous allons utiliser l'image R, V, B puis initié manuellement les centres de classes, pour utiliser l'algorithme de segmentation K-means ou en utilise les attributs qui sont extrais a partir de la matrice de cooccurrence afin d'intégrer la texture.

Pour rendre compte de la pertinence de ces paramètres, nous allons réaliser une comparaison de la segmentation par l'algorithme de K-means obtenue par ces paramètres avec celle obtenue par les paramètres de texture extraits de la matrice de cooccurrence.

### III.1 Description la méthode de segmentation

La méthode proposée pour la segmentation d'image acquise par la Kinect se déroule en deux étapes successives :

- Extraction des attributs
- Classification des pixels

#### III.1.1 Extraction des attributs

Il s'agit dans cette étape de caractériser chaque pixel de la scène par un ensemble d'attributs. Plusieurs stratégies sont utilisées dans ce but :

La première consiste à caractériser chaque pixel soit par les 3 attributs couleurs R, V, B extraits à partir de l'image couleur, soit par la profondeur D fournie l'image de profondeur ou soit par l'association des informations couleur R, V, B et profondeur D. Dans cette stratégie, aucun calcul n'est nécessaire.

La deuxième stratégie consiste à extraire des attributs de texture pour chaque pixel à partir de l'image couleur et/ou de l'image profondeur.

Comme attributs de texture, nous avons utilisé les attributs de Haralick. Ces derniers sont extraits localement pour chaque pixel à partir des matrices de cooccurrence ; ces matrices sont déterminées pour chaque pixel à partir de ses pixels voisins situés dans une fenêtre de taille  $(2w+1) \times (2w+1)$  centrée sur le pixel en question

Rappelons qu'une matrice de cooccurrence d'une image I en niveau de gris est définie par le nombre de couple de pixels séparé d'une distance (d) d'orientation ( $\theta$ )  $M(d, \theta)$ .

La méthode de la matrice de cooccurrence, dite aussi méthode de dépendance spatiale des niveaux de gris, permet de déterminer la fréquence d'apparition d'un "motif" formé de deux pixels séparés par une certaine distance "d" dans une direction particulière (voir figure III.1). Afin de limiter le nombre de calculs, on prend généralement comme valeurs  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  et  $135^\circ$  pour la direction  $\theta$  et 1 pour la valeur de 'd'.

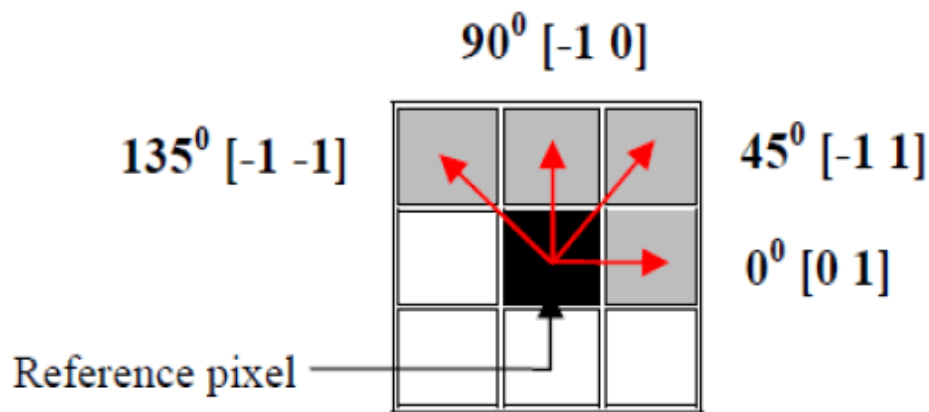


Figure III. 1 : Image de la distance 'd'.

La difficulté à surmonter lors de l'application de cette technique réside dans le choix du vecteur de déplacement et de la taille de la fenêtre du voisinage sur laquelle s'effectue la mesure car ils varient selon le type d'images et de textures.

A chaque direction  $\theta$  et pour chaque valeur de  $d$  correspond une matrice de cooccurrence  $M(d, \theta)$ .

On définit généralement les matrices symétriques de cooccurrence. Elles sont construites à partir des constatations suivantes :

$$M(d, 0^\circ) = M^t(d, 180^\circ)$$

$$M(d, 45^\circ) = M^t(d, 225^\circ)$$

$$M(d, 90^\circ) = M^t(d, 270^\circ)$$

$$M(d, 135^\circ) = M^t(d, 315^\circ)$$

### III.1.1.1 Matrices de cooccurrence pour des images couleur

Pour une image RVB, 6 matrices de cooccurrences sont définies en fonction des composantes couleurs considérées (RR, VV, BB, RV, RB et VB).

Pour notre application, nous avons utilisé que les matrices intra-composantes soient (RR, VV, BB).

A partir de chaque matrice de cooccurrence, on peut extraire plusieurs attributs. Les plus utilisés sont l'énergie, le contraste, la corrélation et l'homogénéité.

**III.1.1.1.1 L'énergie ou moment angulaire d'ordre deux:** elle est donnée par la relation

$$E = \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} p(i, j)^2 \quad (\text{III.1})$$

L'énergie mesure l'homogénéité de l'image. Elle a une valeur d'autant plus faible qu'il y a peu de zones homogènes : dans ce cas, il existe beaucoup de transitions de niveaux de gris.

**III.1.1.1.2 Le contraste ou (l'inertie) :** il est défini par la relation

$$\text{CONT} = \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} (i - j)^2 p^2(i, j) \quad (\text{III.2})$$

Il mesure les variations locales des niveaux de gris. Si elles sont importantes (c'est à dire s'il existe peu de régions homogènes), alors le contraste sera élevé. Ce paramètre permet aussi de caractériser la dispersion des valeurs de la matrice par rapport à sa diagonale principale.

**III.1.1.1.3 La corrélation :** Elle est définie par la relation

$$\text{COR} = \frac{\sum_{i=1}^{Ng} \sum_{j=1}^{Ng} (i - \mu_x)(j - \mu_y)p(i, j)}{\sigma_x \sigma_y} \quad (\text{III.3})$$

où  $\mu_x$  et  $\mu_y$  représentent les moyennes respectivement des lignes et des colonnes de la matrice et  $\sigma_x$  et  $\sigma_y$  représentent les écarts types respectivement des lignes et des colonnes de la matrice.

La corrélation a une forte valeur quand les valeurs sont uniformément distribuées dans la matrice de cooccurrence, elle mesure la dépendance linéaire de niveau de gris dans l'image.

**III.1.1.1.4 L'homogénéité :** Elle est définie par la relation

$$\text{HOM} = \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} \frac{1}{1+(i-j)^2} p(i, j) \quad (\text{III.4})$$

Elle a un comportement inverse du contraste. Pour une texture grossière, les valeurs de la matrice sont concentrées sur la diagonale principale. Au contraire, pour une texture fine, les

valeurs de la matrice seront dispersées : en effet, pour une telle texture il existe beaucoup de transitions de niveaux de gris.

### III.1.2 L'algorithme de classification

Une fois l'étape de caractérisation de chaque pixel, pas un ensemble d'attributs selon la stratégie utilisé est achevée, nous procédons au regroupement des pixels en  $k$  classes de sorte que chaque classe contient les pixels ayant des attributs identiques. Cette étape est réalisée à l'algorithme K-means.

Dans ce travail, nous avons utilisons l'algorithme standard des K-means, qui dans sa version la plus simple nécessite la connaissance préalable du nombre  $K$  de classes. Son principe consiste à choisir initialement  $K$  centres définissant ainsi une partition  $C = \{C_1, C_2, \dots, C_K\}$  de  $K$  classes. Chaque pixel à classer est assignée à l'une des  $C_k$  classes selon une mesure de similarité entre chaque centre des classes et le pixel considérée. Une nouvelle partition  $C$  est ainsi obtenue et permet de redéfinir les nouveaux centres des classes. Ce processus itératif, prend fin lorsqu'un critère d'arrêt est satisfait. La méthode des K-means, également connue sous le nom de procédure de minimisation de l'erreur quadratique [11], fonctionne suivant l'algorithme de K-means. Il s'agit de classer un ensemble des pixels  $Y = (y_1, y_2, \dots, y_n)$  avec  $y_i \in \mathbb{R}^D$  en  $K$  classes ( $D$  étant le nombre d'attribut caractérisant le pixel) où chaque classe  $C_k$  est définie par son centre  $\bar{y}_k = (\bar{y}_{k1}, \bar{y}_{k2}, \dots, \bar{y}_{kp})$ , avec :

$$\bar{y}_{kj} = \frac{1}{N_k} \sum_{y_i \in C_k} y_{ij} \quad (\text{III.5})$$

$j$  étant l'indice de l'attribut,  $N_k$  le nombre de pixels de la classe  $k$  et  $y_{ij}$  l'attribut  $j$  du pixels  $y_i$ . Chaque pixels  $y_i$  se trouvant dans la classe  $k$  est individuellement réassigné à la classe dont la distance euclidienne par rapport à son centre est la plus faible. La distance du pixels  $y_i$  par rapport au différentes classes sont définie par :

$$d(y_i) = d(y_i, \bar{y}_k) = \|y_i - \bar{y}_k\|^2 \quad (\text{III.6})$$

Le critère à optimiser est défini par :

$$J(C) = \sum_{k=1}^K \sum_{y_i \in C_k} d(y_i, \bar{y}_k)^2 = \sum_{k=1}^K J_k(C) \quad (\text{III.7})$$

La partition  $C^*$  qui optimise ce critère est définie par :  $J(C^*) = \min_C J(C)$ .

L'algorithme de K-means est le suivant :

1. Donner le nombre  $K$  de classes, ainsi qu'une partition initiale  $C$  des  $n$  pixels à classer.
2. Calculer  $J(C)$  et  $\bar{y}_k$ ,  $k = 1, 2, \dots, K$ .
3. Pour chaque pixel  $y_i$  affectée initialement à la classe  $C_k$ ,  $i$  variant de 1 à  $n$  :
  - Calculer  $d_k(y_i)$ ,  $\forall k = 1, 2, \dots, K$ .
  - Affecter  $y_i$  à la classe  $C_s$  telle que :  $d_s(y_i) \leq d_k(y_i)$ ,  $\forall k = 1, 2, \dots, K$ .
  - Calculer  $\bar{y}_k$  et  $\bar{y}_s$
  - Recalculer  $J(C)$
4. Si  $J(C)$  a changé durant l'étape 3, revenir en 3, sinon arrêter.

Parmi les distances les plus utilisées, on peut citer : la distance euclidienne, la distance mahalanobis et la distance city block. [III.5]

La distance Euclidienne est la distance la plus employée. Elle est définie par :

$$d_e(y_1, y_2) = \|y_1 - y_2\| = \sqrt{\sum_{d=1}^D (y_{1d} - y_{2d})^2} \quad (\text{III.8})$$

Où :  $y_1$  et  $y_2 \in \mathbb{R}^D$

La distance City block entre deux vecteurs  $y_1$  et  $y_2$ , elle est donnée par :

$$d_c(y_1, y_2) = \sum_{i=1}^p |y_{1p} - y_{2p}| \quad (\text{III.9})$$

où  $p$  est le nombre de paramètres.

La distance Mahalanobis différencie des deux premières distances, par l'utilisation de la matrice de covariance des sommes. Comme elle prend aussi en compte les différentes formes des classes.

$$d_k(Y) = (Y - Y_k)^t T_k^{-1} (Y - Y_k) \quad (\text{III.10})$$

$Y$  est un vecteur d'attributs,  $Y_k$  est la moyenne des vecteurs  $Y$  sur la classe  $C_k$ .

$T_k^{-1}$  est la matrice inverse de la matrice covariance associé à cette classe.

$(Y - Y_k)^t$  est le transposé du vecteur  $(Y - Y_k)$ .

Dans notre travail, nous avons utilisé la distance euclidienne

### III.2 Tests et résultats

Pour évaluer la méthode de segmentation d'images de la Kinect décrite précédemment, nous avons utilisé des images tests issues du site web

([http://cs.nyu.edu/~silberman/datasets/nyu\\_depth\\_v2.html2](http://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html2)).

La figure (III.1.a) et (III.1.b) montre un exemple d'une image couleur et l'image profondeur.

La figure (III.1.c) montre le résultat de la segmentation de l'image couleur en utilisant seulement les attributs couleur R V B.

La figure (III.1.d) contient le résultat de la segmentation de l'image profondeur en utilisant seulement l'attribut profondeur D.

La segmentation associant les attributs couleur R V B et l'attribut profondeur D aboutit au résultat affiché sur la figure (III.1.e).

Notons le nombre de classes utilisées dans l'algorithme K-means a été fixé à 5 pour cette image. La première correspond à la table, la seconde c'est le sol, la troisième au mur en blanc, la quatrième à l'étagère et la cinquième au mur en gris.

L'initialisation des centres des classes a été réalisée d'une manière manuelle

Au vu des résultats, nous remarquons que l'utilisation des attributs couleur seuls n'aboutit pas à une segmentation correcte puisqu'elle permet pas, par exemple, de détecter toute les régions du sol ou celle de la porte du pond même classe.

Par contre l'utilisation de la profondeur seulement ne permet pas de différencier le sol des murs par exemple

Cependant, la combinaison des attributs couleur et profondeur a permis de détecter correctement la surface du sol et celle de la porte.

Elle permet également de différencier les murs, le sol, la porte et les étager.



Image couleur(a)

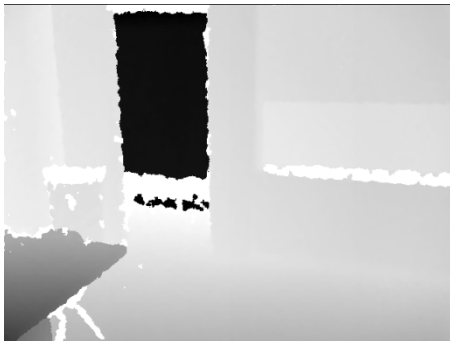
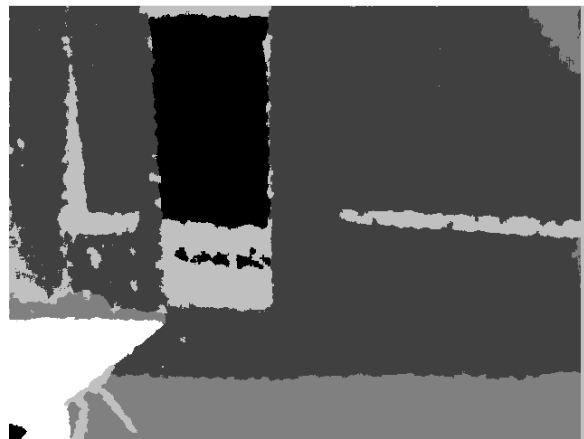


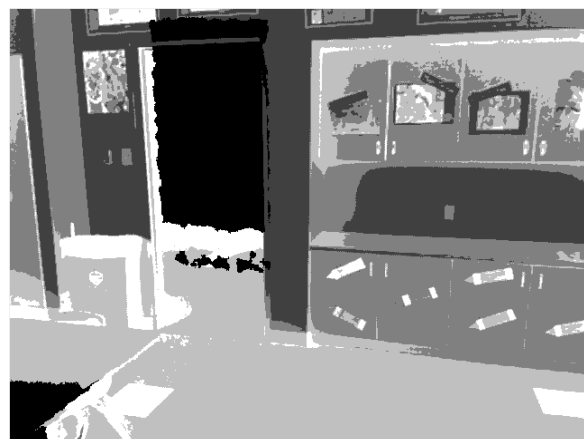
Image profondeur(b)



Segmentation de l'image couleur (c)



Segmentation de l'image profondeur (d)



Segmentation de l'image couleur et profondeur (e)

**Figure III.2:** Résultat de la segmentation d'image de la kinect

### III.3 Conclusion

Nous avons décrit dans ce chapitre une méthode simple et rapide d'une méthode de segmentation d'image de la Kinect. Cette méthode consiste à combiner les attributs couleur et profondeur dans le processus de classification. Les résultats ont montré l'intérêt d'associer ces attributs.

Notons également que nous avons aussi proposé d'extraire des attributs de texture sur l'image couleur et profondeur en vue de les combiner dans le processus de classification.

Cependant, les résultats obtenus et qui n'ont pas été rapportés dans ce chapitre n'ont pas été convainquants. Des tests supplémentaires incluant le choix des attributs de texture, le réglage des paramètres de l'algorithme (taille de la fenêtre  $W$ ) doivent être menés.

### Conclusion Générale

Nous avons abordé dans ce mémoire la segmentation d'images issues du système kinect. La kinect est un dispositif qui contient principalement deux caméras, la première fournit une image couleur de la scène et la seconde une image de profondeur mesurant la distance de chaque point de la scène avec la caméra.

La technique proposée pour segmenter les images issues de la kinect consiste à caractériser chaque pixel de la scène par ces trois composantes couleurs et l'information profondeur, ces pixels sont ensuite regroupés en classes via l'algorithme k-means.

Les tests effectués montrent que l'association des attributs couleur avec celui de la profondeur améliore considérablement les résultats de la segmentation comparativement à ceux obtenus avec les attributs couleur seuls ou l'attribut profondeur seul.

Nous avons également proposé, d'extraire des attributs de texture sur l'image couleur et profondeur en vue de les combiner dans le processus de classification. Ce pendant, les résultats obtenus et qui n'ont pas été convainquant. Des tests supplémentaires incluant le choix des attributs de texture, le paramètre de l'algorithme (taille de la fenêtre  $W$ ) doivent être menés.

Il serait également intéressant de faire appel à d'autres méthodes de caractérisation de la texture et à une technique de segmentation autre que l'algorithme k-means.

Nous avons remarqué l'apparition d'un décalage entre l'image couleur et l'image de profondeur faussant ainsi le résultat de la segmentation. Ceci est du au positionnement des deux capteurs image. Pour éviter ce problème, nous proposons d'effectuer un recalage d'images.

Finalement, nous avons pu nous rendre en compte du potentiel de la caméra kinect qui ouvre de nouvelles perspectives dans le domaine du traitement d'images.

- [I.1] Li Y(2012), "Hand gesture recognition using kinect". In IEEE 3<sup>rd</sup> international conference on software engineering and service science (ICSESS), June 2012, pp196-199.
- [I.2] Liu X, Fujimura K (2004), "Hand gesture recognition using depth data". In proceeding sixth IEEE international conference on automatic face and gesture recognition, May 2004,pp 529-534.
- [I.3] M.R. Andersen et al. "Kinect Depth Sensor Evaluation for Computer Vision Applications" Electrical and Computer Engineering, Technical report ECE-TR-6, 2012.
- [II.1] Jérémy LECOEUR « Segmentation d'IRM cérébrales multidimensionnelles par coupe de graphe » thèse doctorat, UNIVERSITÉ DE RENNES 1,2010.
- [II.2] J. Dunn : « A fuzzy relative of the ISODATA process and its use in detecting compact wellseparated clusters ». Journal of Cybernetics, 3:32–57, 1973.
- [II.3] J. Bezdek, J. Keller, N. Pal et R. Krisnapuram : « Fuzzy Models And Algorithms For Pattern Recognition And Image Processing ».Kluwer Academic Publishers, 1995.
- [II.4] S. Alkama, thèse de doctorat, Segmentation des images multi-composantes, Ecolenationale polytechnique, 2013.
- [II.5] P. Berkshire, Survey of Clustering Data Mining Techniques, 2002.
- [II.6] V. Faber, Clustering and Continuous K-means, Los Alamos Science, 1994.
- [II.7] J. MacQueen, Some Methods for Classification and Analysis of Multivariate Observations, Proc.5th Berkeley Symp, 1965, pp. 281-297.
- [II.8] A.M. Bensaid, L.O. Hall, J.C. Bezdek, and L.P. Clarke. « Partially supervised clustering for image segmentation ». Pattern Recognition, 29(5):859–871, 1996.
- [II.9] F. Salzenstein and W. Pieczynski. « Parameter estimation in hidden fuzzy Markov random fields and image segmentation. Graph ». Models Process. 59:205–220, 1997.

- [II.10] S. Ruan, B. Moretti, J. Fadili, and D. Bloyet. Fuzzy Markovian « Segmentation in Application of Magnetic Resonance Images. » *Computer Vision and Image Understanding*, 54–69, 2002.
- [II.11] M.C. Clark, L.O. Hall, D. Goldgof, and M. Silbiger. « Automatic tumor segmentation using knowledge-based techniques ». *IEEE Transactions on Medical Imaging*:187–201, 1998.
- [II.12] A. Zijdenbos and B.M. Dawant. « Brain Segmentation and White Matter Lesion Detection in MR Images ». *Critical reviews in Biomedical Engineering*, 441–465, 1994.
- [II.13] A. Mayer et H. Greenspan : « Segmentation of brain MRI by adaptive mean shift ». Dans *International Symposium on Biomedical Imaging : Macro to Nano*, pages 319–322, avril 2006.
- [II.14] V. Vapnik : « The nature of statistical learning theory ». Springer-Verlag New York, Inc., 1995.
- [II.15] M. Beladgham, F. Derraz, M. Khélif : « Segmentation d'images médicales IRM par la méthode d'ensembles de niveaux (Level\_Sets) » Article, Université Abou- Bekr Belkaid – Tlemcen 2005
- [II.16] T. Lindeberg: «Detecting salient blob-like image structures and their scales with a scale-space»: A method for focus-of-attention. *International Journal of Computer Vision*, 11(3):283-318, 1993.
- [II.17] D. Marr et E. Hildreth: « Theory of edge detection. *Proceedings of the Royal Society in Biological Sciences*, 1980.
- [II.18] G. Matheron « *Random Sets and Integral Geometry* ». Wiley, New York, 1965.
- [II.19] J. Serra. « *Image Analysis and Mathematical Morphology* ». Theoretical Advances, volume 2. Academic Press, London, 1988.
- [II.20] H. Digabel et C. Lantuéjoul : « Iterative algorithm ». *Quantitative analysis of microstructures in materials sciences*, 1:85-99, 1978.
- [II.21] M. Kass, A. Witkin et D. Terzopoulos : « Snakes : Active contour models ». *International Journal of Computer Vision*, 1(4):321-331, 1987.

## Références Bibliographiques

---

- [II.22] T. Cootes et C. Taylor : « Active shape models - smart snakes ». Dans Proceedings of British Machine Vision Conference, 1992.
- [II.23] MAHMOUDI Saïd et MERATI Medjeded : « Une Nouvelle Approche de Segmentation d'Images Médicales par combinaison des deux Modèles Déformables : ASM et Snakes »Article, 2009
- [II.24] T. F. Cootes, G. J. Edwards, and C. J. Taylor. « Active appearance models. In Proc ». Européen Conférence on Computer Vision, pp 484–498, June 1998.
- [II.25] Nicolas EVENO : « Segmentation des lèvres par un modèle déformable analytique » thèse doctorat, INSTITUT POLYTECHNIQUE DE GRENOBLE, 2003.
- [II.26] S. Mallat : « A Wavelet Tour of Signal Processing ». Academic Press, 1998.
- [II.27] K. Pohl, J. Fisher, W. Grimson, R. Kikinis et Wells W. : « A bayesian model for joint segmentation and registration ». Neuro Image, 31:228239, 2006.
- [II.28] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. « interactive foreground extraction using iterated graph cuts ». ACM Trans. Graph., 23(3) :309–314, 2004.
- [III.1] D. Aouali, K. Amzal, mémoire de fin d'études, Segmentation d'images texturées par la méthode des matrices de cooccurrence, Université Mouloud Mammeri, 2010.