

N d'ordre : 10/2014-M/INF

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie Houari Boumediene
Faculté D'ÉLECTRONIQUE et D'INFORMATIQUE



M É M O I R E

présenté pour l'obtention du diplôme de **MAGISTER**

En: INFORMATIQUE

Spécialité: INFORMATIQUE MOBILE

Par: Mohamed Lamine **LAOUIRA**

S U J E T

**Fusion des données dans les réseaux
de capteurs Multimédias.**

Soutenu publiquement, le: **27/04/2014**, devant le jury composé de:

M. S. **LARABI**, Professeur à l'USTHB

Président

M. A. **ABDELLI**, Maître de Conférences A à l'USTHB

Directeur de mémoire

M. D. **DJENOURI**, Maître de Recherche au CERIST

Examineur

M. K. **BEGHDAD BEY**, Maître de Conférences A à l'EMP

Examineur

Fusion des données dans les réseaux de capteurs Multimédias.

Soutenue par : Mohamed Lamine **LAOUIRA**.

Spécialité : **INFORMATIQUE MOBILE**.

Directeur de thèse : Abdelkrim **ABDELLI**, Maître de Conférences **A** à l'USTHB.

R É S U M É.

Dans les réseaux de capteurs sans fil multimédia, une quantité importante de données nécessite d'être traitée, transmise et évaluée selon les objectifs de l'application. Ceci implique une consommation accrue des ressources du réseau pouvant compromettre sa durée de vie. Par conséquent la définition de méthodes pour une gestion efficace des ressources du réseaux représente un défi majeur à surmonter. C'est dans ce contexte que la fusion de données intervient comme solution dédiée. L'objectif principal de la fusion d'image est de combiner les informations pertinentes de plusieurs images en une seule image. Dans ce document nous nous intéressons à la fusion d'images en multifocus dans les réseaux de capteurs multimédia . Nous établissons en premier lieu un diagnostic ainsi qu'une classification des méthodes de fusion de données dans les réseaux de capteur et dans les réseaux de capteurs multimédia en particulier. Ensuite, nous proposons des mécanismes de gestion de la transmission de données dans le réseau afin de minimiser la consommation des ressources tels que l'énergie et la bande passante. Par la suite, nous proposons une approche de fusion des images basée sur la technique DCT(Discrete Cosine Transform). L'approche de fusion proposée utilise comme critère de fusion l'énergie contenue dans un bloc DCT. Les résultats de simulation (sous TinyOS) relatifs à la consommation de l'énergie et le temps d'acheminement des paquets montrent l'efficacité des mécanismes proposés par rapport aux méthodes citées dans l'état de l'art. En ce qui concerne la technique de fusion proposée, les simulations sous Matlab ont montré l'efficacité de la méthode comparativement avec d'autres méthodes de fusion récentes.

D E D I C A C E S

Je dédie cette thèse à :

Mes très chers parents, mon père OMAR et ma mère NOUARA qui n'ont pas cessé de m'encourager et de prier pour moi.

Mon frère ABDOU,

Mes sœurs : AMEL, SABRINA, MOUNIA, DALLAL, IMEN,

Ma femme,

Mes deux fillettes AMINA et HIBET EL RAHMANE,

Mon très cher ami MOUNIR,

Mon collègue de sport AHMED,

Tous mes amis HAKIM, SAID, YACINE...

M.L. LAOUIRA.

R E M E R C I E M E N T S

*Mes remerciements les plus sincères s'adressent en premier lieu à mon directeur de thèse M. **Abdelkrim ABDELLI** pour son sérieux, sa disponibilité et pour ses judicieux conseils qui ont permis d'améliorer la qualité de ce document, Ce travail n'aurait pu voir le jour sans ses conseils toujours pertinents. En une seule phrase, c'était un vrai plaisir de travailler avec lui.*

*Je tiens à remercier M.**Abdelkrim NEMRA** maître de conférence à l'EMP d'abord pour sa gentillesse et son énorme patience avec les multiples captures d'images et expériences que nous avons effectué au cours de la partie expérimentation de ma thèse. Je tiens à le remercier aussi pour avoir contribué à la réalisation de certains des travaux présentés dans cette thèse.*

*Mes remerciements s'adressent aussi au M.**Hakim AOUAD** pour les efforts fournis afin d'améliorer la qualité de rédaction du présent document.*

*Je souhaite aussi exprimer mes vifs remerciements à M.**Hamid HENTOUS** (the coach) le chef d'U.E.R informatique de l'EMP, M.**Adnane BENAÏSSA** le chef du laboratoire systèmes au niveau de l'U.E.R informatique de m'avoir accueilli au sein du laboratoire systèmes dans le quel ce travail a été réalisé.*

Je remercie également tous les cadres de l'U.E.R informatique et l'U.E.R automatique de l'EMP qui ont contribué de proche ou de loin à ce travail.

Pour finir, je remercie mes parents et toute ma famille, de m'avoir soutenu et encouragé pendant toutes mes études et surtout pendant ces deux dernières années.

M.L. LAOUIRA.

R É S U M É

Dans les réseaux de capteurs sans fil multimédia, une quantité importante de données nécessite d'être traitée, transmise et évaluée selon les objectifs de l'application. Ceci implique une consommation accrue des ressources du réseau pouvant compromettre sa durée de vie. Par conséquent la définition de méthodes pour une gestion efficace des ressources du réseaux représente un défi majeur à surmonter. C'est dans ce contexte que la fusion de données intervient comme solution dédiée. L'objectif principal de la fusion d'image est de combiner les informations pertinentes de plusieurs images en une seule image. Dans ce document nous nous intéressons à la fusion d'images en multifocus dans les réseaux de capteurs multimédia . Nous établissons en premier lieu un diagnostic ainsi qu'une classification des méthodes de fusion de données dans les réseaux de capteur et dans les réseaux de capteurs multimédia en particulier. Ensuite, nous proposons des mécanismes de gestion de la transmission de données dans le réseau afin de minimiser la consommation des ressources tels que l'énergie et la bande passante. Par la suite, nous proposons une approche de fusion des images basée sur la technique DCT(Discrete Cosine Transform). L'approche de fusion proposée utilise comme critère de fusion l'énergie contenue dans un bloc DCT. Les résultats de simulation (sous TinyOS) relatifs à la consommation de l'énergie et le temps d'acheminement des paquets montrent l'efficacité des mécanismes proposés par rapport aux méthodes citées dans l'état de l'art. En ce qui concerne la technique de fusion proposée, les simulations sous Matlab ont montré l'efficacité de la méthode comparativement avec d'autres méthodes de fusion récentes.

Mots-clés : Réseaux de capteurs sans fil, Multimédia, consommation d'énergie, fusion de données, fusion d'images en Multi focus, DCT (Discrete Cosine Transform), PSNR, Energie.

A B S T R A C T

In multimedia wireless sensor networks a large amount of data needs to be processed, delivered, and assessed according to the application goals. This leads to an increased usage of the resources of the network thus reducing its life time. Consequently, the definition of appropriate methods for managing efficiently the resources of the network becomes a challenging issue to overcome. Within this context, data aggregation techniques have been proposed as dedicated solutions. The main objective of such an approach is to synthesize relevant information from multiple images to obtain a single self-contained image. In this document, we deal with Multi focus image fusion in the context of multimedia wireless sensor network. First of all, we establish a diagnosis and classification of more recent methods of data fusion that have been addressed in the context of sensor networks and in multimedia sensor networks in particular. Afterwards, we put forward several mechanisms for managing efficiently the transmission of image data in the network in order to reduce the resource consumption such as energy and bandwidth. Finally, we propose a novel technique based on DCT (Discrete Cosine Transform) that uses the energy containing in image blocks as a tool to decide data fusion. The simulation results (under tinyOS) related to resource consumption and transmitting delays show the efficiency of the proposed mechanisms comparatively to followed approaches. The same statement is reported when evaluating our proposed fusion technique; simulations under Matlab show good results in terms of image quality.

Keywords : Wireless Sensor Networks, Multimedia, energy consumption, fusion images en multifocus, DCT (Discrete Cosine Transform), Energy, PSNR.

Table des matières

Table des matières	vi
Liste des figures	ix
Liste des tableaux	xii
Introduction générale	1
1 Introduction aux WSN et aux WMSN	5
1.1 Introduction	5
1.2 Présentation des réseaux de capteurs sans fil (WSN)	6
1.2.1 Description d'un nœud capteur	6
1.2.2 Description d'un réseau de capteurs sans fil	8
1.2.3 Caractéristiques et contraintes des WSN	8
1.2.4 Consommation de l'énergie dans un réseau de capteur	10
1.2.5 Domaines d'application des WSN	12
1.2.6 Type des WSN :	16
1.3 Présentation des réseaux de capteurs Multimédia sans fil (WMSN)	17
1.3.1 Architecture des WMSN	17
1.3.2 Facteurs influant la conception des WMSN	19
1.3.3 Défis des réseaux de capteurs multimédia sans fil	21
1.3.4 Domaines d'application des WMSN	23
1.3.5 Problématiques liées aux réseaux de capteurs multimédia	24
1.4 Conclusion	25
2 Fusion de données dans les réseaux de capteurs	26
2.1 Introduction	26
2.2 Présentation du concept : fusion de données	28
2.2.1 Définition	28
2.2.2 Classification de la fusion de données	28
2.2.3 Concepts important dans la fusion de données dans les WSN	30
2.3 La fusion de données dans les WMSN	31
2.3.1 La fusion d'images dans les WMSN	31

2.3.2	Concepts important dans le domaine de fusion d'images :	33
2.3.3	Le concept Multi-focus image fusion (MFIF) :	34
2.3.4	Evaluation des images fusionnées	36
2.4	Conclusion	40
3	Les méthodes de fusion de données dans les réseaux de capteurs	41
3.1	Introduction	41
3.2	Méthodes et algorithmes de fusion dans les WSN	41
3.2.1	Méthodes par inférence :	42
3.2.2	Méthodes par estimation :	49
3.2.3	Méthodes par agrégation :	51
3.3	Méthodes et algorithmes de fusion dans les WMSN	60
3.3.1	Méthodes de fusion niveau décisionnel (Decision Level Based Fusion)	61
3.3.2	Méthodes de fusion niveau caractéristique (Featur Level Based Fusion)	64
3.3.3	Méthodes de fusion niveau Pixel (Pixel Level Based Fusion)	69
3.4	Conclusion	87
4	Conception de la solution proposée	88
4.1	Introduction	88
4.2	Le contexte	89
4.3	Architecture proposée	93
4.3.1	La couche de capture	94
4.3.2	La couche de fusion	95
4.3.3	La couche de traitement et de décision	96
4.4	Description du Fonctionnement de la solution proposée	96
4.4.1	Formalisation	97
4.4.2	Description de la solution au niveau capteur	97
4.4.3	Description de la solution au niveau agent de fusion	106
4.4.4	Description de la solution au niveau décision et traitement	110
4.4.5	Protocole de routage	112
4.5	Organigramme de la solution	114
4.6	Conclusion	115
5	Tests et validation	116
5.1	Introduction	116
5.2	Démarche adoptée pour la simulation	116
5.2.1	Simulation de la couche capteur	117
5.2.2	Evaluation de la qualité des images au niveau couche agent de fusion	117

5.3	Outils de simulation	118
5.3.1	TinyOS	118
5.3.2	Matlab	120
5.4	Description détaillée de la simulation	121
5.4.1	Simulation de la couche capteur	121
5.4.2	Simulation au niveau agent de fusion	131
5.5	Conclusion	151
	Conclusion générale	152
	Bibliographie	154

Liste des figures

1.1	Architecture d'un nœud capteur.	7
1.2	Exemple de réseau de capteurs sans fil.	8
1.3	(a)- Transmissions directes. (b)- Transmission saut par saut. (c) Hiérarchisation en clusters.	12
1.4	Capteur de type Mica2 Mote.	16
1.5	Architecture de base des WMSN.	18
1.6	Les paramètres de la configuration d'un capteur directionnel : α = angle avec la verticale, θ = ouverture de la caméra, R_s = rayon de détection.	22
2.1	Intégration, fusion et agrégation de données.	27
2.2	Etapes du processus d'intégration de données.	27
2.3	Fusion selon les relations entre les entrées.	29
2.4	Exemple de combinaison des trois niveaux de fusion.	32
2.5	Cordonnées cartésiennes 2-D d'une image numérique de taille $M \times N$	33
2.6	L'alignement des images sources dans le processus de fusion.	34
2.7	Evaluation de la qualité des images fusionnées.	37
3.1	Architecture générale des algorithmes de fusion de données.	42
3.2	Processus d'agrégation de données.	60
3.3	Moteur de fusion proposé.	61
3.4	Architecture distribuée du système.	65
3.5	Architecture de fusion proposée.	67
3.6	La DWT avec 2-niveaux de décomposition.	71
3.7	Architecture proposée dans [1]	73
3.8	Approche combinant SFVIVI et SFVISF	76
3.9	Coefficients AC et DC d'un bloc de 8×8 pixels.	79
3.10	Exemple d'une matrice d'entrée et sa matrice DCT.	79
3.11	Exemple d'un bloc DCT.	80
3.12	PSNR obtenue pour WTF, DCT+average et DCT+contrast.	82
3.13	Le processus de fusion DCT+variance.	84
3.14	Échantillon d'images utilisées pour la simulation.	84
4.1	Faible profondeur de champ.	90

4.2	Grand profondeur de champ.	91
4.3	Image zoomée sur la partie gauche.	91
4.4	Image zoomée sur la partie droite.	92
4.5	Image souhaitée après fusion.	92
4.6	Architecture générale de la solution proposée.	94
4.7	Temps d'exécution d'une DCT sur une image découpée en blocs de 8×8 pixels.	98
4.8	Temps d'exécution d'une DCT sur une image entière.	99
4.9	Algorithme de découpage d'une image en blocs de 8×8 pixels.	99
4.10	Bloc de 8×8 pixels d'une image prise avec éclairage élevé.	101
4.11	Bloc de 8×8 pixels d'une image prise avec éclairage faible.	102
4.12	Les quatre blocs de 8×8 pixels de l'image A.	104
4.13	Les quatre blocs de 8×8 pixels de l'image B.	105
4.14	Processus de vérification de la consistance.	110
4.15	Diagramme de la solution proposée.	112
4.16	Processus global de la solution.	115
5.1	GUI développée pour tester les différentes techniques de fusion.	121
5.2	Images utilisées pour la simulation du premier scénario.	122
5.3	Topologie réseau avec 50 nœuds, 1 :Source et 50 :Agent de fusion.	123
5.4	Code Matlab de la fonction de comparaison des images(références et capturées)	125
5.5	Images utilisées pour la simulation du premier (1^{er}) cas du deuxième scénario.	126
5.6	Images utilisées pour la simulation du deuxième (2^{eme})cas du deuxième scénario.	127
5.7	Consommation de l'énergie lors de l'envoi de données.	129
5.8	Temps d'acheminement des blocs de données.	130
5.9	Images sources et images fusionnées pour l'image ' Clock '. (a) : la première image source avec focus à droite. (b) : la deuxième image source avec focus à gauche. (c) : image fusionnée en utilisant la moyenne. (d) : image fusionnée en utilisant l'écart type. (e) : image fusionnée en utilisant la variance. (f) : image fusionnée en utilisant l'énergie d'un block (technique proposée). (g) : la version zoomée de (e). (h) : la version zoomée de (f).	132
5.10	Images sources et images fusionnées pour l'image ' Church '. Même ordre que la figure 5.9	133
5.11	Images sources et images fusionnées pour l'image ' Bottle '. Même ordre que la figure 5.9	134
5.12	Images sources et images fusionnées pour l'image ' Book '. Même ordre que la figure 5.9.	135

5.13	Images sources et images fusionnées pour l'image ' Plane '. (a) : la première image source avec focus en bas. (b) : la deuxième image source avec focus en haut. (c) : image fusionnée en utilisant la moyenne. (d) : image fusionnée en utilisant l'écart type. (e) : image fusionnée en utilisant la variance. (f) : image fusionnée en utilisant l'énergie d'un block (technique proposée). (g) : la version zoomée de (e). (h) : la version zoomée de (f).	136
5.14	Capture des images multi-focus réelles dans le laboratoire robotique et productique de l'EMP.	137
5.15	Camera SONY de type DFW-X710 utilisée pour la capture des images multi-focus réelles.	138
5.16	Images sources et images fusionnées pour l'image ' Marker '. (a) : la première image source avec focus à droite. (b) : la deuxième image source avec focus à gauche. (c) : image fusionnée en utilisant la variance. (d) : image fusionnée en utilisant l'énergie d'un block (technique proposée). .	139
5.17	Images sources et images fusionnées pour l'image ' Box '. Même ordre que la figure 5.16.	140
5.18	Images sources et images fusionnées pour l'image ' Equipement '. Même ordre que la figure 5.16.	141
5.19	Images sources et images fusionnées pour l'image ' Nike '. Même ordre que la figure 5.16.	142
5.20	Images sources et images fusionnées pour l'image ' Quran '. Même ordre que la figure 5.16.	143
5.21	Images sources et images fusionnées pour l'image ' Robot '. Même ordre que la figure 5.16.	144
5.22	Comparaison des résultats des deux techniques <i>DCT+Varaiance (e)</i> et <i>DCT+Energy (f)</i> sur l'image Bottle	145
5.23	Comparaison des résultats des deux techniques <i>DCT+Varaiance (e)</i> et <i>DCT+Energy (f)</i> sur l'image Plane	146

Liste des tableaux

1.1	Caractéristiques physiques des capteurs Mica et Mica2.	7
3.1	Tableau comparatif des méthodes de fusion par inférence.	49
3.2	Tableau Comparatif des méthodes de fusion par estimation.	51
3.3	Tableau comparatif des méthodes de calcul de la moyenne consensus dans un WSN.	56
3.4	Tableau comparatif des protocoles d'agrégation dans un WSN.	59
3.5	Tableau Comparatif des méthodes de fusion niveau décisionnel.	64
3.6	Moyenne et écart-type en pixels de la distance entre les positions estimées et les positions tirées de la carte.	65
3.7	Erreur moyenne et écart type des capteurs séparés et l'approche de fusion.	67
3.8	Erreur moyenne et écart type des trios capteurs dans l'approche de fusion.	68
3.9	Tableau Comparatif des méthodes de fusion niveau caractéristique.	70
3.10	Paramètres de performances et comparaison des méthodes de fusion.	73
3.11	La fréquence spatiale.	77
3.12	La visibilité.	77
3.13	La valeur moyenne du SSIM des algorithmes.	85
3.14	La valeur moyenne de l'exécution des algorithmes en microsecondes par bloc 8×8	85
3.15	Evaluation des deux l'algorithmes proposés par rapport aux autres algorithmes en utilisant les métriques Piella et Petrovic.	85
3.16	Tableau Comparatif des méthodes DWT et DCT.	86
4.1	Interprétation du coefficient $T_{B_k}^i$	103
4.2	Algorithme niveau capteur	106
4.3	Algorithme niveau Agent de fusion	110
5.1	Taille et nombre de blocs 8×8 des images itilisées pour la simulation du premier scénario.	122
5.2	Energie consommée lors de l'envoi de l'image entière (la totalité des blocs).	124
5.3	Temps total d'acheminement des blocs lors de l'envoi de l'image entière.	124
5.4	Valeur du seuil T_G^i dans les deux cas de simulation du deuxième scénario.	127

5.5	Résultats de la fonction <code>imgComp</code> sur un ensemble d'images (références et capturées).	128
5.6	Energie consommée lors de l'envoi de la totalité des blocs, énergie consommée lors de l'envoi des blocs modifiés et le gain en termes d'énergie.	129
5.7	Temps d'acheminement des blocs lors de : l'envoi de la totalité de l'image, l'envoi des blocs modifiés et le gain en termes de temps.	130
5.8	Ecart type des images fusionnées avec les quatre techniques de fusion.	147
5.9	PSNR entre les images fusionnées obtenues par les différentes techniques de fusion et les images de référence 'Church et Bottle'.	147
5.10	Métrique de Petrovic de quelques images fusionnées avec les quatre techniques de fusion.	148
5.11	Temps d'exécution des quatre techniques de fusion (en secondes).	148
5.12	Ecart type des images réelles fusionnées avec les techniques DCT+Mean, DCT+S.D, DCT+Variance et DCT+Energy.	149
5.13	Le PSNR des images réelles fusionnées avec les techniques DCT+Mean, DCT+S.D, DCT+Variance et DCT+Energy.	149
5.14	La métrique de Petrovic pour des images réelles fusionnées avec les techniques DCT+Mean, DCT+S.D, DCT+Variance et DCT+Energy.	150
5.15	Temps d'exécution (en secondes) des techniques DCT+Mean, DCT+S.D, DCT+Variance et DCT+Energy sur quelques images réelles.	150

Introduction générale

Domaine d'intérêt

Dans la plupart des architectures réseaux actuelles, la technologie de communication sans fil est devenue un acteur incontournable grâce à ses diverges avantages, cette nouvelle technologie de communication a donné naissance à diverses architectures à savoir : Les réseaux locaux sans fils, les réseaux cellulaires et autres,....

D'un autre coté, depuis les années 90, la technologie des microsystèmes électromécaniques (*MEMS*) a connu elle aussi un avancement considérable. Ces deux facteurs ont permis l'émergence des *Réseaux de Capteurs Sans Fil (RCSF)* (*Wireless Sensor Networks (WSN)* en anglais) [2]. Un réseau de capteurs sans fil est une collection de nœuds organisés en un réseau coopératif. Un nœud représente un capteur (soit un équipement de petite taille), pouvant détecter, traiter et communiquer des informations telles que la température, la pression atmosphérique, ou d'autres grandeurs physiques.

Les réseaux de capteur sans fil ont intéressé plusieurs domaines de recherche à savoir : la santé, l'industrie et le domaine militaire. Ils peuvent être considérés comme étant les successeurs des réseaux ad hoc (*MANET*) avec lesquels ils partagent plusieurs propriétés en commun, telles que l'absence d'infrastructure et les communications sans fil. Cependant, les protocoles et les algorithmes proposés pour les réseaux Ad hoc traditionnels semblent inappropriés pour les WSN. Concrètement, initialement les capteurs furent principalement dédiés à la mesure de certains paramètres environnementaux mais grâce à l'évolution de la micro électronique, l'intégration par la suite de petites caméras et microphones est devenue possible. Ceci a donné naissance à ce que nous appelons aujourd'hui *un capteur multimédia* ou d'une manière générale aux *Réseaux de Capteurs Multimédia Sans Fil (RCMSF)* (*Wireless Multimedia Sensors Networks (WMSN)* en anglais), [3].

Un *WMSN* est un réseau de dispositifs interconnectés sans fil permettant d'obtenir des flots audio et vidéo, voire d'images inanimées, et ce en plus des données scalaires traditionnelles. Donc, les composantes d'un tel réseau sont appelées à stocker, traiter, corrélérer et fusionner en temps réel des données multimédias provenant de sources hétérogènes [3].

En plus des applications des *WSN*, les *WMSN* peuvent être utilisés pour la surveillance des frontières ainsi que pour le contrôle du trafic routier. Aujourd'hui la plu-

part des réseaux de capteur utilisés pour capter des données scalaires ont des exigences faibles en bande passante et une tolérance considérable de retard contrairement aux réseaux de capteurs multimédia qui exigent des paramètres de QoS et une bande passante importante pour la communication entre nœuds.

Problématique

Le rôle principal d'un réseau de capteurs est de collecter des informations dignes d'intérêt, et ce, soit pour prendre une décision soit pour les envoyer à l'utilisateur final pour traitement [4]. Ces nœuds capteurs sont déployés d'une manière dense dans le champ de captage. En raison de cette forte densité de déploiement, une quantité importante de données est généralement produite ce qui favorise une redondance dans leur transmission et leur stockage, résultant ainsi de l'épuisement des ressources déjà limitées telles que : l'énergie, la mémoire et la bande passante. Par conséquent, la collecte et la fusion des données (*agrégation*), dans les réseaux de capteurs sans fil sont d'une importance capitale.

La fusion de données dans les réseaux de capteurs consiste essentiellement à confronter et à intégrer des informations multiples provenant de plusieurs capteurs, pour former une seule information globale. Par exemple, trouver l'agrégation de plusieurs mesures (température, humidité, etc) prises à différents endroits par les capteurs [4]. En d'autres termes, la fusion des données est le processus qui consiste au traitement de l'information collectée avant de l'envoyer à l'utilisateur final. Ce processus est très utile dans les réseaux de capteurs sans fil. Les nœuds capteurs ont la capacité de transmettre les données entre eux ou vers une station de base ou SINK. Cette transmission de données est pratiquement directe entre capteurs mais à plusieurs sauts pour atteindre la station de base, ce qui la rend coûteuse. De cet fait, les nœuds du réseau ont besoin d'une quantité considérable d'énergie pour véhiculer ces données. Lorsqu'un nombre limité de nœud effectue cette tâche de transmission, on parle alors de *Clustering* et de *Cluster head* pour désigner le partitionnement du réseau.

La fusion des données dans un réseau de capteurs peut aider à améliorer la précision de l'information captée par différents nœuds voisins. Elle permet aussi d'éliminer les redondances existantes dans les données recueillies ce qui réduit le trafic engendré sur le réseau et par conséquence l'énergie consommée.

Dans un réseau de capteurs multimédias, quelque soit la résolution des caméras utilisées pour la capture des images, ces caméras souffrent d'un problème de limitation de la profondeur du champ de vision. Seuls les objets à l'intérieur du focus apparaissent

avec précision, alors que les autres objets en dehors du champs du focus apparaissent flous voire brouillés. Pour produire une image où tous les objets de la scène sont en focale, on utilise plusieurs caméras opérant en *Multi-Focus*. Une opération de fusion devient nécessaire pour synthétiser les différentes images captées par les différentes caméras. Ce processus de fusion de ce type d'images est appelé *Multi-Focus Image Fusion (MFIF)*. Le travail défendu dans ce mémoire vise principalement en la contribution dans ce domaine de recherche.

Objectif et Méthodologie

Le premier objectif de ce document est de faire un diagnostic des méthodes de fusion et d'agrégation de données dans les réseaux de capteur d'une manière générale et dans les réseaux de capteurs multimédia en particulier. A travers cette étude, nous passons en revue et discutons des principales techniques utilisées dans les réseaux de capteur. Nous proposons une classification de l'ensemble de ces méthodes selon plusieurs critères que nous discutons et établissons une comparaison entre ces techniques.

Notre contribution s'oriente vers la proposition de solutions destinées à la fusion d'images en Multifocus dans le contexte d'un réseau de capteurs multimédia. Nos propositions s'articulent autour de deux objectifs : le premier consiste en la proposition de mécanismes de gestion de l'envoi de données multimédias dans un réseau de capteurs afin de réduire la quantité de données circulant sur le réseau ; le second objectif comprend l'élaboration d'une technique de fusion en multi focus pour les images capturées.

A ce jour, la plupart des méthodes de fusion destinées à fonctionner dans un réseau de capteur multimédia donnent de l'importance uniquement au processus de fusion lui même (qualité de l'image fusionnée...) et négligent certains paramètres importants tels que la consommation de l'énergie et les ressources du réseau qui conditionnent sa durée de vie. Cela est remarquable à travers les travaux les plus récents, comme ceux de M.Haghighat 2011 [5] et J. Tang 2004 [6]. où les auteurs ont proposé deux techniques de fusion d'image dans un réseau de capteurs multimédias.

Dans ces travaux aucun mécanisme de gestion ou de contrôle de l'envoi des images entre les nœuds collecteurs (chargés de la capture) et les agents de fusion (chargés de la fusion de données) n'a été introduit. L'image ainsi capturée est envoyée à l'agent de fusion dans sa totalité et aucune mesure de performance sur la gestion des ressources du réseau n'a été formulée.

Dans ce document nous tentons de répondre à ces insuffisances par la proposition de mécanismes de gestion et de contrôle de l'envoi des données capturées afin de mieux gérer les ressources du réseau (énergie, bande passante, mémoire, CPU,..) et rallonger

par conséquent la durée de vie du réseau. Notre deuxième contribution ce résumé en la proposition d'une nouvelle technique de fusion des images en multi-focus basée sur les concepts de la transformée en cosinus DCT (Discrete Cosine Transform) et de l'énergie pour la comparaison entre blocs. Les résultats de simulation obtenus ont montré l'efficacité en termes de gain du mécanisme de gestion et de contrôle que nous avons proposé. De plus la technique de fusion de données proposée a rendu de meilleurs résultats en termes de qualité visuelle comparativement aux techniques citées dans l'état de l'art. Cela est remarquable à vue d'œil et à travers les métriques d'évaluation des images fusionnées (calcul du PNSR, et d'autres métriques).

Organisation du document

Pour atteindre les objectifs mentionnés ci-dessus, nous avons structuré notre document en cinq (05) chapitres en plus de l'introduction générale et de la conclusion générale.

- Le premier chapitre, introduit les réseaux de capteurs sans fil, ainsi que les réseaux de capteurs multimédia sans fil. En premier lieu, nous présentons les WSN d'une manière générale et donnons leurs domaines d'application. En second lieu, nous décrivons l'architecture des WMSN, et discutons des facteurs influant leur conception, de leurs domaines d'application, et des problématiques inhérentes à leur fonctionnement.
- Le second chapitre est consacré aux concepts de fusion de données dans les réseaux de capteurs.
- Le troisième chapitre représente un état de l'art sur les techniques de fusion de données. Nous présentons les différentes méthodes de fusion et d'agrégation des données dans les WSN et les WMSN. Ensuite, nous discutons et comparons ces méthodes et clôturons chaque section par un tableau comparatif pour chaque classe de méthodes.
- Le quatrième chapitre décline notre contribution. Nous y présentons notre solution en donnant une architecture et une description détaillée de sa conception.
- Le cinquième et dernier chapitre est consacré à l'implémentation de notre solution ainsi qu'aux différents tests et simulations. Une évaluation des performances de l'approche de fusion proposée comparativement à d'autres solutions est donnée également.

Finalement, la conclusion générale conclut le travail réalisé et met en relief les perspectives de recherche et son application dans le domaine considéré.

Chapitre 1

Introduction aux WSN et aux WMSN

1.1 Introduction

Les Réseaux de Capteurs Sans-fil (WSN), ou "réseaux de capteurs", est une technologie émergente issue du progrès dans différents domaines. Plusieurs facteurs ont contribué à l'émergence de cette technologie : la miniaturisation des composants électroniques, l'augmentation de leur capacité (puissance de calculs, énergie, etc.), la diminution des coûts de fabrication et le développement des techniques de communication sans fil grâce à l'essor de la téléphonie mobile,... etc .

Par la suite, l'évolution récente de la technologie a permis de passer d'un modèle où les capteurs sans fil étaient principalement dédiés à la mesure de paramètres scalaires environnementaux comme la température ou l'humidité à des capteurs équipés de capacités multimédia. En effet, la disponibilité du matériel à coût abordable tels que les caméras et microphones CMOS (Complementary Metal Oxide Semiconductor) a permis d'embarquer ces dispositifs dans un capteur. Dans ce chapitre, nous présentons les principaux concepts liés aux réseaux de capteurs. Nous donnons une description d'un nœud capteur, d'un réseau de capteur ainsi que les différents domaines d'applications, les facteurs influant sur la conception des réseaux de capteurs et les types des WSN. Nous décrivons aussi les réseaux de capteurs multimédia sans fil en présentant leur architecture, les facteurs influant sur leur conception, leurs domaines d'application et les principales problématiques observées dans ces réseaux.

1.2 Présentation des réseaux de capteurs sans fil (WSN)

1.2.1 Description d'un nœud capteur

Un nœud capteur est défini généralement comme étant un dispositif qui produit une réponse mesurable suite à un changement de l'état physique ou chimique de son environnement. Plus spécifiquement, un nœud capteur est un dispositif qui répond à un stimulus externe (tel que la chaleur, la lumière, la pression, etc.) et qui produit un signal qui peut être mesuré ou interprété. En outre, les nœuds capteurs sont définis comme étant de petits dispositifs sans fil capables de répondre à un ou plusieurs stimuli, de traiter les données et de les transmettre sur une petite distance en utilisant la radio [7]. Les architectures des nœuds capteurs sont quasi-identiques. Comme le montre la figure 1.1, une telle architecture est composée de quatre unités de base, une unité de détection, une unité de traitement, une unité émettrice-réceptrice (Transceiver ou unité de transmission) et une unité d'énergie. Parfois, le type d'application exige l'ajout de composants supplémentaires tels que : un système de localisation, un générateur d'énergie, un mobilisateur [8].

- L'unité de détection ou d'acquisition est composée d'un capteur relevant des mesures sur des paramètres environnementaux et d'un convertisseur analogique-numérique (ADC) convertissant la donnée relevée en une donnée numérique afin de l'envoyer à l'unité de traitement.
- L'unité de traitement est constituée de deux interfaces : une interface pour l'unité d'acquisition et une autre pour l'unité de transmission. Elle reçoit les informations en provenance de l'unité d'acquisition et les envoie, éventuellement après traitement, à l'unité de transmission. Cette unité est également composée d'un processeur sur lequel tourne un système d'exploitation spécifique.
- L'unité de transmission est responsable de toutes les émissions et les réceptions des données via le médium de communication radio.
- L'unité d'énergie est le composant le plus important du capteur, la fonctionnalité de cette unité peut être assurée par des techniques tels que les cellules solaires.
- Le système de localisation est parfois nécessaire pour déterminer la position des nœuds capteurs du moment que ces derniers sont déployés d'une manière aléatoire dans la zone d'intérêt.
- Le générateur d'énergie, comme son nom l'indique a pour rôle de fournir de l'énergie au nœud en cas de besoin.
- Un mobilisateur est nécessaire pour le déplacement d'un nœud afin d'effectuer

une tâche spécifique.

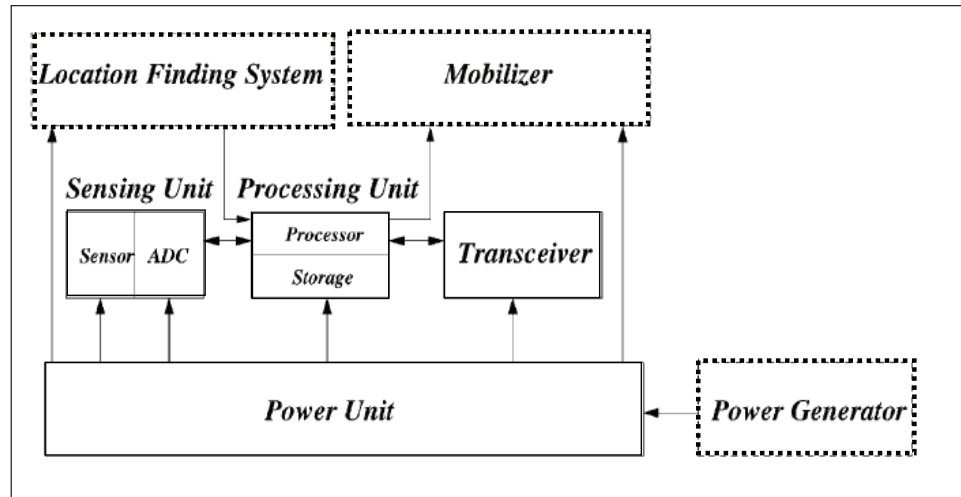


Figure 1.1 – Architecture d'un nœud capteur.

Plusieurs entreprises start-up fabriquent des nœuds capteurs. Nous citons, entre autres, le leader américain Crossbow qui produit différentes marques de nœuds capteurs, à savoir : Mica, Mica2, Telos et Telosb. Le tableau 1.1 illustre les caractéristiques physiques des nœuds capteurs Mica et Mica2 [9].

Caractéristiques	Mica	Mica2
Type du CPU	Atmel 128	Atmel 128
Fréquence du CPU	4 Mhz	8 Mhz
Mémoire de données du CPU	4 KB	4 KB
Mémoire de programmes du CPU	128 KB	128 KB
Type de liaison Radio RFM	TR1000	ChipCon
Fréquence Radio	917 Mhz	917 Mhz, 433 Mhz
Sortie Radio	40 Khz	39.2 Khz
Type du Flash	Atmel AT45DB041	Atmel AT45DB041
Capacité du Flash	512 KB	512 KB
Temps d'écriture du Flash	10 ms / 256 bytes	10 ms / 256 bytes

Tableau 1.1 – Caractéristiques physiques des capteurs Mica et Mica2.

1.2.2 Description d'un réseau de capteurs sans fil

Un réseau de capteurs est un ensemble de nœuds capteurs, déployés dans un environnement d'intérêt, appelée champ de capture, pour capturer des données relatives à des phénomènes physiques. Les données capturées sont acheminées grâce à un routage multi-sauts à un nœud considéré comme "point de collecte", appelé aussi nœud collecteur supposé avoir plus de capacités relativement aux autres nœuds du réseau. Ce nœud peut représenter directement le dépôt final (station de base), comme pour le cas des PDAs, ou bien il peut agir comme une passerelle entre la station de base et le réseau de capteurs en passant par un réseau de liaison intermédiaire. Selon l'application visée, un réseau de capteurs peut éventuellement contenir plusieurs stations de base. Les nœuds capteurs peuvent être déployés de façon déterministe ou aléatoire (largués à partir d'un avion, engin autonome volant ou autre). Ceci en fonction de la nature de l'application, le type du champ de capture, le coût de déploiement, etc. La figure 1.2 schématise un réseau de capteurs connecté à un système de communication global.

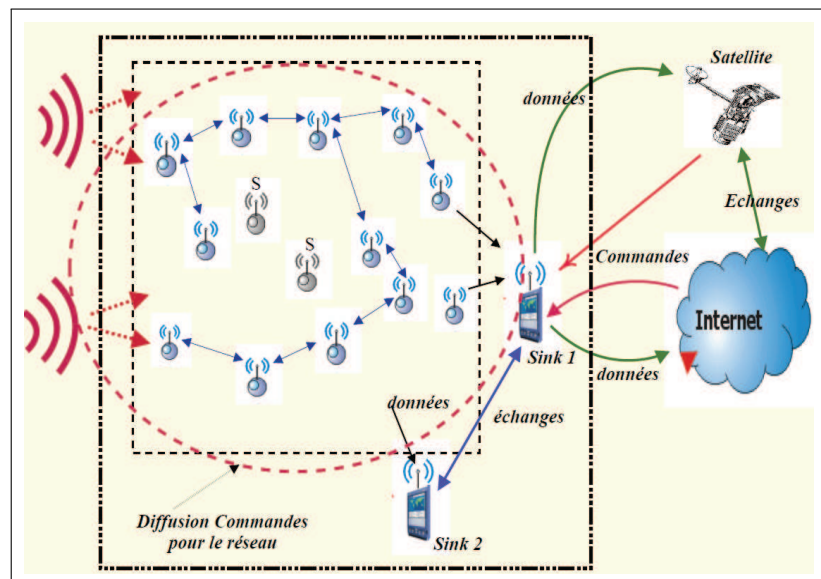


Figure 1.2 – Exemple de réseau de capteurs sans fil.

1.2.3 Caractéristiques et contraintes des WSN

Un WSN présente les caractéristiques et les contraintes suivantes[10] :

- **Absence d'infrastructure** : les réseaux mobiles Ad hoc et particulièrement les WSN se distinguent des autres réseaux par la propriété d'absence d'infrastructure pré-existante.

- **Taille importante (densité) :** la densité des nœuds est plus élevée dans les WSN que dans les réseaux mobiles ad hoc. Le nombre de nœuds capteurs peut atteindre des millions pour permettre une meilleure granularité de surveillance.
- **Modèle de communication :** les nœuds dans les WSN communiquent selon un paradigme plusieurs-à-un (many to one). En effet, les nœuds capteurs collectent des informations à partir de leur environnement et les envoient vers un seul nœud qui représente le centre de traitement.
- **Interférences :** les liens radio ne sont pas isolés ; deux transmissions simultanées sur une même fréquence en utilisant des fréquences proches, peuvent s'interférer.
- **Topologie dynamique :** les nœuds capteurs peuvent être attachés à des objets mobiles qui se déplacent d'une façon libre et arbitraire rendant ainsi la topologie du réseau fréquemment changeante.
- **Sécurité physique limitée :** les WSN sont plus touchés par le paramètre de sécurité que les réseaux filaires classiques. Cela se justifie par des contraintes et limitations physiques, qui font que le contrôle des données transférées doit être minimisé. En plus, les nœuds capteurs eux-mêmes sont des points de vulnérabilité du réseau car, ils peuvent être modifiés, remplacés ou supprimés.
- **Bande passante limitée :** une des caractéristiques primordiales des réseaux basés sur la communication sans fil est l'utilisation d'un médium de communication partagé. Ce partage fait que la bande passante réservée à un nœud est limitée.
- **Passage à l'échelle :** les WSN peuvent contenir des centaines voire des milliers de nœuds capteurs. Un nombre aussi important de nœuds engendre beaucoup de transmissions inter-nœuds et nécessite que le nœud Collecteur soit équipé d'une mémoire importante pour stocker les informations reçues.
- **Contrainte d'énergie, de stockage et de calcul :** la caractéristique la plus critique dans les WSN est la limitation de ses ressources énergétiques car, chaque nœud capteur possède des faibles ressources en termes d'énergie (batterie). Afin de prolonger la durée de vie du réseau, une minimisation des dépenses énergétiques est exigée dans chaque nœud. Ainsi, la capacité de stockage et la puissance de calcul sont limitées dans ce dernier. Nous évoquons en détails la contrainte d'énergie dans la sous-section suivante.

1.2.4 Consommation de l'énergie dans un réseau de capteur

Niveaux de consommation de l'énergie

La durée de vie d'un nœud dépend fortement de la durée de vie de sa batterie, pour cette raison la conservation de l'énergie et la gestion de l'alimentation sont d'une importance capitale. Dans ce contexte, les recherches se sont concentrées sur la conception de protocoles et d'algorithmes économes en énergie pour les réseaux de capteurs. Dans les autres réseaux mobiles, la consommation d'énergie est aussi un facteur important mais de moindre importance car les ressources d'énergie peuvent être remplacées par l'utilisateur. La tâche principale d'un nœud capteur est de détecter les événements, effectuer un traitement local des données, puis transmettre les données. La consommation de l'énergie peut donc être envisagée en trois niveaux : la détection, la communication et le traitement des données.

- **La détection** : Durant cette période la consommation de l'énergie dépend directement du type de l'application en question, une détection qui n'est pas régulière (sporadique), consomme moins d'énergie par rapport à une surveillance continue d'un événement. En plus, la complexité de la détection d'évènement joue un rôle crucial dans la détermination de la consommation de l'énergie, cette complexité est influencée par le bruit ambiant.
- **La communication** : C'est dans cette partie que les nœuds capteurs dépensent plus d'énergie, cela implique la transmission et la réception des données. Dans ce contexte, nous ne considérons pas seulement l'énergie de fonctionnement, mais aussi la consommation d'énergie dans le Transceiver lors de l'opération de démarrage.
- **Traitement de données** : La consommation de l'énergie dans cette phase est beaucoup moindre par rapport à la phase précédente, le traitement local des données est crucial pour réduire au minimum la consommation d'énergie dans un réseau de capteurs multi-sauts. Il a été démontré que l'énergie nécessaire pour transmettre 1 KB sur une distance de 100m est approximativement la même qu'exécuter 3 millions d'instructions par second.

Mécanismes de conservation de l'énergie

Comme nous l'avons déjà signalé, la communication dans un réseau de capteurs est la tâche la plus consommatrice d'énergie par rapport aux autres tâches d'un nœud capteur. Nous introduisons dans le paragraphe suivant certains mécanismes de base pour la conservation de l'énergie au sein d'un réseau de capteurs.

- **Mode d'économie d'énergie** : Cela consiste à éteindre le module de communication dès que possible. Par exemple, des protocoles MAC fondés sur la méthode TDMA (Time Division Multiple Access) offrent une solution implicite puisqu'un nœud n'échange des messages que dans les intervalles de temps qui lui sont attribués. Il peut alors garder sa radio éteinte durant les autres slots. Comme nous l'avons souligné précédemment, il faut toutefois veiller à ce que le gain d'énergie obtenu en mettant en veille le module radio ne soit pas inférieur au surcoût engendré par le redémarrage de ce module[11].
- **Traitement local** : L'idée de cette technique est que la source peut effectuer un prétraitement. Ainsi une programmation événementielle semble bien adaptée aux réseaux de capteurs. Seuls les changements significatifs de l'environnement devrait provoquer un envoi de paquets sur le réseau. Dans le même état d'esprit, une grande collaboration est attendue entre les capteurs d'une même région en raison de leur forte densité et dans la mesure où les observations ne varient presque pas entre des voisins très proches. Cette stratégie de traitement local permet de réduire sensiblement le trafic[11].

Ce mécanisme sera abordé dans la conception de notre solution en chapitres 4 et 5.

- **Organisation des échanges** : Ce procédé revient à limiter les problèmes de retransmission dus aux collisions. La solution extrême consiste à utiliser la technique d'accès au médium TDMA. Les collisions sont ainsi fortement réduites[11].
- **Répartition de la consommation d'énergie** : La formation de clusters permet d'envisager des réseaux comportant un très grand nombre de capteurs. Elle favorise une meilleure répartition de la consommation d'énergie. En effet, dans le cas d'une transmission directe vers l'observateur (voir fig1.3(a)), les capteurs éloignés vont plus rapidement manquer d'énergie. Au contraire, dans le cas d'une transmission par saut (voir fig1.3(b)), les nœuds proches de l'observateur vont être vite en rupture de batterie car ils seront plus sollicités pour relayer les messages des autres. La solution consiste à hiérarchiser les échanges en divisant la zone d'observation en clusters (voir fig1.3(c)).

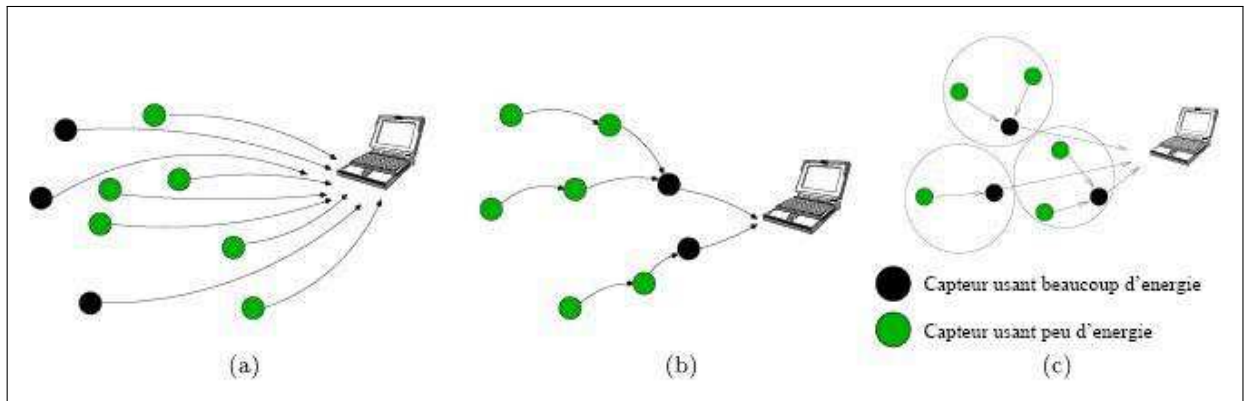


Figure 1.3 – (a)- Transmissions directes. (b)- Transmission saut par saut. (c) Hiérarchisation en clusters.

1.2.5 Domaines d’application des WSN

Les réseaux de capteurs peuvent être constitués de différents types de capteurs (Sismique, Thermique, Visuel, à Infrarouge, etc) qui sont en mesure de surveiller une grande variété de conditions ambiantes (Température, Humidité, Pression, Mouvement de véhicule, Niveau de bruit, etc). De nos jours, l’utilisation des réseaux de capteurs dans le domaine de l’industrie est devenue inévitable vu l’utilité et le grand nombre de services offerts. Nous classons d’abord les WSN par domaine selon l’objectif visé, en trois classes principales, ensuite nous donnons quelques exemples d’applications :

1. **La collecte des mesures** : les réseaux de capteurs ont permis aux scientifiques de surmonter les difficultés liées à l’observation de certains phénomènes. Ces difficultés sont généralement dues à la nature du phénomène observé. Par exemple, il est parfois difficile d’accéder à l’endroit où se déroule le phénomène et que ce phénomène nécessite la présence continue de l’observateur. Le réseau de capteurs déployé dans cet objectif est généralement programmé pour transmettre un certain nombre de mesures de façon périodique ou à la demande des utilisateurs.
2. **La détection des évènements** : Toutes les applications de surveillance font appel à la notion de détection d’évènements. Un évènement peut être simple (dans la mesure où il est détecté par un seul nœud capteur), ou bien composé nécessitant la collaboration de plusieurs nœuds capteurs (détection d’explosions). Les évènements capturés peuvent soit être envoyés directement à une station de base (cas de surveillance d’une zone), ou stockés localement ou quelque part dans le réseau attendant une demande d’un utilisateur (cas de surveillance off-line).

3. **La poursuite de cibles :** L'une des applications émergentes dans le contexte d'un réseau de capteurs est la poursuite de cibles. Le but de ce type d'application est de poursuivre l'évolution dans le temps d'un objet ou d'un phénomène physique, afin de prédire son comportement futur et de prendre les mesures nécessaires. Dans une même application, on peut combiner deux à trois de ces classes. Un réseau de capteurs déployé pour la lutte contre les feux de forêts, doit être capable de détecter un feu, de collecter des mesures demandées par le pompier et de fournir une carte d'évolution du feu à tout instant.

Les exemples d'applications sont nombreuses, nous pouvons citer entre autres :

Applications environnementales : Certaines applications environnementales des réseaux de capteur comprennent : le suivi des mouvements d'oiseaux, les petits animaux et d'insectes ; surveillance des conditions environnementales qui affectent les céréales et le bétail ; l'arrosage des terres agricoles ; détection des incendies de forêt ; la mesure de la pollution. Prenant quelques exemples :

- Détection des incendies de forêt : Les nœuds capteurs peuvent être déployés d'une manière dense et aléatoire dans une forêt, ceci permet aux capteurs d'envoyer à l'utilisateur final l'information sur l'origine exacte de l'incendie avant que le feu ne se propage et devienne incontrôlable. Les nœuds capteurs communiquent entre eux afin d'effectuer une détection distribuée et surmonter les obstacles, tels que les arbres et les rochers, qui bloquent la vue.
- Détection des inondations : Un exemple de détection d'inondation est le système ALERT déployé aux Etats-Unis. Plusieurs types de capteurs sont déployés dans ce système afin de détecter les intempéries, le niveau d'eau et les conditions climatiques. Ces capteurs fournissent des informations à un système de base de données d'une manière prédéfinie.

Applications médicales : Dans le domaine de la santé, les réseaux de capteurs sans fil fournissent des interfaces de communication pour les personnes handicapées, le suivi intégré du patient, l'administration des médicaments dans les hôpitaux, la télésurveillance des données physiologiques de l'être humain, ainsi que le suivi et la surveillance des médecins et des patients à l'intérieur d'un hôpital.

- La télésurveillance des données physiologiques de l'être humain : Les données physiologiques recueillies par les réseaux de capteurs peuvent être stockées pendant une longue durée, et peuvent être utilisées pour l'exploration médicale. Les réseaux de capteurs peuvent également surveiller et détecter les comportements des personnes âgées. Cela donne plus de liberté de mouvement à ces personnes et

permet aux médecins d'identifier les symptômes prédéfinis précédemment. "Health Smart Home" a été conçu à la Faculté de Médecine de Grenoble-France pour valider la faisabilité d'un tel système.

- Suivi et contrôle des médecins et des patients dans un hôpital : Chaque patient est doté d'un ou plusieurs capteurs de petite taille. Chaque capteur remplit une tâche bien déterminée, par exemple un capteur qui contrôle les battements de cœur alors qu'un autre contrôle la tension artérielle. Les médecins aussi peuvent être dotés de capteurs afin de faciliter la localisation à l'intérieur de l'hôpital.

Applications dans les maisons : C'est le domaine le plus touché par les applications des réseaux de capteurs, nous citons :

- Automatisation des maisons : Comme la technologie avance, on peut placer des nœuds capteurs à l'intérieur des équipements électroménagers tels que les micro-ondes, les aspirateurs et les réfrigérateurs. Ces capteurs peuvent communiquer entre eux ou avec des réseaux externes via Internet ou le réseau téléphonique. Ceci permet à l'utilisateur de gérer ces équipements localement ou à distance plus facilement.
- Environnement élégant : Des capteurs placés à l'intérieur des équipements électroménagers peuvent ainsi communiquer entre eux ou avec un serveur de chambre gérant l'ensemble des services disponibles, (exemple l'impression, le fax, et le scannage). Ce serveur peut communiquer avec d'autres serveurs de chambre. Ce type d'application a été implémenté dans "The Residential Laboratory" au niveau de l'institut de technologie de Georgia-USA.

Applications militaires : Les réseaux de capteurs sans fil représentent une partie intégrante dans les commandements militaires, le contrôle, la communication, la reconnaissance, et le renseignement militaire. Le déploiement rapide, l'auto-organisation et la tolérance aux pannes font que les réseaux de capteurs sans fil sont les plus utilisés dans les champs de bataille. Parmi les applications militaires des réseaux de capteurs sans fil, nous citons : le contrôle des forces amies (équipements, munitions) ; Surveillance du champ de bataille ; la reconnaissance du terrain, contrôle des forces ennemies (positionnement et équipements).

- Surveillance des forces amies (équipements et munitions) : Les commandants des forces peuvent à tout moment surveiller l'état (conditions + disponibilité) des troupes amies, par l'utilisation d'un réseau de capteurs sans fil. Chaque véhicule, équipement important peut être doté d'un capteur qui fournit des informations au nœud responsable (Sink) et qui envoie les informations au commandement.
- Surveillance du champ de bataille : Les terrains et les routes critiques peuvent

être rapidement couvertes par des réseaux de capteurs. Une surveillance de près des activités des forces opposées est possible. Comme les opérations sont variables et de nouveaux plans opérationnels peuvent se préparer, de nouveaux réseaux de capteurs peuvent être déployés à tout moment pour la surveillance du champ de bataille.

- Reconnaissance du terrain et des forces ennemies : Connaissant d'avance le terrain à occuper par les forces ennemies, cela permet de déployer des capteurs sans fil à fin de contrôler toutes ses activités et ses mouvements en temps réel.
- L'évaluation des pertes dans le champ de bataille : Juste avant ou après les attaques, les réseaux de capteurs peuvent être déployés dans une zone cible pour recueillir les données d'évaluation des pertes du combat.

Applications commerciales : Parmi les applications commerciales des réseaux de capteurs sans fil, nous citons : le contrôle de la qualité d'un produit, la construction d'espaces de bureaux intelligents, contrôle de l'environnement dans les bureaux, surveillance des régions sinistrées, etc.

- Contrôle de l'environnement dans un bureau : L'air conditionné et la température de la plupart des bâtiments sont contrôlés d'une manière centralisée. Par conséquent, la température à l'intérieur d'une chambre peut varier de quelques degrés ; un côté peut-être plus chaud que l'autre. Comme l'écoulement de l'air provenant du système central peut ne pas être uniformément réparti, un réseau de capteurs sans fil peut être installé pour contrôler l'écoulement de l'air et la température dans différentes parties de la salle. Une telle technologie peut réduire la consommation énergétique.
- Musées interactifs : grâce au réseaux de capteurs, une interaction est créée entre les objets et le public dans les musées. Un objet archéologique sera en mesure de répondre à une personne qui le touche. En plus un WSN peut fournir la localisation à l'intérieur du musée. Un exemple de ces musées est l'*Exploratorium* (un musée de la science et de la technologie) à San Francisco.
- Contrôle des vols de voitures : Les nœuds de capteurs sont déployés pour détecter et identifier les vols de voiture dans une région géographique et signaler ces menaces aux utilisateurs par Internet ou par satellite. En plus des capteurs sans fil peuvent être déployés dans un véhicule afin de localiser et traquer le véhicule volé à distance.

Exemple réel d'applications des WSN

Un exemple de projets récents exploitant la technologie des réseaux de capteurs est celui initié par l'armée américaine dénommé "*Counter-Sniper system*" (système de détection de sniper) basé sur les Motes Mica2 de Crossbow. Le système est développé par l'institut des logiciels et les systèmes intégrés à l'université de VANDERBILT. Un grand succès a été réalisé en identifiant avec succès l'endroit des tireurs agissant comme sniper. La plateforme de test contient soixante nœuds capteurs de type Mica2 équipés de détecteurs acoustiques et déployés sur un terrain urbain d'opérations militaires. Les nœuds capteurs collaborent entre eux pour déduire par triangulation les coordonnées du sniper. Les nœuds capteurs ont été testés sur un système de coordonnées 3D et 2D. L'exactitude du résultat est de moins d'un mètre pour les coordonnées en 3D et de 0,6 mètre pour le cas de 2D. Le programme d'analyse a montré de manière exacte le point de tir sur une fenêtre spécifique. Le système est très sensible avec une latence globale de moins de deux secondes.



Figure 1.4 – Capteur de type Mica2 Mote.

1.2.6 Type des WSN :

Les réseaux de capteurs peuvent avoir plusieurs architectures physiques dictées par les caractéristiques de l'application ciblée. Ces architectures peuvent être classées selon deux critères différents à savoir : la mobilité des nœuds et la nature des collecteurs.

- *Selon la mobilité des nœuds* : Les WSN peuvent être classés en de deux types :
 - WSN mobiles** : Le réseau est constitué d'un ensemble de nœuds capteurs mobiles évoluant dans un environnement statique. Le but est souvent l'exploration de zones inaccessibles ou dangereuses. Les travaux de recherche sont souvent orientés robotique ; le nœud jouant à la fois le rôle de capteur et d'actionneur.
 - WSN stationnaires** : Le réseau est constitué de nœuds capteurs fixes servant à la surveillance de l'occurrence d'événements dans une zone géographique. Dans ce cas, le réseau n'effectue que la surveillance et les données mesurées sont transmises en mode multi-sauts au nœud collecteur. Ce dernier est chargé, après la réception

des données, de mettre en œuvre les actions nécessaires. Ce collecteur peut être connecté, de manière filaire à un autre réseau.

- *Selon la nature de la station de base* : Les WSN peuvent être aussi répartis en deux classes :

WSN avec station de base fixe : Les positions des nœuds collecteurs dans le réseau sont connues à l’avance par tous les nœuds du réseau.

WSN avec station de base non-fixe : Les nœuds collecteurs, qui peuvent être statiques ou mobiles, apparaîtront momentanément pour interroger le réseau. Ce type de réseau est utilisé souvent pour la surveillance off-line.

1.3 Présentation des réseaux de capteurs Multimédia sans fil (WMSN)

Comme il a été déjà signalé dans la section précédente, l’amélioration et la miniaturisation a permis d’équiper un simple nœud capteur avec un module de collecte d’information audio et visuelle. A titre d’exemple, on peut citer le module Cyclops comme étant un module de captage et d’inférence d’image qui est conçue pour l’imagerie. Il est extrêmement léger et peut être interfacé avec des motes comme le MICA2 de Crossbow. En plus de la possibilité de la récupération des données multimédias, un réseau de capteurs multimédias sans fil sera également en mesure de stocker, traiter en temps réel, corréliser et surtout fusionner les données multimédias provenant de sources hétérogènes[3].

1.3.1 Architecture des WMSN

Nous introduisons une architecture de référence pour les WMSN dans la figure 1.5, où les utilisateurs se connectent à travers Internet à un réseau de capteurs déployés [12]. Les fonctionnalités des différents composants du réseau sont résumées comme suit :

- **Standard Video and Audio Sensors** : Se sont généralement des capteurs de faible résolution, ils peuvent être disposés dans un réseau à un-tier, comme indiqué dans le nuage **a**, ou dans une structure hiérarchique, comme indiqué dans la troisième nuage **c**.
- **Scalar Sensors** : Se sont généralement des dispositifs (capteurs) avec ressources limitées en termes de stockage et capacité de traitement et d’énergie, ils sont utilisés pour capturer des données scalaires telles que la température, l’humidité,

la pression afin de les envoyer aux stations de base (Sink).

- **Multimedia Processing Hubs** : Ces dispositifs disposent relativement d'importantes ressources de calcul et ils sont adaptés pour agréger des flots multimédia à partir des nœuds de capteurs. Ils font partie intégrante de la réduction à la fois de la dimension et de la quantité de données transmises au SINK et de dispositifs de stockage.
- **Storage Hubs** : Selon l'application, les flots multimédia peuvent être exploités en temps réel ou après un traitement supplémentaire. Ces plates-formes de stockage disposent d'algorithmes d'extraction, de datamining et de fonction pour identifier les caractéristiques importantes de l'événement, avant même que les données soient transmises à l'utilisateur final.
- **Sink** : Le Sink (puits) est responsable de la communication des requêtes utilisateurs au réseau et retourne les parties filtrées des flots multimédia à l'utilisateur. Plusieurs puits peuvent être nécessaires dans un grand réseau hétérogène.
- **Gateway (passerelle)** : Elle contient une estimation géographique de la zone couverte par son cadre de détection, et répartit les tâches aux puits appropriés pour le transfert des données captées.
- **Users (utilisateurs)** : Ils sont identifiés par leurs adresses IP. Ils sont initiateurs des requêtes envoyées au réseau qui leur retourne les résultats obtenus.

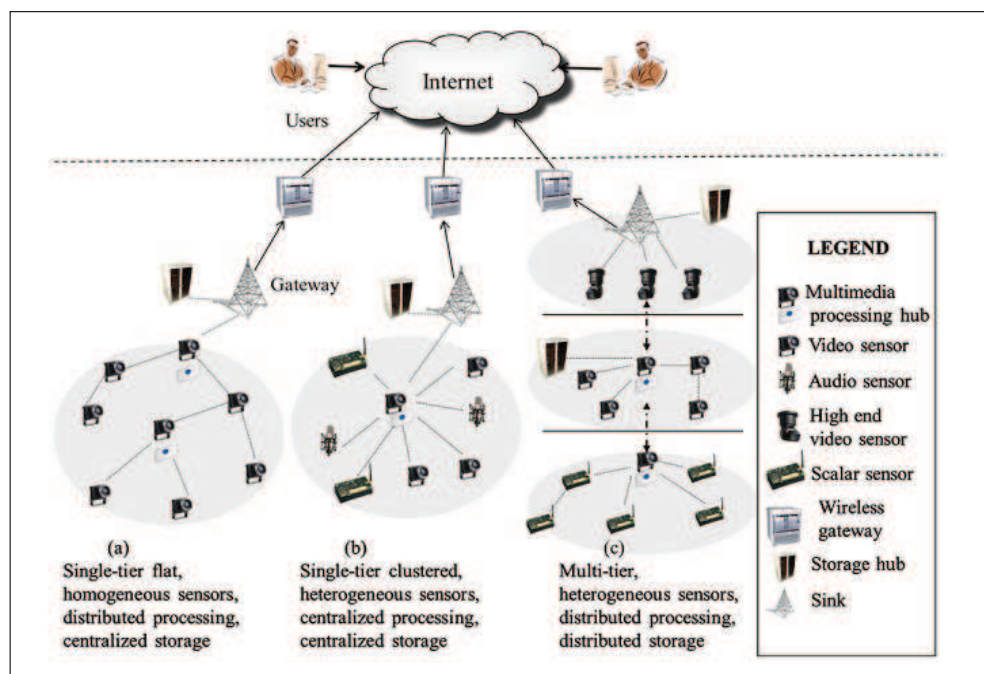


Figure 1.5 – Architecture de base des WMSN.

1.3.2 Facteurs influant la conception des WMSN

Nous discutons dans cette section des paramètres à prendre en considération lors de la phase de conception d'un réseau de capteurs sans fil multimédia. Dans ces réseaux, un grand nombre de flots audio - vidéo (en plus des données scalaires), est transmis au centre de contrôle (Sink ou station de base) pour analyse et prise de décision ce qui oblige de considérer d'autres facteurs. Il y a un besoin important en termes de bande passante, avec surtout une forte contrainte en termes de délai de transmission. Les solutions existantes ont été développées pour des WSN mais ne prennent pas en compte la génération à grande échelle de données multimédia, ce qui les rend inadaptées aux WMSN. Pour ces raisons, le paradigme des WMSN doit être repensé et doit intégrer des mécanismes permettant la transmission de contenu multimédia avec un certain degré de qualité de service. Dans ce qui suit nous présentons les facteurs influents dans la conception des WMSN.

- **Exigences de la qualité de service (QoS)** : L'ensemble des applications envisagées dans les WMSNs aura des exigences différentes. En plus de la délivrance des données scalaires comme dans le cas des réseaux de capteurs simple, les données multimédias englobent des images et vidéos. Par conséquent, des exigences sont nécessaires en termes de matériels et algorithmes de haut niveau pour fournir la qualité de service requise. Ces exigences peuvent toucher plusieurs domaines à savoir : les retards, la fiabilité, la distorsion des données, voire la durée de vie du réseau.
- **Demande élevée en bande passante** : Les données multimédia, en particulier les flots (stream) de vidéo, nécessitent une bande passante de transmission supérieure en grandeur à celle supportée par des capteurs scalaires. Dans ce cas, un débit plus élevé est nécessaire pour ce type de capteurs, avec une consommation d'énergie comparable. A cet égard, la technique de transmission UWB (Ultra Wide Band) ou ultra large bande en français semble particulièrement la plus adaptée pour les WMSN car elle est basée sur la transmission d'impulsions de très courte durée (souvent inférieure à la nanoseconde), ce qui fournit une bande passante importante.
- **Techniques de codage des sources multimédias** : Les flots de vidéos non compressés exigent une bande passante excessive dans un environnement multi-sauts sans fil. A titre d'exemple un flot de vidéo exige une bande passante de 5 Mbits/s. Par conséquent, il est évident que des techniques de traitement efficaces pour la compression avec perte (lossy compression) sont nécessaires pour les réseaux de capteurs multimédias. Les techniques de codage vidéo traditionnelles utilisées pour les communications filaires et sans fil sont basées sur l'idée de réduire le débit binaire généré par le codeur de source en exploitant les statistiques

de la source. A cet effet, les codeurs reposent sur des techniques de compression intra-trame afin de réduire la redondance au sein d'une trame, tandis qu'ils utilisent une compression inter-trame (aussi connue sous le nom de codage prédictif ou d'estimation de mouvement). Ceci permet d'exploiter la redondance entre les trames suivantes pour réduire la quantité de données devant être transmise et stockée ; augmentant ainsi les performances du réseau.

- **Consommation de l'énergie** : La consommation d'énergie est une préoccupation fondamentale dans les réseaux de capteurs multimédias sans fil, plus que dans les réseaux traditionnels de capteurs sans fil. En effet, les capteurs sont des équipements alimentés par batteries limitées et la durée de vie d'un capteur est étroitement liée à sa batterie. Les applications multimédias produisent un volume de données important nécessitant une vitesse de transmission élevée et un traitement extensif. Dans les nœuds de capteurs traditionnels, la consommation d'énergie est plus ou moins estimée et maîtrisée par les fonctionnalités de communication, mais ce n'est pas toujours le cas dans les WMSNs. Par conséquent, les protocoles, les algorithmes et les architectures de maximisation de la durée de vie du réseau tout en assurant la qualité de service requise par l'application dénotent un enjeu crucial à relever.
- **Architecture flexible pour supporter des applications hétérogènes** : Les architectures des WMSN doivent supporter plusieurs applications hétérogènes et indépendantes avec des exigences différentes. Il est nécessaire de développer des architectures hiérarchiques, flexibles, qui peuvent répondre aux exigences de toutes ces applications.
- **Couverture multimédia** : Certains capteurs multimédias, en particulier des capteurs vidéo, ont de plus grands rayons de détection et sont sensibles à la direction d'acquisition (directivité). En outre, les capteurs vidéo peuvent capturer des images uniquement lorsqu'il y a une visibilité directe entre l'événement et le capteur. Par conséquent, les modèles de couverture développés pour les réseaux de capteurs sans fil traditionnels ne sont pas suffisants pour la planification avant déploiement d'un réseau de capteurs multimédia.
- **L'intégration avec d'autres technologies sans fil** : Les réseaux de capteurs à grande échelle peuvent être créés à travers l'interconnexion locale (îles), de capteurs par l'intermédiaire d'autres technologies sans fil. Cela doit se faire sans sacrifier de l'efficacité de l'opération au sein de chaque technologie. En plus de ces facteurs, il existe d'autres facteurs qu'on doit prendre en considération lors de la phase de conception d'un réseau de capteurs multimédia, nous citons :
 - la tolérance aux pannes ;
 - la scalabilité (passage à l'échelle) ;
 - le coût de production ;
 - l'environnement d'exploitation ;

- la topologie du réseau ;
- les contraintes matérielles ;
- le support de transmission.

1.3.3 Défis des réseaux de capteurs multimédia sans fil

En plus des données scalaires telles que la température, la pression,...etc, on manipule des données multimédia telles que les images, la vidéo et l'audio. Les particularités de ce type de données exigent des performances accrues nécessitant de relever des défis lors de la mise en oeuvre des réseaux de capteurs multimédia :

- **Lien radio** : Les capteurs possèdent le matériel nécessaire pour effectuer des communications par ondes radio. Toutefois, la diffusion de l'information via ces liens est peu aisée à cause de l'instabilité et du manque de fiabilité qu'ils présentent. En plus, l'utilisation d'un médium commun de communication induit des interférences radio, réduisant considérablement la capacité d'exploitation du canal.
- **La dynamicité** : La topologie des réseaux de capteurs sans fil est en constante évolution à cause de l'état d'activité des capteurs (extinction, mise en veille et actif), ainsi que leurs déplacements indépendamment des uns et des autres. La conception d'un protocole d'auto-organisation qui s'adapte continuellement et rapidement aux changements, s'avère nécessaire pour assurer le bon fonctionnement du réseau.
- **La limitation des ressources** : Les capteurs se caractérisent par une limitation de ressources en termes de calcul, de stockage et d'autonomie d'énergie. La contrainte énergie dans ce type de réseaux étant très forte, puisque l'on considère généralement qu'il est difficile voire impossible de recharger les batteries des capteurs (capteurs déployés dans des zones hostiles et inaccessibles). L'ordonnement d'activité et l'auto-organisation sont considérés comme des solutions de premier choix, dans le but d'éteindre l'équipement radio de certains capteurs et permettre seulement aux capteurs disposant de plus d'énergie de transmettre les données captées à la station de base.
- **Localisation des décisions** : Un réseau de capteurs est potentiellement constitué d'un grand nombre de capteurs ayant une portée de transmission très limitée. D'où, un processus d'acheminement d'une information se faisant généralement en multi-sauts par le biais de plusieurs capteurs. Par conséquent, il est inconcevable de faire impliquer un grand nombre de capteurs pour une prise de décision. Tout processus de prise de décision doit être localisé.
- **Routage du trafic hétérogène** : Dans le réseau, circulent différents types de

trafic. Par ailleurs, les protocoles qui ont été créés pour faire la différenciation des divers types de trafic, ne tiennent pas compte des particularités et spécificités de l'information multimédia. Par conséquent, de nouveaux protocoles de routage sont nécessaires et devraient être capables de manipuler et classifier les types de trafic, et d'établir les priorités en fonction des exigences des applications [3].

- **L'incapacité de supporter la communication en temps réel** : La diffusion de la vidéo ou de l'audio dans les WMSN est en mode continu (streaming), cela exige le besoin d'implanter des schémas de transmission en temps réel, assurant des communications avec des délais stricts, et avec prise en charge des différents types de trafic [3].
- **Le déploiement de capteurs multimédia** : Un autre élément différenciateur entre un WSN et un WMSN est le modèle de captation qui est utilisé par les capteurs qui font partie de ces réseaux. Les capteurs scalaires, c'est-à-dire, les capteurs de température ou de pression ou de lumière, suivent le modèle omnidirectionnel. Par contre, un capteur d'audio ou de vidéo a une perception sectorielle (voir figure 2.5). Donc les réseaux de capteurs multimédia sont par essence hétérogènes (i.e.

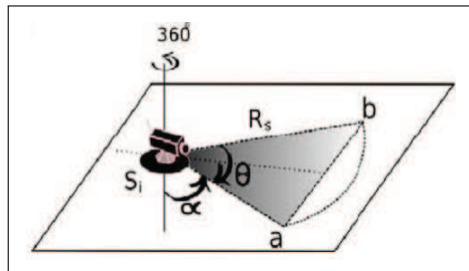


Figure 1.6 – Les paramètres de la configuration d'un capteur directionnel : α = angle avec la verticale, θ = ouverture de la caméra, R_s = rayon de détection.

Puissance du microphone et zoom de la caméra) et directionnels (i.e. angle d'ouverture de la caméra ou du microphone) ce qui les rend complètement différents des capteurs scalaires. Ces deux caractéristiques (hétérogénéité et capture directionnelle) rendent le problème de placement plus complexe qu'avec les capteurs scalaires. Un schéma spécifique pour le placement optimisé de capteurs multimédia directionnels en fonction de leurs capacités est donc nécessaire [3].

- **Multimedia in-network processing** : Dans de nombreuses applications de WMSN, un seul nœud capteur n'est pas en mesure de répondre à une question posée, mais plusieurs doivent collaborer pour y répondre. Par exemple, une seule caméra surveillant un ensemble de personnes, ne peut compter leur nombre exact et les localiser ; la collaboration avec plus de capteurs en proximité est nécessaire afin de couvrir toute l'étendue du groupe de personnes [13].

1.3.4 Domaines d'application des WMSN

Les réseaux de capteurs multimédia vont non seulement améliorer les applications existantes des WSN telles que le dépistage, la domotique et la surveillance de l'environnement, mais introduisent également de nouvelles applications telles que [14] :

- **Surveillance vidéo** : Un réseau vidéo sans fil sera composé de plusieurs caméras vidéo interconnectées, miniaturisées et alimentées par batterie. Chacune des caméras est dotée d'un module émetteur-récepteur (Transceiver) capable de traiter, envoyer et réceptionner de l'information. Les capteurs vidéo et audio sont utilisés afin d'améliorer et compléter les systèmes de surveillances existants contre le crime et les attaques terroristes. Les WMSN permettent le contrôle des frontières, les événements publics et les propriétés privées. Ces capteurs peuvent inférer et enregistrer des activités potentiellement pertinentes (telles que les vols, les accidents de voitures,...etc) et produire des flots audio-vidéo pour de futures enquêtes.
- **Gestion du trafic routier** : Grâce aux réseaux de capteur multimédias, il est possible de contrôler la circulation automobile dans les grandes villes ainsi que sur les autoroutes, il est également possible de déployer des services qui offrent des conseils routiers afin d'éviter la congestion. Une autre application des WMSN est le contrôle des places disponibles dans un parking, on parle alors de parkings intelligents où le conducteur est renseigné automatiquement ce qui organise la mobilité des véhicules dans les zones urbaines. Enfin, les capteurs multimédias peuvent contrôler le flot de véhicules qui circulent sur une autoroute renseignant sur leurs vitesses moyennes de circulation et leur nombre.
- **Prestation des soins de santé** : En télémédecine, un réseau de capteurs peut être intégré avec un réseau multimédia de troisième génération pour fournir un service omniprésent de soin de santé. Les patients peuvent être dotés de capteurs médicaux afin de contrôler certains paramètres tels que : la température du corps humain, la tension artérielle, l'oxymétrie de pouls et l'activité de respiration. En plus, des centres médicaux peuvent effectuer un contrôle à distance de leurs patients à travers les capteurs audio et vidéo.
- **Service de localisation des personnes** : Le contenu multimédia comme la vidéo ou les images statiques combiné avec des techniques avancées de traitement de signal peuvent être utilisés pour localiser les personnes perdues ou identifier des criminels ou terroristes.
- **Le contrôle d'environnement** : Plusieurs projets de surveillance de l'environnement utilisant l'acoustique et la vidéo sont envisagés ; l'information doit être transportée en temps réel. A titre d'exemple, un ensemble de capteurs vidéo sont déjà utilisés par les océanographes afin de déterminer l'évolution du sable.
- **Le contrôle de processus industriels** : La vision industrielle est l'application

des techniques de vision par ordinateur pour la fabrication où l'information peut être extraite et analysée par des capteurs multimédias. Ceci permet d'assister un processus de fabrication, tel que ceux utilisés dans les puces semi-conductrices, les voitures, les aliments ou des produits pharmaceutiques. Par exemple, dans le contrôle de qualité des processus de fabrication, des produits finis sont automatiquement contrôlés pour trouver des défauts de fabrication. En outre, les systèmes de vision industrielle sont capables de détecter la position et l'orientation des pièces de produits afin d'être levées par un bras robotisé. Donc, l'intégration d'un système de vision industrielle avec les WMSN ajoute une flexibilité aux systèmes de contrôle visuel et aux actions automatisées (qui exigent une vitesse élevée et un fonctionnement continu).

1.3.5 Problématiques liées aux réseaux de capteurs multimédia

Ces dernières années, les réseaux de capteurs sans fil ont intéressé plusieurs domaines de recherches, et ce pour les deux catégories de capteurs (scalaires ou multimédia). En raison des caractéristiques particulières des réseaux de capteurs multimédias, les chercheurs se sont attaqués à plusieurs problématiques, que nous citons ci-après [15] :

- **Routage** : Concevoir un protocole de routage performant en termes de minimisation de la consommation de l'énergie, du choix des routes optimales pour l'acheminement de l'information d'un capteur à la station de base et vice versa, et de réduction du délai de délivrance des paquets. Les solutions de routages doivent passer à l'échelle sans dégrader les performances du réseau .
- **Couche MAC** : La spécificité de ces réseaux nécessite le développement de nouveaux protocoles MAC qui s'adaptent aux contraintes imposées. Ceci dans le but d'améliorer le débit, minimiser la consommation d'énergie, optimiser le partage du médium ainsi que minimiser le délai de délivrance des paquets.
- **Diffusion de l'information** : Les protocoles de diffusion conçus pour les réseaux de capteurs doivent tenir compte de leurs spécificités ainsi que de leurs contraintes intrinsèques imposées. Ainsi, pour concevoir un protocole efficace, il faudrait assurer une couverture maximale des capteurs composant le réseau (taux d'accessibilité supérieur à 90%), minimiser le nombre des ré-émission et des réceptions redondantes ainsi que la consommation d'énergie.
- **Sécurité** : Du point de vue sécurité, il n'y a pas de différences entre les deux types de réseaux de capteurs (scalaires et multimédias). Par conséquent, la plupart des solutions définies pour les WSN peuvent être facilement adaptées pour les WMSN. Cependant, les WMSN ont aussi quelques nouvelles fonctionnalités qui découlent du fait que certains des nœuds auront en plus une caméra vidéo et des capacités

de calcul supérieures. Cela soulève de nouveaux défis de sécurité et de protection [16] du matériel.

- **Hétérogénéité des données** : Dans un réseau de capteurs multimédia, plusieurs types de données sont générés, à savoir les données scalaires (Températures, pression, etc) et les données multimédias (images, flots vidéo et audio, etc). Plusieurs applications hétérogènes et avec des exigences différentes sont mises en oeuvre sur de tels réseaux impliquant des réponses concrètes.
- **Redondance de l'information** : Dans un réseau de capteurs multimédia, la prise de décision concernant le phénomène observé est basée sur les informations collectées par l'ensemble des nœuds. Ces derniers sont dispersés dans la zone d'intérêt. En raison de cette manière de déploiement, un volume important de données est engendré ce qui favorise une transmission redondante et par la suite l'épuisement des ressources telles que : l'énergie et la bande passante. Dans ce contexte plusieurs solutions sont proposées, à savoir, la fusion ou l'agrégation des données.
- **Consommation de l'énergie** : Les capteurs scalaires ou multimédias sont conçus pour être capables de fonctionner durant des mois, voire même des années. Ainsi, la capacité énergétique des nœuds capteurs doit être utilisée de manière rationnelle afin de maximiser la durée de vie du réseau. Un nœud capteur qui a consommé toute son énergie est un élément défaillant, et il peut ainsi influencer négativement sur la connectivité du réseau. Cette ressource rare, qui est l'énergie, doit donc être bien prise en charge dans la fabrication des composants d'un capteur multimédia (camera, microphone, etc), ainsi que dans la conception des applications exploitant ces nœuds capteurs.

1.4 Conclusion

Nous avons présenté dans ce chapitre les réseaux de capteurs scalaires et les réseaux de capteurs multimédias. Nous avons donné d'abord une description d'un nœud capteur ainsi que d'un réseau de capteur scalaire et multimédia, et précisé leurs différents domaines d'applications et les facteurs influant leur conception. Par la suite nous avons discuté des défis à relever et des principales problématiques rencontrées pour la mise en oeuvre des réseaux de capteurs multimédia. Dans le chapitre suivant nous abordons les concepts de fusion et d'agrégation de données dans les réseaux de capteurs d'une manière générale et particulièrement dans les réseaux de capteurs multimédias.

Chapitre 2

Fusion de données dans les réseaux de capteurs

2.1 Introduction

Avec plusieurs sources d'information disponibles via des connexions réseaux non coûteuses que ce soit sur Internet ou sur les intranets des entreprises, le besoin d'accéder à toutes ces sources par le biais d'une interface cohérente a été la force motrice derrière beaucoup de recherches dans le domaine de l'intégration de l'information. Au cours des trois dernières décennies, plusieurs systèmes ont été développés afin de répondre à ce besoin. Donc le mot intégration est devenu un mot clé dans le domaine de traitement des données au sein des systèmes d'information constitués de plusieurs sources de données.

D'après [12], un scénario d'intégration de données est composé de trois (03) étapes comme le montre la figure 2.2. Tout d'abord, nous devons identifier les attributs correspondants pour décrire les éléments d'information dans les sources de données. Le résultat de cette étape est un schéma de mappage qui est utilisé pour transformer les données présentes dans les sources en une représentation commune (re-nommage, restructuration). Deuxièmement, les différents objets qui sont décrits dans les sources de données doivent être identifiés et alignés. De cette façon, en utilisant des techniques de détection de doublons, les représentations incohérentes des mêmes objets du monde réel sont détectées. Troisièmement, dans une dernière étape, les représentations dupliquées sont combinées et fusionnées en une représentation unique alors que les incohérences dans les données sont résolues. Cette dernière étape est connue sous le nom de Fusion (agrégation) de données. Néanmoins et selon [17], il faut distinguer entre fusion

et agrégation, cette dernière représente la capacité à synthétiser les données pour en réduire la taille.

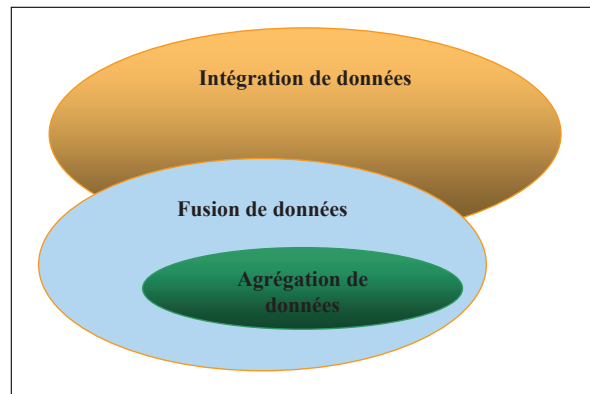


Figure 2.1 – Intégration, fusion et agrégation de données.

Dans le contexte des réseaux de capteurs, le but de fusionner les données est d'éliminer la transmission redondante des données ce qui améliore la durée de vie du réseau (en termes d'énergie, bande passante, etc). La transmission redondante des données est provoquée par le fait qu'un réseau de capteurs est constitué d'un ensemble de nœuds déployés d'une manière condensée pouvant capter la même information. Par exemple un ensemble de nœuds capteurs voisins peut mesurer la même température de la zone dans laquelle ils ont été déployés, ou bien deux capteurs multimédia contrôlant les mêmes parties d'une zone d'intérêt. Cette notion de fusion ou d'agrégation de données ainsi que les concepts clés dans ce domaine représentent l'objectif principal de ce chapitre.

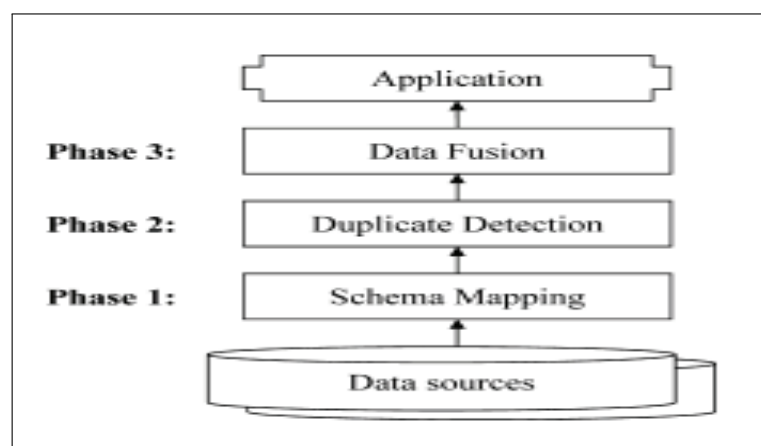


Figure 2.2 – Etapes du processus d'intégration de données.

2.2 Présentation du concept : fusion de données

2.2.1 Définition

La fusion de données est un concept utilisé dans les processus de traitement des informations. Ce concept n'est pas lié aux réseaux de capteurs seulement, mais il a été utilisé dans plusieurs domaines de recherche caractérisés par l'hétérogénéité des sources de données, à savoir la robotique, le traitement d'images, les systèmes d'information. D'après [18] et [19], *la fusion de données consiste à combiner des informations issues de plusieurs sources afin d'améliorer la prise de décision*. Cela veut dire que la fusion de données consiste au traitement de l'information collectée avant de l'envoyer à l'utilisateur final. D'après [20], *un processus de fusion de données permet, grâce à la combinaison d'informations hétérogènes provenant de différents capteurs pouvant être géographiquement répartis, de fournir une représentation synthétique de l'univers d'intérêt*. Dans la littérature, le processus de fusion de données ou en anglais *data fusion* est appelé aussi *data merging*, *data consolidation*, *entity resolution*, or *finding representations / survivors*. D'après [21], *trois éléments sont toujours considérés dans le cadre de la fusion de données : les sources, les algorithmes et le résultat*. Le résultat sera composé de données ayant une valeur ajoutée par rapport aux données initiales en provenance de la source. Cette nature de cette valeur ajoutée dépendra de l'application.

2.2.2 Classification de la fusion de données

D'après [22] et [17], la fusion de données peut être catégorisée en fonction de plusieurs caractéristiques, on peut classer la fusion de données selon les relations qui existent entre les données d'entrée (données redondantes, données complémentaires), comme on peut fusionner les données selon le niveau d'abstraction des données manipulées (bas niveau, niveau moyen, niveau haut et multi niveaux).

– Classification selon les relations entre les entrées (Input) :

1. Données redondantes : Si deux ou plusieurs sources indépendantes offrent la même portion de données, ces portions peuvent être fusionnées pour obtenir une donnée plus précise.
2. Données complémentaire : Si par exemple, les données fournies par les sources représentent différentes parties d'une grande scène, la fusion de données peut être appliquée pour obtenir une donnée plus complète de la scène.

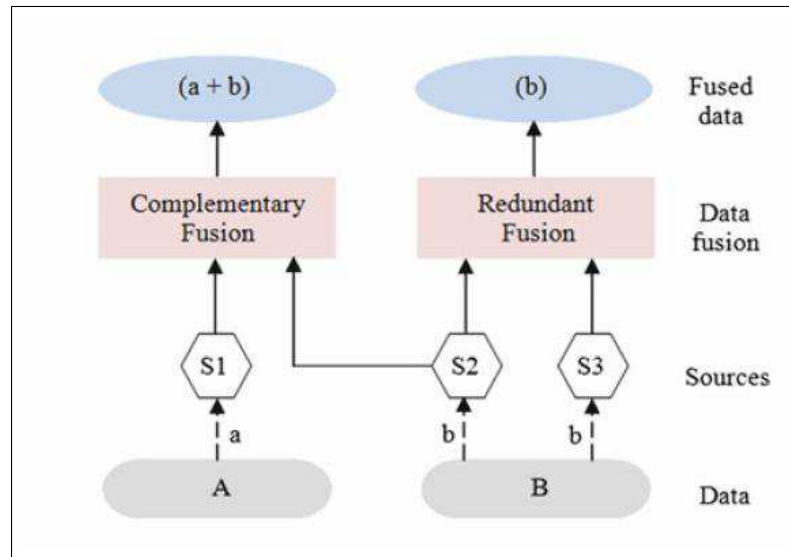


Figure 2.3 – Fusion selon les relations entre les entrées.

- Classification selon le niveau d'abstraction des données manipulées :
 1. Fusion bas niveau : Dite aussi fusion niveau pixel (Pixel Level Fusion), les données brutes sont fournies comme entrées et combinées en de nouvelles données qui sont plus précises que les entrées individuelles. Dans [23] un exemple de fusion bas niveau a été présenté où il est estimé le bruit ambiant afin de déterminer si le canal de communication est de bonne qualité.
 2. Fusion moyen niveau : Les caractéristiques et attributs d'une entité sont fusionnés pour obtenir un ensemble de nouvelles caractéristiques pouvant être exploitées pour d'autres tâches. Ce niveau de fusion est également connu sous le nom '*niveau caractéristiques / attributs*' (feature/attribute fusion level).
 3. Fusion haut niveau : Connue aussi sous le nom '*niveau symbole/décision*', il prend des décisions ou des représentations symboliques en entrée et les combine pour en déduire une décision globale la plus pertinente. Un exemple de ce niveau de fusion est l'approche Bayésienne pour la détection d'événements binaires proposée dans [24].
 4. Fusion multi niveaux : C'est une combinaison des trois niveaux cités ci-dessus, par exemple on peut fusionner une mesure avec une caractéristique pour obtenir une décision.

2.2.3 Concepts important dans la fusion de données dans les WSN

- **Le Clustering dans les réseaux de capteurs :** Le déploiement à grande échelle de réseaux de capteurs sans fil (WSN) et la nécessité d'agrégation de données nécessitent une organisation efficace de la topologie du réseau dans le but d'équilibrer la charge et de prolonger la durée de vie du réseau [25]. En plus, le déploiement condensé des nœuds capteurs dans une région d'intérêt signifie que des données très similaires sont produites par l'environnement physique dans le réseau de capteurs. Cela favorise un taux plus ou moins important de redondance. Donc, tous ces faits encouragent les regroupements de nœuds capteurs, un groupement de nœuds peut combiner ou compresser les données et ne transmettre que leur synthèse. Ce processus de groupement dans un réseaux a grande échelle est connu sous le nom de *Clustering*. Le Clustering dans les réseaux de capteurs comprend le regroupement de nœuds capteurs en clusters et l'élection d'un *Cluster-head* (CH). Ainsi, la collection des CHs dans le réseau forme un ensemble connecté dominant. Cependant, plusieurs défis font face à la technique du clustering [26] :

1. Combien de nœuds capteurs peut regrouper un cluster et quelle est la procédure de sélection qu'on doit appliquer pour déterminer le cluster head (CH),
2. Hétérogénéité dans le réseau, qui veut dire que l'utilisateur peut mettre quelques nœuds à pleine puissance en termes d'énergie qui vont se comporter en tant que Cluster Heads, les autres nœuds dans le groupe vont se comporter comme étant des membres seulement.

Pour remédier à ces préoccupations, plusieurs protocoles et algorithmes ont été proposés. La méthode de formation de cluster la plus répandue s'exécute comme suit :

- Chaque noeud devra connaître son voisinage par le biais des messages Hello,
 - Chaque noeud prend la décision selon sa connaissance locale de la topologie pour être cluster-head ou non,
 - Le noeud choisi comme cluster-head diffuse son statut dans son voisinage et invite ses voisins qui ne sont pas encore affiliés à d'autres clusters de le rejoindre.
- **La synchronisation des horloges des noeuds capteurs :** La synchronisation du temps est un challenge important et coûteux dans les réseaux de capteurs à communication multi-saut. De nombreuses applications de réseaux de capteurs exigent une synchronisation des horloges locales des noeuds. Les noeuds capteurs collaborent entre eux pour parvenir à une tâche de détection plus complexe. La fusion de donnée est un exemple d'une telle collaboration où les données collectées par différents nœuds sont regroupées en un seul résultat significatif. Par exemple

dans le cas où nous voulons tracer le chemin d'un véhicule, les noeuds captent la position et le temps de détection du véhicule et les envoient à une station de base qui à son tour combine toutes les informations pour tracer le chemin complet. Il est évident que si les noeuds ne sont pas synchronisés, le chemin sera inexact. Cependant, la synchronisation peut être utilisée dans la gestion d'économie d'énergie pour augmenter la durée de vie du réseau. Par exemple, les capteurs peuvent être mis en veille à des instants déterminés, et se réveiller en cas de besoin [4].

2.3 La fusion de données dans les WMSN

Dans un réseau de capteurs multimédia, les informations relatives à la même zone d'intérêt et parvenues de différents capteurs peuvent être utilisées pour obtenir une meilleure connaissance de la zone d'intérêt que l'utilisation d'une seule information parvenue d'un seul capteur. Dans un WMSN, les données peuvent être de type Image, vidéo, son,.... Une séquence vidéo représente une succession d'un ensemble d'images capturées dans un intervalle de temps bien déterminé. Pour cette raison, nous présentons dans cette section une vue globale sur les techniques de fusion des images.

2.3.1 La fusion d'images dans les WMSN

D'après [27], la fusion d'images est le processus consistant à combiner des informations provenant de deux ou plusieurs images d'une scène en une seule image plus informative et plus adaptée pour la perception visuelle et le traitement informatique. Dans le domaine de la fusion des images, les données sont représentées sous forme de tableaux contenant des valeurs numériques qui représentent l'intensité lumineuse, la couleur, la température, la distance et d'autres paramètres d'une scène bien déterminée [27]. Dans un réseau de capteurs multimédias, la fusion d'images est basée sur l'utilisation des informations capturées par les différents capteurs déployés dans un environnement, pour cette raison la fusion d'images dans un réseau de capteurs est appelée parfois la fusion d'image multi-capteurs ou en anglais Multi-sensor image fusion [28]. Dans les dernières années, plusieurs techniques de fusion d'images ont été développées. Il serait impossible de discuter toutes ces approches, mais une chose est sûre est que le processus de fusion d'images est généralement effectuée à l'un des trois différents niveaux de fusion : signal, caractéristique et de décision.

- La fusion niveau signal (signal level fusion) : Appelée également fusion au niveau pixels, représente le niveau de fusion le plus bas, où un certain nombre de si-

gnaux d'entrée brutes d'une image sont combinés pour produire un seul signal qui représente le signal de l'image fusionnée [28].

- La fusion niveau caractéristiques (features level fusion) : Cette approche de fusion est appelée aussi la fusion niveau objet, elle est basée sur l'extraction de caractéristiques à partir des images sources de telle sorte que les caractéristiques de chaque source peuvent être utilisées conjointement pour certains usages [28].
- La fusion niveau décision (decision level fusion) : Elle est également connue sous le nom de fusion niveau symbole, cette catégorie de fusion permet la combinaison des informations parvenues de différentes sources afin de produire une décision sur une situation bien déterminée.

La fusion d'images au niveau pixel (signal) reste le domaine le plus abordé par les travaux de recherches. Mais, des fois et selon le type d'application, les trois niveaux de fusion peuvent être combinés ensemble afin de fusionner des données capturées par plusieurs sources. La figure 2.4 [28] illustre la combinaison des trois niveaux de fusion.

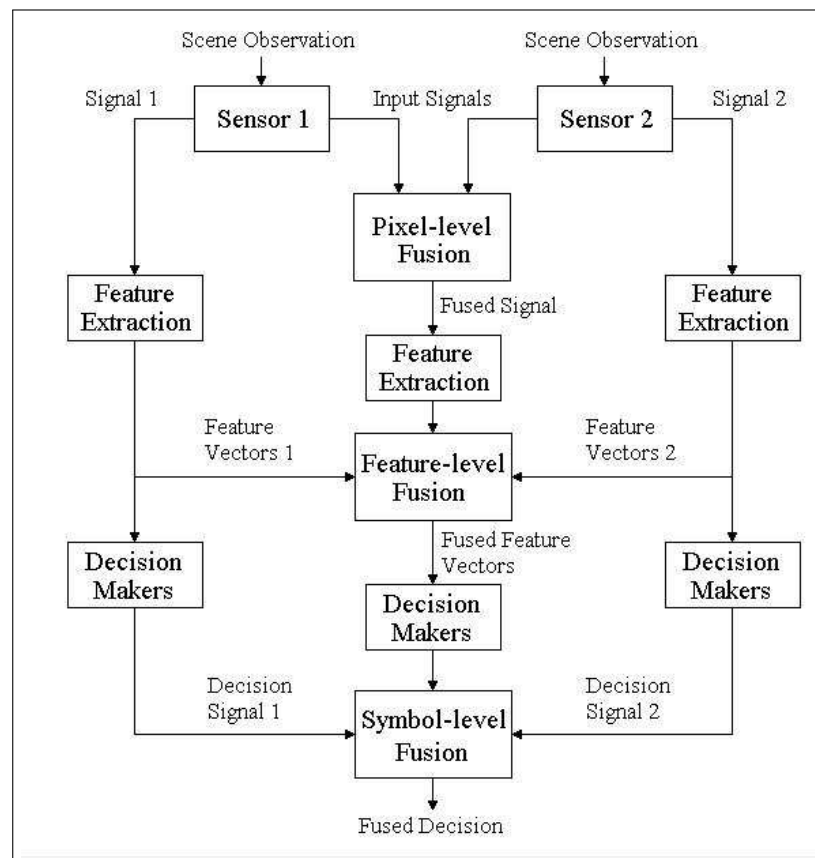


Figure 2.4 – Exemple de combinaison des trois niveaux de fusion.

2.3.2 Concepts important dans le domaine de fusion d'images :

Dans les prochains chapitres, nous utilisons un ensemble de notions propres à la fusion d'images. Pour cette raison, nous avons jugé utile de donner un aperçu sur chaque concept que nous utilisons par la suite.

– **Qu'est-ce qu'une image informatique ?**

Une image informatique est constituée de points de couleurs différentes. L'association (point, couleur) est appelée pixel. La mémoire utile pour stocker un pixel peut varier de 1 bit (cas des images monochromes) à 24 bits (images en 16 millions de couleurs).

Les informations sur la luminance (paramètre Y) et la chrominance (I et Q) sont des combinaisons linéaires des intensités de rouge (R), vert (G), et bleu (B) :

$$Y = 0.30 R + 0.59 G + 0.11 B, I = 0.60 R - 0.28 G - 0.32 B, Q = 0.21 R - 0.52 G + 0.31 B$$

Soit une image 640×480 RGB 24 bits/pixel. Chacune de ces trois variables est reprise sous forme de matrice 640×480 . Comme chaque point de chaque matrice est une info codée sur 8 bits, il y a chaque fois 256 niveaux possibles (0-255). En soustrayant 128 à chaque élément, on met à zéro le milieu de la gamme de valeur possible :-128 à +127.

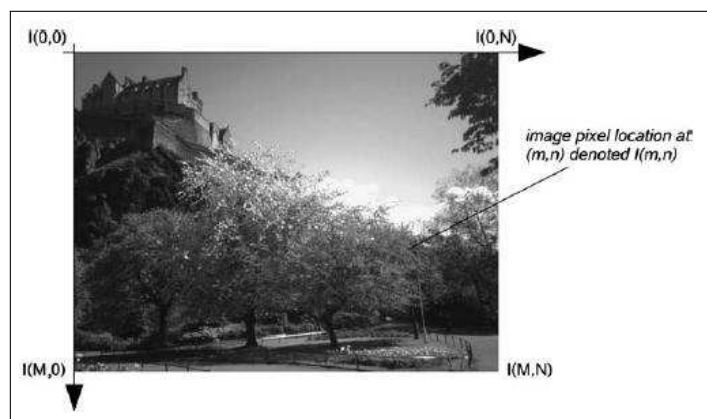


Figure 2.5 – Cordonnées cartésiennes 2-D d'une image numérique de taille $M \times N$.

En termes de fréquence, une image est généralement subdivisée en zones basses fréquences (Low Frequencies), zones hautes fréquences (High Frequencies) et des zones intermédiaires (Medium Frequency).

- **Image registration IR (alignement des images) :** Dans les réseaux de capteurs multimédias, la mise en œuvre d'une fusion d'images nécessite plusieurs manipulations préalables qui interfèrent directement sur la qualité du produit fusionné. Avant de procéder à la fusion d'image proprement dite, un certain nombre de traitements doit être effectué sur les images sources.

Dans la pratique, les capteurs multimédias utilisés pour la capture des images sont soit non identiques, soit identiques avec des paramètres de configuration différents. Ce concept est primordial avant de commencer le processus de fusion. L'alignement des images sources appelé aussi *la correction géométrique* des images consiste à la transformation de plusieurs images dans un même système de coordonnées. Ceci permet d'éliminer le décalage dit à une rotation ou bien un découpage des images sources. Il est à signaler que le processus d'alignement est difficile à atteindre dans certaines situations. La figure 3.6 montre la position de l'alignement des images sources dans le processus de fusion.

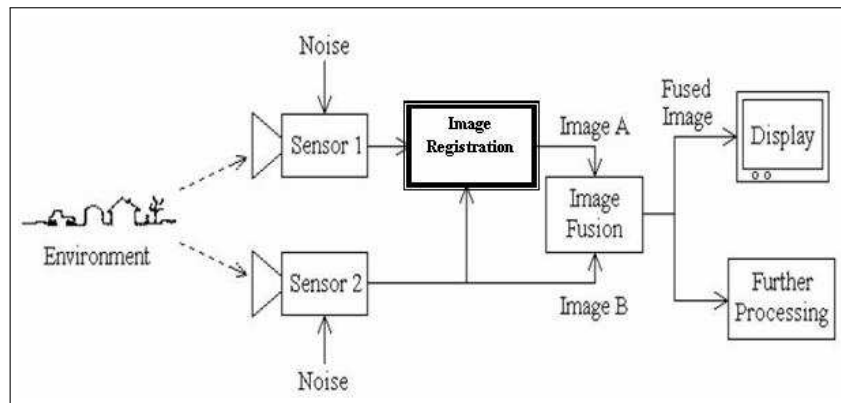


Figure 2.6 – L'alignement des images sources dans le processus de fusion.

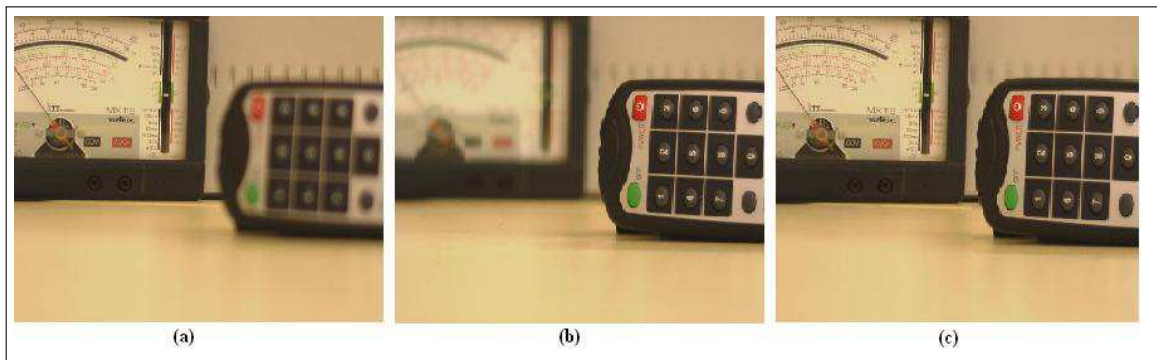
En général, il y a deux types de différences entre les images capturées et celles qu'on veut aligner. Le premier type est dû aux changements dans l'acquisition qui peut provoquer un décalage (translation, rotation, échelle) dans ce cas une transformation spatiale est nécessaire, pour le deuxième type de différences le traitement par une transformation spatiale ne suffit pas, ce type de différences peut être provoqué par un changement de luminosité ou l'utilisation des capteurs avec de paramètres différents. Le processus d'alignement des images sources assure que le même pixel dans les images sources correspond au même point.

Dans le contexte de notre solution nous supposons que les images sont déjà bien alignées avant d'être fusionnées. Plus de détails sur le concept d'alignement des images sont disponibles dans les deux références [29] et [30].

2.3.3 Le concept Multi-focus image fusion (MFIF) :

La plupart des capteurs multimédia utilisés pour la capture des images sont limités en termes de focus (profondeur de champ), c'est pour cette raison que l'utilisation de

plusieurs CM pour couvrir une région est devenue incontournable. Dans cette situation il est souvent impossible d'obtenir une image qui contient tous les objets pertinents dans une scène. Seuls les objets à l'intérieur de la profondeur de champ sont concentrés, alors que d'autres objets sont flous ou brouillés. Pour obtenir une image claire de tous les objets, un processus de fusion des images multi-focus est nécessaire. L'image fusionnée donne une meilleure vue pour la perception humaine ou machine. Le mécanisme de fusion 'multi-focus image fusion' consiste à la combinaison de deux images sources ou plus. Dans la figure ci-dessous, nous supposons que les deux images (a) et (b) sont capturées par deux capteurs visualisant la même région mais chaque capteur est focalisé sur une partie bien déterminée de la scène appelée zone d'intérêt du capteur. Donc un capteur se focalise sur la partie droite (partie droite claire / partie gauche floue), et l'autre se focalise sur la partie gauche (partie gauche claire / partie droite floue) de la scène. L'image (C) représente l'image résultante (fusionnée), il est clair que cette image fusionnée est plus parlante par rapport aux deux images (a) et (b) car sa totalité est claire.



Dans ce mémoire, le mécanisme de fusion d'images que nous proposons est basé sur le concept *Multi-focus image fusion* dans un réseau de capteur.

La technique de fusion dite multi-focus images fusion se base sur plusieurs approches. Dans ce qui suit, nous allons donner un aperçu sur le principe de base de ces approches, un état de l'art sur les travaux de recherches "multi-focus image fusion" est présentée dans le chapitre qui suit.

- **Simple Average** (moyenne simple) Il est clair que la région zoomée d'une image est la région qui contient une intensité élevée de pixels. Le principe de base est de procéder pixel par pixel, c'est-à-dire qu'un pixel de l'image résultante (fusionnée) est obtenu par le calcul de la moyenne des pixels correspondants dans les images sources selon la formule :

$$P_f = \frac{1}{N} \sum_{s=1}^N P_s(i, j) \quad (2.1)$$

Avec N : le nombre d'images sources à fusionner. L'opération est répétée pour chaque pixel [31].

- **Simple Greatest Pixel** (le grand pixel) Au lieu de prendre la moyenne des pixels comme dans l'algorithme précédent, cette méthode considère la valeur la plus élevée du pixel. Si nous avons N images sources, chaque pixel $P(i,j)$ de chaque image source est comparé avec les autres pixels qui ont les mêmes coordonnées (i,j) dans le reste des images sources, la valeur la plus grande est retenue et constitue la valeur du pixel dans l'image résultante [31].
- **Simple block replace** (remplacement de bloc) Contrairement aux deux approches précédentes, cette approche procède par bloc de pixels, pour chaque pixel $P(i,j)$ elle considère les pixels voisins et calcule la valeur moyenne de ces pixels. Le pixel qui dispose d'une moyenne de bloc supérieure est copié vers l'image résultante [31].

N.B : Il est à signaler que les trois techniques de fusion d'image citées ci-dessus utilisent la fusion dite niveau pixel car l'image fusionnée est constituée pixel par pixel ou par bloc de pixels.

2.3.4 Evaluation des images fusionnées

Le processus de fusion d'image doit répondre à certaines exigences, parmi ces exigences nous citons [32] :

- L'image fusionnée doit préserver le maximum possible d'information pertinente,
- Le processus de fusion ne doit jamais introduire de l'inconsistance qui peut induire une erreur dans l'observation humaine,
- Dans l'image fusionnée, le bruit et l'information inutile doivent être éliminés.

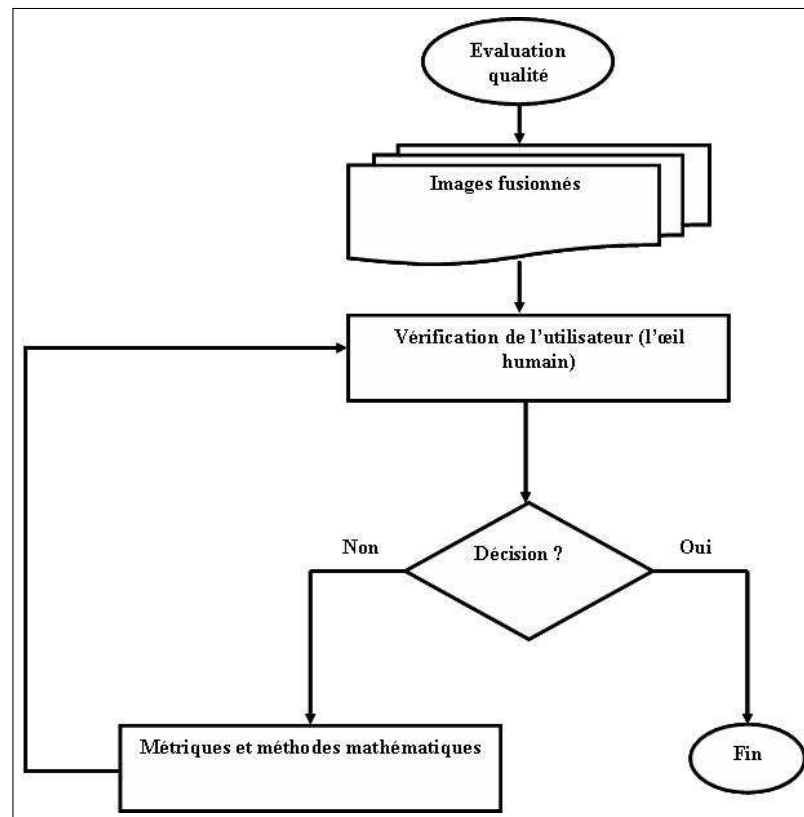


Figure 2.7 – Evaluation de la qualité des images fusionnées.

Dans le processus d'évaluation de l'image fusionnée, la vérification humaine (l'œil de l'utilisateur) est la première étape dans ce processus, après si l'œil humain n'arrive pas à évaluer la qualité de l'image, des outils mathématiques et des métriques sont utilisés pour prendre des décisions et valider le rendu visuel.

Dans la littérature, les méthodes d'évaluation de qualité sont classées en deux volets : méthodes d'évaluation subjectives c'est-à-dire l'évaluation est faite par l'être humain ; et méthodes d'évaluation objectives où l'évaluation se fait par des algorithmes et métriques. Les algorithmes d'évaluation de la qualité de fusion d'images sont regroupés en deux classes :

- **Techniques exigeant une image de référence** : Dans certaines applications, il est possible de générer une image fusionnée dite idéale. Cette image idéale est utilisée par la suite comme étant une image de référence pour la comparer avec l'image fusionnée en utilisant l'algorithme de fusion choisie. Parmi ces techniques on peut citer [33] :

1. The Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{1}{MN} \sum_{i=1}^N \sum_{j=1}^M |R(i, j) - F(i, j)|^2}, \text{ Avec :} \quad (2.2)$$

R : image de référence, F : l'image fusionnée, (i,j) : un pixel donné et $N \times M$: la taille de l'image.

2. The correlation (CORR)

$$CORR = \frac{2R_{R,F}}{R_R + R_F}, \text{ Avec :} \quad (2.3)$$

$R_R = \sum_{i=1}^N \sum_{j=1}^M R(i, j)^2$, $R_F = \sum_{i=1}^N \sum_{j=1}^M F(i, j)^2$ et
 $R_{R,F} = \sum_{i=1}^N \sum_{j=1}^M R(i, j)F(i, j)$.

3. The Peak Signal to Noise Ratio (PSNR)

$$PSNR = 10 \log_{10} \left(\frac{L^2}{\frac{1}{MN} \sum_{i=1}^N \sum_{j=1}^M |R(i, j) - F(i, j)|^2} \right), \quad (2.4)$$

Avec : L : le nombre de niveaux de gris dans l'image.

– **Techniques n'exigeant pas une image de référence :**

1. The standard deviation (SD) [33]

$$\sigma = \sqrt{\sum_{i=0}^L (i - \bar{i})^2 h(i)}, \quad \bar{i} = \sum_{i=0}^L i h(i) \quad (2.5)$$

Avec : h est l'histogramme normalisé de l'image.

2. The entropy (H) [33]

$$H = - \sum_{i=0}^L h(i) \log_2 h(i) \quad (2.6)$$

Avec : h est l'histogramme normalisé de l'image.

3. The overall cross entropy (CE) of the source images X, Y and the fused image F [33]

$$CE(X, Y; F) = \frac{CE(X; F) + CE(Y; F)}{2} \quad (2.7)$$

Avec : $CE(X; F), CE(Y; F)$, les entropies croisées des images sources X, Y et de l'image fusionnée F.

$$CE(X; F) = \sum_{i=0}^L h_X(i) \log_2 \frac{h_X(i)}{h_F(i)} \quad (2.8)$$

4. Métrique de Piella Q_w [34] Dans son papier, l'auteur a proposé une nouvelle métrique pour évaluer les performances des algorithmes de fusion des images. Cette métrique est basée sur l'estimation de la quantité d'information importante qui a été transférée à partir des images sources vers l'image fusionnée. Il est à signaler que cette métrique n'exige pas la présence d'une image de référence afin d'évaluer les performances de l'algorithme de fusion.
5. Métrique de Petrovic $Q^{AB/F}$ [35] Cette métrique mesure la quantité relative de bordures (contours) qui a été transférée à partir des images sources vers l'image fusionnée. Dans la première étape du processus, l'algorithme extrait l'information visuelle à partir des images sources et l'image fusionnée, sous forme de contours (l'intensité et l'orientation du contour). Par la suite ces informations sont utilisées pour estimer la quantité d'information transférée à partir des images sources A et B vers l'image fusionnée. Cela est effectué à l'aide d'une comparaison entre les contours contenus dans les images sources et ceux contenus dans l'image fusionnée. Le résultat de cette étape est la production de deux paramètres Q^{AF} et Q^{BF} qui indiquent la préservation de l'information dans l'image fusionnée F par rapport aux deux images sources A et B. Le paramètre de performance de l'algorithme de fusion $Q^{AB/F}$ représente la somme des deux paramètres Q^{AF} et Q^{BF} en introduisant une notion de pondération sur les contours. La valeur de $Q^{AB/F}$ est incluse entre 0 et 1, une valeur de 0 indique une perte totale d'information entre les images sources et l'image fusionnée, par contre une valeur proche de 1 signifie une fusion idéale.
6. Feature Mutual Information(FMI) [36] Le principe de cette métrique est basé sur l'information mutuelle qui calcule la quantité d'information menée à partir des images sources à l'image fusionnée. L'information considérée est représentée par des caractéristiques (features) d'image comme le dégradé (the gradient) ou les bordures, qui sont souvent sous la forme de signaux bidimensionnels. L'information mutuelle MI mesure le degré de dépendance entre deux variables A et B, en mesurant la distance entre la densité de probabilité jointe donnée par $P(A,B)$ et les distributions marginales $P(A)$ et $P(B)$ (le cas d'une indépendance complète). Cette métrique d'évaluation récente (proposée en 2011) et qui n'exige pas l'existence d'une image de référence est classée parmi les métriques d'évaluation basées sur la quantité d'information transférée à partir des images sources vers l'image fusionnée.

2.4 Conclusion

Dans le présent chapitre, nous avons présenté quelques concepts importants dans le domaine de la fusion de données. Nous avons axé sur la fusion d'images dans un réseau de capteurs où nous avons mis l'accent sur la fusion basée sur le multi-focus image que nous considérons dans notre solution présentée en Chapitre 4. Dans le prochain chapitre nous présentons un état de l'art sur les méthodes de fusion de données utilisées dans les deux types de réseaux de capteurs (scalaires et multimédias).

Chapitre 3

Les méthodes de fusion de données dans les réseaux de capteurs

3.1 Introduction

Durant les dernière années, la fusion de données a trouvé ses applications dans plusieurs domaines tels que la médecine, la vidéo surveillance, etc...

Dans ce contexte, plusieurs approches et techniques de fusion de données ont vu le jour, allant de l'utilisation de plusieurs capteurs complémentaires et/ou concurrents au raisonnement symbolique destiné à l'interprétation de données. Dans le présent chapitre nous présentons un état de l'art récent sur les techniques de fusion de données. Nous classons ces techniques selon différents critères et modalités. Nous explorons d'abord les techniques de fusion des données dans les réseaux de capteurs scalaires WSN, avant celles dédiées pour les réseaux de capteurs multimédias WMSN. Chaque approche présentée est discutée et critiquée.

3.2 Méthodes et algorithmes de fusion dans les WSN

La fusion de données dans les réseaux de capteurs sans fil a constitue de nos jours un domaine de recherche en vogue. Dans ce contexte plusieurs algorithmes et techniques de fusion de données ont été proposés afin de minimiser le volume de données circulant dans le réseau et réduire la consommation d'énergie au niveau des nœuds capteurs. Ces algorithmes fonctionnent comme illustré sur la figure 3.1 ; l'algorithme utilise les

données du capteur à partir du nœud, puis les fusionne en utilisant un algorithme de fusion. Les données fusionnées sont transférées par la suite à la station de base, par la sélection d'un chemin optimal.

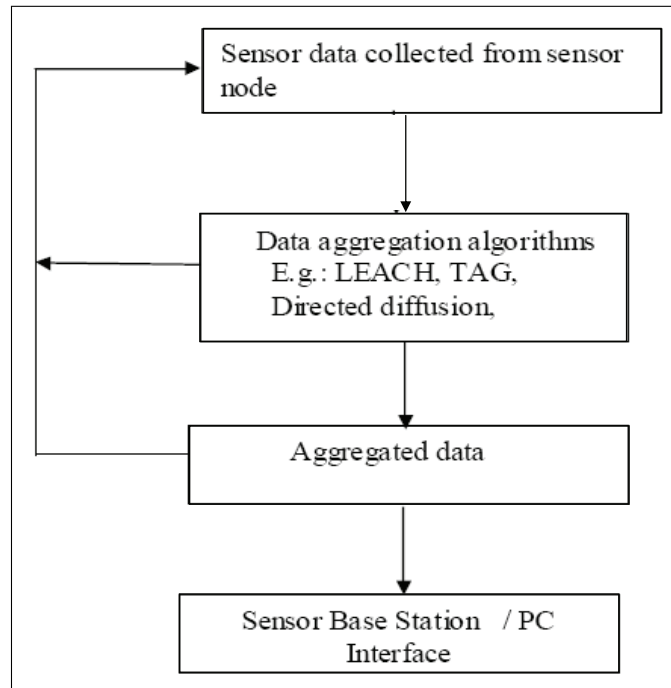


Figure 3.1 – Architecture générale des algorithmes de fusion de données.

Les algorithmes de fusion peuvent être classés en fonction de plusieurs critères, tels que le niveau d'abstraction de données, les paramètres, ou sur une base mathématique, voire le type des données fusionnées. La fusion des données peut être réalisée avec des objectifs différents tels que l'inférence, l'estimation, l'agrégation, la compression, etc. Dans cette section nous allons évoquer un ensemble de ces approches, nous ciblons les méthodes les plus utilisées et les plus citées dans la littérature[17] .

3.2.1 Méthodes par inférence :

La décision est prise sur la base de la connaissance de la situation perçue. L'inférence désigne le passage d'une proposition probablement vraie à une autre dont sa vérité dépend directement de la première proposition.

– **Méthodes Bayésiennes :**

Les méthodes Bayésiennes sont classées parmi les théories probabilistes, elles sont les méthodes les plus exploitées dans le domaine de fusion de données. L'incertitude est représentée en termes de probabilités conditionnelles décrivant la croyance, et elle prend ses valeurs dans l'intervalle $[0, 1]$, où 0 est l'incrédulité absolue et 1 est la croyance absolue. L'inférence Bayésienne est fondée sur la règle Bayes [37]. Dans cette approche, les noeuds capteurs sont considérés comme un ensemble d'entités capables de fournir une décision à tout instant. Chaque capteur est alors vu comme un estimateur Bayésien. Ainsi, les distributions de probabilités associées à chaque capteur sont combinées dans une seule fonction où les probabilités sont jointes a posteriori en utilisant la règle de Bayes donnée par [21] :

$$Pr(Y/X) = \frac{Pr(X/Y)P(Y)}{Pr(X)}, \text{ Avec :} \quad (3.1)$$

- $Pr(Y)$: la probabilité à priori de l'hypothèse Y,
- $Pr(X)$: la probabilité à priori de l'événement X, peut être considérée comme étant une constante normalisée,
- $Pr(X/Y)$: la probabilité d'observer X sachant que l'hypothèse Y est vraie,
- $Pr(Y/X)$: la probabilité à posteriori, représentant la croyance de l'hypothèse Y sachant l'information X.

Formalisation :

Pour formuler mathématiquement cette approche on procède comme suit : soit $E_1=w_1,w_2, \dots,w_i, \dots i = 1, \dots N$, l'ensemble des hypothèses et soit $E_2=x_1,x_2, \dots,x_k, \dots k = 1, \dots K$, l'ensemble des observations, la relation entre les observations x_k et les hypothèses w_i est donnée par un jeu de probabilités conditionnelles comme il est indiqué par l'équation 3.1 où Y représente l'ensemble des hypothèses et X l'ensemble des observations. L'approche Bayésienne pour la fusion de données dans les réseaux de capteurs sans fil à été implémentée dans plusieurs travaux de recherche. Dans [38], il est question de développer un nouveau système de captage pour ADAS (Advanced Driver Assistance Systems). Il a été proposé une technique de fusion de données basée sur le principe Bayésien, cette technique est connue sous le nom de " *Bayesian Programming* ".

Etant donné un ensemble de capteurs qui fournissent des informations sur un environnement de cibles potentielles, le problème est d'estimer certaines caractéristiques de ces cibles à savoir leurs positions et leurs vitesses de déplacement. L'étude a considéré une cible et trois noeuds capteurs différents fournissant des informations sur la distance et la position de la cible. Les différents composants du programme Bayésien sont :

- La distance actuelle z et la position de la cible,
- la distance et la position mesurées par les capteurs $S_i, i=1, \dots 3$.

En appliquant les règles de bayes, on peut écrire la formule suivante :

$$Pr(z\alpha/\rho_1\theta_1\rho_2\theta_2\rho_3\theta_3) = Pr(z\alpha) \prod_{i=1}^3 Pr(\rho_i\theta_i/z\alpha), \text{ Avec :} \quad (3.2)$$

- $P(z\alpha)$: la probabilité à priori de la distance et la position de la cible,
- $P(\rho_i\theta_i/z\alpha)$: la réponse du capteur S_i à la cible localisée à (z, α) ,
- $P(z\alpha/\rho_1\theta_1\rho_2\theta_2\rho_3\theta_3)$: l'estimation de la distance et la position (z, α) sachant les observation $\rho_i\theta_i$.

Résultats de l'expérience :

Le groupe de travail a représenté les résultats de cette expérience par des graphes, ils ont montré graphiquement l'estimation de la distance et la position (z, α) de la cible sachant l'observation de chaque capteur séparément $P(z\alpha/\rho_1\theta_1)$, $P(z\alpha/\rho_2\theta_2)$, $P(z\alpha/\rho_3\theta_3)$ ainsi que l'estimation résultante c'est-à-dire $P(z\alpha/\rho_1\theta_1\rho_2\theta_2\rho_3\theta_3)$. Cette dernière était proche de l'observation retournée par le capteur S_2 (le capteur le plus précis). Pour plus de détails consulter la page 4 du [38].

Discussion :

L'approche Bayésienne est une approche probabiliste qui veut dire qu'elle repose sur une base mathématique solide éprouvée par de nombreuses études depuis plusieurs années. Néanmoins, il est difficile de modéliser des connaissances par des probabilités, en plus de l'existence de certaines contraintes sur les décisions (les classes) telles que que l'exhaustivité et l'exclusivité. Il reste aussi le problème délicat de l'estimation des probabilités à priori où il est souvent nécessaire de faire appel à une connaissance plus subjective sur l'application. Malgré ces limites, cette approche forme une base pour d'autres méthodes plus récentes, telles que les Modèles de Markov Cachés (HMM : Hidden Markov Models), ou encore les réseaux Bayésiens, permettant justement la modélisation de systèmes dynamiques.

– Inférence de Dempster-Shafer :

Connue aussi sous le nom de théories de l'évidence ou des fonctions de croyance. C'est une théorie mathématique introduite par Dempster et Shafer, elle généralise la théorie Bayésienne. Au lieu de manipuler des probabilités comme dans le cas de la règle de Bayes, cette méthode traite des croyances ou des fonctions de masse, elle est utilisée pour la représentation des connaissances incomplètes et pour la combinaison des preuves [39]. Un concept fondamental dans cette méthode est le cadre de discernement (the frame of discernment), qui est dénoté comme suit : Soit $S = (\theta_1, \theta_2, \dots, \theta_N)$, l'ensemble de tous les états possibles qui décrivent le système (les propositions possibles du problème posé), de tel manière que le système est certainement dans un, et un seul état $\theta_i \in S$ et $1 \leq i \leq N$.

Nous appelons l'ensemble S le cadre de discernement parce que ses éléments sont utilisés pour distinguer les états du système réel. Les éléments de l'ensemble de puissance 2^S représentent les hypothèses (toutes les disjonctions possibles des hypothèses ainsi que l'ensemble vide), cet ensemble d'hypothèses est défini comme suit :

$$2^S = (A/A \subseteq S) = \{\emptyset, \{\theta_1\}, \dots, \{\theta_N\}, \{\theta_1, \theta_2\}, \dots, S\} \quad (3.3)$$

Selon une probabilité d'affectation de base, ou une fonction de masse m , une probabilité est affectée à chaque hypothèse $H \in 2^S$. La fonction m est définie comme suit : $m : 2^S \rightarrow [0, 1]$, elle satisfait les conditions :

- $m(\emptyset) = 0$,
- $m(H) \geq 0, \forall H \in 2^S$,
- $\sum_{H \in 2^S} m(H) = 1$

Et pour exprimer la croyance d'une hypothèse H , Dempster-Shafer propose la fonction de croyance appelée **bel**, tel que : $bel : 2^S \rightarrow [0, 1]$

$$bel(H) = \sum_{A \subseteq H} m(A), \text{ Avec : } bel(\emptyset) = 0 \text{ et } bel(S) = 1$$

Le degré de doute dans H peut être intuitivement exprimée en termes de la fonction de croyance $bel : 2^S \rightarrow [0, 1]$

$$dou(H) = bel(\neg H) = \sum_{A \subseteq \neg H} m(A)$$

Pour exprimer la fonction de plausibilité de chaque hypothèse, la fonction $pl : 2^S \rightarrow [0, 1]$ est définie comme :

$$pl(H) = 1 - dou(H) = \sum_{A \cap H = \emptyset} m(A)$$

La plausibilité indique intuitivement qu'il y a moins de doute dans l'hypothèse H . Donc, l'intervalle de confiance $[bel(H), pl(H)]$ définit la vraie croyance de l'hypothèse H . Pour combiner deux fonctions de masse m_1 et m_2 de deux sources d'information S_1 et S_2 , Dempster-Shafer propose la règle de combinaison $m_1 \oplus m_2$ tel que :

$$m_1 \oplus m_2(\emptyset) = 0,$$

$$m_1 \oplus m_2(H) = \frac{\sum_{X \cap Y = H} m_1(X)m_2(Y)}{1 - \sum_{X \cap Y = \emptyset} m_1(X)m_2(Y)}.$$

L'inférence de Dempster-Shafer a été utilisée pour résoudre plusieurs problèmes de fusion de données. Dans [40] cette méthode a été exploitée pour construire (fusionner) des photos dynamiques opérationnelles d'un champs de bataille pour évaluation de la situation. Dans [41] les auteurs l'ont appliquée dans le domaine de traitement de la parole. La théorie de croyance a été employée et adaptée pour la localisation de robots (à partir d'images à ultrason), mais aussi pour l'exploitation d'un réseau autoroutier [42]. Nous énumérons ci-après les situations dans lesquelles la théorie des croyances pourrait s'appliquer [43], :

- Dans le cas idéal où toute l'information est connue; la théorie des croyances est alors optimale et se ramène à l'approche Bayésienne.
- Lorsqu'une source donne des informations seulement sur certaines décisions, par exemple un sonar peut fournir une information sur le sédiment de surface, alors qu'un sondeur renseigne sur les différentes couches de sédiments.
- Lorsqu'une décision ne peut pas différencier deux hypothèses, il faut alors considérer la disjonction de ces deux éléments, ce qui permet de ne pas introduire d'information supplémentaire forçant leur séparation.

Discussion :

La théorie de Dempster-Shafer nous permet de fusionner les données fournies par différents types de capteurs. En outre, dans l'inférence de Dempster-Shafer nous n'avons pas besoin d'attribuer des probabilités a priori à des propositions inconnues comme dans le cas des méthodes Bayésienne. Au lieu de cela, les probabilités sont affectées uniquement lorsque l'information supplémentaire est disponible. Elle est fondée sur les mathématiques ce qui représente un point fort de cette théorie. Néanmoins, l'estimation des fonctions de masse dépend souvent de l'application en question.

– Méthodes de vote :

Ce type de méthodes est simple à mettre en oeuvre, mais plus adapté pour la prise de décision. Elles sont basées sur le principe de combinaison des sorties des nœuds capteurs en employant un procédé de vote. Dans ce contexte on parle de classes et de classifieurs et dans le cas des réseaux de capteurs sans fil, les classifieurs représentent les nœuds capteurs et les classes l'ensemble de valeurs ou événements captés par les nœuds. Les méthodes de vote consistent à interpréter chaque sortie d'un classifieur (nœud) comme un vote pour l'une des classes possibles (valeur de mesure, type d'événement, etc). La classe ayant un nombre de votes majoritaire ou supérieur à un seuil préfixé est retenue comme décision finale [44].

Formalisation :

Dans le contexte d'un réseau de capteur sans fil qui contient m nœuds, l'ensemble de ces nœuds est représenté par :

$$S_j = S_1, S_2, \dots, S_m. \quad (3.4)$$

Notons $S_j(x) = i$, le fait que le nœud S_j attribue la classe C_i à l'observation x . Nous supposons ici que les classes C_i sont exclusives. A chaque source nous associons la fonction indicatrice : $M_i^j(x)$ est la réponse du nœud j pour l'observation x : 0 si x n'appartient pas à la classe i , 1 sinon. Donc il est noté :

$$M_i^j(x) = \begin{cases} 1 & \text{si } S_j(x) = i, \\ 0 & \text{sinon} \end{cases}$$

La combinaison des nœuds capteurs s'écrit par la formule : pour tout k .

$$M_K^E(x) = \sum_{j=1}^m M_K^j(x) \quad (3.5)$$

L'opérateur de combinaison est donc associatif et commutatif. La règle du vote majoritaire consiste à choisir la décision prise par le maximum de sources, c'est-à-dire le maximum de M_K^E . Cependant cette règle simple n'admet pas toujours de solutions dans l'ensemble des classes $D = C_1, \dots, C_n$. En effet, par exemple si le nombre de sources m est pair et que $m/2$ sources décident C_{i_1} et $m/2$ autres sources disent C_{i_2} , ou encore dans le cas où chaque source affecte à x une classe différente. Nous sommes donc obligés d'ajouter une classe C_{n+1} qui représente l'incertitude totale liée au conflit des sources sous l'hypothèse de l'exhaustivité des classes $C_{n+1} = C_1, \dots, C_n$. La décision finale de l'expert prise par cette règle s'écrit donc par :

$$E(x) = \begin{cases} k & \text{si } \max_k M_K^E(x), \\ n+1 & \text{sinon} \end{cases}$$

Cette règle est cependant peu satisfaisante dans les cas où deux sources donnent le maximum pour des classes différentes. La règle la plus employée est la règle du vote majoritaire absolu qui s'écrit :

$$E(x) = \begin{cases} k & \text{si } \max_k M_K^E(x) > m/2, \\ n+1 & \text{sinon} \end{cases}$$

A partir de cette règle il a été démontré plusieurs résultats prouvant que la méthode du vote permet d'obtenir de meilleures performances que toutes les sources prises séparément. Ceci montre de manière théorique l'intérêt de la fusion d'informations. Il est possible de généraliser le principe du vote majoritaire afin de supprimer le conflit. Au lieu de combiner les réponses des sources par une somme simple comme dans l'équation 3.5, l'idée est d'employer une somme pondérée [45] :

$$M_K^E(x) = \sum_{j=1}^m \alpha_{j_k} M_K^j(x) \quad (3.6)$$

avec : $\sum_{j=1}^m \sum_{k=1}^n \alpha_{jk} = 1$. Les poids α_{jk} représentent la fiabilité d'une source pour une décision donnée, et l'estimation de ces poids peut se faire à partir des taux normalisés de réussite pour chaque classe et chaque classifieur ; notons qu'alors une connaissance à priori est introduite. Les différentes règles de décision possibles peuvent se résumer par la formule suivante :

$$E(x) = \begin{cases} k & \text{si } M_K^E(x) = \max_i M_i^E(x) \geq cm + b(x), \\ n + 1 & \text{sinon} \end{cases}$$

Où c est une constante de $[0,1]$ et $b(x)$ est une fonction de $M_K^E(x)$.

Discussion :

- La méthode de vote est une méthode simple à mettre en œuvre. Dans sa version du vote majoritaire absolu ou non il n'y a pas besoin de connaissance à priori sur les nœuds capteurs (les sources) contrairement à d'autres méthodes telles que les méthodes Bayésiennes qui exigent une connaissance à priori. Dans [46] une comparaison de cette méthode avec les différentes approches Bayésiennes et l'approche de Dempster-Shafer pour de la reconnaissance manuscrite a été réalisée. Sa conclusion montre que l'approche par vote reste moins bonne que les deux autres mais moins coûteuse.
- Cependant [47] montre que dans le cas de la reconnaissance de cibles sur des images captées par de nœuds capteurs, la fusion des résultats de quatre classifieurs (nœuds) par la méthode du vote donne des résultats équivalents à l'approche Bayésienne.
- De nombreuses études ont comparé le vote majoritaire dans le cas de la classification ou de la reconnaissance. Même si cette règle reste simple, elle donne souvent des résultats comparables à des approches plus complexes.

-Tableau comparatif des méthodes de fusion par inférence

Dans cette section nous comparons les trois méthodes suscitées à travers le tableau qui suit :

	Méthodes Bayésiennes	Inférence Dempster-Shafer	Méthodes de vote
Connaissance à priori sur les sources	Avec connaissance à priori	Sans connaissances à priori	Sans connaissances à priori
Domaine d'application	Très utilisée dans le traitement d'images et la reconnaissance de formes	Téledétection, identification de cibles, imagerie, médicale	lorsque l'information s'exprime sous forme d'hypothèses (reconnaissance, détection, identification)
Niveau de fusion	Bas, Moyen niveau	Niveau moyen	Haut niveau (décision)
Complexité de calcul	Calcul de distributions à posteriori compliqué	complexité à croissance exponentielle avec la taille du cadre de discernement	Simple à mettre en œuvre
Scalabilité	Scalable (un nouveau nœud est vu comme un estimateur Bayésien)	Le passage à l'échelle augmente le cadre de discernement ce qui augmente la complexité	La scalabilité influe directement sur la dimension de l'ensemble des classifieurs (nœuds)
Qualité des résultats (performances)	Optimale lorsque l'apriori est parfaitement connu	Meilleur résultats (généralisation des méthodes Bayésiennes)	Moins bons que les deux autres méthodes
Modélisation	Fondée sur les probabilités conditionnelles	Basée sur les fonctions de représentation de l'imprécision	fondée sur la définition des fonctions indicatrices
Avantages majeurs	repose sur une base mathématique solide	Espace d'hypothèses composées 2D au lieu de D	-Bien adaptée à la prise de décision - Sans apprentissage
Inconvénients majeurs	-Connaissances qui ne se traduisent pas par des Probabilités, -Connaissances à priori.	Une phase d'apprentissage est souvent indispensable	Exige un nombre impair de sources

Tableau 3.1 – Tableau comparatif des méthodes de fusion par inférence.

Commentaires : Pour les trois méthodes de fusion discutées ci-dessus, l'utilisation de l'une ou de l'autre pour une situation donnée dépend de l'application. Ainsi pour une distribution de probabilité avec valeurs connues on utilisera l'approche Bayésienne, pour prendre des décisions il faut utiliser le vote, pour mesurer l'incertain on optera pour Dempster et Shafer..., mais cela reste toujours dépendant du contexte ou du domaine d'application en question.

3.2.2 Méthodes par estimation :

La fusion par estimation utilise les lois de probabilité pour calculer un vecteur d'état de processus à partir d'un vecteur de mesure ou une séquence de vecteurs de mesure.

– **Maximum likelihood estimation (MLE) :**

Une fois un système spécifié avec ces paramètres et les données déjà collectées nous sommes en mesure d'évaluer sa qualité d'ajustement, et de l'adapter aux données observées. La qualité de l'ajustement est évaluée par la recherche de valeurs de paramètres d'un modèle qui correspond le mieux aux données. Autrement dit

c'est la procédure d'estimation des paramètres. Dans un contexte de fusion de données, on peut formaliser cette approche comme suit : soit x l'état estimé et $z = \{z(1), \dots, z(k)\}$ une séquence de k observations de x , la fonction likelihood (vraisemblance) notée $\lambda(x)$ est définie comme la fonction de densité de probabilité de la séquence d'observation z sachant la valeur vraie de l'état x :

$$\lambda(x) = P(z/x) \quad (3.7)$$

Les estimateurs de vraisemblance maximum cherchent la valeur de x qui maximise la fonction de vraisemblance :

$$\hat{x}(k) = \arg \max_x P(z/x). \quad (3.8)$$

Dans [48] les auteurs proposent une méthode MLE distribuée qui est robuste aux liens de communication non fiables de réseaux de capteurs. Dans cette méthode, chaque nœud calcule une estimation raisonnable locale qui converge vers la solution du maximum de vraisemblance globale. Les auteurs ont étendu cette méthode pour supporter l'asynchronisme et les mesures livrées en temps opportun (mesures prises en différentes périodes de temps d'une manière asynchrone). MLE est utilisée couramment pour résoudre le problème de localisation, pour obtenir des estimations précises de la distance (direction et angle). Le problème qu'on peut reprocher à cette méthode est qu'elle est très coûteuse en termes de temps de calcul.

– **Le filtre de Kalman :**

Le filtre de Kalman permet d'estimer les états d'un système dynamique à partir d'une série de mesures incomplètes ou bruitées. Cette méthode de fusion de données est considérée comme étant la méthode fusion de données la plus populaire. Le filtre de Kalman permet aussi d'avoir une information continue sur la position ou la vitesse d'un objet à travers une série d'observation relative à la position de cet objet avec un taux d'erreurs de mesures.

Il est à noter que le filtre de Kalman est un estimateur récursif utilisé pour fusionner des données redondantes de bas niveau, il donne des résultats optimaux dans le cas où le système est décrit par un modèle linéaire et l'erreur est modélisée par un bruit Gaussien. Dans les WSN, le filtre de Kalman a été appliqué pour affiner les estimations de positionnement et de distance [49].

Une préoccupation importante est la perte de données due à des canaux de communication non fiables dans les réseaux de capteurs sans fil. Dans ce contexte, [50] a évalué la performance du filtre de Kalman dans un scénario avec des observations intermittentes et a montré l'existence d'une valeur critique du taux d'arrivée des observations, au-delà duquel le filtre de Kalman devient instable. Un autre problème d'utilisation du filtre de Kalman dans les WSN est qu'il exige une synchronisation des horloges des nœuds capteurs.

-Tableau comparatif des méthodes de fusion par estimation

	Maximum likelihood estimation (MLE)	Le filtre de Kalman
Principe de base	repose sur la notion de vraisemblance d'un ensemble donné d'observations relatives à un modèle	C'est un estimateur récursif
Domaine d'application	utilisée couramment pour résoudre les problèmes de localisation par des capteurs	l'association de données qui permet de trouver parmi différents signaux ceux qui sont émis par la même source (radar, vision électronique, communication)
Niveau de fusion	Niveau signal (bas niveau)	Bas niveau (numériques)
Complexité de calcul	Calcul intensif	Le coût de calcul devient une contrainte très forte pour des systèmes de grande taille et non linéaires
Qualité des résultats (performances)	Résultats dépendent du modèle d'estimation	- Dans le cas gaussien et linéaire il est optimal dans le sens Bayésien, - Dépend des modèles mathématiques et stochastiques utilisés dans le procédé de filtrage
Modélisation	L'état à estimer est modélisé par une fonction de densité de probabilité de la séquence d'observation de l'état	Le système est décrit par un modèle linéaire, le bruit est Gaussien
Avantages majeurs	- Utilise toute l'information contenue dans les données, - Statistiquement fondée	- Prédiction des paramètres et rectification des erreurs, - Considérée comme étant la méthode de fusion de données la plus populaire, - Capacité à déterminer l'erreur moyenne de son estimation.
Inconvénients majeurs	- Nécessite des hypothèses assez fortes , - La fiabilité des résultats dépend directement du modèle choisi	- Permet de prendre en compte uniquement un modèle de bruit Gaussien

Tableau 3.2 – Tableau Comparatif des méthodes de fusion par estimation.

Commentaire : Le filtre de Kalman est donc une méthode d'estimation intéressante, mais qui n'est utilisable que lorsque l'on peut décrire assez précisément notre système. S'il est impossible de trouver une modélisation correcte du système, il est alors préférable de se tourner vers d'autres méthodes (comme la méthode de Monté-Carlo par exemple qui est une méthode statistique, mais qui requiert une importante puissance de calcul.

3.2.3 Méthodes par agrégation :

Les techniques d'agrégation sont les fonctions de synthèse de données telles que la moyenne, le maximum, etc. L'agrégation de données représente un cas particulier de la fusion de données permettant de réduire le volume de données par rapport au volume initial. Dans ce qui suit nous présentons quelques méthodes d'agrégation des données dans les WSN.

A)- Méthodes de calcul de la moyenne consensus dans un WSN :

Dans un réseau de capteurs, chaque nœud capteur prend une mesure donnée, l'objectif ensuite est de trouver la moyenne de ces mesures mesurées par l'ensemble des nœuds, ce problème est connu sous le nom de la moyenne consensus dans les réseaux de capteurs. A titre d'exemple on peut citer la mesure de la température moyenne dans une zone d'intérêt. Nous classons ces méthodes parmi les méthodes d'agrégation. Dans ce contexte, plusieurs méthodes et algorithmes ont été proposés, dans cette section nous abordons les plus usités.

Formalisation : La formalisation suivante est souvent utilisée pour montrer l'ensemble de méthodes et d'algorithmes de calcul de la moyenne consensus dans les réseaux de capteurs [4] :

- Soit $G = (V, E)$ le graphe connexe non-orienté qui représente le réseau de capteurs sans fil, avec : V : Ensemble de nœuds du réseau (les sommets du graphe), marqués par $i = 1, 2, \dots, n$. et E : Ensemble de liaisons entre ces nœuds (les arcs), une liaison entre i et j est désignée par (i, j) .
- Soit $G(t) = (V, E(t))$ le graphe du réseau de capteurs, variable dans le temps, où : $E(t)$ est l'ensemble des liaisons actives à l'instant t , et V est l'ensemble de nœuds qui restent invariables pour l'instant t .
- Soit $N_i(t) = \{j \in V | (i, j) \in E(t)\}$: l'ensemble des voisins du nœud i à l'instant t et $n_i(t)$ le degré (nombre de voisins) du nœud i tel que $n_i(t) = |N_i(t)|$. On considère que chaque message envoyé par i à l'instant t est en fait diffusé à tous les nœuds de $N_i(t)$. Il est supposé que les communications sont bidirectionnelles et que chaque nœud du réseau maintient un état dynamique de la moyenne, notée $x_i(t)$, sachant qu'initialement $x_i(0) = z_i$ (la mesure initiale du nœud i). Donc, il est clair que l'état $x_i(t)$ du nœud i représente une estimation de la moyenne consensus $\sum_{i=1}^n \frac{z_i}{n}$, où n est le nombre total des nœuds dans le réseau. L'objectif de l'algorithme de fusion est de faire tendre tous les états des nœuds vers $\sum_{i=1}^n \frac{z_i}{n}$, quand $t \rightarrow \infty$.

1. **Méthodes par inondation (Flooding) :** Ces méthodes utilisent des algorithmes distribués. Généralement elles sont utilisées pour calculer la moyenne consensus des données mesurées par les nœuds capteurs. La méthode d'agrégation de données par inondation est parmi les premières méthodes utilisées dans ce domaine de recherche. Via cette méthode, chaque nœud capteur diffuse toutes ses données stockées et reçues à ses voisins. Comme il doit stocker les nouvelles données reçues, il aura à la fin toutes les données du réseau et agit comme un centre de fusion pour calculer la moyenne consensus [48]. Néanmoins, cette tech-

nique a plusieurs inconvénients. Par inondation, des messages peuvent être dupliqués. De plus, cette technique est coûteuse en bande passante (donc en énergie) et en mémoire de stockage, ce qui diminue considérablement la durée de vie du réseau.

2. **Le travail de Olfati-Saber et Murray :** Olfati-Saber et Murray [51] proposent le système discret suivant pour calculer la moyenne consensus dans un réseau : $x(t+1) = x(t) - \gamma Lx(t)$, où γ est un paramètre et L est la matrice Laplacienne associée au graphe G représentant le réseau. Elle est définie comme suit :

$$L_{ij} = \begin{cases} n_i & \text{si } i = j, \\ -1 & \text{s'il existe une liaison entre } i \text{ et } j, \\ 0 & \text{sinon.} \end{cases}$$

Il a été démontré que tous les états x_i de tous les nœuds tendent vers la moyenne consensus lorsque γ satisfait la condition suivante : $0 < \gamma < \frac{1}{2n_{max}}$, Où n_{max} est le degré maximal de tous les degrés n_i . Dans la section 5 du [51] l'auteur a donné un contre exemple montrant que pour certains graphes, la solution au problème de la moyenne consensus dans un réseau de capteur est non réalisable. La simulation de la technique a montré l'efficacité des résultats théoriques de l'algorithme proposé.

Discussion :

L'approche proposée représente un avantage du fait qu'elle est distribuée et non pas centralisée ; C'est à dire que chaque nœud du réseau détient dans son état la valeur de la moyenne consensus et non pas un seul nœud qui détient la valeur de la moyenne de tout le réseau. Néanmoins, l'auteur de cet algorithme n'a pas abordé la Scalabilité de son algorithme. Il semble que lors du passage à l'échelle, le graphe G devient volumineux et par conséquent cela induit un fort coût de modélisation. Cette méthode est une méthode itérative ce qui induit un risque de divergence lorsque le réseau change de topologie par exemple. En plus, l'auteur n'a pas abordé les situations de pannes où les noeuds ne fournissent pas de valeurs de mesure.

3. **Le travail de Xiao et al :** Dans [48], les deux auteurs ont proposé un algorithme distribué pour le calcul de la moyenne consensus, c'est un algorithme itératif, où après un certain nombre d'itérations tous les états des noeuds $x_i(t)$ convergent vers la moyenne consensus. Ils ont considéré un réseau de n capteurs, où chaque noeud i prend une mesure initiale z_i , échange son état instantané avec ses voisins et met à jour son état $x_i(t)$ selon le système discret suivant :

$$x_i(t+1) = x_i(t) - \sum_{j \in N_i} \alpha_{ij}(t)(x_i(t) - x_j(t)), i = 1, \dots, n. \quad (3.9)$$

Où : α_{ij} , représentent le poids de nœud j sur le nœud i , et $\alpha_{ij}(t) = 0$ pour $j \notin N_i(t)$; $N_i(t)$ est l'ensemble des voisins du nœud i à l'instant t . Dans un second lieu les auteurs présentent leur système sous la forme :

$$x(t+1) = \alpha(t)x(t), \quad (3.10)$$

$$(3.11)$$

Où x est le vecteur ayant comme composantes tous les x_i tel que $x(0) = z$, et $\alpha \in R_{n \times n}$ est la matrice des poids. Une condition est nécessaire pour la convergence de l'algorithme :

$$\lim_{t \rightarrow \infty} \alpha^t = \frac{1}{n} \mathbf{1}^t,$$

Où : $\mathbf{1}$ représente le vecteur dont toutes les composantes sont égale à un. Pour le choix des poids α_{ij} , les auteurs ont proposé deux méthodes : maximum-degree et Metropolis. La matrice de poids Metropolis est définie comme suit :

$$\alpha_{ij} = \begin{cases} \frac{1}{1+\max(\eta_i, \eta_j)} & , si \quad j \in N_i, \\ 1 - \sum_{i,k \in N_i} \alpha_{ik} & , si \quad i = j, \\ 0 & sinon. \end{cases}$$

Suivant les poids définis ci-dessus, l'équation est considérée comme un mécanisme pour la fusion de données, où chaque nœud met à jour son état instantané suivant la différence entre sa propre donnée et les données de ses voisins.

Discussion :

L'approche proposée exige une synchronisation des nœuds capteurs, chose qui n'est pas réalisable à cause des délais de transmission. En plus durant l'échange de messages entre nœuds il est fort possible qu'il y ait perte de messages à cause des interférences.

4. **Le travail de Abdallah Makhoul :** L'auteur du travail [4] a proposé un algorithme basé sur la diffusion pour calculer la moyenne consensus dans un réseau de capteurs. Chaque nœud prend une mesure, puis collabore avec ses voisins afin d'estimer la moyenne de toutes les mesures collectées. L'algorithme proposé s'inspire de l'algorithme de Bertsekas et Tsitsiklis [52] qui est un algorithme d'équilibrage de charge, il a été adapté pour qu'il soit robuste aux changements dynamiques de la topologie et aux pertes des messages. Du moment que l'algorithme proposé supporte les délais de transmission, l'auteur a considéré qu'à l'instant t un nœud i reçoit l'état de son voisin j envoyé à l'instant $d_j^i(t)$, tel que : $0 \leq d_j^i(t) \leq t$, et $t - d_j^i(t)$ représente le délai de transmission entre les nœuds i et

j . Pour ce faire, il a été considéré deux scalaires importants pour mettre à jour l'état instantané du noeud en retranchant le premier et en ajoutant le deuxième à l'état courant. Les deux scalaires sont : $s_{ij}(t)$ calculé par le noeud i et envoyé à son voisin j tandis que le second $r_{ji}(t) = s_{ji}(d_j^i(t))$ est calculé par le noeud j à l'instant $d_j^i(t)$ et reçu par son voisin i à l'instant t . L'algorithme proposé est le suivant :

- (a) Chaque noeud maintient un état instantané $x_i(t) \in \mathbf{R}$, et à l'instant $t = 0$ (après que tous les noeuds aient pris les mesures initiales), il initialise son état $x_i(0) = z_i$,
- (b) A chaque instant t et pour chaque noeud i :
 - Compare son état aux états de ses voisins,
 - Calcule $s_{ij}(t)$, ils doivent être calculés d'une manière à assurer la convergence de l'algorithme,
 - Diffuse ses données,
 - Reçoit le scalaire calculé par ses voisins $r_{ji}(t)$,
 - Met à jour son état selon l'opération suivante :

$$x_i(t+1) = x_i(t) - \sum_{j \in N_i(t)} S_{ij}(t) + \sum_{j \in N_i(t)} r_{ji}(t). \quad (3.12)$$

Pour le choix de $s_{ij}(t)$, l'auteur a considéré une différence pondérée entre les états des noeuds i et j par une valeur $\alpha_{ij}(t)$ comme suit :

$$S_{ij}(t) = \begin{cases} \alpha_{ij}(t)(x_i(t) - x_j^i(t)), & \text{si } x_i(t) > x_j^i(t), \\ 0 & \text{sinon.} \end{cases}$$

Pour le choix du coefficient α_{ij} , l'auteur a proposé un autre algorithme où chaque noeud i met à jour ses poids à chaque instant t .

Les résultats de simulation ont montré qu'après un certain nombre d'itérations tous les états des noeuds convergent vers la moyenne consensus. L'auteur a même donné une preuve de convergence de son algorithme.

Discussion :

L'algorithme proposé par [4] est un algorithme distribué qui ne se base sur aucune synchronisation entre les noeuds, en plus il ne nécessite pas de connaissance de la topologie globale du réseau. L'auteur a proposé également un mécanisme qui prend en charge le cas de perte des messages, ce mécanisme est basé sur l'envoi de la somme des $S_{ij}(t)$ au lieu du scalaire $S_{ij}(t)$. Donc chaque noeud garde la somme de tous les scalaires calculés jusqu'à l'instant t , et à son tour le noeud j recevant les messages du noeud i garde la somme de tous les scalaires reçus par le noeud i avant l'instant t . Néanmoins, l'algorithme est itératif et lors du passage à l'échelle, le nombre d'itérations devient considérable, ce qui augmente la consommation de l'énergie.

-Tableau comparatif des méthodes de calcul de la moyenne consensus dans un WSN

	Alg. de Olfati-Saber et Murray	Alg. de Xiao et al	Alg. de Abdallah Makhoul
Asynchronisme (délais)	Fusion de données synchrone	Fusion de données synchrone	Fusion de données asynchrone
Topologie réseau	Considère une architecture statique	Considère une architecture statique	Tolérant à la topologie dynamique du réseau
Pertes des messages	Non pris en charge	Non pris en charge	Robuste à la perte de messages
Convergence de l'algorithme	Converge après un certain nombre d'itération	Converge après un certain nombre d'itération	Converge avec moins d'itérations que les deux autres
Scalabilité	Testé pour 10 nœuds	Testé pour 50 nœuds	Testé pour un nombre de nœuds allons de 20 à 200
Complexité de calcul	Dépend du nombre d'itération avant la convergence	Dépend du nombre d'itération avant la convergence	L'optimisation du nombre d'itération influe sur le temps de réponse de l'algorithme
Modélisation	Le WSN est modélisé par un graphe orienté	Le WSN est modélisé par un graphe orienté	Le WSN est modélisé par un graphe orienté

Tableau 3.3 – Tableau comparatif des méthodes de calcul de la moyenne consensus dans un WSN.

Commentaires : Il est évident d'après ce tableau que l'algorithme proposé par Abdelah Makhoul dans sa thèse de doctorat représente un bon choix par rapport aux deux autres algorithmes proposés pour le calcul de la moyenne consensus dans un réseau de capteurs sans fil.

B)- Protocoles d'agrégation de données :

Dans cette section, nous présentons les techniques utilisées pour la distribution et le transport des données à agréger.

– L'approche EEIA (Energy Efficient Indexed Aggregation) :

Cette approche d'agrégation de données dans les réseaux de capteurs sans fil a été proposée par [53]. EEIA est un algorithme distribué destiné au traitement des requêtes dans les WSN afin de diminuer la consommation d'énergie et des ressources. Les trois objectifs visés par cet algorithme sont : une faible consommation d'énergie, stockage et utilisation limités de la mémoire des nœuds capteurs. Le premier souci est de réduire la taille des paquets ainsi que le nombre de paquets envoyés. Pour diminuer la taille des paquets, chaque capteur doit connaître tous ses nœuds enfants pour effectuer l'agrégation partielle avant d'envoyer les données à son parent cela permet de réduire la taille des paquets circulant dans le réseau. Afin de pouvoir interroger les données avec un nombre

minimum de paquets échangés, l'auteur a proposé d'indexer le réseau en utilisant un arbre d'index. Le principe d'indexation est le suivant :

Chaque noeud dans l'arbre envoie son index à son parent. De là, le parent déduit le nombre d'enfants qu'il possède, ainsi il compare les index reçus de chaque noeud enfant : si ils sont égaux, il envoie son index avec ceux de ses enfants à son propre parent. Sinon il ne fait rien. Lors d'un changement d'index d'un noeud, ce noeud envoie son nouveau index à son parent, ce dernier vérifie de nouveau si les index sont égaux. Si ce n'est pas le cas et qu'il est indexé, il met à jour son index et informe son parent si ce dernier est indexé, sinon il ne l'informe pas.

Pour minimiser la consommation d'énergie, l'auteur a proposé une indexation en fonction de l'évaluation de l'énergie disponible à chaque noeud. On peut utiliser par exemple un index de 0 pour noter que certains noeuds sont faibles en termes d'énergie et qu'ils seront éliminés lors de l'exécution de la requête.

Dans la section 5 [53], l'auteur a fourni une simulation et une analyse des résultats de son algorithme. Il a utilisé son propre simulateur développé en VB pour évaluer son algorithme en termes de consommation d'énergie, durée de vie et latence. Une comparaison des résultats avec ceux de l'algorithme naïve et l'algorithme TinyDB simple a été présentée. Il a été constaté qu'en termes de consommation d'énergie et de durée de vie du réseau, l'algorithme EEIA donne de meilleurs résultats que les deux autres approches.

Discussion :

L'algorithme EEIA présente une approche distribuée pour l'agrégation des données dans un WSN, cet algorithme vise à réduire la taille, le nombre de paquets circulants dans le réseau ainsi que la consommation d'énergie. L'auteur a montré à travers des simulations l'efficacité de cette approche par rapport à d'autres, néanmoins cette approche doit faire face aux problèmes de topologie, routage de données et tolérance aux pertes par l'introduction de plusieurs techniques d'optimisation. Par ailleurs, l'évolution du réseau ainsi que l'effet de quelques paramètres externes tels que la mobilité des noeuds et des obstacles ont été ignorés.

– **L'approche EECD (Energy-Efficient Clustering & Data Aggregation) :** EECD est une approche d'agrégation de données dans les WSN [54]. Présentée récemment (2011), elle est destinée au WSN hétérogènes, elle équilibre la consommation d'énergie et prolonge la durée de vie du réseau par un facteur de 51% par rapport à LEACH (Low-Energy Adaptive Clustering Hierarchy) qui est un protocole d'agrégation de données proposé en 2000. L'objectif principal du protocole EECD est de maintenir efficacement la consommation d'énergie des noeuds capteurs en les impliquant dans une communication à un seul saut au sein d'un cluster. L'agrégation des données et la technique de fusion sont utilisées pour réduire le nombre de messages transmis à la station de base cela permet d'économiser l'énergie et d'éviter la congestion. Pour la mise en oeuvre de

ce protocole, l'auteur a adopté les hypothèses suivantes :

- N nœuds capteurs sont uniformément dispersés dans un champ carré ;
- Tous les nœuds capteurs et la station de base sont à l'arrêt après le déploiement ;
- Le WSN se compose de nœuds hétérogènes en termes d'énergie ;
- Le CH (Cluster head) effectue l'agrégation des données ;
- La station de base n'est pas limitée en termes d'énergie contrairement aux autres nœuds du réseau.

EECDA effectue l'agrégation en trois phases :

1. **Election du CH** :EECDA considère trois types de nœuds (normal, avancé et super), les nœuds avec une énergie plus élevée sont de type avancé et super, les autres nœuds sont des nœuds normaux. Intuitivement, les nœuds de type avancé et super doivent devenir des CHs plus souvent que les nœuds normaux.
2. **Sélection de la route** : Chaque CH doit estimer son résidu d'énergie et diffuser cette information aux nœuds voisins.
3. **Communication de données** : Chaque nœud non-CH transmet ses données au CH associée. Chaque CH recevra tous les données détectées par ses noeuds associés (non-CH) et l'envoi à la station de base. Par conséquent, chacun des CH procède à une agrégation de données reçues et transmet ensuite les données agrégées à la station de base (BS).

Discussion :

Afin de comparer les performances de cette approche avec d'autres approches d'agrégation de données dans un WSN hétérogène tel que l'approche LEACH, l'auteur a présenté deux scénarios de simulation, un premier scénario qui est un réseau avec 100 nœuds déployés dans une surface carrée de 100×100 mètres, le deuxième scénario est un réseau de 200 nœuds déployés dans une surface carrée de 200×200 mètres. Les résultats ont montré que EECDA prolonge la durée de vie du réseau de 51% et que le nombre de nœuds en vie est important par rapport au protocole LEACH. Pour le résidu d'énergie, LEACH donne un résidu faible par rapport à EECDA. Cependant, EECDA ignore les problèmes liés à la mobilité des nœuds et le changement de topologie du réseau de capteurs.

- **Algorithme de C. Asgari et J.A.Torkestani** :

Cet algorithme présenté dans [55], considère le réseau de capteurs comme étant un graphe dont les sommets sont les nœuds capteurs et les arcs sont les liaisons entre ces nœuds. L'algorithme proposé est basé sur un automate d'apprentissage distribué implémenté au niveau de chaque nœud. D'après l'auteur, il n'est pas nécessaire d'utiliser tous les nœuds du réseau pour effectuer une agrégation, il suffit uniquement d'utiliser les nœuds Backbones qui envoient un seul paquet à la station de base. L'ensemble de nœuds constituant le Backbone est appelé *Connected Dominating Set*, chaque nœud de cet ensemble est appelé *dominant*. La station de base est considérée comme étant le

premier dominant, par la suite l'automate d'apprentissage sélectionne le suivant d'une manière aléatoire, ce processus continue jusqu'à la couverture de tous les réseaux où un ensemble de dominants constituant le Backbone est formé. A la fin, la station de base envoie un message " *data agregation*" aux nœuds dominants à l'intérieur du Backbone. Les nœuds dominants envoient directement le message à leurs parents, chaque parent doit patienter jusqu'à l'arrivée des messages de tous les fils. Par la suite il effectue une agrégation de données reçus et envoie ces données agrégées à son parent jusqu'à l'arrivée des données à la station de base sous forme de paquets uniques. Dans la section 5, l'auteur a présenté une simulation de son algorithme avec 50, 100 et 150 nœuds déployés dans une zone carrée de 100x100m. La comparaison de son algorithme avec [56], [57] ont démontré que la durée de vie du réseau diminue si le nombre de nœuds augmente ou si l'intervalle de transmission est allongé. Enfin, l'auteur a montré que son algorithme est meilleur que les deux autres algorithmes en termes de durée de vie.

Tableau comparatif des Protocoles d'agrégation de données dans un WSN

	EEIA	EECDA	C. Asgari et J.A.Torkestani
Nombre de messages circulant	L'utilisation d'un arbre d'index a réduit le nombre de messages	Le nombre de messages est réduit grâce à la communication à un seul saut au sein d'un cluster	Nombre de message réduit grâce à la constitution des Backbones
Taille de messages	La taille des messages est réduite grâce à une agrégation partielle (avant l'envoi au parent)	Non pris en charge	Non discutée
tolérance aux pannes	Non pris en charge	Non pris en charge	Non pris en charge
Consommation de l'énergie	Evaluation d'énergie disponible à chaque nœud	Réduite de 51% par rapport à LEACH	Mieux que d'autres algorithmes présentés en [56] et [57]
Mobilité des nœuds	Non pris en charge	Non pris en charge	Non pris en charge
Scalabilité	Simulé avec 500 nœuds, il a été concluant	Simulé avec 100 et 200 nœuds	Simulé avec 50, 100 et 150 nœuds
Topologie du réseau	Ne supporte pas une topologie dynamique	Ne supporte pas une topologie dynamique	Ne supporte pas une topologie dynamique

Tableau 3.4 – Tableau comparatif des protocoles d'agrégation dans un WSN.

Commentaires :

Parmi ces trois protocoles d'agrégation de données dans un réseau de capteurs sans fils et d'après les résultats présentés par les auteurs de ces algorithmes à travers les différentes simulations, il est clair que le protocole **EECDA** réduit considérablement la consommation d'énergie par rapport au protocole **LEACH** d'un rapport de 51% rallongeant ainsi la durée de vie du réseau.

3.3 Méthodes et algorithmes de fusion dans les WMSN

Dans un réseau de capteurs multimédias, la transmission des données multimédias exige une bande passante considérable et par conséquent induit une consommation d'énergie supplémentaire. L'un des aspects pertinents dans les WMSN est le problème des données redondantes au moment de la collecte de l'information par les nœuds capteurs. Cette contrainte exige l'existence des mécanismes de fusion et d'agrégation de données. Le processus de fusion passe par trois étapes comme montré en figure 3.2 : L'acquisition de données, le traitement de données et la transmission de données. Dans la partie traitement de données, différentes techniques sont proposées[58]. Parmi les méthodes étudiées se trouvent celles définies initialement pour les WSN et adaptées aux WMSN. Par exemple, la méthode de Dempster et Shafer a été utilisée pour fusionner des images dans un réseau de capteurs multimédias. Dans cette section nous présentons les techniques les plus récentes dédiées à la fusion des données dans les WMSN.

La plupart des méthodes que nous allons discuter dans cette section, s'intéressent à la fusion des images statiques (still images), la fusion d'une séquence vidéo consiste à la fusion de l'ensemble des images capturées durant un intervalle de temps pour reconstituer la vidéo en question.

La fusion d'image dans les réseaux de capteurs multimédias peut être classée selon le niveau d'abstraction de données sur lequel la fusion est effectuée (niveau Pixel, Niveau Feature (caractéristique) et niveau symbolique ou décisionnel). La troisième section du chapitre précédent donne une explication de chaque niveau d'abstraction.

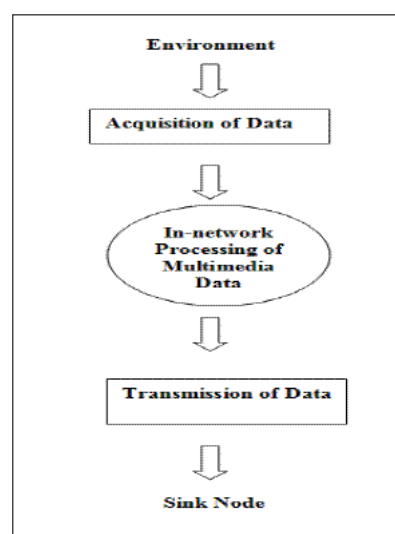


Figure 3.2 – Processus d'agrégation de données.

3.3.1 Méthodes de fusion niveau décisionnel (Decision Level Based Fusion)

1. Travail de Soukaina Messaoudi et al

Dans [59], un système de traitement de l'information qui offre une fusion de données dans un milieu hétérogène a été proposé. Dans un environnement intelligent, plusieurs types de capteurs sont déployés, à savoir des capteurs de température, des détecteurs de mouvement et des capteurs visuels tels que les caméras de sécurité. Pour fusionner les données des différents capteurs, les auteurs ont opté pour l'approche Bayésienne. L'auteur a motivé le choix d'une telle approche par la pertinence des entrées et des sorties utilisées dans ce modèle ainsi que son utilisation dans la littérature pour des problèmes similaires. Le principe de base de la théorie Bayésienne est que toutes les inconnues sont traitées comme des variables aléatoires et que la connaissance de ces quantités peut être représentée par une distribution de probabilité. En outre, la méthodologie Bayésienne réclame que la probabilité d'un certain événement représente le degré de croyance qu'un tel événement se produira. Le degré de croyance est associé à une mesure de probabilité qui peut être mise à jour par des données supplémentaires observées. Toutes les nouvelles observations sont ajoutées pour mettre à jour la probabilité antérieure et donc pour obtenir une distribution de probabilité postérieure. Le moteur de fusion proposé est utilisé pour fusionner les données visuelles et non visuelles et fournit un ensemble de décisions appropriées (outputs).

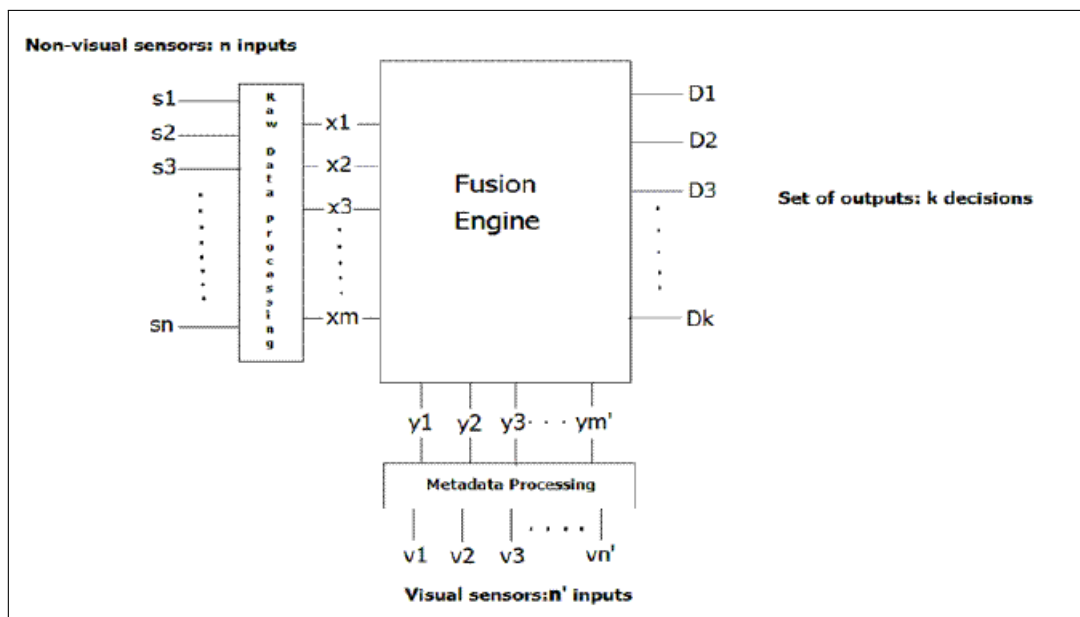


Figure 3.3 – Moteur de fusion proposé.

Comme il est illustré sur la figure 3.3), S_1, S_2, \dots, S_n sont les entrées des capteurs non visuels. Ces entrées passent d'abord par un modèle de corrélation (Raw Data Processing sur la figure), qui détermine les corrélations parmi les entrées des capteurs et transmet les m sorties au moteur de fusion. Ces sorties (entrées de moteur de fusion), sont marquées : $x_1, x_2, x_3, \dots, x_m$. Pour les données provenant des capteurs visuels (caméras optiques et infrarouges ($v_1, v_2, \dots, v_{n'}$), un prétraitement (extraction d'objets mobiles et leurs limites (Metadata processing sur la figure)), est nécessaire avant le passage au moteur de fusion. Les données visuelles extraites (méta-données) sont communiquées directement au moteur de fusion afin d'être fusionnées avec les autres types de données non visuelles; les données visuelles traitées sont notées : $y_1, y_2, y_3, \dots, y_{m'}$. Les sorties du moteur de fusion représentent un ensemble de décisions dénotées par D_1, D_2, \dots, D_k .

Discussion :

L'auteur a présenté une méthode de fusion de données hétérogènes basée sur une approche Bayésienne appliquée dans un environnement intelligent. L'auteur a développé une plateforme de simulation en JAVA afin de tester l'efficacité de son approche. Il a simulé l'environnement intelligent par des capteurs de températures, des détecteurs d'incendies, et des caméras de sécurité. L'outil de simulation offre aussi la possibilité de simuler les cas d'urgence comme les inondations et les incendies. Dans les dix expériences effectuées, il a calculé la valeur de la probabilité qui va être utilisée par le moteur de fusion pour des valeurs de $m = 10; 8; 6$ et 4 . Il a jugé les résultats raisonnables surtout pour le cas $m = 8$ et 6 pouvant renseigner sur un mouvement accru dans une zone déterminée. La méthode Bayésienne pour la fusion de données repose sur une base mathématique solide éprouvée depuis des années, mais des fois il est impossible de représenter certaines connaissances par des probabilités d'autant plus que cette méthode exige l'estimation des valeurs de probabilité a priori. L'approche proposée a été testée dans un simulateur développé par les auteurs eux même et non pas par des simulateurs homologués dans ce domaine tel que le TinyOS ou d'autres simulateurs, en plus l'auteur n'a pas présenté le degré de précision de son approche car les méthodes Bayésiennes sont basées sur le calcul et l'estimation probabiliste.

2. Travail de Damien Brulin

Dans sa thèse de doctorat [60], l'auteur a présenté une architecture de fusion de données composée de quatre modules où les données proviennent de différentes sources pour un habitat intelligent (smarthome). Cette thèse s'inscrit dans le cadre du projet CAPTHOM qui est un projet qui vise à la détection de la présence des individus. Plusieurs types de capteurs ont été utilisés pour la détection de la présence humaine, sa localisation et sa posture. Dans la partie expérimentation, l'auteur a opté pour trois types de capteurs : détecteurs infrarouges passifs, les thermopiles et les caméras. Il est à signaler qu'aucun capteur n'est porté par la

personne. L'architecture proposée est composée de quatre modules de fusion permettant respectivement de détecter la présence humaine, de localiser en 3D la personne, de déterminer la posture et d'aider à la prise de décision finale selon l'application visée. Le module de détection de présence fusionne les informations des trois capteurs : les détecteurs IRP pour la détection du mouvement, les thermopiles pour la présence en cas d'immobilité de la personne et la caméra pour identifier l'entité détectée. La localisation 3D de la personne est réalisée grâce à l'estimation de la position sur horizon glissant. Cette méthode, nommée *Visual Receding Horizon Estimation* (VRHE), formule le problème d'estimation de position en un problème d'optimisation non linéaire sous contraintes dans le plan image. Le module de fusion pour la détermination de posture s'appuie sur la théorie des ensembles flous. Il assure la détermination de la posture indépendamment de la personne et de sa distance vis à vis de la caméra. Enfin, un module d'aide à la décision fusionne les sorties des différents modules et permet de déclencher des alarmes dans le cas de la surveillance de personnes âgées ou de déclencher des applications domotiques (chauffage, éclairage) pour la gestion énergétique de bâtiments.

La théorie des ensembles flous est un très bon outil pour représenter explicitement des informations imprécises sous la forme de fonctions d'appartenance. C'est à dire une mesure M_i^j qui prend la forme $M_i^j = \mu_i^j$ où μ_i^j désigne par exemple le degré d'appartenance d'un élément à la classe C_i selon la source S_j [61].

Discussion :

L'approche basée sur la théorie des ensembles flous est l'approche adaptée pour représenter explicitement des informations imprécises. Ces fonctions ne souffrent pas des contraintes axiomatiques imposées aux probabilités et offrent donc une plus grande souplesse lors de la modélisation. L'inconvénient majeur de la théorie des ensembles flous est qu'elle représente essentiellement le caractère imprécis des informations, l'incertitude étant représentée de manière implicite et n'étant accessible que par déduction à partir des différentes fonctions d'appartenance.

-Tableau Comparatif des méthodes de fusion niveau décisionnel

	Damien Brulin	S.Messaoudi
Niveau de fusion	Décisionnel(haut)	Décisionnel(haut)
Environnement de fusion	Hétérogène	Hétérogène
Technique utilisée	Théorie des ensembles flous pour le module de détermination de la posture	Règles de Bayes
Scalabilité	Testée sur 3 type de capteurs	Simulé pour 4,6,8 et 10 capteurs
Complexité	Complicé vu la fusion de données sur 4 modules (détection, localisation, détermination posture et décision)	Complicé à cause du module de corrélation pour les données non visuelles et le module de prétraitement pour les données visuelles et en fin les calcul du moteur de fusion lui même
Avantages majeurs	Visé la présence, la localisation et la posture des personnes dans un espace	Basée sur des modèles mathématiques solides
Inconvénients majeurs	La poursuite n'est pas prise en charge par les algorithmes de fusion	Connaissances à priori

Tableau 3.5 – Tableau Comparatif des méthodes de fusion niveau décisionnel.

Commentaires :

Les deux méthodes de fusion de données niveau décisionnel que nous avons présentées sont basées sur des outils mathématiques solides à savoir la théorie des ensembles flous et la théorie de Bayes. Le choix entre tel et tel outil dépend directement du type d'application que nous allons modéliser. Si le problème peut être modélisé par des probabilités et qu'on arrive à les estimer à priori alors il est recommandé d'utiliser la méthode Bayésienne. Dans le cas contraire, la théorie des ensembles flous est mieux appropriée.

3.3.2 Méthodes de fusion niveau caractéristique (Featur Level Based Fusion)**1. Travail de Snidaro, Foresti et Varshney, 2004**

Le travail présenté dans [62] représente un système de vidéo surveillance intégrant des capteurs (caméras) optiques et des caméras Infra rouges (IR). Les signaux vidéo de chaque capteur sont traités afin d'extraire les régions mobiles (appelées "blobs"), de l'image. Dans un premier niveau, des nœuds de traitement spécialisées (PN) dépistent chaque blob détecté sur la surface de l'image et transforme sa position 2D (dans le système de coordonnées du capteur), en une position 3D

(dans le système de coordonnées de la carte de l'environnement surveillé) ; ceci représente la position de l'objet.

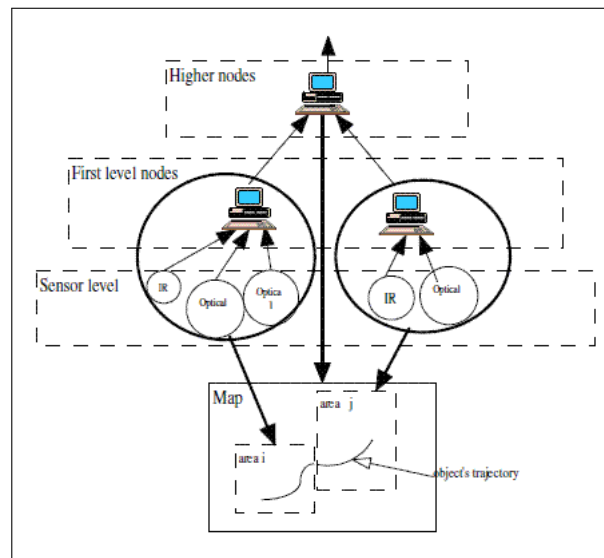


Figure 3.4 – Architecture distribuée du système.

	<i>Moyenne</i>	<i>Ecart-type</i>
<i>Premier capteur couleur</i>	3.46	1.12
<i>Deuxième capteur couleur</i>	2.92	1.38
<i>Fusion de données</i>	2.12	1.02

Tableau 3.6 – Moyenne et écart-type en pixels de la distance entre les positions estimées et les positions tirées de la carte.

L'architecture du système proposé est composée d'un ensemble de capteurs statiques et des PN pour chaque région d'intérêt de l'environnement surveillé. Les capteurs surveillant la même zone sont connectés au même PN qui est responsable de la poursuite des objets dans cette zone. En particulier les PN sont responsables de l'exécution des algorithmes d'identification des objets en mouvement.

Dans le système proposé, le mécanisme de fusion est basé sur le *filtre de Kalman* expliqué dans l'une des sections précédentes. Le but est d'améliorer l'estimation de la position des objets. Deux plans de fusion ont été proposés : fusion dite de mesure et fusion dite piste à piste. L'approche proposée a été testée avec des séquences vidéos réelles. Deux caméras couleur ont été utilisées pour suivre les mouvements de trois personnes dans une cour, le but était d'évaluer la précision des trajectoires. Ainsi les trajectoires calculées par les capteurs séparément ont été comparées aux trajectoires calculées par l'approche de fusion. Les résultats

sont donnés en Tableau 3.6 consistant en la moyenne et l'écart type de la distance en pixels entre la position mesurée et la vraie position tirée de la carte).

Discussion :

Il est clair que les résultats obtenus en appliquant une fusion par filtre de Kalman sont plus précis que les résultats obtenus individuellement par chaque capteur. Mais il faut s'assurer d'abord de la fiabilité des canaux de transmission lors de l'utilisation du *filtre de Kalman* car la perte de données est à tenir en compte. Une autre contrainte lors de l'utilisation du filtre de Kalman est l'assurance d'une synchronisation des horloges des nœuds capteurs.

2. Travail de Snidaro, Visentini et Foresti, 2012

Dans [63], les auteurs ont présenté une approche de fusion de données multimédias basée sur le principe d'extraction des caractéristiques (features), à partir des données captées par plusieurs capteurs similaires ou hétérogènes déployés dans une zone d'intérêt pour assurer une vidéo surveillance. Ce travail représente une amélioration du travail présenté précédemment. D'après les auteurs, dans un réseau de capteurs multimédias, il est impossible d'effectuer une fusion dite "fusion niveau image" car chaque nœud capteur peut avoir un champ de vision différent que ces nœuds voisins (pas de chevauchement entre les champs de vision). Dans ce cas il est nécessaire de faire une extraction de certaines caractéristiques à partir de la vue donnée par chaque capteur et faire par la suite une fusion pour obtenir une vision commune. Une probabilité projetée (projected likelihoods) a été utilisée pour fusionner les différents champs de vision des capteurs. Les auteurs ont montré comment un système de surveillance multi-caméra peut exploiter la fusion de données pour obtenir des estimations plus précises sur la position d'une cible. L'architecture de cette approche est montrée en figure 3.5, où un ensemble de classificateurs a été employé pour chaque cible dans chaque image.

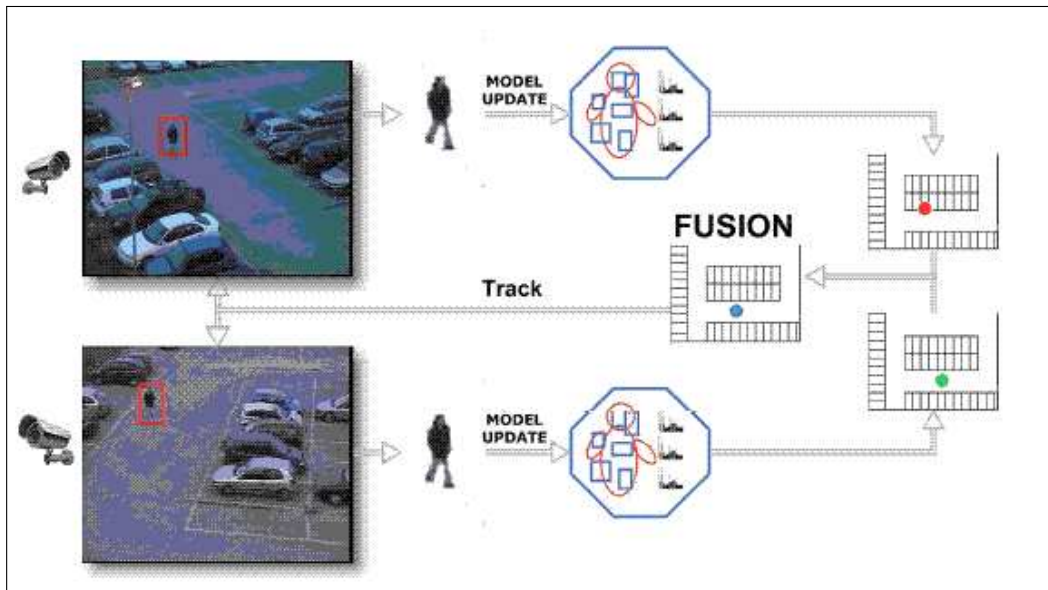


Figure 3.5 – Architecture de fusion proposée.

Pour la détection et la poursuite, chaque classificateur dans l'ensemble est établi à partir des valeurs fournies par une caractéristique (feature) extraite à partir de l'image. A chaque capture (frame), l'ensemble fournit la position la plus probable sur l'image de la cible pour laquelle il a été employé. Cette position avec sa probabilité (likelihood) est ensuite projetée à la vue de dessus de la carte de la zone à travers une transformation homographique. Les auteurs ont décrit comment la projection de la probabilité peut être effectuée de deux manières : par une approximation gaussienne ou par une projection pleine de probabilités (full likelihood projection). Les simulations ont montré un gain apporté par cette approche de fusion en termes de précision de la poursuite. Deux scénarios ont été présentés, le premier considère deux caméras ciblant un objet (une personne) en déplacement dans un parking. Le tableau 3.7 représente les résultats de l'expérience où une comparaison entre les données (l'erreur moyenne et l'écart type), est fournie par chaque capteur (caméra) séparément et les données obtenues après fusion des deux capteurs. Le deuxième scénario considère deux personnes rentrant dans une

—	Erreur moyenne(m)	Ecart type(m)
Caméra 1	0.92	0.38
Caméra 2	1.43	0.44
Fusion Gausienne	0.81	0.40
Fusion Likelihood	0.79	0.44

Tableau 3.7 – Erreur moyenne et écart type des capteurs séparés et l'approche de fusion.

scène à partir de deux directions opposées. En plus des deux cameras précédentes caméra1 et caméra2, deux autres caméras ont été employées (caméra3 et caméra4), la caméra numéro 2 est exclue car son champ de vision est loin des deux personnes. Les résultats obtenus sont présentés dans le tableau 3.8.

—	Erreur moyenne(m)	Ecart type(m)
Caméra 1	1.45	0.19
Caméra 3	0.61	0.17
Caméra 4	0.89	0.15
Fusion Gaussienne	0.53	0.17
Fusion Likelihood	0.50	0.18

Tableau 3.8 – Erreur moyenne et écart type des trios capteurs dans l’approche de fusion.

Discussion :

L’auteur a présenté une approche de fusion de données appliquée à un système de vidéo surveillance, il a utilisé une approche de fusion dite fusion niveau caractéristiques (feature level fusion). A travers les résultats des tests effectués, l’auteur a montré l’avantage de la fusion de plusieurs capteurs par rapport à une considération individuelle de chaque capteur. Néanmoins, l’auteur n’a pas évalué le comportement de son approche lors du passage à l’échelle (scalability). D’un autre coté il serait préférable d’abord d’appliquer une étape de fusion au niveau des capteurs (caméras) avant d’appliquer la fusion proprement dite (likelihood or gaussian fusion), afin de ne pas encombrer le réseau.

3. Le Système d’Aide à la Maîtrise de la Sécurité Individuelle dans les Transports publics (SAMSIT)

Dans [64], l’auteur a présenté une méthode de fusion de données basée sur les matrices de combinaison. Cette approche a été proposée dans le cadre du projet SAMSIT, où un algorithme de fusion a été implémenté afin de fusionner les résultats de détection et de suivi de deux caméras. La fusion des données des deux caméras, quand celles-ci sont synchronisées, a lieu à l’aide d’une matrice de combinaison. La matrice de combinaison est une matrice $n \times p$ contenant les n objets mobiles détectés par la première caméra suivant les lignes et les p objets mobiles détectés par la seconde caméra suivant les colonnes. Pour chaque objet mobile d’une caméra, la matrice de combinaison exprime le taux de correspondance avec tous les objets mobiles détectés par l’autre caméra. Le taux de correspondance est calculé sur la base de la position 3D des objets mobiles [64]. Dans un premier temps, les valeurs de correspondance sont comparées à un seuil prédéfini t . En suite, une recherche est effectuée sur les objets mobiles voisins inter-caméras de forte correspondance ($>t$) pour construire des sous-matrices de taille $m \times k$. Pour chaque sous-matrice, deux cas se présentent :

- Le cas non ambigu (1×1) : si les objets mobiles ont la même classe (personne, groupe de personnes, groupe) alors les deux mobiles sont fusionnés en opérant une moyenne de leurs positions et dimensions 3D. Dans le cas contraire, on sélectionne l'objet mobile étant plus proche de la caméra respective.
- Le cas ambigu : le jeu de règles suivant est utilisé pour associer ou éliminer les objets mobiles d'une des deux caméras :
 - (a) 2×2 : la diagonale (/ ou \) de la sous-matrice est sélectionnée telle que la somme des correspondances est la plus grande : Les deux paires d'objets mobiles obtenus par mises en correspondance, sont fusionnées de la même manière que dans le cas 1×1 .
 - (b) $m \times m$: La liste des objets mobiles contenant le plus grand nombre de mobiles est gardée.

Discussion :

L'approche présentée dans cette section vise la fusion des résultats de détection et de suivi de deux caméras. Grâce à cette approche, à chaque itération il est possible soit de privilégier la meilleure caméra observant la scène soit de fusionner les données provenant des deux caméras. L'approche emploie la fusion de deux caméras uniquement, car elle modélise l'ensemble des objets détectés par une caméra sur les lignes de la matrice et l'ensemble des objets détectés par la deuxième caméra sur les colonnes de la matrice.

Tableau Comparatif des méthodes de fusion niveau caractéristique

Commentaires :

Parmi ces trois méthodes de fusion de données multimédias dite 'Feature Level Fusion', la méthode SAMSIT est à écarter car elle est basée sur des combinaisons matricielles compliquées et en plus elle est limitée à la fusion de données de deux capteurs au maximum. Par contre le travail de Snidaro représente une amélioration du travail effectué en 2004 et qui consiste à l'estimation de la position et le trajet des objets dans un réseau de capteurs multimédias.

3.3.3 Méthodes de fusion niveau Pixel (Pixel Level Based Fusion)

1. **Multi-Focus Image Fusion basée sur la DWT (Discrete Wavelet Transform) :**

	Snidaro 2004	Snidaro 2012	SAMSIT
Niveau de fusion	Moyen(features)	Moyen (features)	Moyen(features)
Environnement de fusion	Homogène	Hétérogène	Homogène
Technique utilisée	Filtre de kalman	Projected likelihood	Matrice de combinaison
Scalabilité	Testé avec 2caméras pour suivre 3 personnes	Testé sur 2 caméras pour suivre une personnes et sur 3 caméras pour suivre 2 personnes	Considère 2 caméras uniquement
Complexité	Complicé à cause du transfert de position 2D en position 3D	Calcul compliqué car il est basé sur le traitement des séquences vidéos	compliqué (calcul matriciel)
Avantages majeurs	Les capteurs surveillant la même zone sont connectés au même nœud de traitement	Les résultats de fusion ont été meilleurs que les mesures individuelles des capteurs	Offre la possibilité de la détection et le suivi des objets en mouvement
Inconvénients majeurs	L'auteur n'a pas inclus le bruit	La vraisemblance projetée n'a pas été utilisée dans le domaine de fusion de données	Dans une matrice on peut représenter 2 caméras seulement

Tableau 3.9 – Tableau Comparatif des méthodes de fusion niveau caractéristique.

La DWT ou la transformée en ondelette est un outil mathématique qui peut effectuer une décomposition multi-échelle (Multi-scale décomposition) d'un signal numérique. Dans notre cas le signal numérique représente une image en 2D. Le principe de base de la DWT est qu'elle décompose chaque image source en lignes et en colonnes par des filtres de type passe-haut(H) (**high-pass** filtering), et des filtres de type passe-bas(L) (**low-pass** filtering). Lorsque on applique un filtre de type **high-pass** nous atténuons les signaux de faible fréquence et lorsque on applique un filtre **low-pass** c'est les hautes fréquences qui seront affaiblies. Le résultat de cette opération est d'obtenir le **coefficient d'approximation** (LL) appelé aussi la bande de fréquence Low-Low ainsi que les **coefficients détails** (Low-High,High-Low,High-High) cela signifie que l'image source est décomposée en quatre (04) sous-images dont la taille de chaque sous-image est égale à la moitié de la taille de l'image originale. Supposons que nous avons deux images sources à fusionner A et B , nous allons obtenir HH_a, HL_a, LH_a, LL_a pour l'image A et HH_b, HL_b, LH_b, LL_b pour l'image B . LH veut dire qu'un filtre Low-pass est appliqué selon l'axe X suivie par le filtre High-pass selon l'axe Y . Cette bande de fréquence contient les **coefficients détails horizontaux** de l'image, par contre HL contient les **coefficients détails verticaux** et HH contient les **coefficients détails diagonaux**. Il est à signaler que la transformée en ondelettes peut être appliquée en plusieurs niveaux (la figure 3.6 montre 2 niveaux de décompositions). Le prochain niveau de décomposition est appliqué uniquement sur la sous-image LL , le résultat est toujours quatre sous-images dont la taille est égale à la moitié

de la taille de l'image LL [65].

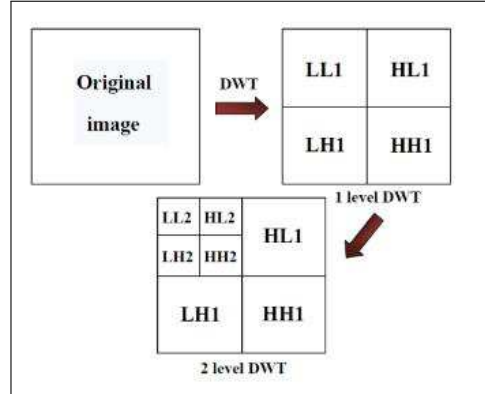


Figure 3.6 – La DWT avec 2-niveaux de décomposition.

D'après [66], la transformée en ondelettes d'une fonction $f = f(x, y)$ d'une taille $M \times N$ est donnée par les formules 3.13 et 3.14. Pour un signal de deux dimensions, une fonction de mise en échelle $\varphi(x, y)$ et trois ondelettes 2-D : $\psi^H(x, y)$, $\psi^V(x, y)$ et $\psi^D(x, y)$ sont nécessaires.

$$W_{\varphi}(j_0, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f \varphi_{j_0, m, n} \quad (3.13)$$

$$W_{\psi}^i(j, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f \psi_{j, m, n}^i \quad (3.14)$$

Avec : $j, m, n, M, N \in \mathbb{Z}$ (ensemble des entiers), $i = H, V, D$, j_0 une échelle arbitraire, les coefficients w_{φ} définissent l'approximation de la fonction f à l'échelle j_0 .

$$\varphi_{j, m, n}(x, y) = 2^{j/2} \varphi(2^j x - m, 2^j y - n) \quad (3.15)$$

$$\psi_{j, m, n}^i(x, y) = 2^{j/2} \psi^i(2^j x - m, 2^j y - n) \quad (3.16)$$

La transformée en ondelettes est devenue un outil très en vogue dans le domaine de traitement des images. Cette technique est utile pour la compression des images, le bureau fédéral américain utilise cette technique pour compresser l'image scannée des empreintes digitales. Dans la norme JPEG2000, la DWT est utilisée pour la compression au lieu de la DCT utilisée dans la compression JPEG. La DWT peut être utilisée aussi pour réduire le bruit dans un signal à l'aide des filtres DWT [67].

Dans ce qui suit, nous donnons un aperçu de l'ensemble des travaux ayant utilisé la DTW dans la fusion de données multimédia. Nous mettons l'accent sur deux travaux de recherche récents dans ce domaine.

- (a) Dans [68], l'auteur a présenté une technique de fusion basée sur la DWT appelée MultiWavelet qui est une extension de la DWT. Cette technique a été appliquée au niveau pixel pour fusionner des images obtenues par plusieurs capteurs. Une pyramide est obtenue pour chaque image source à partir d'une décomposition dite MultiWavelet. Les coefficients MultiWavelet obtenus sont ensuite combinés par une règle de fusion afin de construire l'image fusionnée.
- (b) Dans [69], l'auteur a proposé deux algorithmes basés sur la DWT afin de fusionner des images sources. Dans le premier algorithme appelé **Maximum Pixel replacement**, les quatre sous-bandes de fréquences (**LL,HL,LH,HH**) de l'image fusionnée sont constituées en prenant la valeur maximale de chaque pixel des quatre bandes des images sources. Le deuxième algorithme appelé **Pixels Averaging** prend la valeur moyenne des pixels de chaque sous-bande. A la fin, l'auteur a donné une comparaison des deux algorithmes selon plusieurs critères tels que le PSNR (Peak Signal to Noise Ratio), RMSE (Root Mean Square Error) ou il a montré que l'algorithme Maximum Pixel replacement donne des bonnes performances que l'algorithme Pixels Averaging.
- (c) **Travail de T.Sahoo et al,2011** [1]
 T.Sahoo et al ont proposé une nouvelle technique de fusion de données niveau pixel basée sur la transformée en ondelette DWT et le calcul de la variance dans un domaine spatial. La variance nous donne l'information sur la netteté d'un bloc d'image, à travers une comparaison des blocs des images sources. Seuls ceux qui possèdent une valeur de "netteté" la plus élevée sont utilisés pour constituer les blocs de l'image fusionnée. La variance est calculée avec la formule suivante :

$$VAR = \frac{1}{MN} \sum_i \sum_j (f(i,j) - \bar{f})^2 \quad (3.17)$$

Avec : $\bar{f} = \frac{1}{MN} \sum_i \sum_j f(i,j)$ est la valeur moyenne des niveaux de gris de la région de l'image, $f(i,j)$ est le niveau de gris du pixel (i,j) .

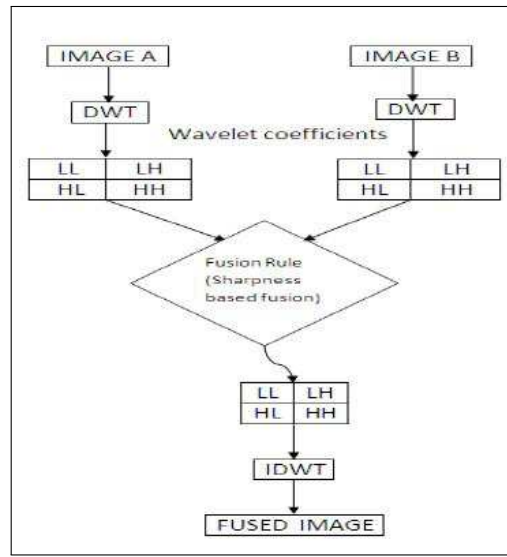


Figure 3.7 – Architecture proposée dans [1]

Pour rappel, dans la DWT nous obtenons une approximation de la partie basses fréquences et des parties en hautes fréquences. Dans ce cas, si nous avons deux images à fusionner A et B, les blocs de ces images sont dénotés A_i , B_i et la valeur de la netteté (Sharpness) des blocs A_i , B_i est notée BSV_i^A et BSV_i^B respectivement. Une comparaison des valeurs de la netteté est utilisée pour former les blocs de l'image fusionnée comme il est illustré par la formule ci-après :

$$F_i = \begin{cases} A_i & \text{si } BSV_i^A > BSV_i^B, \\ B_i & \text{si } BSV_i^A < BSV_i^B, \\ (A_i + B_i)/2 & \text{sinon} \end{cases}$$

Résultats expérimentaux : Dans ce travail, l'auteur a testé sa méthode proposée sur un ensemble d'images de taille 512×512 , sur les images fusionnées et il a calculé trois métriques d'évaluation des performances comme il est illustré sur le tableau 3.10. L'approche proposée dite "hybride" (combinaison entre la variance et la DWT), a été comparée avec les méthodes basées sur la variance dans un domaine spatial et les méthodes basées sur la DWT.

Method	PSNR	Correlation Coefficient	Deviation Index
Wavelete (DWT)	13.1262	0.9645	1.2564
Variance based spatial domain	32.7139	0.9926	0.9802
Proposed Hybrid	47.1393	0.9932	0.0475

Tableau 3.10 – Paramètres de performances et comparaison des méthodes de fusion.

Discussion : Cette technique hybride qui combine la DWT et la variance

dans un domaine spacial permet de préserver l'information spectrale des images sources. Cela est indiqué à travers la valeur faible de l'index de déviation qui est pratiquement insignifiant par rapport aux deux autres méthodes.

(d) **Travail de S.V.More et al, 2012** [70]

Dans ce travail, l'auteur a présenté une méthode de fusion d'images basée sur le principe de l'ondelette discrète DWT. Les images sources doivent être d'abord décomposées en bandes de haute et de basse fréquences. Ensuite, les composantes de basses fréquences sont combinées par une règle dite règle d'énergie maximale et les composantes hautes fréquences sont combinées par une règle de variance. Enfin, l'image fusionnée est construite par transformée en ondelettes inverse IDWT. Si nous avons deux images sources $I_1(x, y)$ et $I_2(x, y)$, la DWT procède à la décomposition de chaque image source en bandes basses et hautes fréquences. Les coefficients des deux bandes sont ensuite combinés à l'aide d'une règle de fusion Φ . L'image fusionnée est obtenue par la DWT inverse IDWT selon la formule 3.19 :

$$I_f(x, y) = IDWT [\Phi \{DWT(I_1(x, y), DWT(I_2(x, y)))\}] \quad (3.18)$$

Notons que dans les images sources, la plupart de l'information est concentrée dans la bande de basses fréquences. Lorsqu'une image contient plusieurs informations dans certaines directions (bandes de fréquences), la DWT de ces bandes génère une grande énergie. Le pixel ayant une grande énergie possède une forte luminosité qu'un pixel ayant une faible énergie. Partant de ce principe, l'auteur a utilisé une technique de sélection des coefficients de basses fréquences basée sur la quantité d'énergie émise, donnée par la formule suivante 3.19.

$$E = \frac{\sum_{i=1}^M \sum_{j=1}^N [f(i, j)]^2}{M \times N} \quad (3.19)$$

Avec : $f(i, j)$ est la valeur du niveau de gris du point(i,j), $M \times N$ est la taille de l'image.

La technique de fusion utilisée pour les basses fréquences est illustrée par la formule ci-dessous.

$$f_L(i, j) = \begin{cases} A_L(i, j) & \text{si } E_A \geq E_B, \\ B_L(i, j) & \text{sinon} \end{cases}$$

$f_L(i, j), A_L(i, j), B_L(i, j)$ représentent respectivement les coefficients basses fréquences de l'image fusionnée, image A et l'image B au point (i, j) . E_a, E_b

représentent l'énergie des coefficients basses fréquences des image A et B au point (i, j)

Sélection des bandes de haute fréquence Dans une image numérique, les bandes de hautes fréquences contiennent le détail de l'image. Pour la sélection de ces coefficients, l'auteur a calculé la variance. La sélection est faite selon la formule ci-dessous.

$$f_H(i, j) = \begin{cases} A_H(i, j) & \text{si } \sigma_A \geq \sigma_B, \\ B_H(i, j) & \text{sinon} \end{cases}$$

$f_H(i, j), A_H(i, j), B_H(i, j)$ représentent respectivement les coefficients hautes fréquences (HL, LH, HH) de l'image fusionnée, l'image A, l'image B au point (i, j) . σ_A et σ_B représentent les variances des coefficients hautes fréquences des images A et B au pixel (i, j) .

Résultats expérimentaux : Dans ce travail, l'auteur a présenté une comparaison de la méthode proposée avec d'autres méthodes telles que le pixel Averaging. Les critères de comparaison ciblés par l'auteur sont : l'Entropy, Standard deviation (Ecart type) et le RMSE.

Discussion : Malgré que la technique proposée donne de bons résultats sur l'image fusionnée (bonne qualité d'image fusionnée), l'auteur n'a pas estimé la complexité de l'approche.

2. Multi-Focus Image Fusion basée sur le calcul de niveau d'information dans des régions des images :

Dans ce type de fusion, les images sources sont décomposées en régions, ensuite un processus de calcul de niveau d'information par région est réalisé. L'image fusionnée est constituée par une règle de fusion zone par zone en utilisant le niveau d'information calculé pour chaque zone dans les images sources.

Dans [71], R.Maruthi et al ont proposé une méthode de fusion basée sur le calcul du niveau d'information appelé *niveau d'activité* (Activity leve(AL)).

Dans cette méthode, les images sources sont subdivisées en régions, l'information calculée dans chaque région est la fréquence spatiale (spatial frequency) et la visibilité(Visibility). Ces deux critères sont par la suite combinés pour obtenir le niveau d'information de l'image fusionnée. La fréquence spatiale est calculée par la formule suivante :

$$SF = \sqrt{(RF^2) + (CF^2)} \quad (3.20)$$

RF, CF représentent les fréquences lignes et les fréquences colonnes respective-

ment.

$$RF = \sqrt{\frac{1}{MN} \sum_{m=1}^m \sum_{n=2}^n [F(m, n) - F(m, n - 1)]^2} \quad (3.21)$$

$$CF = \sqrt{\frac{1}{MN} \sum_{n=1}^n \sum_{m=2}^m [F(m, n) - F(m - 1, n)]^2} \quad (3.22)$$

Les auteurs ont démontré à travers des mesures sur des images que la fréquence spatiale est inversement proportionnelle au niveau du brouillage de l'image. Autrement dit, si la fréquence spatiale d'une image est élevée cela indique une bonne qualité et une valeur de contraste élevée. Pour le deuxième critère qui est la visibilité, il est calculé comme suit :

$$VI = \sum_{M=1}^M \sum_{N=1}^N \omega(\mu) \frac{f(m, n - \mu)}{\mu} \quad (3.23)$$

Avec : μ la valeur moyenne de l'intensité du bloc, $\omega(\mu) = (1/\mu)$.

Il a été montré aussi que la visibilité est inversement proportionnelle au niveau du brouillage de l'image. Donc si nous avons deux images sources A et B subdivisées en i régions, et soit F l'image fusionnée des deux images A et B. ILA_i et ILB_i sont les deux niveaux d'information ou d'activité des régions A_i et B_i respectivement. Si $ILA_i > ILB_i$ alors $F_i = A_i$, et si $ILA_i < ILB_i$ alors $F_i = B_i$. Dans ce cas le niveau d'information représente **la fréquence spatiale** (SF) et **la visibilité**(VI). Donc les régions de l'image fusionnée sont obtenues par la comparaison des niveaux d'information (fréquence spatiale ou visibilité) des régions des images sources. Les auteurs ont proposé une approche de combinaison entre ces deux critères. Dans cette combinaison, les images sources sont d'abord fusionnées en se basant sur la fréquence spatiale et par la suite en utilisant la visibilité de manière indépendante. Les deux images obtenues en appliquant les deux critères de fusion sont à nouveau fusionnées en utilisant l'un des deux critères. Autrement dit on aura les deux séquences suivantes : SFVVISF ou SFVIVI. Le diagramme de cette approche combinée est illustré sur la figure ci-dessous.

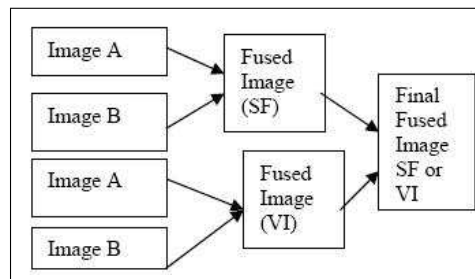


Figure 3.8 – Approche combinant SFVIVI et SFVISF.

Dans la partie expérimentation, les auteurs ont fusionné plusieurs images en utilisant la technique SF et VI. Les images résultantes ont été par la suite fusionnées en utilisant l'une des techniques de combinaison SFVIVI et SFVISF.

Pour évaluer l'approche, les auteurs ont calculé la fréquence spatiale et la visibilité des images fusionnées. Les résultats sont reportés sur les deux tableaux donnés ci-après :

S .No	SF	VI	SFVIVI	SFVISF
Book.jpg	1.5E+12	1.46E+12	1.39E+12	1.39E+12
Clock.jpg	1.21E+11	2.37E+11	1.39E+11	1.39E+11
Bottle.jpg	3.09E+12	4.46E+12	3.43E+12	3.43E+12
Disk.gif	3.95E+11	6.76E+11	5.16E+11	5.16E+11
Cup.gif	2.97E+11	4.11E+11	3.15E+11	3.15E+11

Tableau 3.11 – La fréquence spatiale.

S .No	SF	VI	SFVIVI	SFVISF
Book.jpg	2.88E-09	7.89E-09	2.76E-09	2.76E-09
Clock.jpg	-6.41E-08	-6.17E-08	-6.41E-08	-6.41E-08
Bottle.jpg	2.21E-08	2.78E-08	2.21E-08	2.21E-08
Disk.gif	1.03E-08	1.16E-08	1.01E-08	1.01E-08
Cup.gif	-6.35E-08	-6.06E-08	-6.35E-08	-6.35E-08

Tableau 3.12 – La visibilité.

Discussion :

Le tableau 3.11 montre que la fréquence spatiale pour les images fusionnées est la plus élevée dans les cas de la méthode VI pour l'ensemble des images (book, clock,...). Cependant les auteurs signalent que malgré cela, l'image obtenue n'est pas vraiment claire et reste contrastée.

Dans la méthode SF, la fréquence spatiale est faible et dans le cas des méthodes combinées SFVIVI et SFVISF la fréquence spatiale est presque la même ; l'approche combinée est la meilleur dans ce cas. Cela est confirmé à travers le tableau 3.12 qui contient les valeurs de la visibilité pour les deux méthodes SF et VI et les méthodes combinées SFVISF et SFVIVI.

Critiques :

Les auteurs ont essayé dans leur méthode de fusion de combiner des critères importants dans le domaine de traitement d'images, il s'agit de la fréquence spatiale et la visibilité. Mais malheureusement, dans la pratique une bonne visibilité et une valeur élevée de la fréquence spatiale ne signifie pas une image claire. Des fois il est préférable d'utiliser l'œil humain au lieu de calculer le niveau d'information.

De plus, l'auteur n'a pas précisé la manière de subdiviser les images source en sous-régions et a laissé le choix à l'utilisateur. La plupart des méthodes de fusion comme la DCT, la DWT indiquent la manière de subdivision des images sources en sous régions ou en blocs de $N \times N$ pixels.

Dans la partie expérimentation et pour évaluer la qualité des images fusionnées, les auteurs ont utilisé le même mécanisme que celui utilisé pour la fusion c'est-à-dire qu'il ont calculé la SF et la VI pour les images fusionnées malgré l'existence de plusieurs métriques et méthodes mathématiques pour l'évaluation des images fusionnées à savoir : la similarité structuré SSIM et d'autres méthodes citées dans les sections précédentes.

3. Multi-Focus Image Fusion basée sur la DCT (Discrete Cosine Transform) :

La DCT est une transformation importante dans le domaine de traitement d'images [72]. La DCT permet de passer d'une représentation spatiale d'un signal à une représentation fréquentielle. Les coefficients larges de la DCT sont concentrés dans la région de basses fréquences. Autrement dit, ces coefficients sont situés dans la partie haute gauche de la matrice DCT.

L'avantage essentiel de cette technique est qu'elle permet de compacter l'énergie d'un signal dans certaines composantes tant dis que les autres composantes sont négligeables ou nulles. Cela facilite la séparation d'une image en plusieurs parties (sous-bandes avec la conservation de la qualité visuelle de l'image).

Selon [5], la DCT en deux dimensions d'un bloc de taille $N \times N$ d'une image $x(m, n)$ est donnée par la formule suivante :

$$d(k, l) = \frac{2\alpha(k)\alpha(l)}{N} \times \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x(m, n) \times \cos \frac{(2m+1)\pi k}{2N} \times \cos \frac{(2n+1)\pi l}{2N}, \text{ Avec :} \quad (3.24)$$

$k, l = 0, 1, \dots, N-1$ et

$$\alpha(k) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{si } k = 0, \\ 1 & \text{sinon.} \end{cases}$$

Le coefficient $d(0,0)$ est appelé le DC(Direct Component), les autres coefficients $d(1,k)$ sont appelés les AC(Alternative Component) voir figure 3.9.

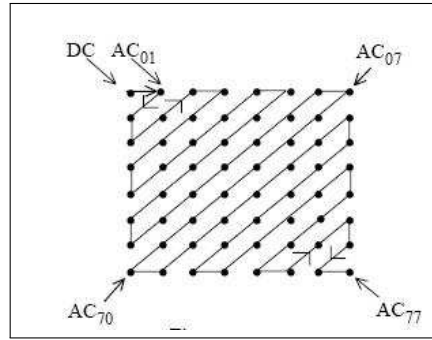


Figure 3.9 – Coefficients AC et DC d’un bloc de 8 × 8 pixels.

Sur la figure 3.10, on remarque une tendance générale des valeurs de la matrice DCT à s’approcher de 0 lorsqu’on s’éloigne du coin supérieur gauche, c’est-à-dire lorsqu’on monte dans les plus hautes fréquences. Cela traduit le fait que l’information effective de l’image est concentrée dans les basses fréquences. C’est le cas de la majorité des images.

Matrice de pixels d’entrée							
140	144	147	140	140	155	179	175
144	152	140	147	140	148	167	179
152	155	136	167	168	162	152	172
168	145	156	160	152	155	136	160
168	148	156	148	140	136	147	168
147	167	140	165	165	140	136	162
136	156	123	167	162	144	140	147
148	155	136	155	152	147	147	186
Matrice DCT							
1210	-18	15	-9	23	-9	-14	-19
21	-34	26	-9	-11	11	14	7
-10	-24	-2	6	-18	3	-20	-1
-8	-5	14	-15	-8	-3	-3	8
-3	10	8	1	-11	18	18	15
4	-2	18	8	8	-4	1	7
9	1	-3	4	-1	-7	-1	-2
0	-8	-2	2	1	4	-6	0

Figure 3.10 – Exemple d’une matrice d’entrée et sa matrice DCT.

- Travaux de recherches sur la fusion Multi-Focus basée sur la DCT

(a) **Travail de J.Tang, 2004[6]** :

Dans ce papier, les auteurs ont présenté deux approches de fusion d’image appelées *DCT+average* et *DCT+Contrast*. Afin de calculer les coefficients DCT de chaque bloc, les auteurs ont utilisé la formule 3.24 avec N=8 ; ce qui a donné la formule suivante :

$$d_{k,l} = \frac{c(k)c(l)}{4} \sum_{i=0}^7 \sum_{j=0}^7 x_{i,j} \cos \frac{(2i+1)k\pi}{16} \cos \frac{(2j+1)l\pi}{16}, \text{ Avec : } \quad (3.25)$$

$k, l = 0, 1, \dots, 7$ et

$$c(k) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{si } k = 0, \\ 1 & \text{sinon.} \end{cases}$$

L'image d'origine peut être reconstituée à partir des coefficients DCT en appliquant la DCT inverse dite IDCT qui est donnée par la formule :

$$x_{i,j} = \sum_{k=0}^7 \sum_{l=0}^7 \frac{c(k)c(l)}{4} d_{k,l} \cos \frac{(2i+1)k\pi}{16} \cos \frac{(2j+1)l\pi}{16}, \text{ Avec :} \quad (3.26)$$

$i,j=0, 1, \dots, 7$.

La DCT décompose chaque bloc d'entrée en une série de formes d'onde, chacune avec une fréquence spatiale particulière. Les auteurs ont utilisé cette propriété des coefficients DCT pour définir une mesure de contraste. Pour cette raison, les coefficients DCT sont divisés en 15 bandes de fréquences différentes. Le contraste au niveau de chaque coefficient est calculé par la formule suivante :

$$C_{i,j} = \frac{d_{i,j}}{\sum_{k=0}^{n-1} E_k}, \text{ Avec :} \quad (3.27)$$

$E_k = \frac{\sum_{t+p=k} |d_{p,t}|}{N}$ est l'amplitude moyenne à travers une bande spectrale, et

$$N = \begin{cases} k + 1 & \text{si } k < 8, \\ 14 - k + 1 & \text{si } k \geq 8. \end{cases}$$

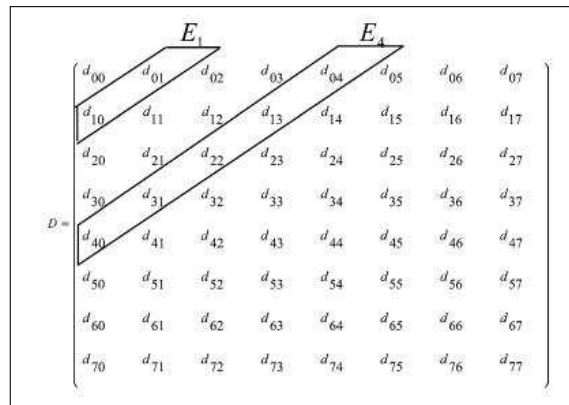


Figure 3.11 – Exemple d'un bloc DCT.

La figure 3.11, illustre la première et la quatrième bande spectrale du bloc DCT.

Supposons que nous avons une image $Y = \{y_{i,j}\}$ avec $(i = 0, \dots, N - 1 \text{ et } j = 0, \dots, M - 1)$ qui peut être divisée en blocs 8×8 pixels. Supposons aussi que $X_n = \{x_{n,k,l}\}$ ($k = 0, \dots, 7, l = 0, \dots, 7, n = 0, \dots, Q - 1$) est le $n_{i\text{eme}}$ bloc de 8×8 . Le bloc DCT correspondant à X_n est $D_n = \{D_{n,k,l}\}$.

L'ensemble : $D = D_0, D_1, D_2, \dots, D_{Q-1}$ est appelé la représentation DCT de

l'image $Y = \{y_{i,j}\}$. Dans ce cas :

$D^t = \{D_0^t, D_1^t, D_2^t, \dots, D_{Q-1}^t\}$ est la représentation DCT de la t^{ieme} image source, la représentation DCT de l'image fusionnée est obtenue par la fusion des coefficients DC et les coefficients AC respectivement.

Les coefficients AC de l'image fusionnée sont obtenus par comparaison du contraste des coefficients AC correspondants des différentes images et le coefficient AC avec le plus grand contraste est retenu en tant que coefficient AC de l'image fusionnée, mathématiquement on peut écrire :

$$D_{n,k,l} = D_{n,k,l}^T, \text{ Avec :} \quad (3.28)$$

$$T = \operatorname{argmax} \{C_{n,k,l}^t\}, \quad (3.29)$$

$k, l \neq 0$ et $C_{n,k,l}^t$ est le contraste AC du coefficient $D_{n,k,l}^t$ de la t^{ieme} image. Pour les coefficients DC nous avons :

$$D_{n,0,0} = \frac{1}{W} \sum_{t=1}^W D_{n,0,0}^t, \quad (3.30)$$

Avec : W est le nombre d'images à fusionner.

Pour effectuer une comparaison, l'auteur a proposé un autre algorithme de fusion toujours basé sur le principe de la DCT. Dans cette nouvelle approche, la moyenne de la DCT de toutes les images sources est calculée pour obtenir l'image fusionnée, cela est donné par l'équation suivante :

$$D_{n,k,l} = \frac{1}{W} \sum_{t=0}^W D_{n,k,l}^t, \quad (3.31)$$

L'auteur a appelé la technique de fusion décrite par 3.28, 3.29 et 3.30 la DCT+Contrast et celle décrite par 3.31 la DCT+average.

Expérimentation : Dans les expérimentations menées par les auteurs, il a été comparé les deux techniques proposées avec la WTF (Wavelet Transform). En premier lieu, il a été utilisé des images créées artificiellement à partir d'une image de référence par brouillage de différentes parties de l'image de référence. La technique utilisée pour évaluer les algorithmes proposés est PSNR (entre l'image fusionnée et l'image de référence), les résultats sont donnés dans la figure 3.12.

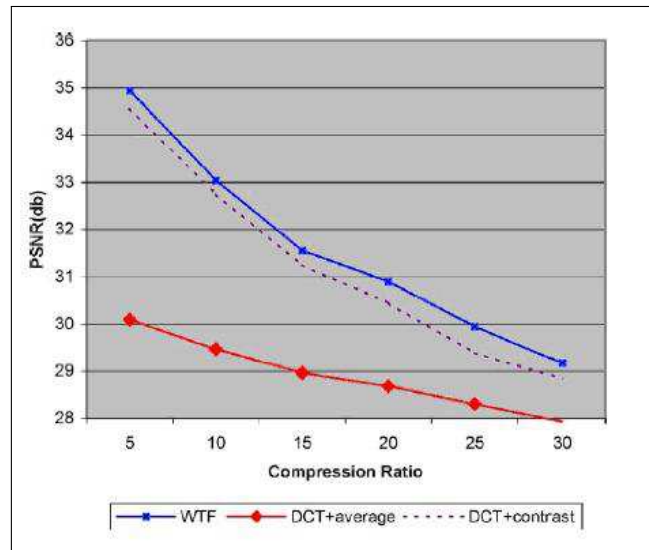


Figure 3.12 – PSNR obtenue pour WTF, DCT+average et DCT+contrast.

Cette figure montre que la technique DCT+average est la plus faible en termes de performance, contrairement à la deuxième technique proposée par l'auteur et qui dispose d'une PSNR proche de celle de la WTF. Par ailleurs, l'image fusionnée obtenue par la WTF et celle obtenue par la DCT+contraste ne montre aucune différence en termes de qualité visuelle.

Dans un second lieu, l'auteur a utilisé des images réelles capturées par des caméras avec des focus sur différents objets de la scène (une image avec un zoom sur la partie gauche de la scène et une autre image avec un zoom sur la partie droite de l'image). Les résultats obtenus montrent (visuellement) que l'image fusionnée par la techniques DCT+average souffre d'un manque de contraste contrairement aux image fusionnées par les deux autres techniques (DCT+contrast et WTF) qui sont claires et semblables.

Pour le temps d'exécution des algorithmes, la méthode proposé DCT+contrast est plus rapide que la WTF. Dans la première expérimentation (images sources produites artificiellement), la DCT+Contrast a consommé **27.5 S**, alors que la WTF **47.6 S**. Dans la seconde expérimentation (images sources réelles), la DCT+Contrast a nécessité **13.9 S** et la WTF **36.3 S**. Ces mesures sont obtenues en utilisant un Pentium4 2GHz avec une RAM de 1 GB.

Critiques : Dans la partie expérimentation, l'auteur a utilisé la PSNR pour évaluer l'image fusionnée par l'algorithme proposé. Cependant, la PSNR exige l'existence d'une image de référence, chose qui est difficile dans le cas réel. Il aurait été souhaitable que l'auteur utilise une technique d'évaluation qui n'exige pas une image de référence. En plus l'auteur n'a pas exprimé les valeurs obtenues dans le cas des images réelles, il a donné uniquement des représentations visuelles des images fusionnées. Pour les temps d'exécution

les résultats sont reportés sans donner de détail sur la manière avec laquelle ils ont été obtenus.

(b) **Travail de M. B.A Haghghat et al, 2011[5]**

Comme son nom l'indique, cette méthode de fusion s'intéresse à la fusion des images capturées avec des capteurs multimédias (CM) avec un degré de zoom (focus). Le principe de la DCT est basé sur la subdivision de chaque image source en blocs de 8×8 pixels comme dans le cas des compresseurs standards tels que JPEG, MPEG, ... En suite, un coefficient DCT est calculé pour chaque bloc et dans chaque image source en utilisant la formule 3.24. Comme dans la méthode précédente, la dimension des blocs $N = 8 \times 8$ pixels. L'auteur a défini un coefficient normalisé comme suit :

$$\tilde{d}(k, l) = \frac{d(k, l)}{N}, \quad (3.32)$$

Par la suite il calcule la moyenne μ et la variance σ^2 . Il a démontré que la variance pour un bloc de $N \times N$ pixels peut être calculé en fonction du coefficient DCT du même bloc selon la formule :

$$\sigma^2 = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} \frac{d^2(k, l)}{N^2} - \tilde{d}^2(0, 0) \quad (3.33)$$

L'auteur considère que la zone zoomée (focused area), est la plus riche en information et c'est la zone qui possède la plus grande variance, car cette dernière est considérée comme contrastée dans le processus de traitement d'image. Donc la variance peut être considérée comme étant le niveau d'information d'un bloc de 8×8 pixels.

Supposons que nous avons deux images sources A et B capturées par deux CM est que l'image résultante (fusionnée) est F. D'après l'auteur, chaque image A, B est subdivisées en blocs de 8×8 pixels. Un coefficient DCT est calculé pour chaque bloc, par la suite la variance de chaque bloc DCT est calculée. Enfin, le bloc qui contient la variance la plus élevée est sélectionné pour l'image résultante, pour cette raison la méthode est appelée DCT+variance. L'image résultante ou fusionnée subit un contrôle de consistance qui corrige certaines anomalies de sélection, supposons par exemple que quelques blocs de la zone d'intérêt ce trouvent dans la profondeur de champ de la caméra A, donc tous les blocs de la même zone dans l'image fusionnée sont sélectionnés à partir de l'image de la caméra A. Parfois et à cause de quelques erreurs dues au bruit, un bloc de la caméra B est sélectionné. Ceci donne dans l'image résultante un bloc B entouré par des blocs A comme il est illustré sur la figure ci-dessous. Dans ce cas, ce bloc B est remplacé par le bloc correspondant dans l'image A.

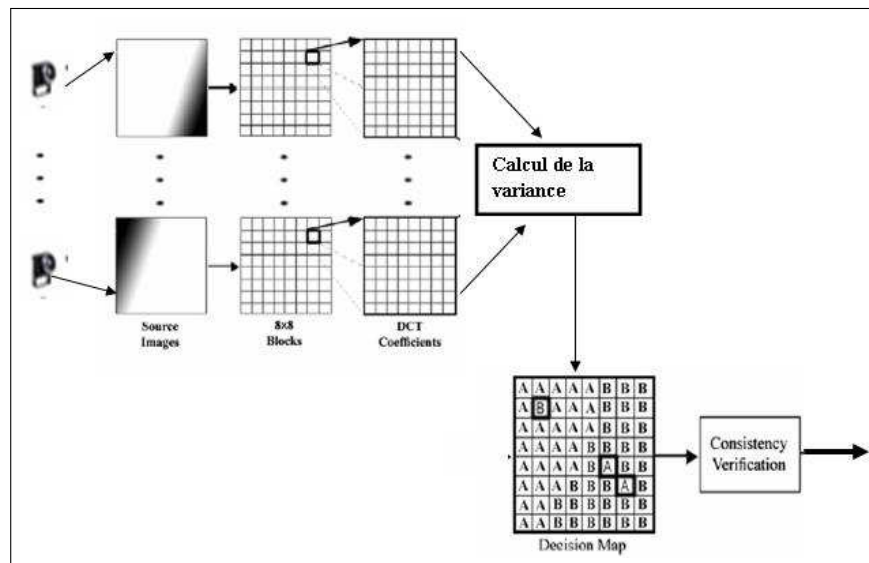


Figure 3.13 – Le processus de fusion DCT+variance.

Expérimentation

Les auteurs ont testé l'algorithme proposé sur 18 images créées artificiellement par brouillage de (06) six images standards présentées en figure 3.14. Le brouillage a été produit sur les deux moitiés gauches et droites des images.



Figure 3.14 – Échantillon d'images utilisées pour la simulation.

Les auteurs ont donné la valeur moyenne de la SSIM (Structural Similarity) pour l'algorithme proposé et d'autres algorithmes. Ainsi que la valeur moyenne de l'exécution des algorithmes en microsecondes par bloc de 8×8 pixels. Les résultats sont présentés dans les tableaux 3.13 et 3.14.

DCT+Average	DCT+Contrast	DWT	SIDWT	DCT+Variance	DCT+Variance+CV
0.90787	0.94672	0.98112	0.98389	0.97654	0.99872

Tableau 3.13 – La valeur moyenne du SSIM des algorithmes.

DCT+Contrast	DCT+Variance	DCT+Variance+CV
264.35004	16.14041	64.56163

Tableau 3.14 – La valeur moyenne de l'exécution des algorithmes en microsecondes par bloc 8×8 .

Afin d'évaluer l'algorithme proposé sur des images réelles, les auteurs ont utilisés quelques techniques de mesure de performance très utilisées dans le domaine de la fusion d'images, il s'agit des deux métriques Piella (Q_w) et Petrovic ($Q^{AB/F}$). La première donne une indication sur la quantité d'information pertinente contenue dans chaque image source et qui a été transférée vers l'image fusionnée. La deuxième métrique mesure la quantité d'information relative aux bordures transférée à partir des images source A et B vers l'image fusionnée F.

—	DCT+Average	DCT+Contrast	DWT	SIDWT	DCT+Variance	DCT+Variance+CV
Q_w	0.8545	0.9073	0.9364	0.9449	0.9275	0.9579
$Q^{AB/F}$	0.6431	0.6696	0.7266	0.7506	0.7716	0.7909

Tableau 3.15 – Evaluation des deux l'algorithme proposés par rapport aux autres algorithmes en utilisant les métriques Piella et Petrovic.

Il est à signaler que les auteurs ont implémenté leurs algorithmes en utilisant l'outil Matlab car il offre une bibliothèque très riche en fonction de traitement d'images.

Analyse des résultats obtenus : D'après le tableau 3.13 il est clair que l'algorithme proposé est plus efficace (similarité élevée) que les autres algorithmes même sans vérification de la consistance. En plus, après la vérification de la consistance (CV) une amélioration considérable est enregistrée. En termes de temps d'exécution, le tableau 3.14 montre que l'algorithme proposé est très rapide même avec la vérification de la consistance. En ce qui concerne les deux métrique Piella et Petrovic il est clair que l'algorithme proposé est meilleurs que les autres algorithmes (tableau 3.15).

Critiques

Malgré les résultats obtenus et qui montrent l'efficacité de l'algorithme proposé, les auteurs n'ont pas utilisé des images réelles capturées par des capteurs multimédia pour évaluer le comportement de l'algorithme. Les auteurs indiquent que la méthode de fusion proposée est destinée aux réseaux de capteurs multimédia mais aucune implémentation sur des capteurs réels n'a été réalisée.

D'autre part, dans l'architecture de la solution proposée, les auteurs parlent d'agent de capture qui est le capteur multimédia lui-même et l'agent de fusion qui est une station de fusion. L'agent de capture envoie l'image source (à fusionner), entière à la station de fusion ce qui pose un problème de bande passante, et d'énergie car l'énergie nécessaire pour la transmission de données est plus importante que celle du traitement.

Tableau Comparatif des méthodes DWT et DCT

	DWT	DCT+Average	DCT+Contrast	DCT+Variance	DCT+Variance+CV
SSIM (%)	98.112	90.787	94.672	97.654	99.872
Qw (%)	93.64	85.45	90.73	92.75	95.79
QAB/F (%)	72.66	64.31	66.96	77.16	79.09
T.exe	47.6 s	11.39323(μ S /bloque 8×8)	264.35004(μ S /bloque 8×8)	16.14041 (μ S /bloque 8×8)	64.56163 (μ S /bloque 8×8)
PSNR(db)	35	30	34.5	—	—
Scalabilité	Testé sur 2 images	Testé sur 2 images	Testé sur 2 images	Testé sur 2 images	Testé sur 2 images
Complexité / bloque $N \times N$	N^2	N^2	N^2	N^2	N^2
Consommation d'énergie	Non pris en charge	Non pris en charge	Non pris en charge	Non pris en charge	Non pris en charge
Niveau de fusion	Pixel/région	Pixel (Bloque de 8×8)	Pixel (Bloque de 8×8)	Pixel (Bloque de 8×8)	Pixel (Bloque de 8×8)
Représentation du signal	Temporel - fréquentiel	Fréquentiel	Fréquentiel	Fréquentiel	Fréquentiel
Catégorie des images sources	Multi-focus images	Multi-focus images	Multi-focus images	Multi-focus images	Multi-focus images
Réversible ?	OUI	OUI	OUI	OUI	OUI

Tableau 3.16 – Tableau Comparatif des méthodes DWT et DCT.

En plus des critiques individuelles que nous avons formulées à la fin de la présentation de chaque méthode, nous donnons un ensemble de remarques communes à toutes les méthodes de fusion.

- Aucune méthode n'aborde le problème de consommation d'énergie lors de la transmission de données (images). Dans la totalité des méthodes de fusions d'images

multi-focus citées dans l'état de l'art le capteur envoie la totalité de l'image ce qui consomme beaucoup d'énergie et réduit la durée de vie du réseau.

- Nous avons constaté aussi que dans la partie expérimentation de chaque méthode, aucune n'a estimé la quantité d'énergie consommée lors de la transmission de données malgré que la consommation d'énergie représente un facteur de base que nous devons prendre en considération dans le domaine de fusion de données du moment que la durée de vie d'un capteur dépend directement de sa batterie.
- Enfin, dans les deux méthodes de fusion de données par DCT, les auteurs ont considéré que les blocs constituant l'image possèdent le même poids (la même pondération) mais en réalité, certains blocs sont plus importants que d'autres. Autrement dit, les blocs qui constituent l'objet sur le quel la caméra du capteur est zoomée auront un poids important que les blocs constituant le reste de la zone d'intérêt, donc il est intéressant de penser à une pondération des blocs.

3.4 Conclusion

Dans le présent chapitre, nous avons évoqué les méthodes et les techniques de fusion et d'agrégation utilisées dans les réseaux de capteurs. Nous avons ciblé en premier lieu les méthodes de fusion définies dans les WSN et en second lieu celles dédiées aux WMSN. Une comparaison des méthodes a été menée. Dans le chapitre suivant, nous exposons notre propre solution qui tiendra compte des avantages des méthodes sus-citées et évitera leurs inconvénients.

Chapitre 4

Conception de la solution proposée

4.1 Introduction

Dans un réseau de capteurs, la consommation d'énergie et l'occupation de la bande passante sont les principaux facteurs déterminants de la durée du vie du réseau. En particulier, un réseau de capteurs multimédias (RCM) est caractérisé par un important volume de données (images, vidéos,...) échangé et par un délai de transmission rallongé. Un RCM est constitué principalement d'un ensemble de caméras alimentées par des batteries spécifiques, difficilement rechargeables. L'ensemble de ces caractéristiques place les nœuds capteurs dans un contexte où ils sont appelés donc à exploiter les ressources limitées du réseau d'une manière efficace. Par conséquent, un mécanisme de contrôle de l'énergie et de la bande passante est fortement indispensable dans un tel réseau.

Durant ces dernières années, le concept de fusion de données a été largement utilisé dans les réseaux de capteurs afin de réduire le volume de données circulant sur le réseau et par conséquent réduire la quantité d'énergie consommée. Dans une architecture de réseau de capteurs, la fusion de données ne suffit pas. Une étape préalable à la fusion est primordiale, déterminant l'information utile à envoyer du nœud capteur jusqu'à l'agent de fusion. En raison des limites en énergie, il est pratiquement impossible de combiner (fusionner) les données (dans notre cas des images) au niveau des capteurs. L'idée est de transférer ces données à travers le réseau à un centre de fusion (agent de fusion) doté de ressources nécessaires à l'accomplissement de cette tâche.

La question est comment transférer ces données afin de réduire la consommation d'énergie et minimiser l'occupation de la bande passante ?
Dans un réseau de capteurs, la consommation de l'énergie lors de la transmission de

données est beaucoup plus importante que la consommation d'énergie lors d'un traitement. La plupart des techniques de fusion d'images dans les réseaux de capteurs se focalisent sur la qualité de l'image fusionnée et ignorent d'autres critères clés, tel que la consommation de l'énergie et l'utilisation accrue de la bande passante. La monopolisation du support de transmission est causée par l'envoi de la totalité de l'image capturée vers l'agent de fusion.

Dans le présent chapitre nous détaillons notre approche qui a pour objectifs d'abord la minimisation du nombre de blocs à envoyer à l'agent de fusion par le nœud capteur. Nous ciblons un type bien déterminé de fusion d'images, appelé, **Multi-Focus Image Fusion**, déjà discuté dans les chapitres précédents.

4.2 Le contexte

Notre contribution s'articule autour de deux objectifs, le premier consiste à la proposition d'un mécanisme de gestion de l'envoi de données multimédias dans un réseau de capteurs afin de réduire la quantité de données circulant sur le réseau et le deuxième objectif comprend l'élaboration d'une technique de fusion de données (images) . Avant de détailler l'architecture de notre solution, nous avons jugé utile d'évoquer le contexte dans lequel notre solution devra être considérée, en tenant compte des contraintes suivantes :

- Nous considérons ici la fusion des images et non pas des séquences vidéos (le principe peut être généralisé car une séquence vidéo est un ensemble d'images capturées dans un intervalle de temps très réduit) ;
- Le type de fusion d'images utilisé est appelé **Multi-Focus image fusion**. Ce type de fusion est caractérisé par le fait que deux ou plusieurs capteurs surveillent la même zone d'intérêt, mais chaque capteur se focalise sur un objet bien déterminé de la zone. Sur l'image capturée, l'objet en focale sera clair alors que les autres objets de la même zone (qui ne sont pas dans la profondeur du champ), apparaissent flous. La profondeur du champ d'une caméra est définie comme l'intervalle de profondeur sur lequel les objets sont projetés avec une netteté suffisante. Cette contrainte rend la capture d'une image nette contenant tous les objets d'une zone d'intérêt impossible. D'ou la nécessité d'un mécanisme de fusion de données afin d'obtenir une image fusionnée améliorant la netteté de tous les objets de la zone d'intérêt. Cette image fusionnée est obtenue à partir des images sources capturées par chaque capteur. Par exemple, sur la figure 4.1 nous voyons une image capturée par une caméra de faible profondeur de champ, contrairement à la figure 4.2 qui

représente une image capturée par une caméra de grande profondeur de champ. Sur la figure 4.3, la caméra se focalise sur la bouteille d'eau minérale (Alma) ; l'inscription **Eau minérale naturelle non gazeuse** apparaît clairement par rapport aux autres objets présents sur l'image comme l'inscription **Eau minérale naturelle**(en arabe) sur la bouteille d'eau minérale (Ifri). Par contre, sur la figure 4.4 c'est l'inverse, la caméra se focalise sur la bouteille d'eau minérale (Ifri) ; l'inscription **Eau minérale naturelle**(en arabe) apparaît nettement que l'inscription **Eau minérale naturelle non gazeuse** sur la bouteille d'eau minérale (Alma). Le but de la fusion est d'obtenir l'image illustrée en figure 4.5 dans laquelle la netteté des deux bouteilles d'eau minérale (Ifri et Alma) est augmentée.

- Dans notre cas, nous supposons que les images sources sont bien alignées (en anglais *registred*) car cette contrainte est primordiale pour la réussite du processus de fusion d'images.
- Les images manipulées sont des images en niveau de gris (monochromes). C'est à dire que l'image est représentée par une matrice 2-D dans laquelle chaque pixel est représenté par une seule valeur qui varie entre 0 et 255 (0 : noir, 255 : blanc) ce qui donne 256 valeurs possibles. Si on veut fusionner des images en couleurs, le pixel sera représenté par trois composantes définissant les couleurs de bases (RGB : Rouge, Vert et Bleu), constituant la couleur du pixel. Le principe est le même, il suffit de faire une fusion au niveau de chaque composante.
- Nous supposons que chaque capteur garde une image de référence de sa zone d'intérêt, cette image de référence est mise à jour chaque fois que le capteur détecte des changements entre l'image de référence et l'image capturée.

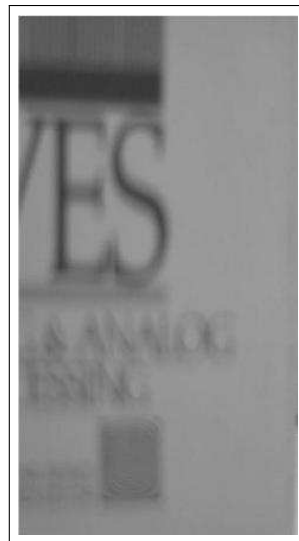


Figure 4.1 – Faible profondeur de champ.

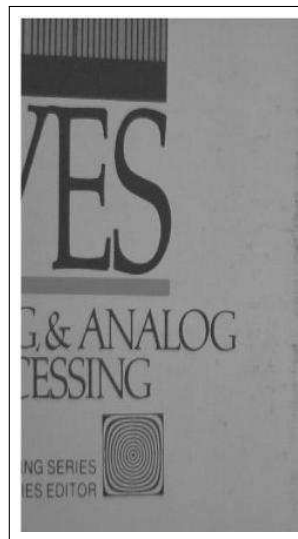


Figure 4.2 – Grand profondeur de champ.



Figure 4.3 – Image zoomée sur la partie gauche.



Figure 4.4 – Image zoomée sur la partie droite.



Figure 4.5 – Image souhaitée après fusion.

4.3 Architecture proposée

Afin de répondre aux objectifs ciblés par notre approche, nous avons considéré une architecture constituée de trois (03) couches, comme il est illustré sur la figure 4.6 :

1. Une couche de base appelée *couche de capture*, composée de l'ensemble de capteurs ;
2. Une couche intermédiaire dite *couche de fusion*. Elle englobe l'ensemble d'agents de fusion ; et
3. Une couche supérieure représente la *couche de traitement et de décision* qui contrôle la qualité des images fusionnées avant une éventuelle exploitation.

Dans ce qui suit, nous allons détailler le fonctionnement de chacune de ces couches.

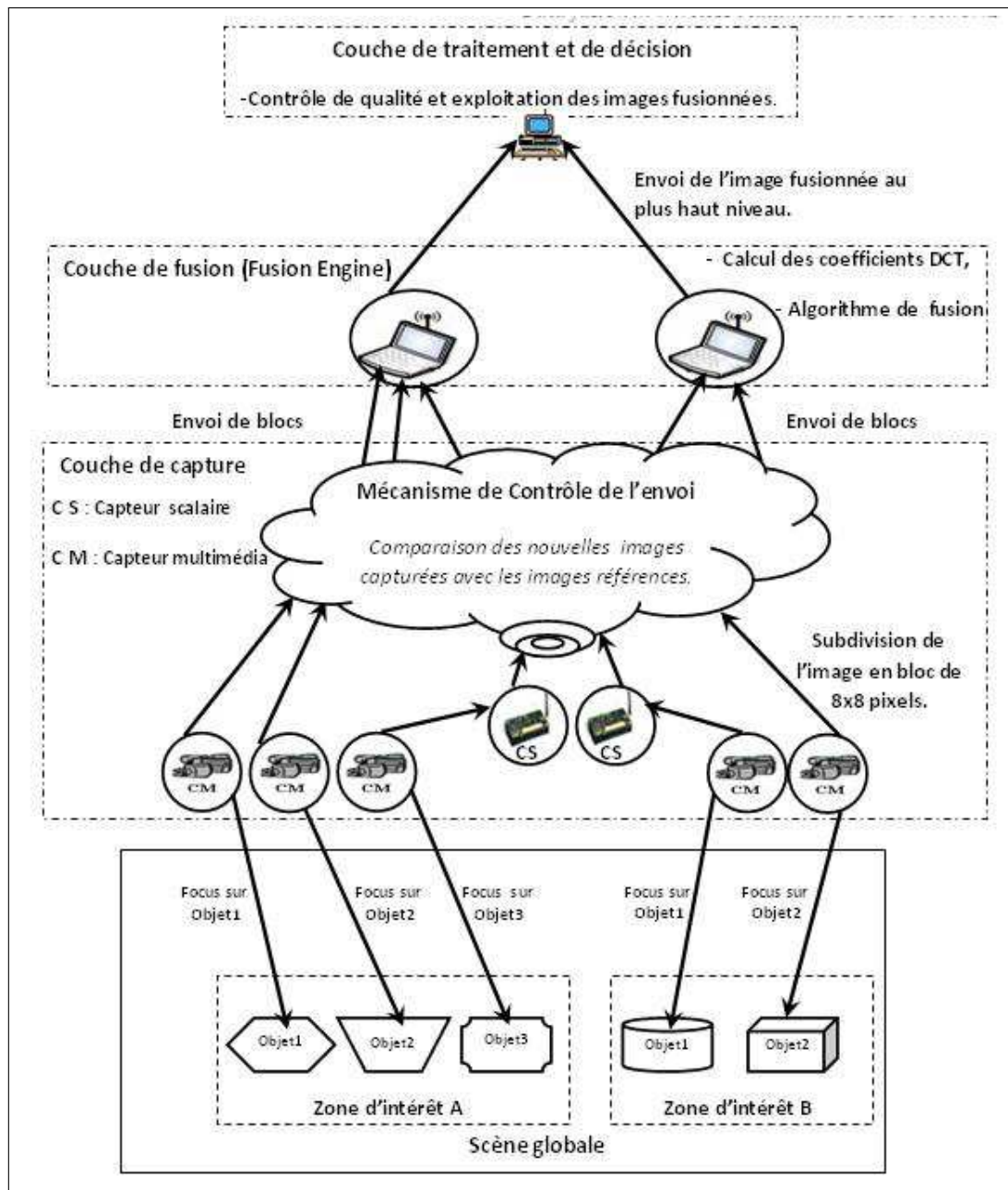


Figure 4.6 – Architecture générale de la solution proposée.

4.3.1 La couche de capture

Cette couche est constituée d'un ensemble de capteurs multimédias ainsi qu'un ensemble de capteurs scalaires surveillant des zones d'intérêts ; chaque capteur multimédia se focalise sur un objet bien déterminé de sa zone d'intérêt. Dans l'image capturée, l'ob-

jet en focale sera clair et les autres objets seront brouillés (floues). Le capteur multimédia aura pour tâches :

1. la capture des images ;
2. le découpage des images en bloc de 8×8 pixels ;
3. le stockage et la mise à jour de l'image de référence de sa zone d'intérêt ;
4. la détection d'un changement entre l'image de référence et la nouvelle image capturée ;
5. et enfin l'envoi des blocs d'images affectés par des changements à l'agent de fusion après que ce dernier ait autorisé l'opération d'envoi.

Chaque fois qu'il y a un changement entre l'image de référence et l'image capturée, le capteur en question met à jour son image de référence qui sera remplacée par l'image capturée (uniquement les blocs affectés et non pas la totalité de l'image). Le capteur multimédia peut envoyer les blocs de données directement à son agent de fusion comme il peut les envoyer à un capteur scalaire ou multimédia voisin. Ce dernier peut jouer le rôle d'un relai, il aura pour tâche l'envoi des paquets vers l'agent de fusion (voir figure 4.6). Il est à signaler que chaque capteur est supervisé par un agent de fusion (noeud de fusion). Initialement, chaque capteur effectue ce que nous appelons, *une phase d'initialisation*. Dans cette phase, le capteur envoie une première image dans sa totalité à son agent de fusion qui sera considérée comme étant son image de référence. Notons que, le capteur multimédia n'a pas besoin d'être renforcé au niveau de son unité de traitement, car la fusion des images est effectuée au niveau du noeud de fusion. L'unité de traitement de chaque capteur sera donc uniquement chargée des tâches susmentionnées. Dans le but d'évaluer le degré de changement, le capteur aura à calculer deux seuils notés respectivement T_{B_k} et T_G . Le premier seuil T_{B_k} donne une estimation sur le taux de différence entre deux blocs en comparaison (bloc de l'image référence et bloc de l'image capturée). Le deuxième seuil T_G représente un seuil global qui estime le taux de différence entre les deux images entière (référence et capturée). Nous avons opté pour ces deux seuils car ils sont complémentaires, pour une image (référence/capturée) constituée de N blocs, nous avons N seuils T_{B_k} (un seuil pour chaque bloc B_k) et un (01) seul seuil T_G calculé à partir des seuils T_{B_k} (pour plus de détails, voir la section calcul de T_{B_k} et T_G).

4.3.2 La couche de fusion

Comme son nom l'indique, cette couche est constituée d'agents de fusion. Chaque agent de fusion supervise l'ensemble de capteurs surveillant la même zone d'intérêt. L'agent de fusion lui-même garde une image de référence de chaque capteur dont il

supervise, cette image est mise à jour chaque fois que l'agent de fusion réceptionne des blocs de données de la part du capteur en question. Autrement dit, l'agent de fusion met à jour l'image de référence lorsqu'il reçoit des blocs d'image envoyés par le capteur concerné.

Les tâches principales de l'agent de fusion sont :

1. d'effectuer la fusion des images selon le principe de la DCT qui sera détaillée par la suite ;
2. l'envoi de l'image fusionnée à un niveau supérieur pour un contrôle de la qualité ainsi qu'un éventuel traitement ou prise de décision.

Pour ces besoins, un agent de fusion représente un nœud qui doit être renforcé au niveau de son unité de traitement et doté de ressources énergétiques importantes. D'autre part l'agent de fusion est débarrassé de la tâche de capture de données. L'agent de fusion peut être vue en tant que station de base qui reçoit les paquets de l'ensemble des nœuds capteurs (scalaires ou multimédias). Cette couche représente le moteur de fusion (Fusion Engine).

4.3.3 La couche de traitement et de décision

Cette couche supérieure reçoit les images fusionnées à partir des différents agents de fusion. Au niveau de cette couche, les images fusionnées sont soumises à un processus de contrôle de qualité où sont calculés certaines métriques d'évaluation de l'opération de fusion tel que le PSNR. Eventuellement, les images fusionnées peuvent passer par un autre processus de traitement, tel que l'extraction de quelques objets afin d'être utilisées pour prendre des décisions sur des situations bien déterminées. Cette couche peut être représentée par une station de traitement d'images ou un centre de prise de décision. Le nombre de stations de travail au niveau de cette couche dépend du nombre d'agents de fusion existants sur le réseau ainsi que le type d'application en question. Cette station est connectée aux agents de fusion à travers une connexion Internet ou satellitaire.

4.4 Description du Fonctionnement de la solution proposée

Dans cette section, nous décrivons les techniques proposées au niveau de chaque couche dans le cadre de notre solution.

4.4.1 Formalisation

Dans ce qui suit nous formulons les notations mathématiquement de l'ensemble des actions prévues dans la solution.

- Soit $N = \{S_1, S_2, \dots\}$ un ensemble de capteurs multimédias (CM) déployés dans une scène qu'on désire contrôler, notée S ;
- La scène S est subdivisée en plusieurs zones d'intérêt, $Z = \{A, B, \dots\}$;
- Chaque zone d'intérêt Z_i contient un ensemble d'objets $O = \{O_{i1}, O_{i2}, \dots\}$;
- Soit $F = \{F_1, F_2, \dots\}$ un ensemble d'agents de fusion ;
- L'ensemble de CM qui surveille la même zone d'intérêt Z_i envoie les blocs de données au même agent de fusion F_i ;
- Soit $(Iref_i)$ l'image de référence du CM S_i et $(Icap_i)$ l'image capturée par le CM S_i ;
- Chaque image de référence $Iref_i$ est subdivisée en blocs $Bref_1, Bref_2, \dots$ de 8×8 pixels ;
- Chaque image capturée $Icap_i$ est subdivisée en blocs $Bcap_1, Bcap_2, \dots$ de 8×8 pixels ;
- Chaque bloc $(Bcap_i)$ possède un poids $(Wcap_i)$. Ce poids représente le degré d'importance du bloc dans l'image ;
- Soit $(IrefAF[i])$ l'image de référence au niveau de l'agent de fusion envoyée par le capteur S_i ;
- $T_{B_k}^i$ est le seuil de différence calculé entre un bloc k de l'image de référence $Iref_i$ et son homologue dans l'image capturée $Icap_i$ du capteur S_i ;
- T_G^i est le seuil de différence global entre l'image de référence $Iref_i$ et l'image capturée $Icap_i$ du capteur S_i , ce seuil sera envoyé à l'agent de fusion concerné.

4.4.2 Description de la solution au niveau capteur

1. **Découpage des images (références et capturées) en blocs de 8×8 pixels**
 Chaque capteur doit garder une image de référence de sa zone d'intérêt. Cette image de référence est initialisée à la première image capturée $Iref_i \leftarrow Icap_i$. Afin d'éviter à chaque capture l'envoi de la totalité de l'image comme le cas des techniques citées dans l'état de l'art[5],[6], un mécanisme de subdivision de l'image de référence et l'image capturée en blocs de 8×8 pixels est prévu. Le choix de la taille 8×8 est justifié par le fait que la variation des fréquences dans un bloc de 8×8 est pratiquement négligeable contrairement à un bloc de 256 ou 512 pixels où la variation des fréquences est significative, c'est-à-dire que dans un bloc de taille importante (de 128, 256 ou 512) les pixels n'ont plus que peu de relation

entre eux. L'objectif d'un tel découpage se résume comme suit :

- Réduction de la complexité, car il est plus facile de traiter des matrices de 64 valeurs que de traiter des matrices de 240.000 valeurs (pour une image de taille 600×400);
- En plus notre approche de fusion est basée sur la construction de l'image résultante bloc par bloc à partir des images sources subdivisées aussi en blocs de 8×8 pixels;
- Il n'est pas intéressant d'appliquer une DCT sur une image de taille importante. Par exemple si nous appliquons une DCT sur une image entière de dimension 512×512 pixels, les informations spatiales de l'image seront mélangées car la DCT divise le signal de départ en deux signaux (base et haute fréquences) et lorsque cette transformée est appliquée sur un bloc de taille réduite, l'information spatiale de l'image entière ne sera pas mélangée (le traitement se fait à l'intérieure de chaque bloc);
- Enfin, la rapidité en termes de temps d'exécution, la DCT sur une image découpée en blocs est plus rapide que la DCT sur la même image entière (non découpée en blocs). Nous avons testé cette contrainte sur une image de taille 512×512 en utilisant l'outil de simulation Matlab (voir figures 4.7 et figure 4.8). Dans le cas où l'image est découpée en blocs de 8×8 , le temps d'exécution été de **0.021744** secondes, par contre dans le cas où l'image na pas été découpée en blocs, le temps d'exécution fut de **0.103388** secondes.

```
>> clear all
>> img_color=imread('c:\clock1.jpg');
I=rgb2gray(img_color);
[m,n]=size(I);
I=double(I)-128;
tic
for i=1:m/8
    for j=1:m/n
        Isub=I(8*i-7:8*i,8*j-7:8*j);
        IsubDCT=dct2(Isub);
    end
end
toc
Elapsed time is 0.021744 seconds.
fx >>
```

Figure 4.7 – Temps d'exécution d'une DCT sur une image découpée en blocs de 8×8 pixels.

```

>> clear all
>> img_color=imread('c:\clock1.jpg');
I=rgb2gray(img_color);
[m,n]=size(I);
I=double(I)-128;
tic
    j=dct2(I);
toc
Elapsed time is 0.103388 seconds.
fx >>

```

Figure 4.8 – Temps d'exécution d'une DCT sur une image entière.

La figure 4.9 illustre l'algorithme de découpage d'une image numérique en blocs de 8×8 pixels.

```

% Algorithme Decompose(img)%
Begin
Buffer = Read('image.jpg'); // Lecture et récupération des pixels dans un tampon,
Get(width),Get(height); // Récupération de la dimension de l'image,
for(i=0; i < height; i++)
{
for(int j = 0; j < width;j++)
{
image[j][i] = buffer[(i*width)+j]; //Récupération des pixels
}
}

//Découpage de l'image en blocs de 8x8 pixels
for(int i = 0; i < height/8; i++)
{
for(int j = 0; j < width/8;j++)
{
for(int k = i*8, l = 0; k < (i*8)+8; k++,l++)
{
for(int m = j*8, n = 0; m < (j*8)+8; m++,n++)
{
dividedimage[(i*(width/8))+j][n][l] = image [m][k];
}
}
}
}
End.

```

Figure 4.9 – Algorithme de découpage d'une image en blocs de 8×8 pixels.

A la fin, chaque image $Iref_i$ respectivement, $Icap_i$ est subdivisée en $k \geq 1$ blocs de 8×8 pixels appelés $Bref_1, Bref_2, \dots, Bref_k$ respectivement, $Bcap_1, Bcap_2, \dots, Bcap_k$ ne se chevauchant pas ($Bref_i \cap Bref_j$ or $Bcap_i \cap Bcap_j = \emptyset$. Pour chaque $i \neq j$) et dans ce cas : $Iref$ et $Icap = \cup_{k=1}^K Bref_k, Bcap_k$.

Dans la solution que nous proposons, la notion de pondération des blocs est introduite; chaque bloc dans l'image possède un poids selon son importance dans l'image. Ce poids est pris en considération lors du calcul du seuil global T_G^i qui donne le taux de différence entre l'image de référence et l'image capturée. Considérons l'architecture générale de la solution illustrée dans la figure 4.6, il

est clair que les blocs qui contiennent l'objet sur le quel le capteur est en focus auront un poids important que le reste des blocs de la zone d'intérêt. Autrement dit, un changement au niveau d'un bloc qui appartient à l'objet en focus est plus significatif qu'un changement au niveau d'un autre bloc qui n'appartient pas à l'objet en focus.

2. Comparaison de l'image capturée avec l'image de référence

Une fois l'image de référence et l'image capturée découpées en blocs, l'unité de traitement de chaque capteur procède à une comparaison bloc par bloc de ces images (image de référence et l'image capturée). En premier, le taux de différence $T_{B_k}^i$ entre un bloc de l'image de référence et son homologue dans l'image capturée est calculé. Si ce taux dépasse un certain seuil maximum, noté T_B^{max} alors les deux blocs sont considérés différents sinon ils sont déclarés identiques. Le deuxième taux noté T_G^i représente le taux global de différence entre l'image de référence et l'image capturée. Plus de détails sur ces deux taux sont données dans les sections qui viennent. A la fin du calcul, le capteur envoie la valeur de T_G^i à l'agent de fusion, ce dernier décide de l'envoi des blocs affectés par le changement au lieu de l'envoi de la totalité de l'image.

3. Mise à jour de l'image de référence

Les capteurs sont appelés à envoyer les blocs modifiés lorsque l'agent de fusion le décide. Avant de procéder à l'envoi, chaque capteur doit effectuer une opération de mise à jour de son image de référence afin de ne pas perdre l'information. La mise à jour est effectuée par un écrasement des anciens blocs de l'image de référence par les nouveaux blocs de l'image capturée. Cela se fait dans le cas où il y a un changement significatif entre les blocs des deux images.

4. Calcul des seuils de différence $T_{B_k}^i$, T_G^i

Afin d'atteindre l'objectif principal de notre solution qui est la limitation de la taille et du nombre de paquets circulants sur le réseau, une image de référence est gardée au niveau de chaque capteur, cette image sera par la suite comparée avec l'image capturée chaque fois qu'il y a une nouvelle capture. La problématique est quand est ce qu'on peut dire qu'il y a une différence significative entre l'image de référence et l'image capturée. Cela est réalisé par mesure de similarité entre les images successives :

Concernant le premier seuil $T_{B_k}^i$ qui donne un degré de similarité entre deux blocs, la première réflexion qui vient à l'esprit est de comparer les pixels des deux blocs en niveau de gris, autrement dit nous allons estimer le nombre de pixels différents entre les deux blocs. Si ce nombre dépasse une certaine valeur alors les deux blocs sont déclarés différents. Autrement, ils sont considérés comme identiques. Le problème avec cette méthode est qu'elle est très sensible aux petites variations de pixels ; si on prend par exemple le cas où deux images identiques sont prises

avec un taux d'éclairage différent elles sont automatiquement considérées comme étant non identiques et cela engendre un trafic inutile sur le réseau. Pour confirmer cela, nous avons effectué l'expérience suivante :

Nous avons pris deux images identiques (même scène) avec une camera **FujiFilm** de résolution **10.0 Mega Pixels** mais avec un taux d'éclairage différent. Par la suite nous avons converti ces images en niveau de gris et nous avons pris deux blocs de 8×8 pixels. Le premier bloc a été pris à partir de l'image dont l'éclairage est élevé. Le deuxième bloc représente le bloc homologue dans l'image prise avec un taux d'éclairage faible. Les valeurs des pixels des deux blocs sont illustrées dans les figures 4.10 et 4.11

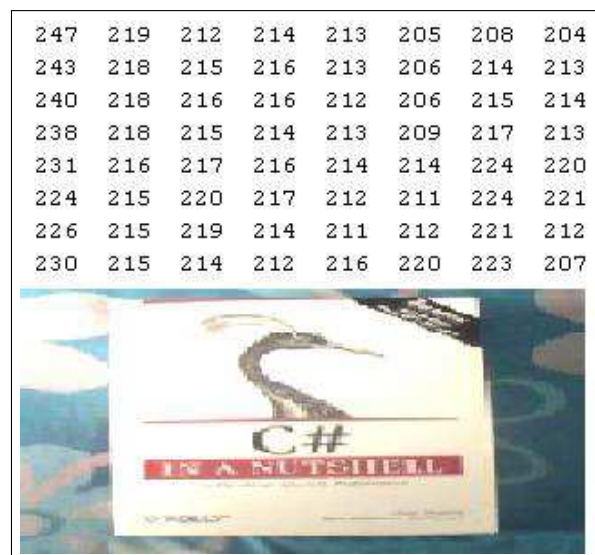


Figure 4.10 – Bloc de 8×8 pixels d'une image prise avec éclairage élevé.

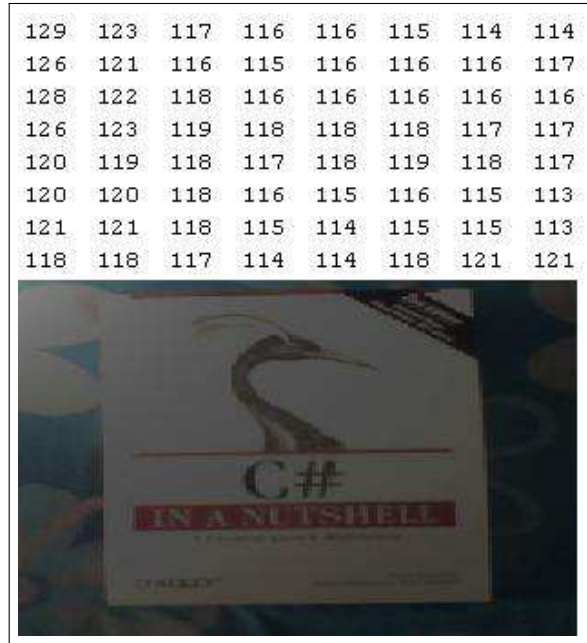


Figure 4.11 – Bloc de 8×8 pixels d'une image prise avec éclairage faible.

On constate que les deux blocs possèdent des valeurs différentes de pixels (la moyenne du premier bloc est 217, la moyenne du deuxième bloc = 118) malgré qu'elles représentent la même scène. Ceci nous pousse à écarter cette solution et penser à un autre mécanisme moins sensible au variation de l'éclairage. Pour ce faire, nous avons opté pour un outil mathématique qui nous permet d'estimer la similarité entre deux blocs ; il s'agit du coefficient de corrélation entre deux matrices carrées. Le choix d'un tel outil est motivé par le faite que la corrélation est basée sur des principes mathématiques solides déjà reconnus tels que la variance et la covariance. De plus, elle ne dépend pas de l'échelle (les unités de mesure des variables qu'on veut corrélérer), et enfin, la corrélation entre deux images n'est pas sensible aux changement d'éclairage (chose que nous avons vérifié à travers un test).

Le coefficient de corrélation noté τ entre deux matrices A et B de même taille est donnée par la formule 4.1.

$$\tau = \frac{\sum_m \sum_n (A_{mn} - \bar{A}) \cdot (B_{mn} - \bar{B})}{\sqrt{(\sum_m \sum_n (A_{mn} - \bar{A})^2) \cdot (\sum_m \sum_n (B_{mn} - \bar{B})^2)}} \quad (4.1)$$

Avec : \bar{A} est la moyenne du bloc A , \bar{B} est la moyenne du bloc B .

Dans notre contexte les deux blocs A et B ont la même taille $m = n = 8$ et dans ce cas le premier seuil $T_{B_k}^i$ qu'on veut calculer peut être donné par la formule

suivante :

$$T_{B_k}^i = 1 - \left| \frac{\sum_{i=1}^8 \sum_{j=1}^8 (A_{ij} - \bar{A}) \cdot (B_{ij} - \bar{B})}{\sqrt{(\sum_{i=1}^8 \sum_{j=1}^8 (A_{ij} - \bar{A})^2) \cdot (\sum_{i=1}^8 \sum_{j=1}^8 (B_{ij} - \bar{B})^2)}} \right| \quad (4.2)$$

Avec : $\bar{A} = \frac{1}{64} \sum_{i=1}^8 \sum_{j=1}^8 A_{ij}$ et $\bar{B} = \frac{1}{64} \sum_{i=1}^8 \sum_{j=1}^8 B_{ij}$

La valeur du coefficient $T_{B_k}^i$ est comprise entre 0 et 1 ($0 \leq T_{B_k}^i \leq +1$). Une valeur du coefficient proche de zéro (0) implique que les deux blocs sont fortement corrélés par contre si la valeur absolue du coefficient est proche de 1, cela indique que les deux blocs ne sont pas corrélés. Pour la valeur de T_B^{max} du coefficient de corrélation à partir de laquelle les blocs sont considérés comme non identiques, nous avons préféré qu'elle soit dynamique et non pas statique. Elle sera dynamique en fonction de l'énergie résiduelle du capteur. Par exemple si l'énergie résiduelle du capteur est de $2000mA$ et le seuil est fixé à 0.4 c'est-à-dire que les blocs qui possèdent un coefficient de corrélation supérieur ou égale à 0.4 sont considérés comme différents, et pourront faire l'objet d'un envoi vers l'agent de fusion. Si l'énergie résiduelle du capteur devient faible (par exemple $800mA$), le seuil doit être revu à la hausse (par exemple 0.6) afin de réduire le nombre de blocs considérés non identiques (réduire le nombre de blocs à envoyer à l'agent de fusion), et par conséquent conserver l'énergie du capteur. Lorsque $T_{B_k}^i$ est inférieur à **0.5** cela signifie une forte corrélation. La corrélation est considérée faible pour une valeur supérieur à **0.5**. En se basant sur l'interprétation du coefficient de corrélation donnée par *Deborah J. Rumsey* dans [73], nous avons :

0.0	Relation linéaire parfaite
0.3	Relation linéaire forte
0.5	Relation linéaire modérée
0.7	Relation linéaire faible
1.0	Pas de relation linéaire

Tableau 4.1 – Interprétation du coefficient $T_{B_k}^i$.

En ce qui concerne le deuxième seuil noté T_G^i qui représente un seuil global et qui est envoyé à l'agent de fusion concerné, il est calculé en fonction des seuils $T_{B_k}^i$ de chaque bloc et en fonction de la pondération (poids) des blocs. En quelque sorte ce seuil représente une moyenne pondérée et il est donné par la formule 4.3.

$$T_G^i = \frac{\sum_{k=1}^K (W_k \times T_{B_k}^i)}{\sum_{k=1}^K (W_k)} \quad (4.3)$$

Avec : W_k : Le poids du bloc k , le choix de ce poids sera discuté par la suite. Si

K est le nombre de blocs, $T_{B_k}^i$ représente la différence entre le bloc k dans l'image de référence et son homologue dans l'image capturée.

Dans le prochain chapitre, l'implémentation du calcul de ces deux paramètres sera explicité. Notant que seulement les blocs contenant un changement significatif sont envoyés à l'agent de fusion.

Exemple numérique :

Pour illustrer le calcul de ces deux paramètres $T_{B_k}^i$ et T_G^i de notre approche, nous considérons deux images A et B illustrées sur les figures 4.3 et 4.4. Pour ce faire, nous avons subdivisé les deux images en blocs de 8×8 pixels. Afin de faciliter le calcul, nous avons considéré quatre (04) blocs de ces images. Respectivement illustrés sur les figures 4.12, 4.13.

Les coefficients de corrélation entre les blocs des deux images A et B sont : $T_{1A1B} = 0.5034$, $T_{2A2B} = 0.8729$, $T_{3A3B} = 1.0$, $T_{4A4B} = 0.3565$. Soit les poids des blocs associés d'une manière aléatoire (Ex : 0.8 ; 0.6 ; 0.8 ; 0.4) ; nous obtenons comme seuil $T_{G_{AB}}$:

$$T_{G_{AB}} = \frac{(0.8 \times 0.5034) + (0.6 \times 0.8729) + (0.8 \times 1) + (0.4 \times 0.3565)}{(0.8 + 0.6 + 0.8 + 0.4)} = \mathbf{0.72} \tag{4.4}$$

bloc01 =								bloc02 =							
240	240	240	240	240	240	240	240	239	239	239	239	239	239	239	239
240	240	240	240	240	240	240	240	239	239	239	239	239	239	239	239
240	240	240	240	240	240	240	240	239	239	239	239	239	239	239	239
240	240	240	240	240	240	240	240	240	240	240	240	240	240	240	240
240	240	240	240	240	240	240	240	240	240	240	240	240	240	240	240
240	240	240	240	240	240	240	240	241	241	241	241	241	241	241	241
239	239	239	239	239	239	239	239	240	240	240	240	240	240	240	240
239	239	239	239	239	239	239	239	240	240	240	240	240	240	240	240
bloc03 =								bloc04 =							
239	239	239	239	239	239	239	239	238	238	238	239	239	240	240	240
239	239	239	239	239	239	239	239	238	238	239	239	239	240	240	240
239	239	239	239	239	239	239	239	238	239	239	239	240	240	240	240
240	240	240	240	240	240	240	240	239	239	239	240	240	240	241	241
240	240	240	240	240	240	240	240	239	239	240	240	240	240	241	241
241	241	241	241	241	241	241	241	240	240	240	240	241	241	241	241
240	240	240	240	240	240	240	240	239	239	239	240	240	240	241	241
240	240	240	240	240	240	240	240	239	239	239	240	240	241	241	241

Figure 4.12 – Les quatre blocs de 8×8 pixels de l'image A.

bloc01 =								bloc02 =							
238	238	238	239	239	240	240	240	240	240	240	240	240	240	240	240
238	238	239	239	239	240	240	240	240	240	240	240	240	240	240	240
238	239	239	239	240	240	240	240	240	240	240	240	240	240	240	240
239	239	239	240	240	240	241	241	241	241	241	241	241	241	241	241
239	239	240	240	240	241	241	241	241	241	241	241	241	241	241	241
240	240	240	240	241	241	241	242	242	242	242	242	242	242	242	242
240	240	240	241	241	241	242	242	242	242	242	242	242	242	242	242
240	240	240	241	241	242	242	242	242	242	242	242	242	242	242	242
bloc03 =								bloc04 =							
240	240	240	240	240	240	240	240	240	240	240	240	240	240	240	239
240	240	240	240	240	240	240	240	240	240	240	240	240	240	240	239
240	240	240	240	240	240	240	240	240	240	240	240	240	240	240	239
241	241	241	241	241	241	241	241	241	241	241	241	241	241	241	240
241	241	241	241	241	241	241	241	241	241	241	241	241	241	241	240
242	242	242	242	242	242	242	242	242	242	242	242	242	242	242	241
241	241	241	241	241	241	241	241	241	241	241	241	241	241	241	241
241	241	241	241	241	241	241	241	241	241	241	241	241	241	241	241

Figure 4.13 – Les quatre blocs de 8×8 pixels de l’image B.

5. Affectation des poids W_k aux blocs B_k

Dans la solution que nous avons proposée, nous avons jugé utile de considérer la pondération des blocs dans le calcul du seuil global T_G^i . Cette pondération de blocs représente le degré d’importance d’un bloc par rapport à un autre.

Dans le contexte de fusion d’image dit multi-focus, le capteur se focalise sur une partie de l’image ce qui donne une image nette au niveau de la partie qui se trouve à l’intérieur de la profondeur du champ du capteur et floue dans le reste de l’image. Il est clair que la partie de l’image qui est en focus est la plus importante par rapport à l’autre partie de l’image, cela implique que les blocs de données qui appartiennent à la zone de focus doivent avoir une pondération importante par rapport aux autres blocs. Une telle approche implique l’utilisation d’un mécanisme mathématique permettant d’évaluer dynamiquement les zones nettes de l’image (tels que le PSNR). Mais cela impliquerait un surcout en calcul et en consommation d’énergie car ces poids devront être ré-évalués à chaque capture. Une autre considération est que l’utilisateur détermine à l’avance le degré d’importance des zones de la scène d’une manière statique et dans ce cas, chaque bloc prend une pondération selon son appartenance à sa zone dans la scène. L’utilisateur pourra marquer les contours des zones sur l’image de référence initiale décrivant la scène et leurs affecter un poids > 1 , les zones non marqués auront un poids minimal égal à 1. Tous les blocs d’une même zone auront le même poids. Ces poids sont envoyés vers l’agent de fusion qui les diffusera vers tous les capteurs qu’il supervise.

6. Mécanisme de transfert de données

Après que le capteur ait calculé les deux seuils T_B^i et $T_{G_k}^i$, le seuil global est envoyé automatiquement à l’agent de fusion. Ce dernier et lorsque il reçoit les différents seuils de tous les capteurs qu’il supervise, il procède au calcul d’un autre coefficient appelé coefficient de fusion T_F^i . En fonction de la valeur de ce seuil, l’agent de fusion décide de l’envoi des données à partir des capteurs.

Pour le protocole de communication et de routage adopté, plusieurs modèles existent dans la littérature, à savoir les modèles basés requête, basés événement et d'autres. Dans notre cas, nous avons opté pour le modèle basé événement, ce qui implique que la transmission des données est déclenchée par un événement (i.e, lorsque la valeur de T_F^i dépasse une valeur bien déterminée). Cette valeur sera discutée dans la suite du document.

Contrairement aux techniques citées dans l'état de l'art et qui sont basées sur l'envoi de la totalité de l'image, notre approche tente de réduire la charge de données à envoyer (i.e, uniquement des blocs affectés par des changements significatifs). Ceci va permettre de mieux gérer les ressources telles que la bande passante, l'énergie et la *CPU*, augmentant ainsi la durée de vie du réseau.

7. Algorithme global (niveau capteur)

Afin de synthétiser ce que nous avons discuté antérieurement, un algorithme principal de cette couche (niveau capteur) est illustré dans le tableau 4.2.

*** Algorithme Capteurs ***
<pre> Begin -<i>Iref</i> ← <i>Icap</i>, // Initialisation de l'image de référence, -<i>Send_To_Fa(Iref)</i>, // Envoi de l'image de référence à l'agent de fusion, -<i>Decompose(Iref)</i>, // Décompose l'image de référence en blocs de 8 × 8 pixels, -<i>Decompose(Icap)</i>, // Décompose l'image capturée en blocs de 8 × 8 pixels, -<i>Compare(Iref, Icap)</i>, //Compare l'image de référence avec l'image capturée bloc par bloc (calcul de $T_{B_k}^i$ et T_G^i), -<i>if(Compare(Iref, Icap))</i> +<i>Update(Iref)</i>, //mise à jour de l'image de référence, +<i>Send_To_Fa(Ti)</i>, // Envoi du seuil global à l'agent de fusion, -<i>if(Decision_Fa)</i> // l'agent de fusion réclame l'envoi de données. +<i>Send_Blocks_To_Fa()</i>, End </pre>

Tableau 4.2 – Algorithme niveau capteur

4.4.3 Description de la solution au niveau agent de fusion

1. Stockage des images références des capteurs

Chaque agent de fusion chapeaute un ensemble de capteurs surveillant une même zone d'intérêt. L'agent de fusion est amené à stocker une image de référence de chaque capteur qui dépend de lui. Cette image de référence est mise à jour chaque fois que l'agent reçoit des données (blocs) de la part du capteur concerné et elle est initialisée à la première image envoyée par le capteur. Pour ce faire, chaque agent de fusion dispose d'un tableau de taille n , où n est le nombre de capteurs

dont l'agent de fusion est responsable.

$$IrefAF[i] \leftarrow Iref_i$$

2. Réception des seuils de différence globale T_G^i et décision d'envoi

Lorsque le capteur calcule le seuil global T_G^i , ce dernier est envoyé directement à l'agent de fusion. La décision d'envoi des données est prise par l'agent de fusion car la fusion est effectuée à son niveau. Imaginons que le capteur envoie directement les données sans que l'agent l'ordonne, ceci peut provoquer des envois inutiles dans le cas où les images des autres capteurs ont subi peu de modifications (les seuils T_G^i sont faibles). Autrement dit, l'agent de fusion avant qu'il ordonne l'envoi de données, doit d'abord synthétiser les T_G^i reçus à partir des capteurs qu'il supervise. Les valeurs récoltées sont utilisées pour calculer un nouveau paramètre, appelé, seuil de fusion T_F^i , qui représente la moyenne des valeurs reçues (voir formule 4.5).

$$T_F^i = \frac{1}{N} \sum_{s=1}^N T_{G_s}^i. \quad (4.5)$$

Avec N : le nombre de capteurs que l'agent de fusion supervise.

Une fois calculé, la valeur de T_F^i est comparé à un seuil de référence T_F^{max} dont la valeur est adaptée dynamiquement en fonction de deux principaux critères :

- En fonction de l'énergie résiduelle de l'agent de fusion. La valeur T_F^{max} augmente si l'énergie résiduelle baisse. Cela permet de réduire les opérations de fusion et éviter la transmission des données jugées moins pertinentes (des capteurs vers l'agent de fusion, et de ce dernier vers la station de traitement.
- En fonction de l'application implantée. Si l'application est critique, la valeur T_F^{max} sera abaissée ce qui implique la nécessité d'envoyer les images et de les fusionner en présence de modifications légères, alors que pour des cas d'applications non critiques la valeur T_F^{max} sera augmentée pour ne prendre en considération que les changements pertinents.

3. Mise à jour des images de référence

Dès que l'agent de fusion reçoit des blocs de données, il doit impérativement procéder à la mise à jour de l'image de référence qui correspond au capteur qui a envoyé ces blocs. La mise à jour est effectuée par l'écrasement des anciens blocs de l'image de référence $Bref_i$ par les blocs $Bcap_i$ qui viennent d'être reçus.

4. Techniques proposées pour la fusion des images multi-focus

Comme déjà indiqué dans les sections précédentes, notre approche consiste à fusionner des données multimédias dans un contexte appelé Multi-Focus Image Fusion (**MFIF**). Les techniques de fusion que nous utilisons sont basées sur le

principe de la *DCT*. Les images sources sont ainsi découpées en blocs de 8×8 pixels. Par la suite, un coefficient *DCT* est calculé pour chaque bloc. Cette transformée permet de passer d'un domaine spatial à un domaine fréquentiel dans lequel les fréquences sont regroupées par région (bases et hautes fréquences). L'image fusionnée sera construite bloc par bloc à partir des blocs correspondants dans les images sources. Si nous prenons le cas de deux images I_1 et I_2 , la question est comment choisir entre les blocs de ces deux images afin de construire l'image fusionnée?. Autrement dit un bloc B_i de l'image fusionnée est-il ramené de l'image I_1 ou I_2 ?. Dans notre approche, le critère de sélection est le niveau d'information contenu dans le bloc; le bloc le plus informatif sera automatiquement sélectionné. Pour obtenir le niveau d'information d'un bloc *DCT*, nous testons plusieurs métriques à savoir la moyenne, l'écart type, la variance et l'énergie, contenus dans un bloc. Il est à signaler que la moyenne et la variance ont été déjà utilisées et implémentées dans la littérature. Par contre l'utilisation de l'écart type et l'énergie d'un bloc *DCT* représentent une contribution à notre solution globale. Une comparaison entre les performances de ces différentes métriques sera donnée dans le chapitre suivant.

Après la construction des blocs *DCT* de l'image fusionnée, une étape de vérification de la consistance est nécessaire afin d'éliminer les bruits et les parasites survenus lors de la phase de fusion. Pour cela nous avons opté pour la même technique de vérification de la consistance utilisée dans [5] et explicitée dans le chapitre précédent. A la fin du processus, une transformée inverse *IDCT* est appliquée pour obtenir l'image finale dans un domaine spatial, la formule de la transformée inverse *IDCT* utilisée est donnée par l'équation 3.26.

- **Moyenne dans un domaine *DCT*** : La moyenne d'un bloc *DCT* de 8×8 est donnée par la formule suivante :

$$\mu = \frac{1}{N^2} \sum_{k=1}^N \sum_{l=1}^N d(k, l). \quad (4.6)$$

Avec $d(k, l)$: le coefficient *DCT* et $N = 8$.

Pour le cas où nous avons deux images sources **A** et **B**, la moyenne est calculée pour chaque bloc *DCT* de chaque image. Par la suite chaque bloc *DCT* de l'image **A** est comparé à son homologue de l'image **B**. Le bloc *DCT* de l'image fusionnée est le bloc qui possède la valeur de la moyenne la plus élevée. Ce principe de sélection sera considéré pour le reste des techniques utilisées.

- **Ecart type dans un domaine *DCT*** : L'écart type d'un bloc *DCT* de taille 8×8 est donné par la formule suivante :

$$S = \sqrt{\frac{1}{N^2} \sum_{k=1}^N \sum_{l=1}^N (d(k, l) - \mu)^2}. \quad (4.7)$$

- **La variance dans un domaine DCT** : La variance d'un bloc DCT de taille $N \times N$ est donnée par la formule 3.33.
- **L'énergie d'un bloc DCT** : L'énergie d'un signal d'un bloc DCT de taille $N \times N$ est une caractéristique liée à la quantité d'information représentée dans ce bloc. Selon [74], dans un domaine continue, l'énergie d'un signal $x(t)$ fonction complexe de la variable réelle t , est la quantité E définie par :

$$E = \int_{-\infty}^{+\infty} |x(t)|^2 dt. \quad (4.8)$$

Dans notre cas l'énergie est calculée dans un domaine discret pour un signal de deux dimensions (image numérique), elle peut-être donnée par la formule :

$$E = \sum_{k=1}^N \sum_{l=1}^N d(k, l)^2. \quad (4.9)$$

Avec : $d(k, l)$ coefficient DCT d'un bloc de données.

5. **Vérification de la consistance de l'image fusionnée (Consistency Verification)** Supposons que nous avons une région R_1 de la scène qui englobe plusieurs blocs, supposons aussi que R_1 est complètement dans la profondeur de champ d'un capteur multimédia CM_1 . Nous supposons également que le reste de la scène noté R_2 est complètement dans la profondeur de champ d'un deuxième capteur multimédia CM_2 . L'image capturée par CM_1 est notée $image_1$, l'image capturée par CM_2 est notée $image_2$. L'image fusionnée est constituée bloc par bloc et dans ce cas tous les blocs de l'image fusionnée qui correspondent à la région R_1 de $image_1$ doivent obligatoirement être ramenés de $image_1$. Cependant, il peut y avoir des erreurs dues au bruit ou à des effets indésirables parvenus au cours de la procédure de sélection des blocs et qui peut conduire à la sélection erronée de certains des blocs de $image_2$. Ce défaut peut être résolu par une procédure de vérification de la consistance. Cette méthode est basée sur une technique dite le filtre de majorité. Si un bloc vient de $image_2$ alors que la majorité des blocs voisins viennent de $image_1$, ce bloc central est tout simplement basculé au bloc correspondant dans $image_1$. A la fin, l'image fusionnée est obtenue à base de cette carte de décision (voir figure 4.14). Il est à signaler que la vérification de la consistance est appliquée dans une fenêtre de taille $m \times n$ blocs voisins. Dans [5] il a été montré que si la taille de la fenêtre augmente, la qualité de l'image fusionnée sera évidemment augmentée au détriment de la complexité. Il a été démontré expérimentalement que la taille idéale de la fenêtre est de 3×3 blocs chose que nous avons considérée dans nos expérimentations.

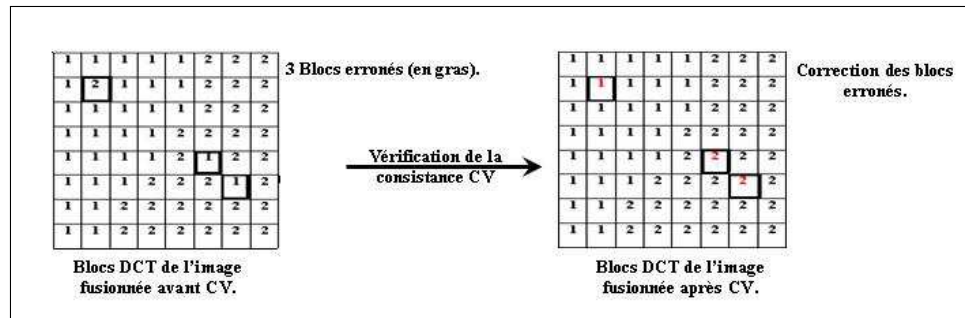


Figure 4.14 – Processus de vérification de la consistance.

6. **Envoi des images fusionnées au niveau supérieur**

Après que l'image ait été contrôlée, elle est envoyée au niveau supérieur pour un autre éventuel traitement. Par exemple on peut procéder à une extraction d'objets ou de contours (features extraction), on peut aussi intégrer les images fusionnées dans un processus de prise de décision. Cela reste dépendant de l'application en question.

7. **Algorithme (niveau Agent de fusion)**

L'algorithme de ce niveau (niveau agent de fusion) se résume dans le tableau 4.3.

```

*** Algorithme Agent de fusion ***

Begin

- IrefAF[i] ← Irefi, // Initialisation de l'image de référence,
- Recept(TGsi), // Réception des seuils globaux envoyés par les capteurs Ss,
- Calculate(TFi), // Calcul du seuil de fusion TFi à partir des seuils globaux TGsi,
- If(TF ≥ TFmax // Dans le cas où TFi est significatif,
  + Request_Data_Sending(), // Réclame l'envoi de données (blocs),
  + UpdateIrefAF[i], // Mettre à jour l'image de référence,
  + ImageFusion(SourcesImages), // Déclencher le processus de fusion,
  + SendFusImgToSupLevel(FusImg), // Envoi de l'image au niveau supérieur.

End
    
```

Tableau 4.3 – Algorithme niveau Agent de fusion

4.4.4 **Description de la solution au niveau décision et traitement**

– **Evaluation de la qualité de fusion :**

Le processus de contrôle de qualité de l'image fusionnée donne une idée sur l'efficacité de la technique utilisée. Ce contrôle est réalisé par le centre de traitement

et de décision situé à la couche supérieure de notre architecture. Comme déjà expliqué dans la section *Evaluation des images fusionnées* du troisième chapitre, deux classes de techniques d'évaluation d'images existent. La première classe exige la présence d'une image de référence sur laquelle l'image fusionnée est évaluée et la seconde classe utilise des métriques d'évaluations. Dans notre cas, nous pouvons utiliser des image de référence et exploiter aussi des métriques afin d'évaluer les image fusionnées. Dans [5], l'auteur a utilisé des métriques pour évaluer les performances de son algorithme de fusion, comme la métrique de Petrovic ($Q^{AB/F}$). Dans notre cas, nous utilisons cette métrique ainsi que d'autres afin de comparer nos résultats avec les résultats obtenus par [5]. Plus de détails sont donnés dans le chapitre suivant concernant le processus d'évaluation.

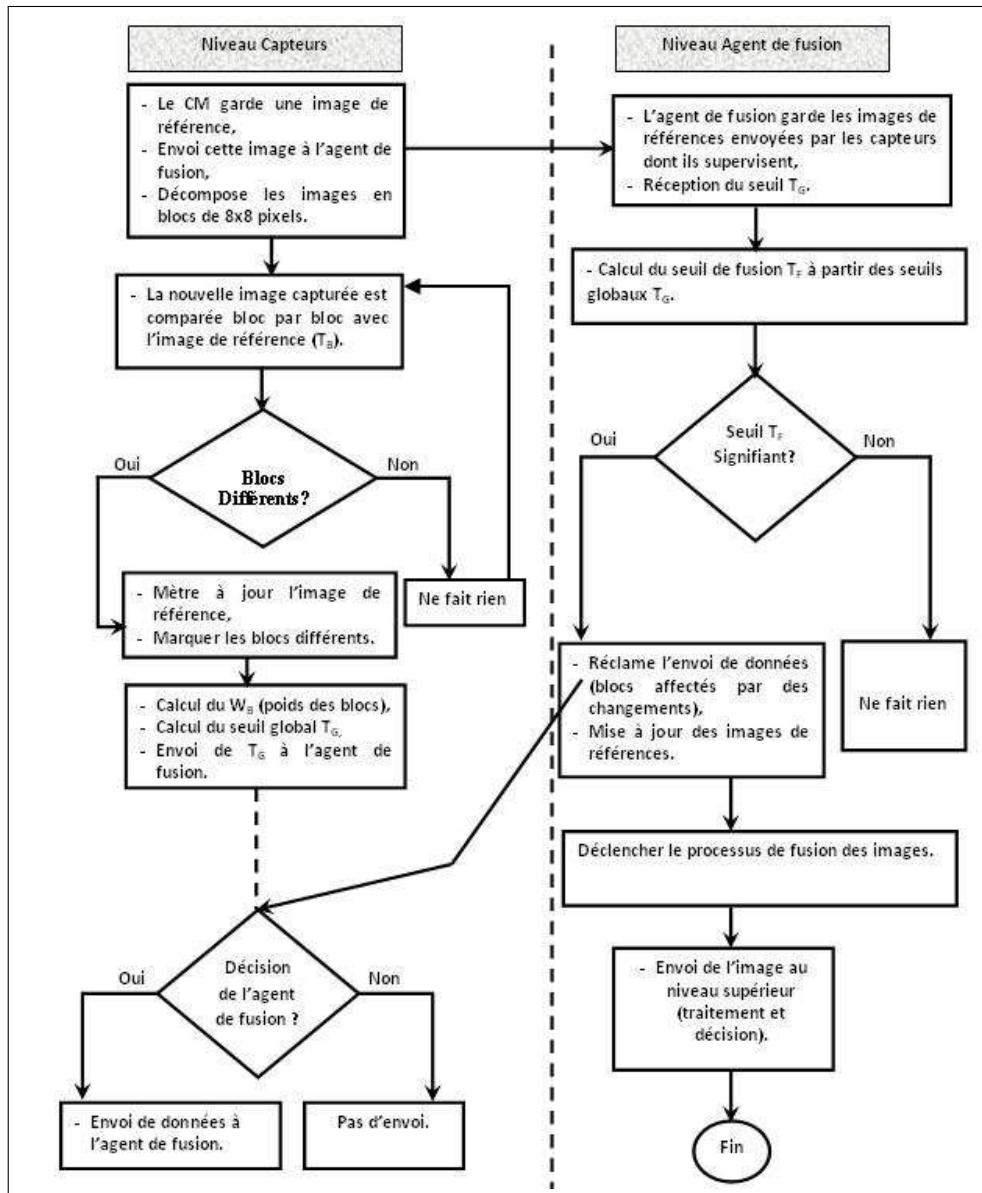


Figure 4.15 – Diagramme de la solution proposée.

4.4.5 Protocole de routage

Afin de s'assurer une bonne qualité de service et une durée de vie du réseau, plusieurs facteurs doivent être prises en considération. Théoriquement, la fusion de données permet de réduire la quantité d'énergie consommée par les différents nœuds du réseau en minimisant le nombre et le volume de paquets à envoyer. Pratiquement, une bonne technique de fusion de données ne suffit pas pour atteindre cet objectif, l'existence d'un protocole fiable de routage est primordiale.

La plupart des applications des réseaux de capteurs multimédia sans fil présentent de nombreux défis, tels que : le routage et la sélection des chemins optimaux, l'adaptation du débit audio/vidéo, la gestion de la congestion réseau et la garantie de la qualité de service [75], entre autres.

Dans la littérature, plusieurs protocoles de routage ont été développés pour les WMSN. La plupart de ces protocoles favorisent l'établissement de plusieurs chemins entre le nœud source et le nœud destination afin d'augmenter la fiabilité de la transmission des flux de données. Parmi ces protocoles de routage nous citons : le **TPGF** (Two-Phase Geographical Greedy Forwarding), le **GPSR** (Greedy Perimeter Stateless Routing) et le protocole **AGEM** (Adaptive Greedy-Compass Energy-Aware Multipath).

Le protocole TPGF [76] est considéré comme étant le premier à introduire le concept de multi-chemins dans les WMSN. Cet algorithme se concentre sur l'exploration et l'établissement d'un nombre maximum de meilleures routes disjointes en termes de délai de bout-en-bout et d'énergie consommée le long d'une route. La première phase de l'algorithme explore les chemins possibles vers la destination. La deuxième phase est responsable de l'optimisation des chemins découverts. L'algorithme TPGF peut être exécuté à plusieurs reprises pour découvrir plusieurs chemins disjoints [75]. Le TPGF est un protocole qui n'est pas tolérant aux pannes qui peuvent se présenter lors de la transmission des données.

Le protocole GPSR [77] a été, à l'origine, conçu pour les réseaux Ad-Hoc mobiles (MANET) mais il a été rapidement adapté pour les réseaux de capteurs (WSN). Ce protocole utilise les coordonnées des nœuds pour le routage. Il est basé sur deux stratégies différentes pour le routage : Transmission Gloutonne (Greedy Forwarding) et Transmission de Périmètre (Perimeter Forwarding). Avec le mode de transmission gloutonne, quand un nœud reçoit un paquet, il choisit le voisin le plus proche de la destination pour lui transmettre le paquet. Parfois, le nœud se trouve plus proche de la destination que tous ses voisins mais la destination finale reste inaccessible en un seul saut. Le nœud est, dans ce cas, face à un "trou". Les trous se forment à cause du déploiement aléatoire, de la présence d'obstacles radio, ou de la défaillance des nœuds. Pour palier à ce problème, la transmission de périmètre est utilisée pour router le paquet autour du trou. Les paquets se déplacent, alors, autour du trou jusqu'à arriver à un nœud plus proche de la destination. A partir de là, la transmission gloutonne reprend la main. Le principal inconvénient de ce protocole est que le routage glouton choisit toujours le meilleur voisin. Par conséquent, en utilisant le même chemin, le protocole GPSR conduit à une défaillance prématurée des nœuds constituant le chemin préféré [75].

Le protocole AGEM [75] représente une version améliorée du protocole GPSR afin de supporter la transmission des flux de données dans les WMSN. L'idée de base est de rajouter une fonctionnalité d'équilibrage de la charge pour l'algorithme GPSR

dans le but de maximiser la durée de vie du réseau. Avec le protocole AGEM, les flux de données vont être routés par différents chemins. A chaque saut, un nœud décide à travers quel voisin il transmet le paquet. La politique de routage est basée sur quatre (04) paramètres : (1) l'énergie restante au niveau de chaque voisin, (2) le nombre de sauts effectués par le paquet jusqu'ici, (3) la distance actuelle qui sépare le nœud de ses voisins et (4) l'historique des paquets déjà routés appartenant au même flux. Le protocole AGEM a deux modes opératoires, un routage glouton intelligent et un routage en marche arrière. Le premier mode est utilisé quand il y a toujours un voisin plus proche de la destination que le nœud actuel, alors que le second est utilisé pour contourner les trous.

*Dans notre cas, nous optons pour le protocole **TPGF** vu l'exploitation de la notion de multi-chemins afin d'effectuer un équilibrage de la charge ce qui représente un avantage pour ce protocole. Malgré que le protocole AGEM est plus efficace, sa mise en oeuvre est lourde et sa complexité reste plus élevée. Cette complexité pourrait avoir un effet inverse comparativement aux bénéfices de routage. Enfin, notre choix peut être également motivé par le fait que TPGF est déjà implémenté au niveau de la plateforme de supervision des réseaux de capteurs de type Imote2, ce qui a facilité nos tests de simulation. Toutefois, ce protocole est considéré non tolérant aux pannes car le nœud voisin est sélectionné lors de l'exploration du chemin et non pas au moment de la transmission de données.*

4.5 Organigramme de la solution

Dans ce qui suit nous synthétisons le fonctionnement de la solution proposée. Le processus se déclenche par la capture d'une image au niveau de chaque capteur, l'ensemble de ces images constitue ce que nous avons appelé " **images sources** ". Chaque image source est découpée en blocs de 8×8 et par la suite elle est comparée avec une image de référence stockée au niveau du même capteur, cette comparaison se fait bloc par bloc et à la fin de cette comparaison on obtient l'ensemble de blocs qui ont subi des changements. Cet ensemble de blocs sera envoyé à l'agent de fusion (après que ce dernier décide de l'envoi sur la base du calcul du coefficient T_F^i) au lieu de l'envoi de l'image entière. A la réception des blocs, l'agent de fusion procède directement à la mise à jour de l'image de référence stockée à son niveau (remplacement des anciens blocs par les nouveaux), il calcule les coefficients DCT de chaque bloc d'images sources. L'image fusionnée est construite à partir des blocs DCT . Avant d'appliquer la transformée inverse $IDCT$ pour obtenir l'image finale en mode spatial, une opération de vérification de la consistance est déclenchée afin d'éliminer un éventuel bruit. A la fin et pour évaluer la qualité de l'image fusionnée certains paramètres d'évaluation sont calculés. L'ensemble

de ces actions est illustré sur la figure 4.16.

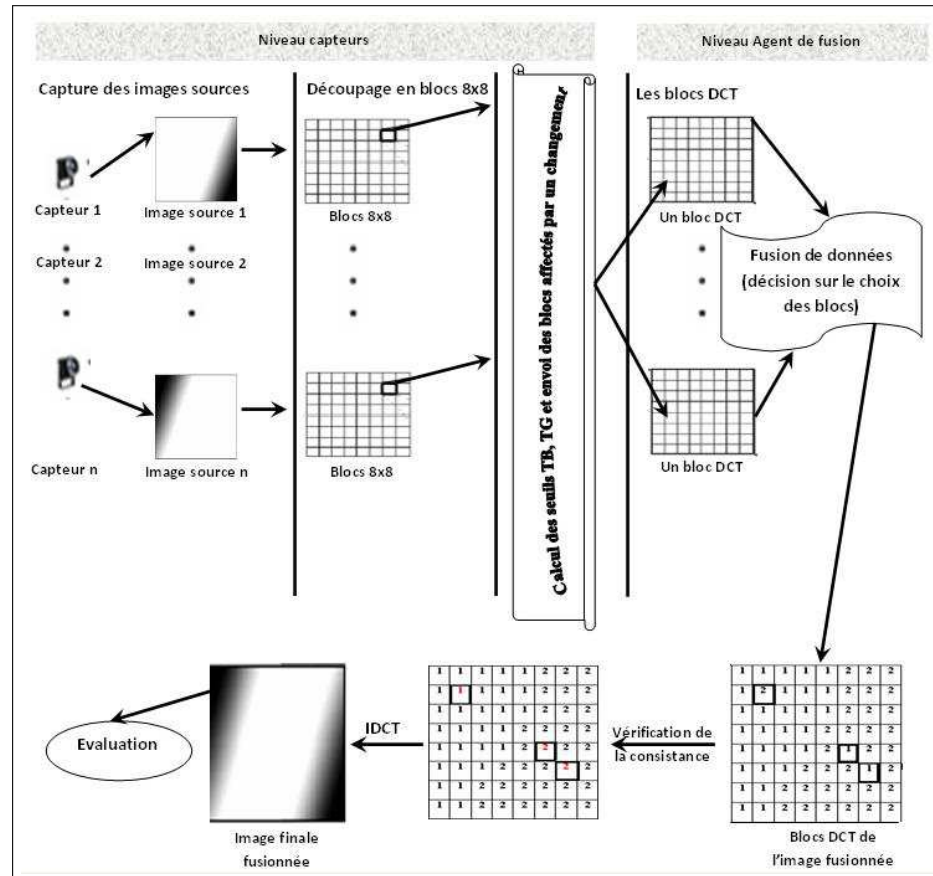


Figure 4.16 – Processus global de la solution.

4.6 Conclusion

La durée de vie d'un réseau de capteurs dépend directement de la durée de vie des batteries de ses nœuds. La consommation d'énergie dans un réseau de capteurs représente l'un des axes de recherche les plus importants dans ce domaine. Dans ce chapitre, nous avons proposé une architecture basée sur trois (03) couches afin de réduire la quantité de données circulant sur le réseau, et par conséquent, la consommation d'énergie. Nous avons décrit le fonctionnement de chaque couche en expliquant les mécanismes utilisés. Nous nous sommes intéressés à une technique de fusion pour un type bien déterminé d'images (Multi-Focus Images) basée sur la *DCT* et combinant plusieurs outils de fusion tels que l'énergie d'un bloc. Dans le prochain chapitre nous validons l'approche proposée à travers différentes simulations et comparons ses performances avec d'autres solutions.

Chapitre 5

Tests et validation

5.1 Introduction

L'objectif de notre contribution est de réduire la consommation des ressources du réseau tout en garantissant des images fusionnées de bonne qualité. Pour évaluer une telle solution, il n'est pas toujours possible d'effectuer une implémentation directe sur le matériel en raison du coût élevé. En contre partie, la simulation permet de tester et de valider à moindre coût certains aspects de notre solution. L'objectif d'une telle simulation est de prévoir le comportement de notre réseau en fonction de plusieurs facteurs avant de passer à un déploiement proprement dit.

Dans le présent chapitre nous présentons la démarche visant à simuler les propositions exprimées dans le chapitre précédent. Notre simulation s'articule sur deux volets : Le premier volet concerne l'évaluation du comportement du réseau en termes de consommation de ressources. Dans le second volet, nous évaluons le rendu visuel de la technique de fusion proposée.

5.2 Démarche adoptée pour la simulation

Dans ce qui suit nous expliquons brièvement la démarche adoptée pour simuler les approches proposées. Les détails de cette démarche seront donnés dans la section *Description détaillée de la simulation*.

5.2.1 Simulation de la couche capteur

Pour montrer l'efficacité de la politique d'envoi de données, nous évaluons le comportement du réseau au niveau de la couche capteur en deux scénarios différents :

- En premier lieu, nous estimons le temps d'acheminement des données d'une image du noeud capteur vers le noeud de fusion ainsi que l'énergie consommée par l'ensemble de capteurs du réseau. Lors de ce premier scénario nous assumons l'envoi de la totalité de l'image après chaque capture. Autrement dit, le capteur procède d'abord au découpage de l'image capturée en blocs de 8×8 pixels avant de l'envoyer directement à l'agent de fusion. Si nous avons une image de taille 600×400 (3750 blocs de 8×8 pixels), l'ensemble de ces blocs sera envoyé à l'agent de fusion par chaque capteur et à chaque fois que le capteur procède à une capture. Cette technique a été adoptée par l'ensemble de méthodes de fusion basées sur le *Multi-Focus*.
- Dans le deuxième scénario, nous estimons les mêmes paramètres que dans le premier scénario mais en considérant l'approche que nous avons proposée. Dans cette optique, nous envoyons uniquement les blocs affectés par des changements au lieu d'envoyer la totalité de l'image.

5.2.2 Evaluation de la qualité des images au niveau couche agent de fusion

Au niveau de la couche fusion, chaque agent de fusion procède à la fusion des images envoyées par l'ensemble de ses capteurs. L'objectif est d'obtenir une image fusionnée plus significatif que les images sources capturées par chaque capteur. Dans cette partie nous évaluons l'efficacité de la technique de fusion d'images que nous avons proposée en considérant différentes métriques (moyenne, écart type, variance, énergie,...). Nous évaluons par la suite la qualité de l'image fusionnée en calculant certains paramètres (Métrique de Petrovic, Ecart type, PSNR). Pour ce besoin, il est important de signaler que nous avons utilisé deux types d'images multi-focus :

- Il s'agit d'un ensemble d'images multi-focus produites artificiellement à l'aide d'un flou gaussien. Ce type d'images a été utilisé par l'ensemble des techniques de fusion d'images multi-focus citées dans l'état de l'art.
- Le deuxième échantillon d'images multi-focus que nous avons utilisé dans notre simulation est constitué d'images réelles capturées par une camera professionnelle.

5.3 Outils de simulation

Afin de tester le comportement de notre solution, nous avons opté pour l'environnement **TinyOS** en ce qui concerne le volet simulation du réseau. Pour l'évaluation de la technique de fusion, nous utilisons l'outil **Matlab**. Par ailleurs, nous avons implémenté une interface graphique (GUI) en langage de programmation **C#.net** qui permet d'exécuter les procédures que nous avons nous même développées en faisant appel aux fonctions de la bibliothèque de **Matlab**.

L'ensemble de procédures et de fonctions que nous avons élaborées dans le cadre de notre simulation ont été implémentées en **Matlab**, **C#.net** et **Tossim/TinyOS** sur un micro portable **FUJITSU SIEMENS ESPRIMO** dont les caractéristiques sont les suivantes :

- Processeur : Intel Core2Duo T1800, 2.0 Ghz,
- RAM : 2 GO,
- OS : Windows XP professionnel SP3.

5.3.1 TinyOS

TinyOS est un système d'exploitation conçu spécialement pour les réseaux de capteurs sans fils. Contrairement aux autres OS, ce système d'exploitation est destiné à un type de matériel bien déterminé connu sous le nom de capteurs. La contrainte d'énergie est pris en compte par l'adoption d'un fonctionnement orienté événement qui permet la mise en veille du capteur en cas d'absence d'un événement à gérer. Un événement qui n'est qu'une interruption matérielle est toujours pris en charge dès son apparition. Cela permet de gérer efficacement les opérations de communications et de capture [78].

Pour mettre en pratique *TinyOS*, les concepteurs ont utilisé un langage de programmation événementiel très proche au langage C, appelé NesC (*Network embedded system C*), avec lequel une application se voit comme une association de plusieurs composants logiciels communicants entre eux. Ces composants représentent des éléments matériels (*Leds, timer, ...etc.*) ou des éléments abstraits situés dans des niveaux plus haut. Le compilateur de nesC connecte et compile les composants en une seule unité. La liaison entre les différents composants d'une application est réalisée via des *interfaces* et elle est décrite par un composant spécial appelé *configuration*. Une *interface* est un ensemble de fonctions décrivant les communications existantes entre deux composants. Ces communications sont modélisées sous forme de *commandes* implémentées par l'un des deux composants, appelé "fournisseur de l'interface", et des *événements* implémentés par

l'autre composant, appelé "usager de l'interface" [79].

Pour simplifier la mise en place d'une application NesC, des composants de base qui se situent généralement juste au dessus de la couche matérielle sont mis à la disposition des développeurs. Ces composants système donnent le premier niveau d'abstraction de la couche matérielle en fournissant les fonctionnalités de base comme la capture, ou la communication à un saut. La tâche du développeur consiste à définir ses propres modules et de les relier avec des composants existants en utilisant les interfaces adéquates.

Pour notre cas, nous avons fait des liens avec plusieurs composants de base fournis par la bibliothèque de TinyOS 2.x. Nous citons essentiellement ceux qui permettent d'accomplir certaines fonctionnalités à savoir :

Effectuer des communications à un saut : Pour pouvoir envoyer un message à un voisin il faut réserver le tampon d'envoi, le remplir et préciser la nature d'envoi (*unicast* ou *broadcast*) et l'identifiant de la destination. Ces fonctions sont à la charge du composant *ActiveMessageC*. Une fois le buffer d'envoi est préparé, le message peut être envoyé par un simple appel à la fonction *send* fournie par le composant *AMSenderC* qui permet de gérer efficacement l'envoi des messages selon leurs types. Coté receveur, la détection de réception d'un message est possible via l'évènement *receive* signalé par le composant *AMReceiverC*. Pour pouvoir analyser le contenu d'un message reçu, l'utilisation du composant *ActiveMessageC* est toujours nécessaire.

Manipuler l'horloge locale et les timer : Certaines tâches, comme l'envoi des messages de positionnement (beacon), sont planifiées à s'exécuter d'une façon périodique. Cette planification nécessite l'utilisation du composant *TimerMilliC* qui donne la possibilité de lancer un timer et exécuter des tâches à son expiration. Pour récupérer la date locale au niveau d'un capteur, on utilise le composant *LocalTimeMilliC*.

Calculer l'énergie résiduelle : Dans le cas d'une implémentation matérielle, il existe des composants système permettant de déduire l'énergie résiduelle du capteur. Pour *Imote2*, le composant à utiliser à cet effet est *DemoSensorC*.

Pour pouvoir simuler le réseau, nous avons utilisé le simulateur de *TinyOS* nommé TOSSIM (*TinyOS SIMulator*). Nous l'avons choisi particulièrement à cause de sa fiabilité. En effet, *TOSSIM* est capable de capturer les événements survenus dans la totalité du réseau sur la base du code source *NesC* chargé dans les nœuds de la scène de simulation [80]. Pour interagir avec TOSSIM, il est nécessaire d'utiliser un script python ou C++. Ce script sert à insérer les paramètres globaux de la simulation comme la

scène et la durée de simulation. On peut aussi utiliser ce script pour injecter des points d'observation permettant de repérer des événements cibles et récupérer les résultats de simulation.

5.3.2 Matlab

Le MATLAB pour **MAT**rix **LAB**oratory, est une application qui a été conçue afin de fournir un environnement de calcul matriciel simple, efficace, interactif et portable, permettant la mise en oeuvre des algorithmes développés dans le cadre des projets *linpack* et *eispack* [74]. MATLAB est constitué d'un noyau relativement réduit, capable d'interpréter puis d'évaluer les expressions numériques matricielles qui lui sont adressées :

- Soit directement au clavier depuis une fenêtre de commande ;
- Soit sous forme de séquences d'expressions ou scripts enregistrées dans des fichiers texte appelés *m-files* et exécutées depuis la fenetre de commande ;
- Soit plus rarement sous forme de fichiers binaires appelés *mex-files* ou fichiers *.mex* générés à partir d'un compilateur **C** ou **fortran**.

Dans notre cas, nous avons utilisé quelques fonctions de la bibliothèque de Matlab à savoir la fonction **DCT2** pour le calcul de la DCT d'un bloc de données, **IDCT2** pour le calcul de la *DCT inverse*, la fonction **CORR2** pour corrélérer deux blocs de données et d'autres fonctions comme *imread*, *imwrite*, *rgb2gray*,...

Pour simplifier l'exécution de notre programme Matlab, nous avons jugé utile le développement d'une interface entre l'utilisateur et le programme développé sous Matlab, en langage *C#.net* comme illustrée sur la figure 5.1.

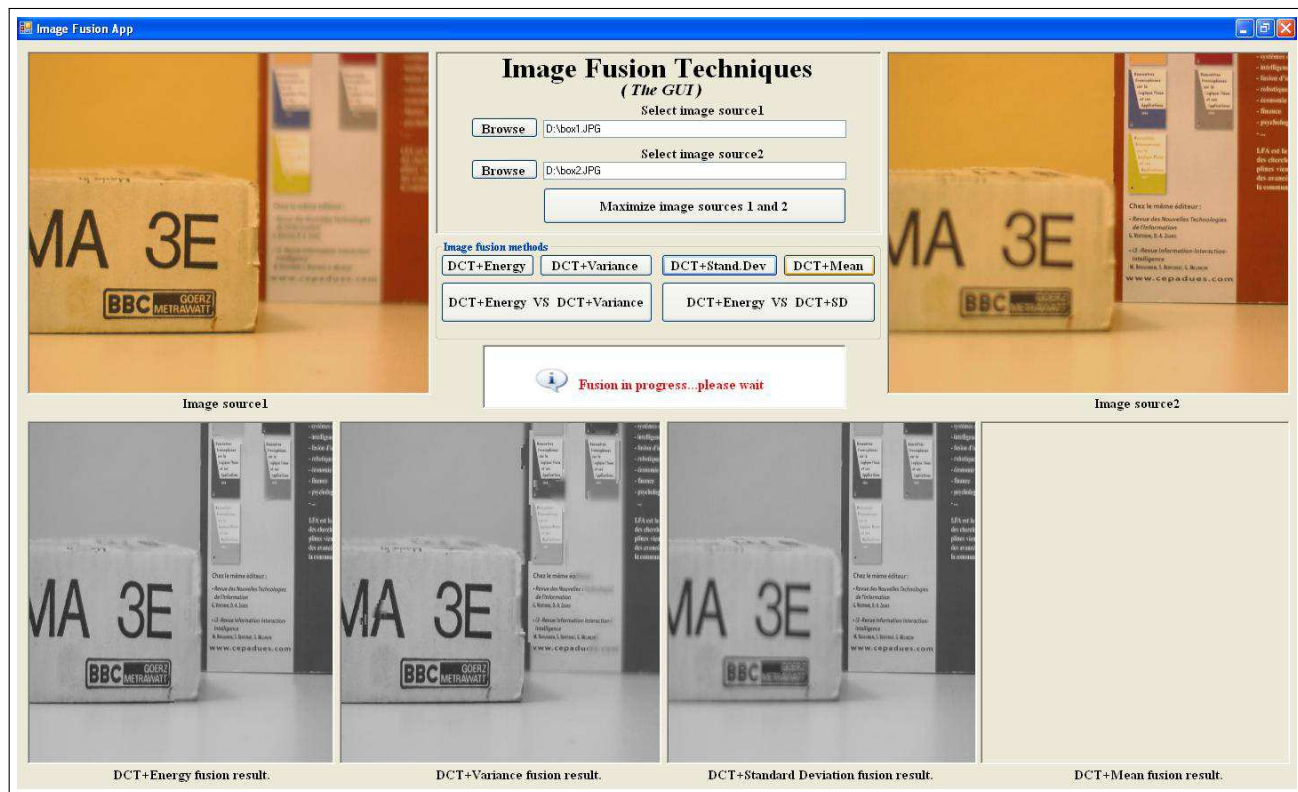


Figure 5.1 – GUI développée pour tester les différentes techniques de fusion.

5.4 Description détaillée de la simulation

Dans cette section, nous abordons d'une manière détaillée la démarche adoptée pour la simulation de notre approche.

5.4.1 Simulation de la couche capteur

1. **Premier scénario** : Pour cette simulation nous considérons l'échantillon d'images donné par la figure 5.2 comme étant des images capturées par les capteurs constituant notre réseau. Nous estimons d'abord l'énergie consommée et le temps nécessaire à l'acheminement de la totalité de l'image pour l'ensemble de capteurs vers l'agent de fusion. La taille de chaque image utilisée dans cette simulation ainsi que le nombre de blocs 8×8 pixels constituant l'image sont données par le Tableau 5.1. Dans ce cas le nombre total de blocs de chaque image est envoyé à l'agent de

fusion par le capteur concerné.

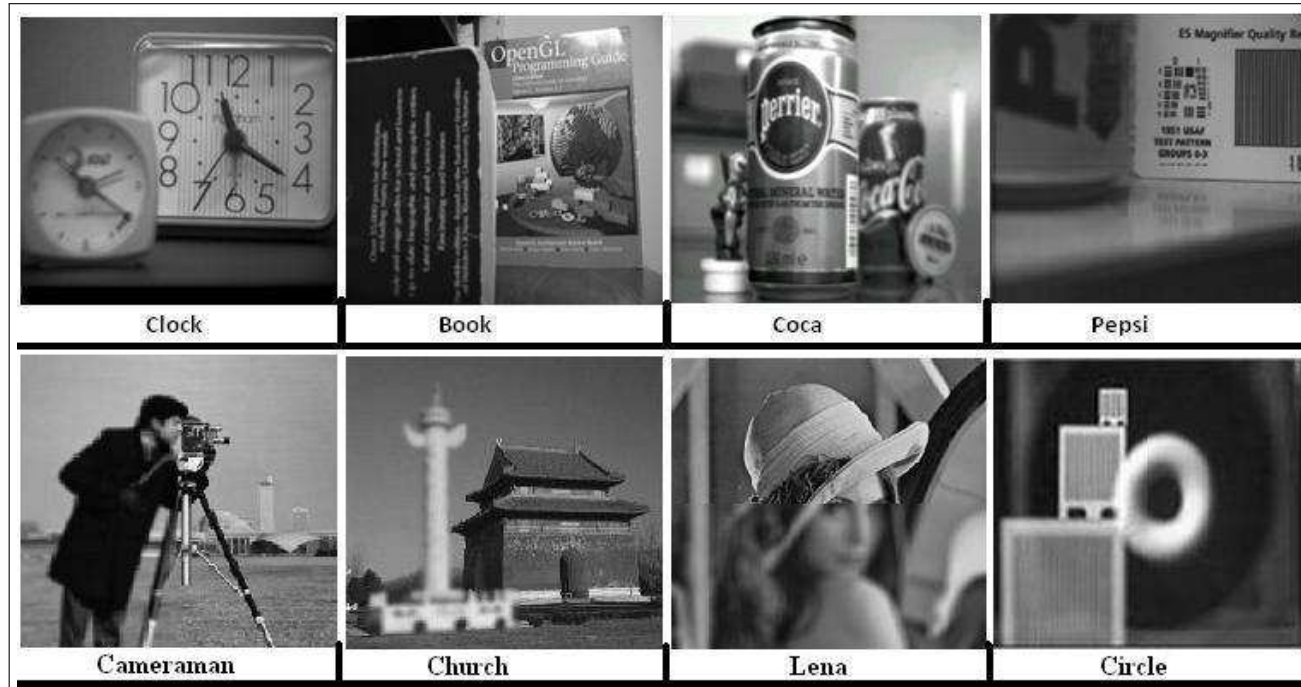


Figure 5.2 – Images utilisées pour la simulation du premier scénario.

- - -	Clock	Book	Coca	Pepsi	Cameraman	Church	Lena	Circle
Taille (Dimension)	512 × 512	1280 × 960	1181 × 960	404 × 404	612 × 612	512 × 512	512 × 512	512 × 512
Nbr de blocs (8 × 8 pixels)	4096	19200	17715	2550	5852	4096	4096	4096

Tableau 5.1 – Taille et nombre de blocs 8 × 8 des images itulisées pour la simulation du premier scénario.

- L'énergie totale consommée :

Pour cette simulation, l'énergie initiale de chaque capteur a été initialisée à **2400mAh**. Cette valeur est motivée par le fait qu'un capteur *IMote2* (que nous disposons au niveau de notre laboratoire), est alimenté par trois batteries de type AAA 1.2V de **800mAh** chacune, ce qui donne un total de **2400mAh**. L'énergie consommée d'un réseau comportant N nœuds capteur est calculée par la formule 5.1 :

$$E_{consum} = E_{init} - E_{remain} \quad (5.1)$$

avec : $E_{init} = 2400 \times N$ et E_{remain} sera donné par le simulateur Tossim.

Nous avons simulé la consommation d'énergie sur une topologie réseau qui comporte **50 nœuds** dont le nœud ayant pour **ID=1** est considéré comme étant le nœud source (capteur) et le nœud ayant pour **ID=50** comme étant l'agent de fusion (voir figure 5.3). Le but ici est d'estimer uniquement la consommation de l'énergie et le temps d'acheminement des paquets de bout en bout à travers l'envoi des images illustrées sur la figure 5.2 à partir du nœud source (ID=1) vers l'agent de fusion (ID=50). Les résultats obtenus sont donnés par le tableau 5.2, ce dernier représente l'énergie totale consommée par l'ensemble des 50 nœuds lors de l'envoi des différentes images considérées dans ce scénario. Ces résultats reflètent uniquement l'envoi des images par le capteur (ID=1), les autres nœuds capteurs participent à l'acheminement des paquets vers l'agent de fusion. Rappelons que le but ici est d'estimer la consommation de l'énergie et le temps d'acheminement et non pas la qualité de l'image fusionnée.

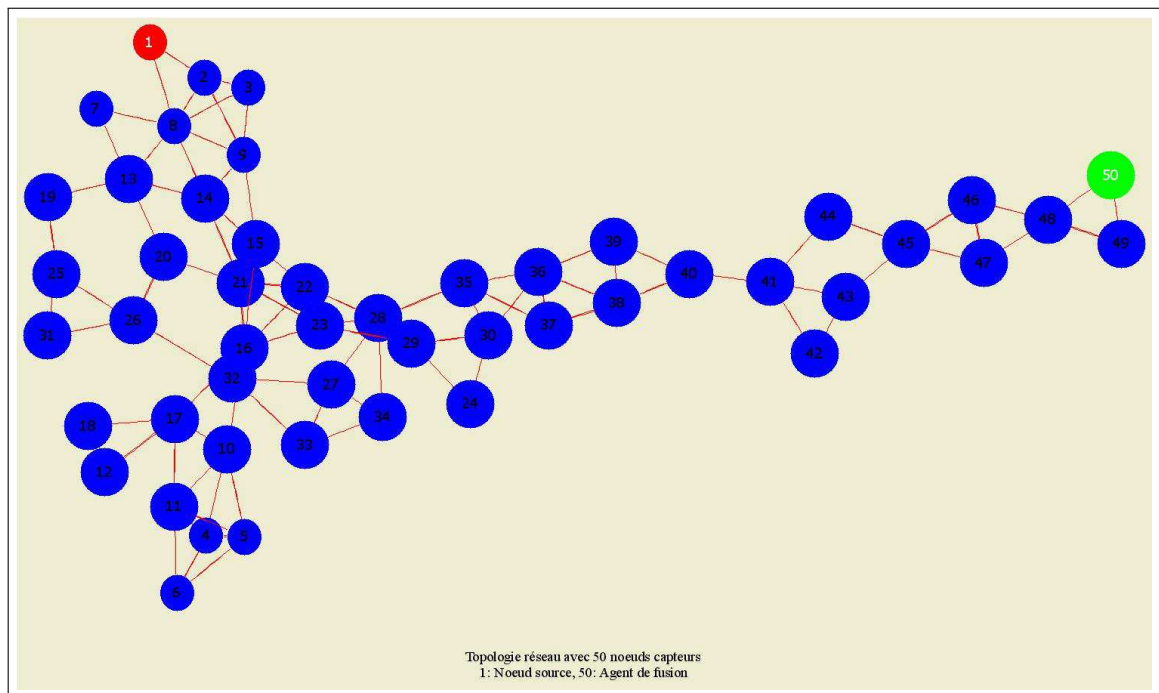


Figure 5.3 – Topologie réseau avec 50 nœuds, 1 :Source et 50 :Agent de fusion.

---	Nbr de blocs total (8 × 8)	Energie totale consommée (en mAh)
Clock	4096	2840.42
Book	19200	13072,76
Coca	17715	12028,12
Pepsi	2550	1753,22
Cameraman	5852	4007.40
Church	4096	2791,58
Lena	4096	2776,80
Circle	4096	2788,88

Tableau 5.2 – Energie consommée lors de l’envoi de l’image entière (la totalité des blocs).

- Temps d’acheminement des blocs :

Pour le temps total d’acheminement des blocs vers l’agent de fusion, les résultats obtenus par le simulateur TOSSIM sont donnés par le tableau 5.3.

---	Nbr de blocs total (8 × 8)	Temps d’acheminement des blocs (en m.s)
Clock	4096	53.6066
Book	19200	244.2569
Coca	17715	225.2743
Pepsi	2550	34.3926
Cameraman	5852	75.1693
Church	4096	53.8826
Lena	4096	53.5790
Circle	4096	53.4231

Tableau 5.3 – Temps total d’acheminement des blocs lors de l’envoi de l’image entière.

- Analyses et commentaires :

D’après les deux tableaux 5.2 et 5.3, il apparaît que la quantité d’énergie consommée et le temps d’acheminement des blocs dépendent directement de la taille de l’image à envoyer, autrement dit la quantité d’énergie consommée et le temps d’acheminement des blocs varient proportionnellement avec le nombre de blocs 8 × 8 pixels constituant l’image. Ces résultats sont par la suite comparés avec les résultats du deuxième scénario.

2. **Deuxième scénario :** Dans le deuxième scénario, nous simulons les mêmes paramètres en considérant notre approche. Durant cette simulation et pour la partie qui comporte la subdivision des images et la comparaison des images bloc par bloc (calcul des taux $T_{B_k}^i$ et T_G^i), ont été obtenues sous Matlab. La fonction qui

subdivise les images en blocs de 8×8 pixels, calcule les taux $T_{B_k}^i$ et T_G^i) et nous donne les indices (ID) des blocs considérés changés est illustrée sur la figure 5.4. Cette dernière possède deux paramètres en entrée (l'image de référence et l'image capturée) et elle nous fournit deux paramètres en sortie (le seuil global T_G^i) et une matrice qui contient les indices des blocs jugés modifiés).

```
function [tGlob, idBlockMat] = imgComp(im_ref, im_cap)

    if nargin ~= 2    % verification du nombre d'arguments
        error('Deux images en entrée sont exigées!!')
    end

    if size(im_ref,3) == 3 % verification si :niveau de gris
        im_ref = rgb2gray(im_ref);
    end
    if size(im_cap,3) == 3
        im_cap = rgb2gray(im_cap);
    end

    if size(im_ref) ~= size(im_cap)    % verification que les images possèdent
    la même taille
        error('la taille des images source doit etre la meme!!')
    end

    % récupération de la taille des images
    [m,n] = size(im_ref);
    %déclaration de tableaux et variables
    corrMat = zeros(m/8,n/8); dcMat=zeros(m/8,n/8);
    wMat=zeros(m/8,n/8); im=0; idBlockMat=zeros(m/8,n/8);
    wCorr=0,sommeW=0,tGlob=0,taille=0;nbBlock=0; bInf_w = 0.0 ; bSup_w = 1.0 ;
    %Affectation des poids aux blocs
    A = bInf_w + (bSup_w-bInf_w).*rand(floor(m/8), floor(n/8));
    sommeW=sum(sum(A)); %calcul de la somme des poids

    % Divise source images into 8*8 blocks and perform the correlation
    for i = 1:floor(m/8)
        for j = 1:floor(n/8)
            im_refSub = im_ref(8*i-7:8*i,8*j-7:8*j);
            im_capSub = im_cap(8*i-7:8*i,8*j-7:8*j);

            %calcul des coefficients DCT des blocs
            im_capSubDct = dct2(im_capSub);
            im_refSubDct = dct2(im_refSub);
            %corrélation de deux blocs 8*8
            im_refCorrim_cap=1-corr2_NaN(im_refSub,im_capSub);
            %constitution de la matrice de corrélation
            corrMat(i,j)=abs(im_refCorrim_cap);
            %matrice des blocs affectés par un changement
            if corrMat(i,j)>0.6
                nbBlock=nbBlock+1;
                idBlockMat(i,j)=1; % marquer le block changé
            end
        end
    end

    %Calcul de la somme (poids*T_B)
    for i = 1:floor(m/8)
        for j = 1:floor(n/8)
            wCorr = wCorr + (corrMat(i,j)*A(i,j));
        end
    end
    tGlob = wCorr/sommeW; %calcul du seuil global
```

Figure 5.4 – Code Matlab de la fonction de comparaison des images(références et capturées)

En premier lieu afin d'illustrer le calcul de T_G^i avec Matlab nous considérons deux cas, la même image comme étant une image de référence et une image capturée à la fois, autrement dit nous simulons le cas où il n'y a pas de changement entre l'image de référence et l'image capturée. Pour cela, nous utilisons l'ensemble des images illustrées sur la figure 5.5. Pour le deuxième cas nous considérons le cas où il y a un changement entre l'image de référence et l'image capturée. L'ensemble des images capturées utilisé dans ce cas est illustré sur la figure 5.6, les images de la figure 5.5 sont considérées images de références dans ce deuxième cas. Afin de simuler le changement entre les images, nous rajoutons un objet dans chacune des images capturées pour voir le comportement de notre algorithme. Le tableau 5.4 donne la valeur du taux T_G^i dans les deux cas.

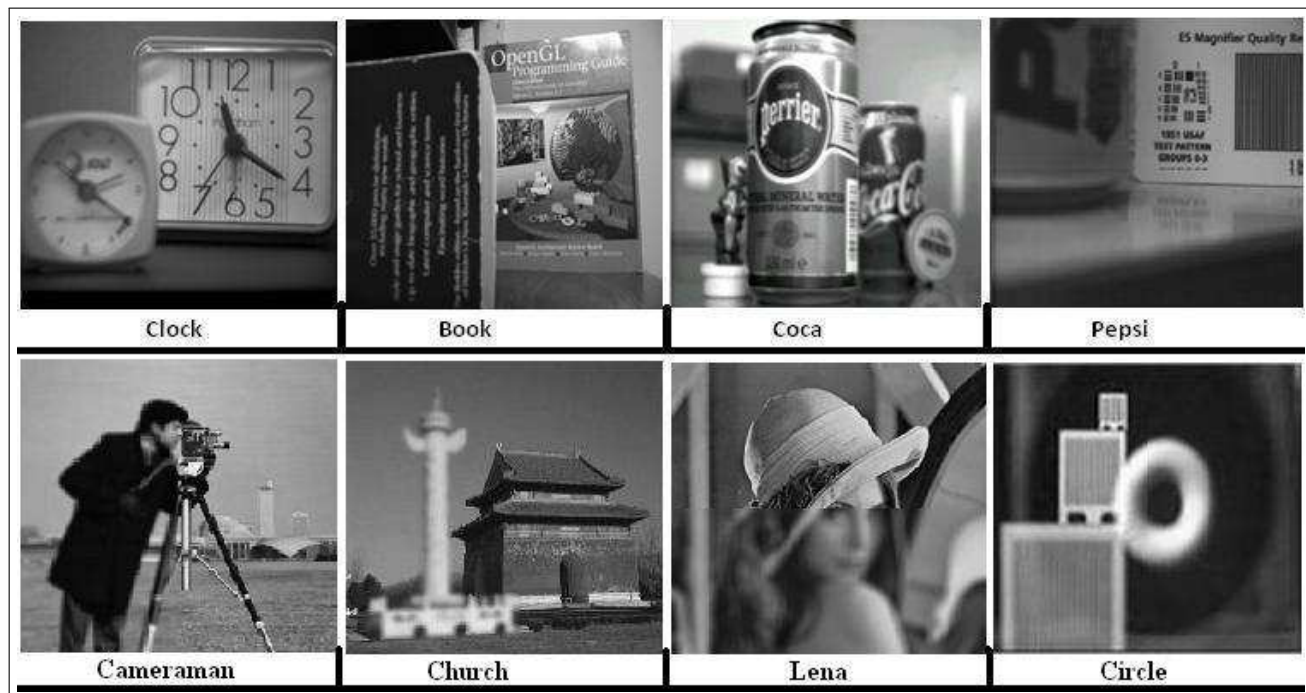


Figure 5.5 – Images utilisées pour la simulation du premier (1^{er}) cas du deuxième scénario.

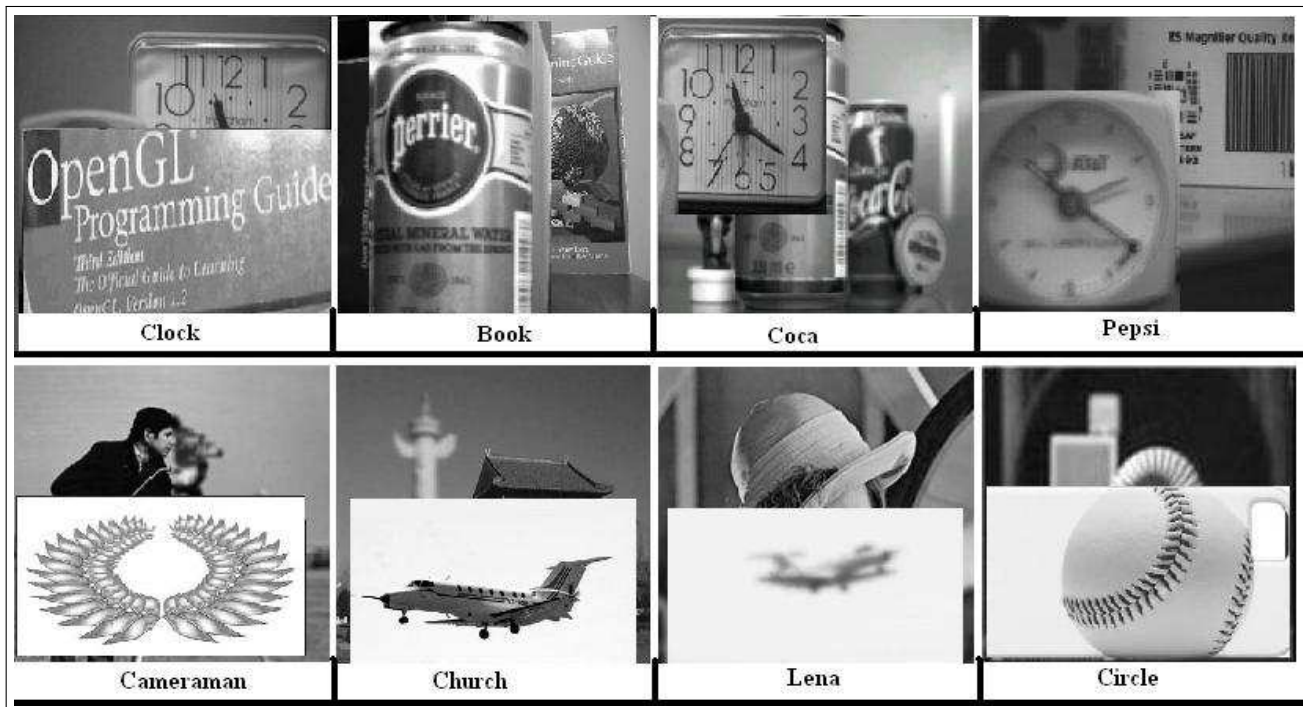


Figure 5.6 – Images utilisées pour la simulation du deuxième (2^{eme}) cas du deuxième scénario.

---	1 ^{er} Cas	2 ^{eme} Cas
Clock	0.0046	0.6416
Book	0.0019	0.6376
Coca	0.0085	0.6393
Pepsi	0.0039	0.6204
Cameraman	0.0061	0.6059
Church	0.0097	0.6568
Lena	0.0003	0.6295
Circle	0.0008	0.6115

Tableau 5.4 – Valeur du seuil T_G^i dans les deux cas de simulation du deuxième scénario.

D'après le tableau 5.4, la fonction qui fournit en sortie le seuil global de comparaison T_G donne de bonnes résultats car ce seuil est proche de **0** pour des images similaires et entre **0.60** et **0.65** pour des images différentes. Sur cette base l'agent de fusion décide de l'envoi des données ou non par le nœud capteur concerné. *En ce qui concerne la valeur de T_B^{max} pour la simulation, (valeur à partir de laquelle les blocs sont considérés non identiques) nous avons effectués plusieurs tests pour voir la variation de cette valeur, à la fin nous avons optés pour un $T_B^{max} = 0.6$. Dans ce cas, si le seuil de différence $T_{B_k}^i$ dépasse **0.6** alors les deux blocs sont considérés non identiques. Sinon ils sont similaires.*

Pour l'estimation de l'énergie consommée ainsi que le temps d'acheminement des

paquets, le même principe de simulation du premier scénario est considéré ici.

En utilisant la fonction *imgComp* (donnée en figure 5.4), avec en entrée les images illustrées sur la figure 5.5 comme étant images de référence et les images illustrées sur la figure 5.6 comme étant les images capturées, nous obtenons en sortie les valeurs de T_G ainsi que les ID des blocs modifiés de chaque image. Par exemple pour l'image nommée **Clock** le nombre de blocs modifiés est de **2722** blocs ce qui représente 66.4% du nombre total de blocs. Ce dernier est égal à $(512/8)*(512/8)=4096$ blocs car l'image est de taille 512×512 . Donc nous avons un gain de 33.6% ou de $4096-2722=1374$ blocs.

Les résultats obtenus pour le reste des images sont donnés par le tableau 5.5, (l'image **Clock** est comprise).

---	Taille ($m \times n$)	Nbr de blocs total (8×8)	Nbr de blocs changés	% de blocs changés	Gain en blocs	Seuil T_G
Clock	512×512	4096	2722	66.4%	1374	0.6416
Book	1280×960	19200	9093	47.35%	10107	0.6376
Coca	1181×960	17715	4740	26.75%	12975	0.6393
Pepsi	404×404	2550	1444	56.62%	1106	0.6204
Cameraman	612×612	5852	2845	48.61%	3007	0.6059
Church	512×512	4096	2065	50.41%	2031	0.6568
Lena	512×512	4096	1826	44.58%	2270	0.6295
Circle	512×512	4096	1920	46.87%	2176	0.6115

Tableau 5.5 – Résultats de la fonction *imgComp* sur un ensemble d'images(références et capturées).

Nous estimons l'énergie consommée et le temps d'acheminement des paquets lors de l'envoi des blocs modifiés (2722, 9093, 4740, 1444,...) pour les images (Clock, Book, Coca, Pepsi,...)de la même manière que dans la simulation du premier scénario.

- L'énergie totale consommée :

En ce qui concerne l'énergie consommée, le Tableau 5.6 illustre les résultats obtenus pour la même topologie du réseau utilisée en scénario 1. Il est a signalé que la valeur T_F^{Max} à partir du quelle l'agent de fusion ordonne l'envoi des données par l'ensemble des capteurs est fixée à 0.6, cela est après plusieurs tests et observations sur les différentes images, dans la réalité cette valeur devra être dynamique en fonction l'énergie résiduelle de l'agent de fusion.

---	Nbr de blocs total (8 × 8)	Energie consommée en mAh (Calculée dans le 1 ^{er} scénario)	Nbr de blocs changés (qui vont être envoyés)	Energie consommée en mAh (Calculée dans le 2 ^{eme} scénario)	Gain en terme d'énergie (en mAh)	Gain d'énergie en (%)
Clock	4096	2840.42	2722	1856,48	983.94	34.64%
Book	19200	13072.76	9093	6195,08	6877.68	52.61%
Coca	17715	12028.12	4740	3217,75	8810.37	73.24%
Pepsi	2550	1753.22	1444	971,95	781.27	44.56%
Cameraman	5852	4007.40	2845	1934,37	2073.03	51.73%
Church	4096	2791.58	2065	1409,32	1382.26	49.51%
Lena	4096	2776.80	1826	1245,58	1531.22	55.14%
Circle	4096	2788.88	1920	1307,30	1481,58	53.12%

Tableau 5.6 – Energie consommée lors de l'envoi de la totalité des blocs, énergie consommée lors de l'envoi des blocs modifiés et le gain en termes d'énergie.

Pour percevoir le gain en terme d'énergie consommée pour les deux scénarios présentés ci-dessus, nous donnons les résultats illustrés sur le Tableau 5.6 sous forme d'histogramme ; voir la Figure 5.7.

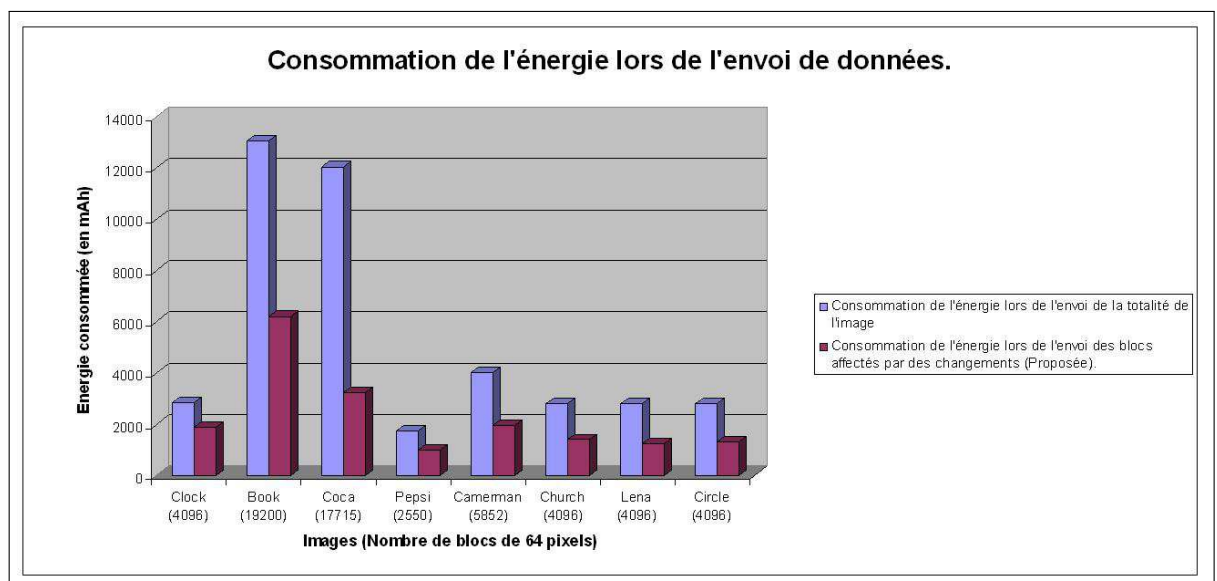


Figure 5.7 – Consommation de l'énergie lors de l'envoi de données.

- Temps d'acheminement des blocs :

Pour le temps total d'acheminement des blocs affectés par des changement vers l'agent de fusion, les résultats obtenus sont donnés par le Tableau 5.7, nous donnons aussi une comparaison avec le premier scénario.

---	Nbr de blocs total (8×8)	Temps d'acheminement des blocs en m.s (Calculée dans le 1 ^{er} scénario)	Nbr de blocs changés (qui vont être envoyés)	Temps d'acheminement des blocs en m.s (Calculée dans le 2 ^{eme} scénario)	Gain en terme de temps d'acheminement (en m.s)
Clock	4096	53.6066	2722	36.5374	17.0692
Book	19200	244.2569	9093	117.0098	127.2471
Coca	17715	225.2743	4740	61.5863	163.688
Pepsi	2550	34.3926	1444	20.1915	14.2011
Cameraman	5852	75.1693	2845	37.9437	37.2256
Church	4096	53.8826	2065	28.4771	25.4055
Lena	4096	53.5790	1826	24.8564	28.7226
Circle	4096	53.4231	1920	26.2106	27.2125

Tableau 5.7 – Temps d'acheminement des blocs lors de : l'envoi de la totalité de l'image, l'envoi des blocs modifiés et le gain en termes de temps.

Pour le temps d'acheminement, nous avons présenté les résultats illustré sur le tableau 5.3 sous forme d'un histogramme comme il est montré sur la figure 5.8.

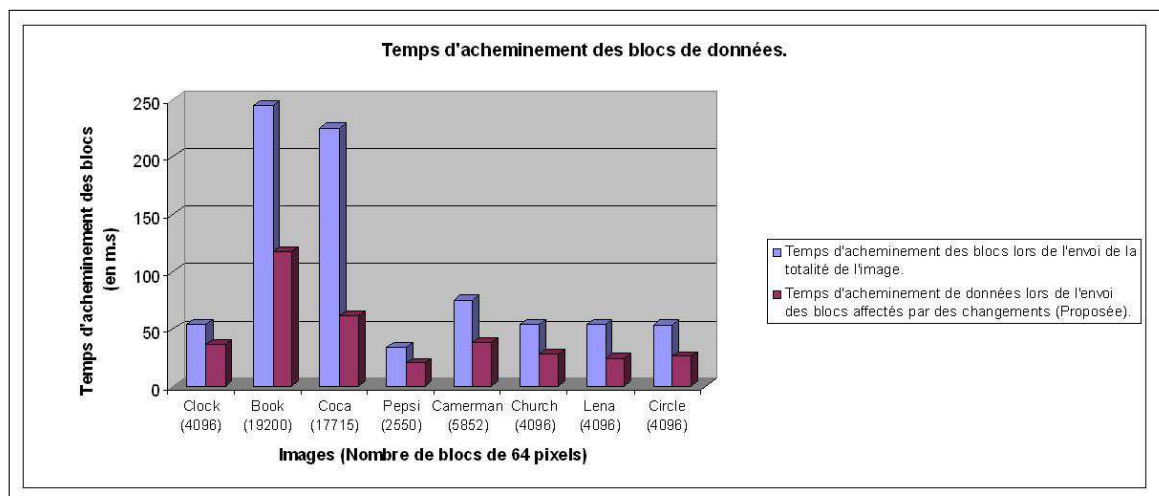


Figure 5.8 – Temps d'acheminement des blocs de données.

- Analyses et commentaires :

- Premièrement, il apparaît clairement d'après le tableau 5.5 que le gain en nombre de blocs est considérable dans tous les cas, il est proportionnel à la taille de l'image. Par exemple pour l'images **book** dont la taille est de 1280×960 , le gain est de **10107**.
- Pour l'énergie consommée, le tableau 5.6 montre que l'approche proposée consomme moins d'énergie par rapport aux approches basées sur l'envoi de la totalité de l'image, ceci et pour l'ensemble des images utilisées dans la simulation. Il est

remarquable aussi à travers ce même tableau que le gain en termes d'énergie est proportionnel au nombre de blocs de l'image que nous voulons envoyer, par exemple pour l'image *Book* constituée de 19200 blocs de 64 pixels, le gain en termes d'énergie était plus de 6877 mAh cela est lors de l'envoi des 9093 blocs affectés par un changement au lieu de l'envoi de la totalité de l'image (Voir la dernière colonne du tableau 5.6).

- Pour le temps d'acheminement des blocs, un gain considérable est aussi enregistré pour l'ensemble des images. Ce gain est proportionnel au nombre de blocs constituant l'image, ceci est remarquable aussi à travers l'histogramme présenté sur la figure 5.8.

5.4.2 Simulation au niveau agent de fusion

C'est à ce niveau que la fusion de données proprement dite est appliquée. Les techniques que nous avons utilisés comme critère de sélection de bloc sont les suivantes :

- *la moyenne et la variance* : qui ont été déjà proposées dans la littérature pour la fusion des blocs DCT.
- L'écart type et l'énergie : que nous proposons pour la fusion des blocs.

Il est à signaler que l'ensemble de ces techniques sera appliqué dans un domaine DCT (en mode fréquentiel et non pas spatial), les résultats de chaque technique sont donnés dans ce qui suit.

- Résultats obtenus

A- Simulation sur des images artificielles

Les quatre techniques ont été testées sur une vingtaine d'images artificielles de différentes tailles et à cause d'un problème d'espace dans ce document, nous avons préféré présenter quelques résultats considérant les images : (*Clock*, *Church*, *Bottle*, *Book*, *Plane*) comme illustré sur les figures ci-après.

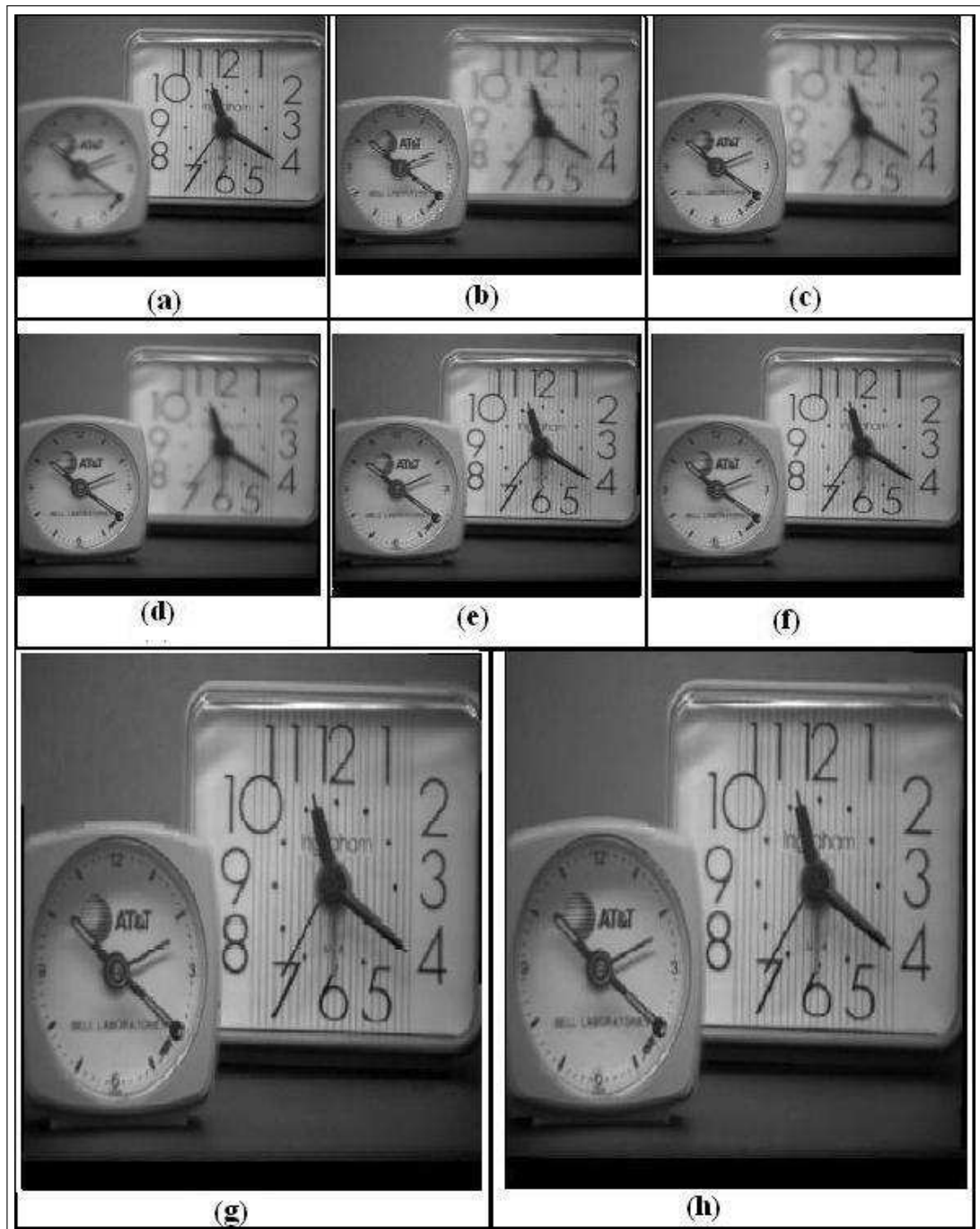


Figure 5.9 – Images sources et images fusionnées pour l'image 'Clock'. (a) : la première image source avec focus à droite. (b) : la deuxième image source avec focus à gauche. (c) : image fusionnée en utilisant la moyenne. (d) : image fusionnée en utilisant l'écart type. (e) : image fusionnée en utilisant la variance. (f) : image fusionnée en utilisant l'énergie d'un block (technique proposée). (g) : la version zoomée de (e). (h) : la version zoomée de (f).

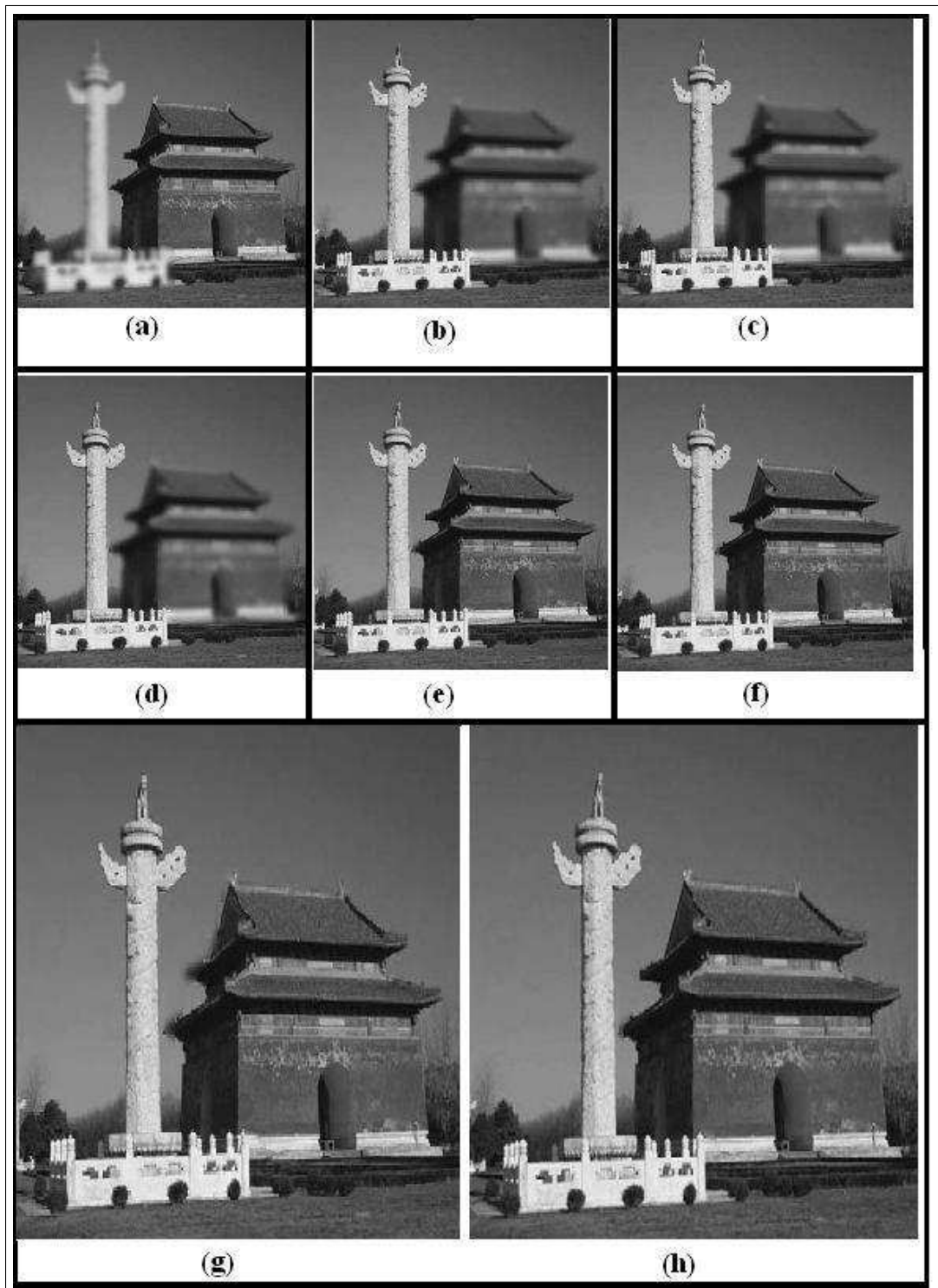


Figure 5.10 – Images sources et images fusionnées pour l'image 'Church'. Même ordre que la figure 5.9

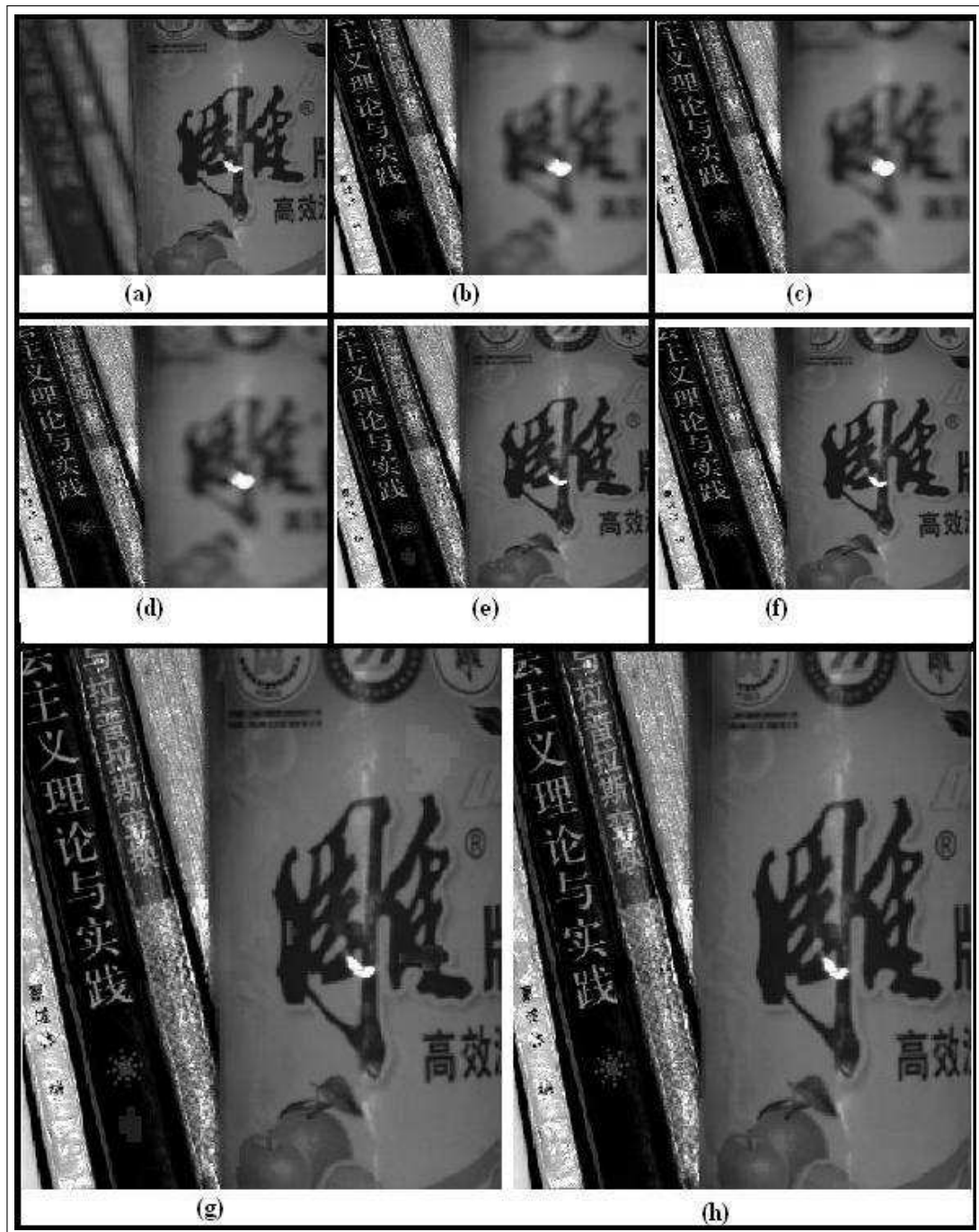


Figure 5.11 – Images sources et images fusionnées pour l'image 'Bottle'. Même ordre que la figure 5.9



Figure 5.12 – Images sources et images fusionnées pour l'image 'Book'. Même ordre que la figure 5.9.

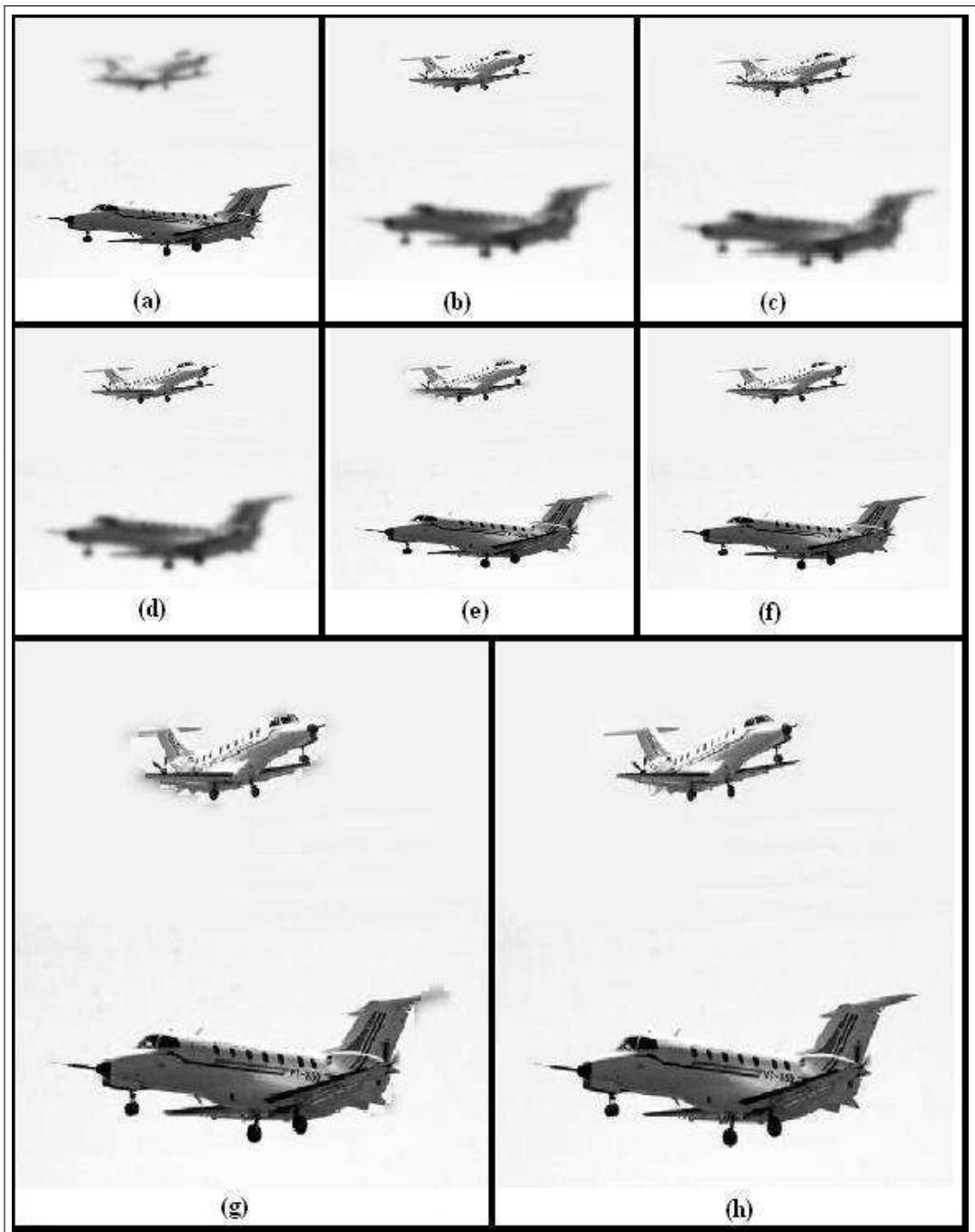


Figure 5.13 – Images sources et images fusionnées pour l'image 'Plane'. (a) : la première image source avec focus en bas. (b) : la deuxième image source avec focus en haut. (c) : image fusionnée en utilisant la moyenne. (d) : image fusionnée en utilisant l'écart type. (e) : image fusionnée en utilisant la variance. (f) : image fusionnée en utilisant l'énergie d'un block (technique proposée). (g) : la version zoomée de (e). (h) : la version zoomée de (f).

B- Simulation sur des images réelles

Pour voir le comportement des différentes techniques de fusion utilisées sur des images réelles, nous avons procédé à la capture de quelques images multi-focus dans le Laboratoire Robotique et Productique de l'Ecole Militaire Polytechnique (voir Figure 5.14). Ces images ont été capturées à l'aide d'une camera SONY de type *DFW-X710* ; voir la Figure 5.15. Les résultats obtenus avec des images réelles sont présentés ci-dessous. Les images sources de chaque cas sont présentées ici en couleur, pour un besoin de présentation. Pour la simulation nous les avons converties en niveau de gris. Les résultats obtenus en utilisant la moyenne et l'écart type ont été médiocres comme c'est le cas pour les images artificielles. Nous présentons ici uniquement les résultats obtenus en appliquant la variance et l'énergie compactée dans un bloc de 8×8 afin de comparer ces deux techniques.

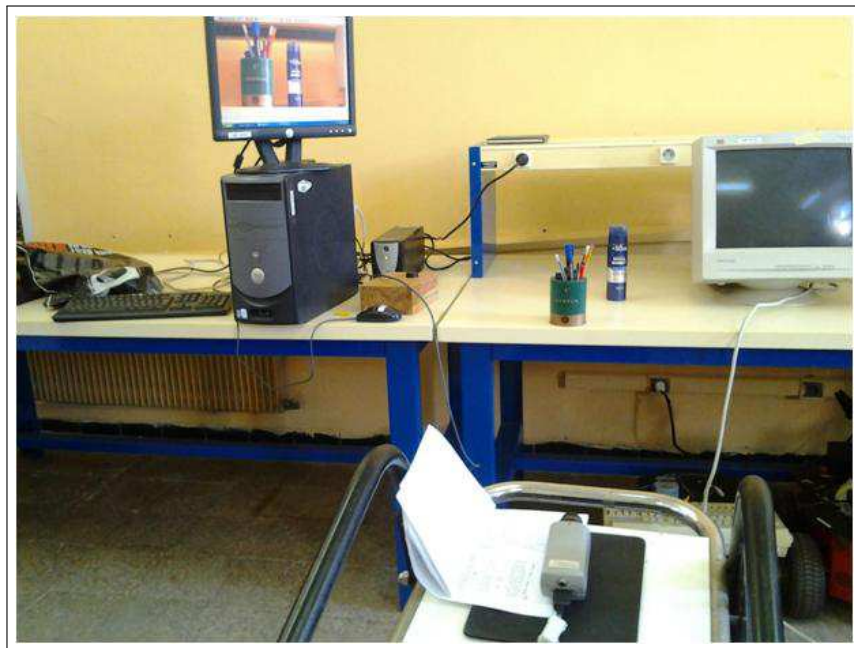


Figure 5.14 – Capture des images multi-focus réelles dans le laboratoire robotique et productique de l'EMP.



Figure 5.15 – Camera SONY de type DFW-X710 utilisée pour la capture des images multi-focus réelles.

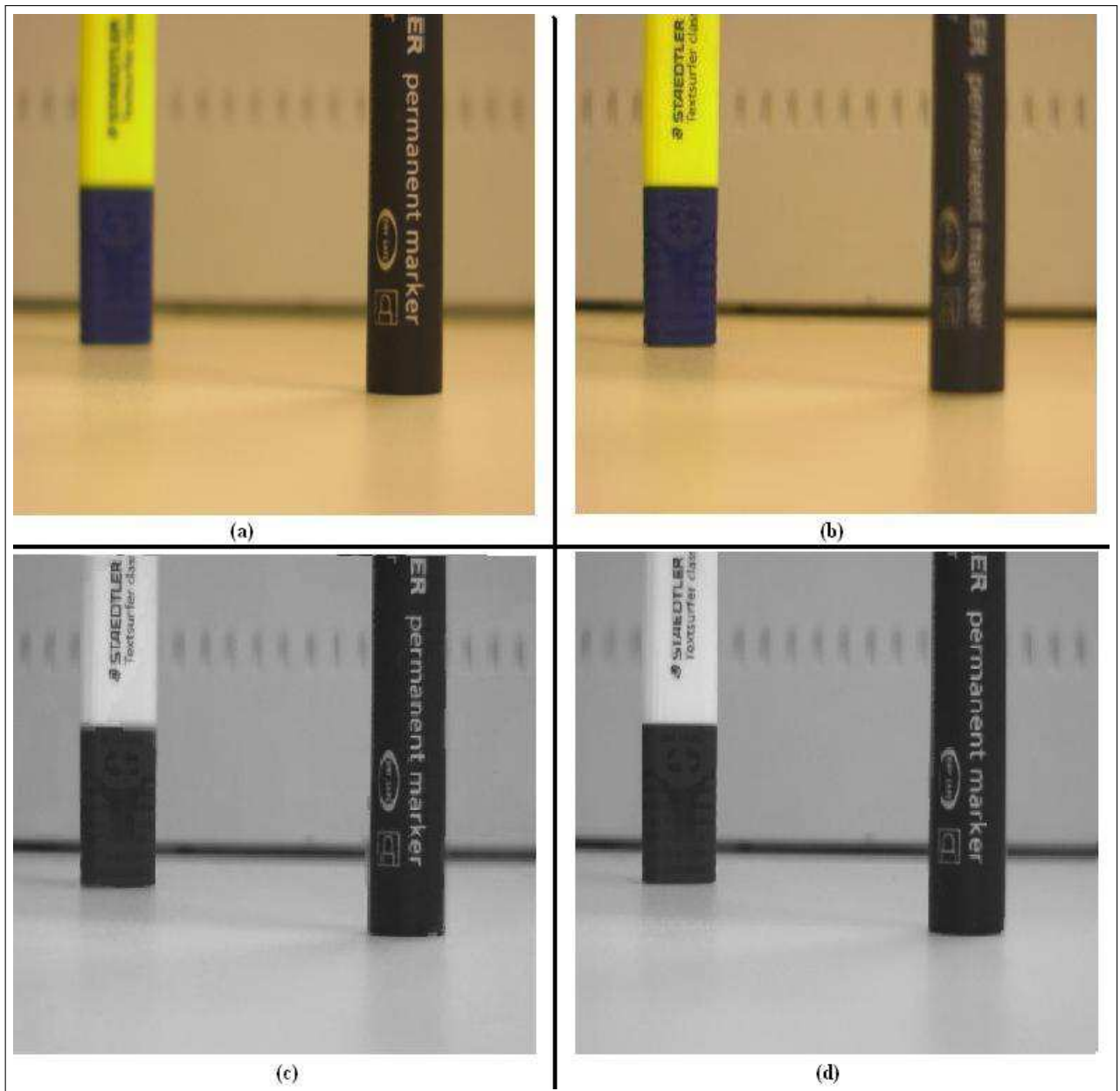


Figure 5.16 – Images sources et images fusionnées pour l'image 'Marker'. (a) : la première image source avec focus à droite. (b) : la deuxième image source avec focus à gauche. (c) : image fusionnée en utilisant la variance. (d) : image fusionnée en utilisant l'énergie d'un block (technique proposée).



Figure 5.17 – Images sources et images fusionnées pour l'image 'Box'. Même ordre que la figure 5.16.

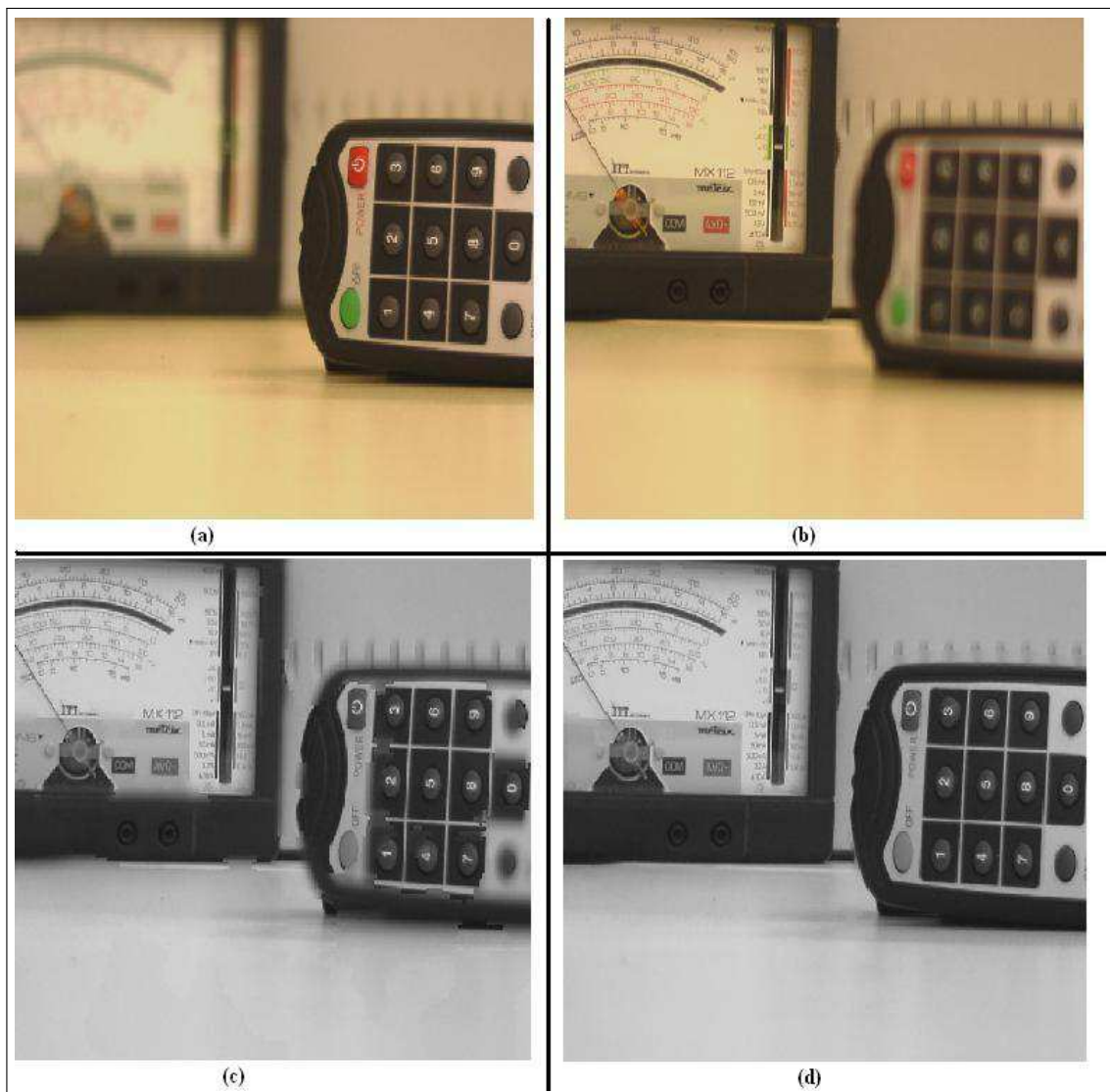


Figure 5.18 – Images sources et images fusionnées pour l'image '**Equipment**'. Même ordre que la figure 5.16.

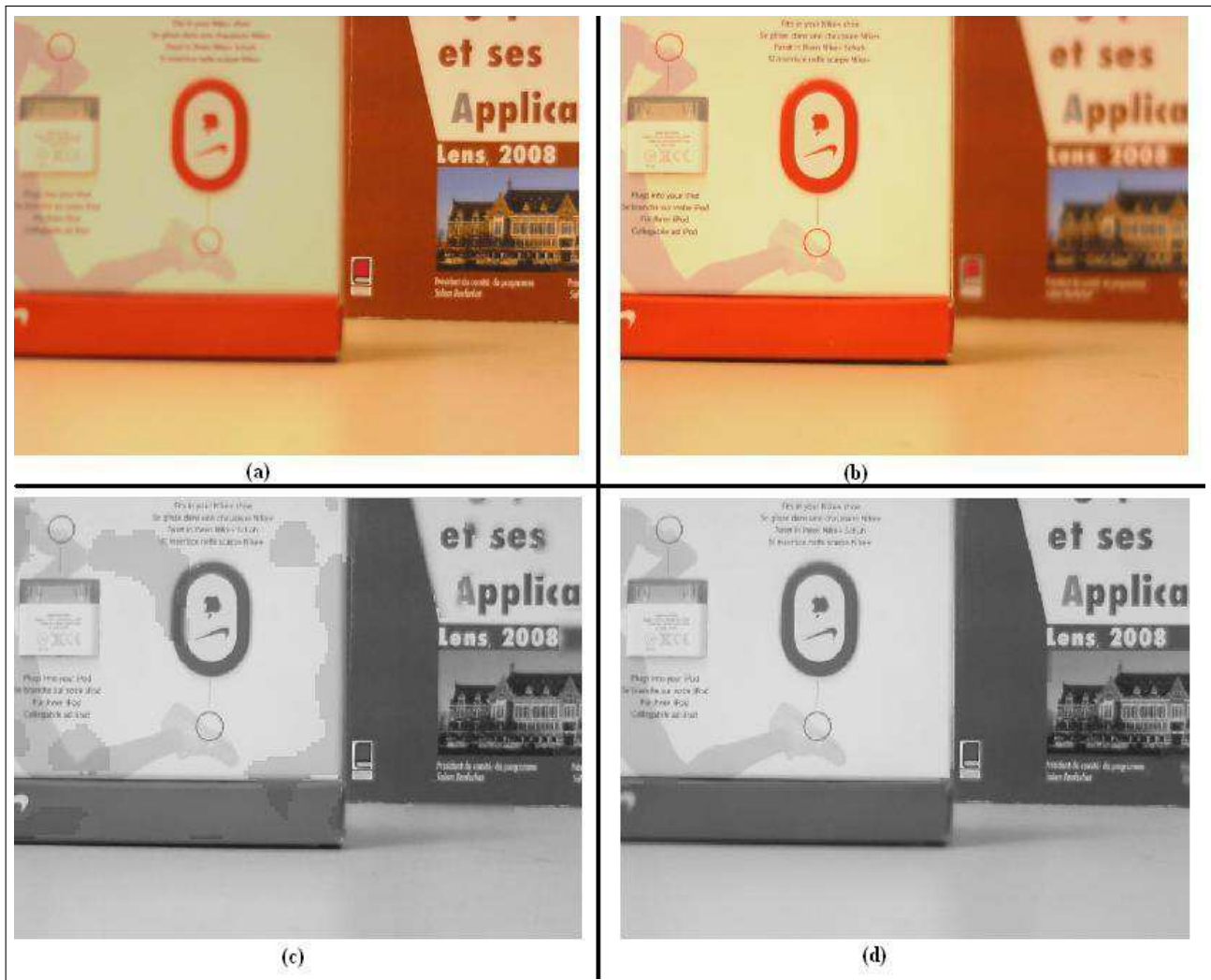


Figure 5.19 – Images sources et images fusionnées pour l'image 'Nike'. Même ordre que la figure 5.16.

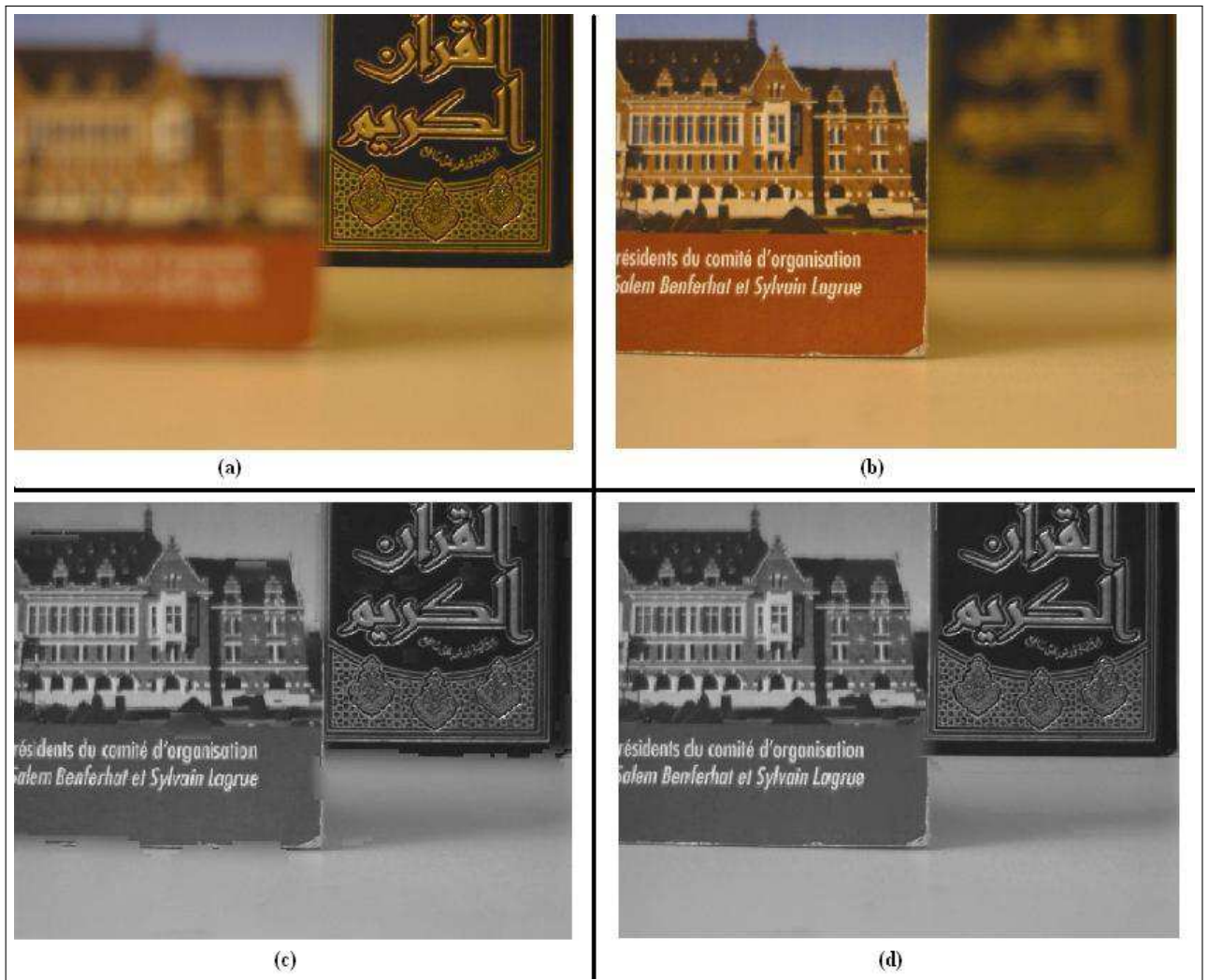


Figure 5.20 – Images sources et images fusionnées pour l'image 'Quran'. Même ordre que la figure 5.16.



Figure 5.21 – Images sources et images fusionnées pour l'image 'Robot'. Même ordre que la figure 5.16.

Commentaire : En analysant les images fusionnées à partir des images sources artificielles en utilisant les techniques DCT+Mean (moyenne) et DCT+S.D (écart type), nous remarquons à vue d'œil que ces deux techniques donnent des résultats médiocres par rapport aux deux autres techniques de fusion. Par ailleurs, nous remarquons que la technique DCT+Energy donne de bons résultats par rapport à la technique DCT+Variance, cela est remarquable à travers une comparaison entre quelques images réelles ou artificielles. Pour illustrer cela considérons les deux images artificielles (e) et (f) de la figure 5.11 où nous remarquons la présence d'une tâche (bruit) sur l'image (e) dans la partie inférieure du livre noir, (voir Figure 5.22 qui est une version zoomée des image (e) et (f) de la figure 5.11). Même constat pour les deux images (e) et (f) de la

figure 5.13 où on voit clairement des tâches autour des deux avions, (voir Figure 5.23 qui est une version zoomée des images (e) et (f) de la figure 5.13. Cela est confirmé aussi à travers une simple comparaison entre les images fusionnées à partir des images sources réelles.

Pour confirmer ce que nous avons vu à l'œil nu, nous avons jugé utile l'utilisation d'autres moyens d'évaluation moins subjectifs, il s'agit des métriques d'évaluation des images fusionnées.

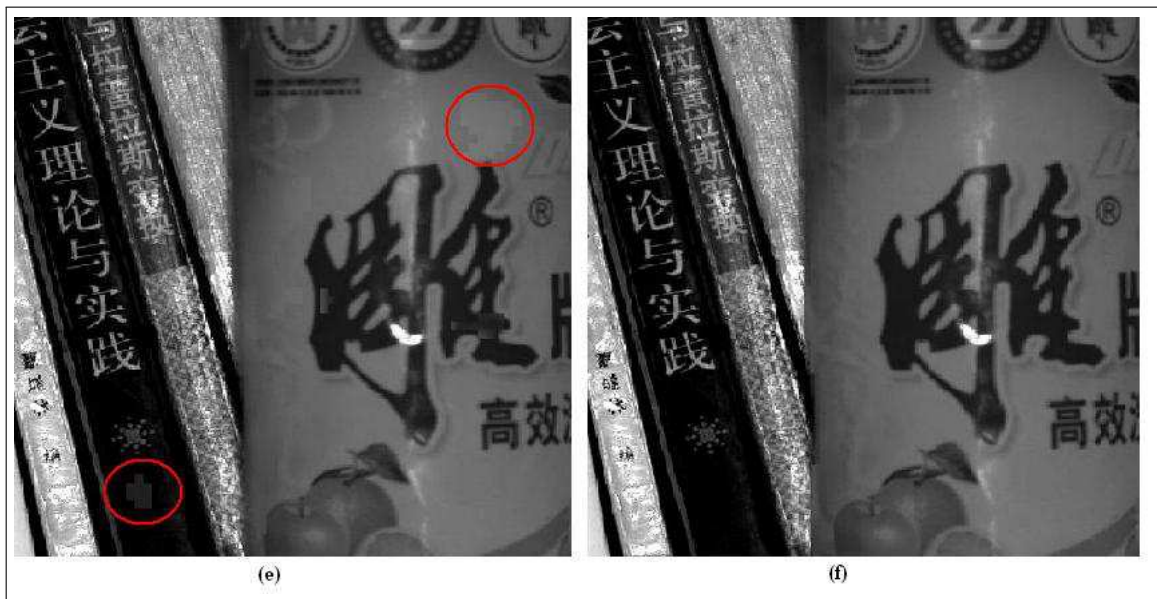


Figure 5.22 – Comparaison des résultats des deux techniques $DCT+Variance$ (e) et $DCT+Energy$ (f) sur l'image **Bottle**.

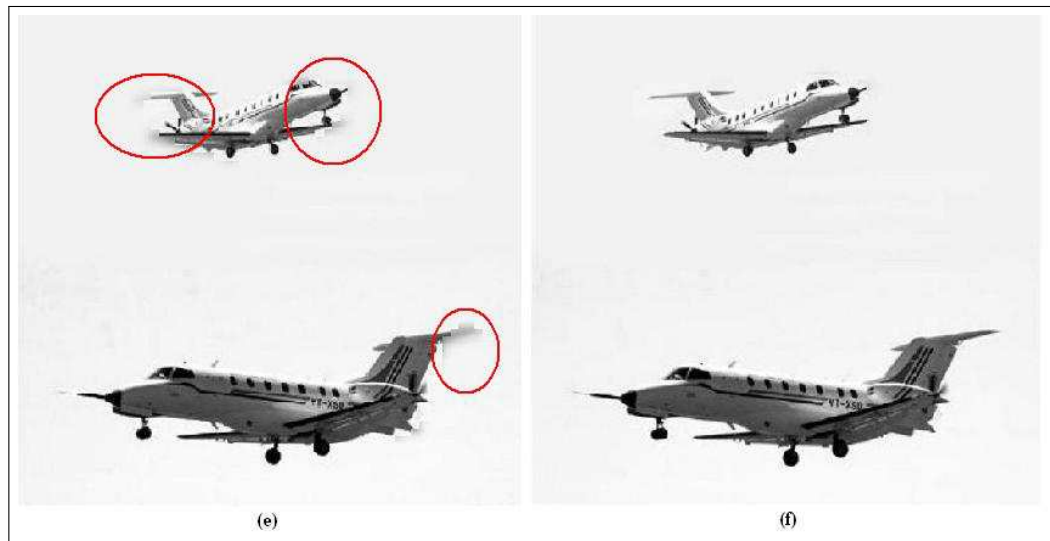


Figure 5.23 – Comparaison des résultats des deux techniques $DCT+Variance$ (e) et $DCT+Energy$ (f) sur l'image **Plane**.

- Evaluation des images fusionnées

Dans cette partie, nous allons évaluer les images fusionnées obtenues par chaque technique en considérant des images artificielles et réelles. Pour ce faire, nous utilisons des métriques d'évaluation détaillées dans ce mémoire. De plus, afin d'estimer la complexité de chaque technique de fusion nous calculons le temps d'exécution de chaque algorithme.

A - Sur des images artificielles

1. Métriques d'évaluation :

- **L'écart type S.D** : L'écart type d'une image numérique donne l'amplitude moyenne de la variation des niveaux de part et d'autre de la moyenne des pixels. La valeur de l'écart type d'une image peut être considérée comme étant le contraste de l'image, c'est-à-dire un écart type faible signifie un contraste faible et un écart type élevé représente un contraste élevé. Nous avons pris un échantillon d'images fusionnées obtenues lors de l'application des différentes techniques de fusion, les résultats obtenus sont donnés par le tableau 5.8.

- - -	DCT+Mean	DCT+S.D	DCT+Variance	DCT+Energy
Clock (512 × 512)	85.6779	85.6779	86.0937	86.4766
Bottle (512 × 512)	51.9757	51.8249	52.0680	52.1424
Book (1280 × 960)	59.4356	62.0492	62.2026	62.3243
Pepsi (404 × 404)	43.6054	43.6054	44.8019	44.9032
Plane (512 × 512)	45.8780	45.9980	49.3338	49.7754

Tableau 5.8 – Ecart type des images fusionnées avec les quatre techniques de fusion.

- **PSNR** : Le PSNR est une métrique qui donne le rapport signal/bruit entre une image fusionnée et une image de référence, autrement dit le PSNR exige l'existence d'une image de référence pour la comparer avec l'image fusionnée. Dans des cas réels, il est difficile voir impossible de disposé d'une image idéal qui peut être considérée comme étant une image de référence. **J. Tang** dans [6] et **H. Zhao** dans [81] ont donné deux images référence pour deux image parmi les images que nous avons utilisées dans nos différents tests ; il s'agit des images **Church** [6] et **Bottle** [81]. Nous avons considéré ces deux images de référence afin de calculer le PSNR entre ces images de référence et les images fusionnées obtenues par les différentes techniques utilisées (DCT+Mean, DCT+S.D, DCT+Variance, DCT+Energy). Les résultats obtenus sont donnés dans le Tableau 5.9.

- - -	DCT+Mean	DCT+S.D	DCT+Variance	DCT+Energy
Church (512 × 512)	20.4334	22.5010	23.1144	23.6548
Bottle (512 × 512)	16.8714	16.8409	17.7598	18.0028

Tableau 5.9 – PSNR entre les images fusionnées obtenues par les différentes techniques de fusion et les images de référence 'Church et Bottle'.

- **Métrique de Petrovic** : Cette métrique nous donne la quantité relative de bordures (contours) qui ont été transférées à partir des images sources vers l'image fusionnée, une valeur élevée de cette métrique signifie que l'image fusionnée contient une quantité conséquente de bordures et contours ramenés à partir des images sources vers l'image fusionnée. Les résultats obtenus sont donnés par le tableau 5.10.

- - -	DCT+Mean	DCT+S.D	DCT+Variance	DCT+Energy
Clock (512 × 512)	0.9945	0.9945	0.9938	0.9938
Bottle (512 × 512)	0.9195	0.9195	0.9071	0.9062
Book (1280 × 960)	0.9769	0.9768	0.9766	0.9756
Pepsi (404 × 404)	0.9998	0.9998	0.9998	0.9998
Plane (512 × 512)	1.0000	1.0000	1.0000	1.0000

Tableau 5.10 – Métrique de Petrovic de quelques images fusionnées avec les quatre techniques de fusion.

2. Temps d'exécution :

Le tableau 5.11 nous donne le temps d'exécution de chaque technique (Moyenne, Ecart type, Variance et Energie contenu dans un bloc) sur l'ensemble des images que nous avons utilisées lors de l'opération de fusion. Il est à signaler que pour chaque technique et sur chaque image nous avons effectué trois (03) mesures du temps d'exécution et nous avons pris la moyenne des trois mesures.

- - -	DCT+Mean	DCT+S.D	DCT+Variance	DCT+Energy
Clock (512 × 512)	2.107102	2.270445	1.874372	1.836416
Book (1280 × 960)	8.742251	9.700201	7.833843	7.774057
Plane (512 × 512)	2.010929	2.297654	1.890139	1.793642
Coca (1181 × 960)	7.842929	8.925756	7.231616	7.125880
Pepsi (404 × 404)	1.174267	1.358992	1.164892	1.141672
Cameraman (612 × 612)	2.603884	3.055785	2.403846	2.349356
Church (512 × 512)	2.054976	2.233892	1.908491	1.823584
Lena (512 × 512)	1.880827	2.146066	1.775523	1.735522
Circle (512 × 512)	1.892163	2.274484	1.760574	1.693885

Tableau 5.11 – Temps d'exécution des quatre techniques de fusion (en secondes).

B - Sur des images réelles

Nous donnons les résultats d'évaluation des images fusionnées, obtenus pour les techniques de fusion dites DCT+Mean, DCT+S.D, DCT+Variance et DCT+Energy, appliquées sur des images réelles.

1. Métriques d'évaluation :

- **L'écart type S.D** : Les résultats obtenus sont donnés par le tableau 5.12.

---	DCT+Mean	DCT+S.D	DCT+Variance	DCT+Energy
Marker (1024 × 768)	46.4976	46.5668	47.7602	48.2143
Box (1024 × 768)	35.5364	35.6115	36.6093	37.7433
Equipment (1024 × 768)	61.5566	61.5411	61.1700	63.8276
Nike (1024 × 768)	47.5071	48.0612	51.4355	53.5317
Quran (1024 × 768)	43.0758	42.9751	46.8255	48.0750
Robot (1024 × 768)	54.7806	56.0993	58.0949	59.7344

Tableau 5.12 – Ecart type des images réelles fusionnées avec les techniques DCT+Mean, DCT+S.D, DCT+Variance et DCT+Energy.

- **PSNR** : Les résultats obtenus sont donnés par le tableau 5.13.

---	DCT+Mean	DCT+SD	DCT+Variance	DCT+Energy
Marker (1024 × 768)	33.6851	33.6851	33.6851	34.4631
Box (1024 × 768)	33.6851	33.6851	34.9939	35.7884
Equipment (1024 × 768)	24.6952	24.3474	27.5462	27.8335
Nike (1024 × 768)	19.5320	19.5921	20.3605	20.6814
Quran (1024 × 768)	19.1746	19.1746	19.5319	19.5419
Robot (1024 × 768)	27.5614	27.4921	29.9958	30.1543

Tableau 5.13 – Le PSNR des images réelles fusionnées avec les techniques DCT+Mean, DCT+S.D, DCT+Variance et DCT+Energy.

- **Métrique de Petrovic** : Les résultats obtenus sont donnés par le tableau 5.14.

- - -	DCT+Mean	DCT+SD	DCT+Variance	DCT+Energy
Marker (1024 × 768)	1.0000	1.0000	1.0000	1.0000
Box (1024 × 768)	1.0000	1.0000	1.0000	1.0000
Equipment (1024 × 768)	0.9914	0.9914	0.9912	0.9883
Nike (1024 × 768)	1.0000	1.0000	1.0000	1.0000
Quran (1024 × 768)	0.9831	0.9800	0.9799	0.9809
Robot (1024 × 768)	0.9963	0.9959	0.9934	0.9937

Tableau 5.14 – La métrique de Petrovic pour des images réelles fusionnées avec les techniques DCT+Mean, DCT+S.D, DCT+Variance et DCT+Energy.

2. **Temps d'exécution** : Les résultats obtenus sont donnés par le tableau 5.15.

- - -	DCT+Mean	DCT+SD	DCT+Variance	DCT+Energy
Marker (1024 × 768)	5.431781	6.417937	5.101760	5.082194
Box (1024 × 768)	5.551333	6.457728	5.101760	5.062437
Equipment (1024 × 768)	5.416333	6.209496	5.115376	5.035175
Nike (1024 × 768)	5.462831	6.152100	5.005409	4.963570
Quran (1024 × 768)	5.421875	6.537723	5.083184	4.997122
Robot (1024 × 768)	5.509567	6.214986	5.138409	5.080626

Tableau 5.15 – Temps d'exécution (en secondes) des techniques DCT+Mean, DCT+S.D, DCT+Variance et DCT+Energy sur quelques images réelles.

- Analyse et commentaires

D'après les résultats obtenus sur des images artificielles ou réelles, il est clair que la technique de fusion basée sur l'énergie contenue dans un bloc donne globalement des résultats meilleurs que les techniques basées sur la moyenne, l'écart type et la variance. Cependant, la mesure de la métrique de Petrovic (donnant le taux de transfert des bordures), favorise légèrement les techniques basées sur la moyenne et l'écart type. A vue d'œil et sans considérer les résultats obtenus des différentes métriques, il apparaît clairement que la fusion basée sur l'énergie contenue dans un bloc DCT donne de loin un

meilleur rendu visuel, par rapport aux autres techniques. En plus et si nous considérant le temps d'exécution de chaque technique comme un critère de complexité, il en ressort que la technique basée sur l'énergie d'un bloc DCT est la plus rapide par rapport aux autres techniques. Ceci est important car la complexité est un facteur déterminant dans la consommation énergétique du réseau et du temps d'acheminement des données de bout en bout.

5.5 Conclusion

Dans ce chapitre, nous avons présenté les résultats de différents tests et simulations réalisés dans le but de valider nos propositions déclinées dans le chapitre 4. Il a été démontré l'efficacité du mécanisme de gestion de l'envoi de données multimédias dans un réseau de capteurs à travers la quantité d'énergie économisée ainsi que le temps d'acheminement des paquets. Il a été démontré également l'efficacité de la technique de fusion des images multi focale que nous avons proposée par rapport aux techniques citées dans l'état de l'art. Les différents tests et simulations ont été effectués d'abord sur un échantillon d'images artificielles et par la suite sur un échantillon d'images multi-focus réelles. D'autres simulations plus exhaustives peuvent être réalisées afin de déterminer d'autres propriétés et le comportement du réseau dans des situations extrêmes (baisse d'énergie, estimer la durée de vie,...etc).

Conclusion générale

Bilan

Nous nous sommes intéressés dans ce mémoire aux réseaux de capteurs sans fil, et particulièrement aux réseaux de capteurs multimédia. En effet, grâce à l'amélioration et la miniaturisation il est devenu possible d'équiper un capteur avec un module de collecte d'information multimédia. Les données multimédias se caractérisent par une consommation importante des ressources du réseau telles que l'énergie et la bande passante. Cette consommation est provoquée par le volume important de données circulant sur le réseau. L'une des méthodes utilisées pour réduire cette consommation de ressources est la fusion de données. Dans le contexte des réseaux de capteurs multimédias, le but de fusionner les données est d'éliminer la transmission redondante tout en synthétisant l'information pertinente. Ceci améliore la durée de vie du réseau et évite le stockage et la manipulation de données inutiles. Dans ce mémoire, nous nous sommes intéressés à la problématique de la fusion d'images en multifocus. Ceci consiste en la fusion de différentes images provenant de différentes caméras surveillant une même zone d'intérêt, telle que chaque caméra se focalise sur un objet particulier de la scène. La fusion des images capturées permet d'obtenir une image de très bonne qualité où tous les éléments de la scène sont dans le focus.

Les travaux menés dans ce contexte se situent à l'intersection des domaines des réseaux de capteurs sans fil, la fusion de données et du traitement d'images. Pour ce faire, nous avons établi un état de l'art exhaustif sur les techniques de fusion de données dans les réseaux de capteurs. Nous avons proposé une classification de ces méthodes et établi une comparaison pour chaque classe. Par la suite, nous avons étudié les principaux travaux de recherche destinés à la fusion de données multimédia. L'intérêt a été porté sur les méthodes de fusion d'images en multifocus.

Par la suite nous avons proposé notre propre solution orientée vers deux objectifs principaux :

1. Le premier est la proposition de mécanismes de gestion de l'envoi de données d'images dans un réseau de capteur afin de minimiser le nombre et la taille des paquets circulant, stocké et traité dans le réseau.
2. Le deuxième objectif est de proposer une technique de fusion d'images dite multifocus basée sur le concept DCT. Plusieurs métriques ont été utilisées pour la comparaison entre blocs à fusionner et l'évaluation de la qualité de l'image obtenue par fusion.

Pour valider nos propositions, nous avons réalisé plusieurs expérimentations et tests de simulations sur les environnements MATLAB et TinyOS. Les résultats obtenus comparant notre approche avec les autres méthodes ont été reportés; ces derniers montrent l'efficacité de notre approche en termes de gestion des ressources du réseau et de la qualité des images en multifocus obtenues. Les tests ont été réalisés sur un échantillon d'images artificielles et sur des images réelles que nous avons capturées au niveau de notre laboratoire.

Perspectives

Plusieurs perspectives peuvent être envisagées sur la base de ces travaux.

- Du point de vue fusion de données multimédias dans un réseau de capteurs, il reste à explorer le volet fusion de vidéos en s'inspirant de la fusion des images.
- Il serait intéressant également d'explorer la fusion d'images en couleurs (en considérant les trois composantes du signal).
- En ce qui concerne la technique de fusion proposée, il serait intéressant d'explorer des techniques hybrides utilisant DCT et ondelettes avec différents mécanismes de comparaison de blocs.
- Elargir les différents tests de simulations (considérer l'envoi de plusieurs images à la fois pour voir le comportement du réseau).

Enfin, nous comptons publier prochainement l'ensemble des résultats obtenus. Par ailleurs, une préparation d'un papier tutorial basé sur la classification des méthodes de fusion dans les réseaux de capteurs est en cours.

Bibliographie

- [1] T. Sahoo, S. Mohanty, and S. Sahu, "Multi-focus image fusion using variance based spatial domain and wavelet transform," *International Conference on Multimedia, Signal Processing and Communication Technologies 17-19 Décembre*, p. 48 to 51, DEC 2011.
- [2] Y. CHALLAL, "Réseaux de capteurs sans fils," Support de cours, Université de technologie de compiègne, 2008.
- [3] L. COBO CAMPO, "Gestion de la qualité de service et planification optimale de réseaux de capteurs multimédia sans fil," Thèse de doctorat, Ecole Polytechnique de Montréal, 2011.
- [4] A. Makhoul, "Réseaux de capteurs : localisation, couverture et fusion de données," Thèse de doctorat, Université de Franche-Comté, France, NOV 2008.
- [5] M. Haghghat, A. Aghagolzadeh, and H. Seyedarabi, "Multi-focus image fusion for visual sensor networks in dct domain." *Computers and Electrical Engineering*, vol. 37, no. 5, p. 789 to 797, SEP 2011.
- [6] J. Tang, "A contrast based image fusion technique in the dct domain." *Digital Signal Processing*, vol. 14, no. 3, p. 218 to 226, 2004.
- [7] V. Rajendran, K. Obracza, and J. Garcia-Luna Aceves, "Energy-efficient, Collision-free Medium Access Control for Wireless Sensor Networks," *U.S. Air Force/OSR, NSFNGI, Jack Baskin Chair of Computer Engineering*, NOV 2003.
- [8] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks : a survey," *Computer Networks*, vol. 38, p. 393 to 422, 2002.
- [9] S. Madden, "The design and evaluation of a query processing architecture for sensor networks," Thèse de doctorat, University of California, Berkeley, 2003.
- [10] R. KOUAH and S. MOUSSAOUI, "ROUTAGE DANS LES RESEAUX DE CAPTEURS SANS FIL MOBILES," Thèse de Magister, University of sciences and technology Houari Boumediene, Algeria, APR 2012.
- [11] R. KACIMI, "Techniques de conservation d'énergie pour les réseaux de capteurs sans fil," Thèse de doctorat, Université de Toulouse, SEP 2009.

- [12] J. Bleiholder and Naumann. (2008, Dec.) F. data fusion. acm comput. surv. [Online]. Available : <http://doi.acm.org/10.1145/1456650.1456651>
- [13] N. Dimokas, D. Katsaros, and Y. Manolopoulos, "Cooperative Caching in Wireless Multimedia Sensor Networks," *Computer Networks*, vol. 13, p. 337 to 356, 2008.
- [14] Akyildiz, I. and Melodia, T. and Chowdhury, R., "A survey on Wireless Multimedia Sensor Networks," *Computer Networks*, vol. 51, p. 921 to 960, 2007.
- [15] M. LEHSAINI, "Diffusion et couverture basées sur le clustering dans les réseaux de capteurs : application à la domotique," Thèse de doctorat, Université A.B Tlemcen Faculté des Sciences pour l'Ingénieur et Université de Franche-Comté U.F.R Sciences et Techniques, 2009.
- [16] M. Zapata, R. Zilan, M. Barcel-Ordinas, K. Bicakci, and B. Tavli, "The future of security in wireless multimedia sensor networks," *Telecommun Syst*, vol. 45, p. 77 to 91, 2010.
- [17] A. Abdelgawad and M. Bayoumi, "Resource-Aware Data Fusion Algorithms for Wireless Sensor Networks," *Electrical Engineering*, vol. 118, p. Chapter 2, 2012.
- [18] I. BLOCH and H. MAITRE, "Fusion de données en traitement d'images : modèles d'information et décisions," *TS.Traitement du signal*, vol. 11, no. 6, p. 435 to 446, 1994.
- [19] V. Venu Dasigi, R. Mann, and V. Protopopescu, "Information fusion for text classification - an experimental comparison," *Pattern Recognition*, vol. 34(12), p. 2413 to 2425, 2001.
- [20] S. S. Gatepaille and S. Brunessaux, "Dossier i.a. et fusion de données," *Bulletin de l'Association Française pour l'Intelligence Artificielle (AFIA)*, vol. 24, p. 21 to 30, JAN 1996.
- [21] D. BELLOT, "Fusion de données avec des réseaux bayésiens pour la modélisation des systèmes dynamiques et son application en télémédecine," Thèse de doctorat, université de Henri Poincaré - Nancy1-France, NOV 2002.
- [22] V. Vincent Ricquebourg, D. Durand, D. Menga, L. Delahoche, B. Marhic, C. Logé, and A. Jolly-Desodt, "La fusion multi-capteurs dans l'habitat communicant : Une approche non-probabiliste," *Proceeding UbiMob '08 Proceedings of the 4th French-speaking conference on Mobility and ubiquity computing*, p. 09 to 16, MAY 2008.
- [23] X. Luo, M. Dong, and Y. Huang, "On distributed fault-tolerant detection in wireless sensor networks," *IEEE Transactions on Computers*, vol. 55, p. 58 to 70, JAN 2006.
- [24] B. Krishnamachari and S. Iyengar, "Distributed bayesian algorithms for fault-tolerant event region detection in wireless sensor networks," *IEEE Transactions on Computers*, vol. 53, p. 241 to 250, MAR 2004.

- [25] O. Younis, M. Krunz, and S. Ramasubramanian, "Node clustering in wireless sensor networks : Recent developments and deployment challenges," *Network, IEEE*, vol. 20, no. 3, p. 20 to 25, MAY 2006.
- [26] K. Maraiya, K. Kant, and N. Gupta, "Study of data fusion in wireless sensor network," *Proc. of the International Conference on Advanced Computing and Communication Technologies (ACCT)*, p. 978 to 981, 2011.
- [27] B. Dasarathy, "Image fusion : Advances in the state of the art." *International journal on Information fusion*, vol. 8, no. 2, p. 114 to 118, APR 2007.
- [28] V. VPetrovic, "Multisensor pixel-level image fusion," Thèse de doctorat, Université de Manchester, APR 2001.
- [29] A. Nevel, "Image registration : a key element for information processing." *Proc. SPIE*, vol. 4471, p. 190 to 200, 2001.
- [30] L. Brown, "A survey of image registration techniques," *ACM Computing Survey*, vol. 24, no. 4, p. 325 to 376, 1992.
- [31] A. Malviya and S. Bhirud, "Multi-focus image fusion of digital images." *International Conference on Advances in Recent Technologies in Communication and Computing*, p. 887 to 889, 2009.
- [32] S. Nikolov, P. Hill, D. Bull, and N. Canagarajah, "Wavelets in signal and image analysis," *Computational Imaging and Vision*, vol. 19, p. 213 to 241, 2001.
- [33] R. Blum and Z. Liu, *Multi-sensor image fusion and its applications*. CRC Press, Taylor and Francis Group, 2006.
- [34] G. Piella and H. Heijmans, "A new quality metric for image fusion," *Proceedings of international conference on image processing (ICIP)*, vol. 3, p. 173 to 176, SEP 2003.
- [35] C. Xydeas and V. Petrovic, "Objective image fusion performance measure," *Electron Lett*, vol. 36, no. 4, p. 308 to 309, FEB 2000.
- [36] J. Hill, "System architecture for wireless sensor networks," PhD Thesis, University of California, Berkeley, 2003.
- [37] E. Nakamura, Manaus, A. Antonio, and B. Horizonte, "Information fusion in wireless sensor networks," *SIGMOD*, vol. 8, JAN 2008.
- [38] C. Coué, T. Fraichard, P. Bessiere, and E. Mazer, "Multi fusion data sensor using bayesian programming : an automotive application." *international conference on intelligent Robots and systems*, OCT 2002.
- [39] E. Nakamura, Manaus, A. Antonio, and B. Horizonte, "Information fusion in wireless sensor networks." *SIGMOD*, vol. 39, no. 3, JUN 2008.
- [40] B. Yu, J. Giampapa, S. Owens, and K. Sycara, "An evidential model of multisensor decision fusion for force aggregation and classification." *the 8th International Conference on Information Fusion*, p. 8 to 13, JUL 2005.

- [41] J. Mauclair and J. Pinquier, "Fusion of descriptors for speech / music classification." *European Signal Processing Conference, EUSIPCO*, p. 1285 to 1288, SEP 2004.
- [42] A. Gattein and P. Vannoorenberghe, "A comparative analysis of two approaches using the road network for tracking ground targets," *International Conference on Information Fusion*, JUN 2004.
- [43] I. Bloch, "Fusion d'informations en traitement du signal et des images," *Hermes Science Publication*, JAN 2003.
- [44] M. Arif, S. Hafid, T. Brouard, and N. Vincent, "Awfo (an aggregation operator) for pattern recognition," *12e Conférence Reconnaissance de Forme et Intelligence Artificielle (RFIA '06)*, JAN 2006.
- [45] A. Martin, "Fusion de classifieurs pour la classification d'images sonar," *CoRR*, JAN 2008.
- [46] L. Xu, A. Krzyzak, and C. Suen, "Methods of combining multiple classifiers and their application to handwriting recognition." *IEEE Transactions on Systems, Man Cybernetics*, vol. 22, no. 3, p. 418 to 435, MAI 1992.
- [47] S. Rizvi and N. Nasrabadi, "Fusion of flir automatic target recognition algorithms." *Information Fusion*, vol. 4, p. 247 to 258, 2003.
- [48] L. XIAO, S. BOYD, and S. LALL, "A scheme for robust distributed sensor fusion based on average consensus," *International Conference on Information processing in Sensor Networks (IPSN)*, p. 63 to 70, 2005.
- [49] F. Nakamura, F. Loureiro, and C. Frery, "Information fusion for wireless sensor networks : Methods, models, and classifications." *ACM Computing Surveys*, vol. 39, no. 3, 2007.
- [50] B. SINOPOLI, L. SCHENATO, M. FRANCESCHETTI, K. POOLLA, M. JORDAN, and S. SASTRY, "Kalman filtering with intermittent observations." *IEEE Trans*, vol. 49, no. 9, p. 1453 to 1464, SEP 2004.
- [51] R. OLFATI-SABER and R. MURRAY, "Consensus problems in networks of agents with switching topology and time-delays." *IEEE Transaction on Automatic Control*, vol. 49, no. 9, p. 1520 to 1533, 2004.
- [52] D. BERTSEKAS and J. TSITSIKLIS, "Parallel and distributed computation : Numerical methods." *Athena Scientific*, 1997.
- [53] M. Watfa, W. Daher, and H. Al Azar, "A sensor network data aggregation technique." *International Journal of Computer Theory and Engineering*, vol. 1, no. 1, APR 2009.
- [54] D. Kumar, T. Aseri, and R. Patel, "Eecda : Energy efficient clustering and data aggregation protocol for heterogeneous wireless sensor networks." *Int. J. of Computers, Communications and Control, ISSN 1841-9836*, vol. 6, no. 1, p. 113 to 124, APR 2011.

- [55] C. Asgari and J. Torkestani, "A new approach to the data aggregation in wireless sensor networks." *IJCSI International Journal of Computer Science Issues*, vol. 9, no. 3, 2012.
- [56] W. Lee and V. Wong, "E-span and lpt for data aggregation in wireless sensor networks." *Computer Communications*, p. 2506 to 2520, 2006.
- [57] S. Upadhyayula and S. Gupta, "Spanning tree based algorithms for low latency and energy efficient data aggregation enhanced convergecast(dac) in wireless sensor networks." *Ad Hoc Networks*, vol. 5, p. 626 to 648, 2007.
- [58] P. Srimani and K. Bhanu, "A survey on data fusion and aggregation technologies of wireless multimedia sensor networks." *International Journal of Current Research*, vol. 4, no. 01, p. 177 to 180, JAN 2012.
- [59] S. Messaoudi, K. Messaoudi, and S. Dagtas, "Bayesian data fusion for smart environments with heterogenous sensors." *Journal of computing Sciences*, vol. 25, no. 5, p. 140 to 146, MAY 2010.
- [60] D. Brulin, "Fusion de données multi-capteurs pour l'habitat intelligent," Thèse de doctorat, Université d'Orléans France, AOU 2010.
- [61] I. BLOCH and H. MAITRE, "Fusion de données en traitement d'images :modèles d'information et décisions." *GRETSI, Saint Martin d'Hères, France*, 1994.
- [62] L. Snidaro, G. Foresti, and k. Varshney, "Sensor fusion for video surveillance." *7th International Conference on Information Fusion*, vol. Paper 108, 2004.
- [63] L. Snidaro, I. Visentini, and G. Foresti, "Fusing multiple video sensors for surveillance." *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, vol. 8, no. 1, 2012.
- [64] INRIA, "D4.3 : Logiciel de fusion multi-caméras." *V1.0*, NOV 2005.
- [65] R. Desale and S. Verma, "Study and analysis of pca, dct and dwt based image fusion techniques," *International Conference on Signal Processing, Image Processing and Pattern Recognition(ICSIPRI)*, p. 66 to 69, 2013.
- [66] P. Akhtar, T. J. Ali, M. Bhatti, and M. Muqeet, "A framework for edge detection and linking using wavelets and image fusion," *Congress on Image and Signal Processing*, p. 273 to 277, 2008.
- [67] I. Selesnick, "Wavelet transforms - a quick study," *Physics Today magazine, Polytechnic University-Brooklyn, NY*, OCT 2007.
- [68] H. Wang, "Multi-sensor image fusion by using discrete multi-wavelet transform," *Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, 26-29 August*, vol. 07, p. 4331 to 4336, AUG 2004.
- [69] R. Desale and S. Verma, "Study and analysis of pca, dct and dwt based image fusion techniques," *International Conference on Signal Processing, Image Processing and Pattern Recognition [ICSIPRI]*, p. 66 to 69, OCT 2013.

- [70] S. More and S. Apte, "Pixel-level image fusion using wavelet transform," *International Journal of Engineering Research and Technology (IJERT)*, ISSN : 2278-0181, vol. 1, no. 5, JUL 2012.
- [71] R. Maruthi and K. Sankarasubramanian, "Multi focus image fusion based on the information level in the regions of the images," *Journal of Theoretical and Applied Information Technology*, p. 80 to 85, 2007.
- [72] V. Naidu and B. Elias, "A novel image fusion technique using dct based laplacian pyramid," *International Journal of Inventive Engineering and Sciences (IJIES)*, vol. 1, no. 2, p. 1 to 9, 2013.
- [73] J. R. Deborah, *Statistics For Dummies, 2nd Edition*, 2nd ed. Wiley publishing, Inc., MAY 2011.
- [74] M. Maurice Charbit and G. Blanchet, *Éléments de base pour le Traitement Numérique du Signal et de l'image*. Télécom ParisTech, MAR 2010.
- [75] S. Medjiah, T. Ahmed, F. Krief, and P. Gélard, "Agem : Un protocole de routage géographique angulaire adaptatif." *Colloque Francophone sur l'Ingénierie des Protocoles (CFIP)*. Strasbourg, France, p. 12 to 15, OCT 2009.
- [76] L. SHU, Y. ZHANG, L. YANG, Y. WANG, and M. HAUSWIRTH, "Geographic routing in wireless multimedia sensor networks." *Second International Conference on Future Generation Communication and Networking*, p. 68 to 73, 2008.
- [77] B. KARP and H. KUNG, "Gpsr : Greedy perimeter stateless routing for wireless networks," in *Mobile Computing and Networking*, p. 243 to 254, 2000.
- [78] D. Gay, P. Levis, R. Behren, M. Welsh, E. Brewer, and D. Culter, "The nesc language : A holistic approach to networked embedded systems," *In Proc. of the conference on Programming language design and implementation*, vol. 38, no. 5, p. 1 to 11, 2003.
- [79] P. Levis and N. Lee. Tossim : A simulator for tinyos networks. [Online]. Available : www.cs.berkeley.edu/~pal/pubs/nido.pdf
- [80] A. Casadevall, "Introduction à matlab," *Université de Paris-Dauphine*, vol. Sciences mention Mass - 2ème année, MAR 2004.
- [81] H. Zhao, Q. Li, and H. Feng, "Multi-focus color image fusion in the hsi space using the sum-modified-laplacian and a coarse edge map," *Image and Vision Computing*, vol. 26, no. 9, p. 1285 to 1295, SEP 2008.