

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

UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU



FACULTE DUGENIE ELECTRIQUE ET D' INFORMATIQUE  
DEPARTEMENT D'ELECTRONIQUE

## Mémoire de Fin d'Etudes De MASTER ACADEMIQUE

Domaine : Sciences et Technologies Filière : Génie électrique  
Spécialité : Télécommunication et réseaux

*Présenté par*  
**Tinhinane ASSAM**

Thème

## Compression d'images omnidirectionnelles à PVU en utilisant le standard JPEG 2000

*Mémoire soutenu publiquement le 10/07/2016. devant le jury composé de:*

**M<sup>me</sup> AMEUR Zohra**  
Professeur, UMMTO, Président

**M ALOUACHE Djamel**  
Maitre de conférences B, UMMTO, Encadreur

**M LAHDIR Mourad**  
Maitre de conférences A, UMMTO, Examinateur

**M<sup>me</sup> HEMDANI Naima**  
Maitre assistante A, UMMTO, Examinatrice

# Remerciements

Ce mémoire a été réalisé au sein du laboratoire d'Analyse et de modalisation des phénomènes Aléatoires (L.A.M.P.A) de la faculté de Génie Electrique et informatique de l'Université Mouloud **MAMMERY** de Tizi-Ouzou

Je tiens à témoigner ma profonde gratitude et mes remerciements les plus sincères à mon promoteur Mr. D. Alouache de m'avoir encadré, suivi et soutenu tout au long de ce travail.

Je tiens également à exprimer ma gratitude à mon co-Encadreur Mr M.Lahdir pour toute son aide, et les conseils qu'il ma prodigué durant la période de préparation de ce mémoire.

Je remercie également les membres du jury qui me feront l'honneur de juger mon travail.

Je tiens également à remercier tout les enseignants qui ont participé à ma formation depuis le début de mon cursus.

Enfin, je tiens à exprimer mon entière reconnaissance à tous ceux qui m'ont encouragé et contribué à la réalisation de ce modeste travail dont ma famille et mes amis (es).

# Dédicace

Ma mère

La source de mon succès

Mon père

Pour les bons moments et la  
spontanéité à rendre service

Mes sœurs

Tu as dépensée toute  
ta vie pour m'assurer une  
éducation exemplaire

Mon frère

Ma réussite est aussi la tienne

Samir

Pour leur soutien et pour  
m'avoir donné le goût  
d'apprendre

Amel et kahina

Ma source de courage

Enfin je dédie ce travail à tous mes amis et à toutes personnes qui sont dans ma mémoire et qui ne sont pas dans ce mémoire.

<b>INTRODUCTION</b> .....	1
<b>CHAPITRE I : La vision omnidirectionnelle</b>	
I.1 Préambule .....	3
I.2 Définition d'une image .....	3
I.2.1 Image binaire .....	4
I.2.2 Image en niveau de gris .....	4
I.2.3 Image couleur .....	5
I.2.4 Image multi spectrale .....	5
I.3 La vision omnidirectionnelle .....	5
I.3.1 Fusion de plusieurs images .....	6
a/ Systèmes rotatifs .....	7
b/ Systèmes de caméras multiples .....	7
I.3.2 Objectifs grand angle .....	8
I.3.4 Système catadioptrique .....	9
I.3.4.1 Caméras catadioptriques centrales .....	10
a- Le miroir plan .....	10
b- Le miroir conique .....	11
c- Le miroir ellipsoïde .....	11
d- Le miroir sphérique .....	12
e- Le miroir hyperboloïde .....	12
f- Le miroir paraboloides .....	13
I.3.4.2 Capteurs catadioptriques non centraux .....	13
I.4 Modélisation d'une caméra .....	14
I.4.1 Modélisation d'une caméra catadioptrique .....	15
I.4.2 Modèle de projection unifié .....	15
I.5 Calibrage des caméras catadioptriques .....	17
I.5.1 Le calibrage avec des mires .....	18
I.5.2 Calibrage avec des points .....	18
I.5.3 Calibrage avec des droites .....	19
I.5.4 Calibrage avec des sphères .....	20
I.6 Discussion .....	21
<b>CHAPITRE II : Généralités sur la compression d'image</b>	
II.1 Préambule .....	22
II.2 Définition de la compression .....	23
II.3 Principe général de la compression d'image .....	23
II.4 Les méthodes de compression sans perte (réversible) .....	24
II.4.1 Codage par symboles individuels .....	24
II.4.1.1 Codage de Shannon-Fano .....	25
II.4.1.2 Codage de Huffman .....	26
II.4.1.3 Codage prédictif .....	27
II.4.1.4 Codage par plan de bit .....	27
II.4.2 Codage de séquences de symboles .....	28
II.4.2.1 Codage Run Length Encoding (RLE) .....	28
II.4.2.2 Méthode par dictionnaire (Lemple-Ziv-Welch) .....	29
II.4.2.3 Codage arithmétique .....	30
II.5 Les méthodes de compression avec perte (irréversibles) .....	32
II.5.1 Méthodes par transformation .....	32
II.5.1.1 La transformée de Fourier Discrète (TFD) .....	32
II.5.1.2 La transformation De Karhunen Loeve .....	33

II.5.1.3 La transformation en cosinus Discrète DCT .....	34
II.5.1.4 Transformée en ondelettes .....	35
II.5.2 Les fractales .....	39
Principe de l’algorithme de compression .....	39
II.5.3 Codage par quantification .....	40
II.5.3.1 La quantification scalaire .....	40
II.5.3.2 Quantification scalaire uniforme .....	41
II.5.3.3 Quantification uniforme à zone morte .....	41
II.5.3.4 La quantification vectorielle .....	42
II.5.4 Méthode hybrides .....	43
II.6 Les normes de compression d’image .....	43
II.6.1 La norme JPEG .....	44
II.6.2 La norme de compression JPEG 2000 .....	45
II.7 Discussion .....	46
<b>CHAPITRE III: Présentation et application du standard JPEG 2000 pour les images omnidirectionnelles</b>	
III.1 Préambule .....	47
III.2 Organisation de l’algorithme JPEG 2000 .....	48
III.2.1 Décomposition en tuiles .....	49
III.2.2 Transformée en ondelettes discrète(TOD) .....	49
III.2.3.1 Quantification .....	52
III.2.3.2 Règle de la quantification inverse (déquantification) .....	53
III.2.4 Codage entropique .....	53
III.2.5 Le codage arithmétique .....	54
III.3 Allocation de débit .....	56
III.4 Organisation de bistream .....	56
III.5 Région d’intérêt .....	57
III.6 Le contrôle de taux .....	57
III.7 Résistance aux erreurs .....	57
III.8 Compression d’image omnidirectionnelle .....	58
III.8. Compression sur la sphère .....	59
III.9 Discussion .....	61
<b>CHAPITRE IV : Tests et résultats</b>	
IV.1 Préambule .....	62
IV.2 Critères d’évaluation .....	62
IV.2.1 Taux de compression .....	62
IV.2.2 Mesure de distorsion .....	63
IV.3 Séquences de synthèses .....	64
IV.4 images réelles .....	66
IV.5 Résultats obtenus et évaluation .....	68
IV.5.1 Résultats obtenus : Séquences de synthèses .....	68
IV.5.2 Résultats obtenus : Séquences réelles .....	70
IV.6 Discussion .....	78
<b>Conclusion</b> .....	79
<b>Annexe</b> .....	80
<b>Bibliographie</b>	

## Liste des figures

<b>Figure I.1</b> : représentation d'une image numérique .....	4
<b>Figure I.2</b> : Image binaire .....	4
<b>Figure I.3</b> : image en niveau de gris .....	4
<b>Figure I.4</b> : Image couleur .....	5
<b>Figure I.5</b> : Image multi spectrale .....	5
<b>Figure I.6</b> : Image panoramique .....	6
<b>Figure I.7</b> : image omnidirectionnelle .....	6
<b>Figure I.8</b> : Exemple de caméra en rotation autour d'un axe vertical fixe .....	7
<b>Figure I.9</b> : Systèmes de caméras multiples .....	8
<b>Figure I.10</b> : Système Fullview .....	8
<b>Figure I.11</b> : Image acquise par une caméra fish-eye .....	9
<b>Figure I.12</b> : système catadioptrique .....	9
<b>Figure I.13</b> : exemple d'image omnidirectionnelle .....	9
<b>Figure I.14</b> : Caméras catadioptriques : Miroir plan. ....	10
<b>Figure I.15</b> : Caméras catadioptriques : Miroir conique. ....	11
<b>Figure I.16</b> : Caméras catadioptriques : Miroir ellipsoïde.....	11
<b>Figure I.17</b> : Caméras catadioptriques : Miroir sphérique.....	12
<b>Figure I.18</b> : Caméras catadioptriques : Miroir hyperboloïde.....	12
<b>Figure I.19</b> : Caméras catadioptriques : Miroir parabolique.....	13
<b>Figure I.20</b> : (a) Système avec un PVU constitué d'un miroir parabolique avec une caméra orthographique (b)- système avec un point de vue non unique composé d'un miroir parabolique et d'une caméra perspective.....	14
<b>Figure I.21</b> : représentation géométrique des repères utilisés pour la détermination d'un modèle de caméra .....	14
<b>Figure I.22</b> : Equivalence entre la projection catadioptrique centrale et la projection sur la sphère unité .....	16
<b>Figure I.23</b> : calibrage avec des mires .....	18
<b>Figure I.24</b> : calibrages avec des points .....	19
<b>Figure I.25</b> : Calibration avec des droites [Barreto 2002] .....	19
<b>Figure I.26</b> : Calibration avec des sphères .....	18
<b>Figure II.1</b> : Schéma synoptique de compression d'images fixe .....	23
<b>Figure II.2</b> : Exemple d'un codage de Huffman .....	26
<b>Figure II.3</b> : schéma de décomposition d'une image en plans de bits.....	28
<b>Figure II.4</b> : application de la DCT sur un bloc de pixels de 8*8 .....	34
<b>Figure II.5</b> : Répartition de l'énergie d'un bloc d'image .....	34
<b>Figure II.6</b> : quantification uniforme avec un pas de quantification =1 .....	41
<b>Figure II.7</b> : synoptique général du codage d'image par quantification vectorielle.....	43
<b>Figure II.8</b> : Principe de compression JPEG avec perte.....	45
<b>Figure II.9</b> : Principe de compression JPEG sans perte.....	45
<b>Figure III.1</b> : blocs constituant la chaine de codage et décodage JPEG 2000 .....	48
<b>Figure III.2</b> : Décomposition multi-résolutions d'une image .....	50
<b>Figure III.3</b> : Décomposition en ondelettes à deux niveaux .....	51
<b>Figure III.4</b> : Reconstruction multi-résolutions d'une image .....	51
<b>Figure III.5</b> : Quantification scalaire uniforme avec zone morte (Dead zone).....	52
<b>Figure 6</b> : Partitionnement en code-blocks .....	55
<b>Figure 7</b> : Représentation en bit-plane d'un code-block.....	55
<b>Figure 8</b> : l'ordre de parcourt d'une stripe.....	55

<b>Figure III.9:</b> Localisation de région d'intérêt.....	57
<b>Figure III.10 :</b> Compression d'images omnidirectionnelles .....	59
<b>Figure III.11 :</b> Equivalence entre un modèle de projection parabolique .....	59
<b>Figure III.12:</b> les coordonnées sphériques .....	60
<b>Figure III.13 :</b> image para-catadioptrique et l'image sphérique équivalente .....	61
<b>Figure IV.1 :</b> Exemple d'image de l'environnement virtuel (3D) généré par le simulateur...	64
<b>Figure IV.2 :</b> image para-catadioptrique de synthèse (moins texturé) .....	65
<b>Figure IV.3 :</b> image para-catadioptrique de synthèse 2(fortement texturé) .....	65
<b>Figure IV.4 :</b> Image omnidirectionnelle et l'image sphérique équivalente de la séquence1...	67
<b>Figure IV.5 :</b> Image omnidirectionnelle et image sphérique équivalente de la séquence 2 ...	67
<b>Figure IV.6 :</b> Image omnidirectionnelle et image sphérique équivalente de la séquence 3 ...	67
<b>Figure IV.7 :</b> Courbe comparative de l'évolution du PSNR moyen en fonction du débit binaire. ....	68
<b>Figure IV.8:</b> Courbe comparative de l'évolution du PSNR moyen en fonction du débit binaire.....	69
<b>Figure IV.9 :</b> Courbe comparative de l'évolution du PSNR moyen en fonction du débit binaire(Séquence1).....	70
<b>Figure IV.10 :</b> Courbe comparative de l'évolution du PSNR moyen en fonction du débit binaire(Séquence 2).....	71
<b>Figure IV.11:</b> Courbe comparative de l'évolution du PSNR moyen en fonction du débit binaire (Séquence3).....	71
<b>Figure IV.11 :</b> restituées et les images erreurs de la 1 <sup>ère</sup> séquence réelle dans le cas plan image .....	73
<b>Figure IV.12 :</b> restituées et les images erreurs de la 1 <sup>ère</sup> séquence réelle dans le cas sphérique.....	73
<b>Figure IV.13 :</b> restituées et les images erreurs de la 2 <sup>ème</sup> séquence réelle dans le cas plan image .....	74
<b>Figure IV.14 :</b> restituées et les images erreurs de la 2 <sup>ème</sup> séquence réelle dans le cas sphérique .....	74
<b>Figure IV.15 :</b> restituées et les images erreurs de la 3 <sup>ème</sup> séquence réelle dans le cas plan image .....	75
<b>Figure IV.16 :</b> restituées et les images erreurs de la 3 <sup>ème</sup> séquence réelle dans le cas sphérique .....	75
<b>Figure IV.17 :</b> restituées et les images erreurs de la 1 <sup>ère</sup> séquence synthèse dans le cas plan image .....	76
<b>Figure IV.18 :</b> restituées et les images erreurs de la 1 <sup>ère</sup> séquence synthèse dans le cas sphérique .....	76
<b>Figure IV.19 :</b> restituées et les images erreurs de la 2 <sup>ème</sup> séquence synthèse dans le cas plan image .....	77
<b>Figure IV.20 :</b> restituées et les images erreurs de la 2 <sup>ème</sup> séquence synthèse dans le cas sphérique .....	77

**Liste des tableaux**

<b>Tableau II.1</b> : Exemple de codage de Shannon-Fano .....	25
<b>Tableau II.2</b> : caractéristiques des principales méthodes de compression réversibles.....	31
<b>Tableau II.3</b> : tableau récapitulatif des caractéristiques de certaines transformations .....	38
<b>Tableau IV.1</b> : PSNR omni moyen et PSNR sphérique de la séquence de synthèse 1.....	69
<b>Tableau IV.2</b> : PSNR omni moyen et PSNR sphérique de la séquence de synthèse 2 .....	69
<b>Tableau IV.3</b> : PSNR omni moyen et PSNR sphérique de la séquence réelle 1 .....	71
<b>Tableau IV.4</b> : PSNR omni moyen et PSNR sphérique de la séquence réelle 2.....	72
<b>Tableau IV.5</b> : PSNR omni moyen et PSNR sphérique de la séquence réelle 3 .....	72

## Introduction

Au cours des deux dernières décennies, nous remarquons un énorme progrès dans le domaine de la vision artificielle, aidé par l'augmentation de la puissance de calcul des ordinateurs et la miniaturisation des caméras; ces derniers permettent une utilisation de plus en plus fréquente des systèmes de vision dans des domaines très variés, tels que la vidéo surveillance, les applications militaires, médicales, robotiques, spatiales et même industrielles.

Pourtant force est de constater que malgré ces progrès, le domaine de la vision est en perpétuelle évolution. Chaque année, de nouvelles méthodes sont proposées, de nouvelles applications imaginées et de nouveaux capteurs développés, faisant de ce domaine une source inépuisable pour les chercheurs. Ce qui a mené à l'apparition de nouveaux capteurs panoramiques et omnidirectionnels. Leur objectif est d'augmenter le champ de vision de la scène observée. Pour y arriver plusieurs solutions ont été imaginées. Parmi lesquelles l'approche catadioptrique qui tient une place avantagée.

L'approche catadioptrique consiste à placer un miroir réfléchissant face à la caméra pour obtenir une image qui est la projection omnidirectionnelle 2D de la scène observée. Elle permet de fournir une vue globale de 360° de l'environnement en une seule acquisition.

Les images catadioptriques présentent des distorsions radiales importantes à cause de l'utilisation des miroirs de révolutions. Tenir compte de ces distorsions lors du traitement est le sujet de nombreux travaux de recherche durant ces dernières années. L'apparition du modèle unifié de Geyer [Geyer 1999] a résolu le problème qui est: toute caméra omnidirectionnelle centrale peut être assimilée à un modèle de projection sphérique. Néanmoins le problème du volume d'information demeure.

Dans ce mémoire nous avons traité des images omnidirectionnelles provenant de la caméra para-catadioptrique, qui représentent des volumes de données numériques important, ainsi que leurs images sphériques équivalentes. Pour archiver, analyser et diffuser ces informations générées par cette technologie on a eu recours aux techniques de compression. Ces dernières, permettent de réduire le nombre de bits représentant un pixel tout en sauvegardant une bonne qualité visuelle de l'image. On distingue deux types de méthodes de compression d'image, la compression sans perte (réversible) et celle avec perte d'information (irréversible)

## Introduction

L'objectif de ce travail est de faire une étude comparative entre la compression des images omnidirectionnelles brutes et la compression des images sphériques équivalentes, pour plusieurs séquences d'images de synthèses et réelles.

Les techniques avec pertes sont les seules qui peuvent répondre à cette exigence et pour cela nous avons dû choisir la technique la plus adéquate pour cette problématique, nous avons choisi le standard JPEG 2000.

Le mémoire est organisé en quatre chapitres répartis comme suit :

Le premier chapitre, ou nous avons présenté la vision omnidirectionnelle, les différentes technologies utilisées ainsi que la modalisation et le calibrage des caméras catadioptriques

Le deuxième chapitre, nous avons exposé les différentes méthodes de compression d'image fixe.

Le troisième chapitre, présente les différentes étapes du standard JPEG 2000 qui seront appliquées aux images omnidirectionnelles brutes ainsi qu'aux images sphériques équivalentes.

Le quatrième et dernier chapitre illustrera les résultats des tests effectués ainsi que leurs interprétations. Nous terminerons ce travail par une conclusion et quelques perspectives.

# *Chapitre I*

## *La vision omnidirectionnelle*

## I.1 Préambule

Quelque soit l'information apportée par une image classique à deux dimensions 2D, sa pauvreté en champ de vision demeure son inconvénient majeur, car toutes les recherches dans les domaines militaire et civil (médecine, biologie, météorologie, agriculture...) se basent sur ce champ de vision large afin de pouvoir fournir une vue globale de l'environnement en une seule acquisition.

Les progrès récents dans la technologie des systèmes de vision ont conduit à l'élaboration des capteurs catadioptriques capables de collecter des images avec un champ de vision large couvrant les 360°. Ces images contiennent des informations globales sur l'environnement 3D. Elles permettent aussi de lever l'ambiguïté due aux mouvements engendrés par les caméras perspectives.

## I.2 Définition d'une image

L'image en termes physique, désigne la représentation d'un objet produite par la réunion des rayons ou faisceaux lumineux qui en proviennent et se reconstituent sur un miroir, écran ou sur l'œil humain qui perçoit cette image.

L'image peut être décrite sous la forme d'une fonction  $I(x, y)$  de brillance analogique continue, définie dans un domaine borné, tel que  $x$  et  $y$  sont les coordonnées spatiales d'un point de l'image et  $I$  est une fonction d'intensité lumineuse et de couleur. L'image numérique est l'image dont la surface est divisée en éléments de tailles fixes appelés cellules ou pixels (Picture élément en anglais). Le pixel est l'élément de base d'une image numérique, ayant chacun comme caractéristique un niveau de gris ou de couleur prélevée à l'emplacement correspondant dans l'image réelle, ou calculé à partir d'une distribution interne de la scène à représenter. La numérisation d'une image est la conversion de son état analogique (distribution continue d'intensités lumineuses dans un plan  $(xOy)$ ) en une image numérique représentée par une matrice bidimensionnelles de valeur numérique  $f(x, y)$  (voir figure 1.1).

Il y a quatre formes physiques d'existence d'une image : forme binaire, en niveau de gris, couleur et multi spectrale.

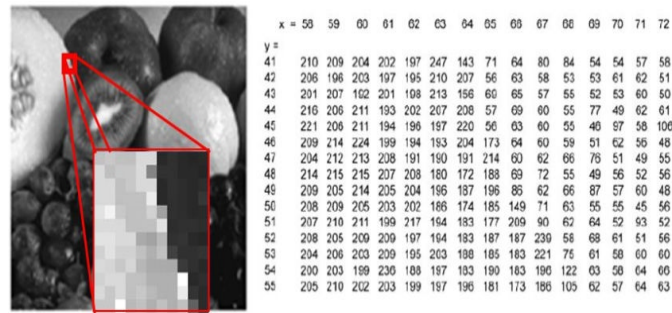


Figure I.1 : représentation d'une image numérique

### I.2.1 Image binaire

Une image binaire est une image dont les pixels ne peuvent avoir pour valeur binaire '0' ou '1'. Le '0' correspond à un pixel noir et le '1' correspond à un pixel blanc comme illustré dans la figure 2 le pixel est codé sur 1 bit, dans ce cas le niveau de gris ( $N_g=2$ ) et la résolution sur les niveaux de gris devient :

$$p(i, j)=0 \text{ ou } p(i, j)=1.$$


Figure I.2 : Image binaire

### I.2.2 Image en niveau de gris

Une image en niveau de gris autorise un dégradé de gris entre le noir et le blanc. En général, on code le niveau de gris sur un octet soit '256' nuances de dégradé. L'expression de la valeur du niveau de gris avec  $N_g=256$  devient  $p(i, j)=[0 - 256]$ . Par convention le '0' représente le noir et la valeur '255' représente le blanc.



Figure I.3 : image en niveau de gris

### I.2.3 Image couleur

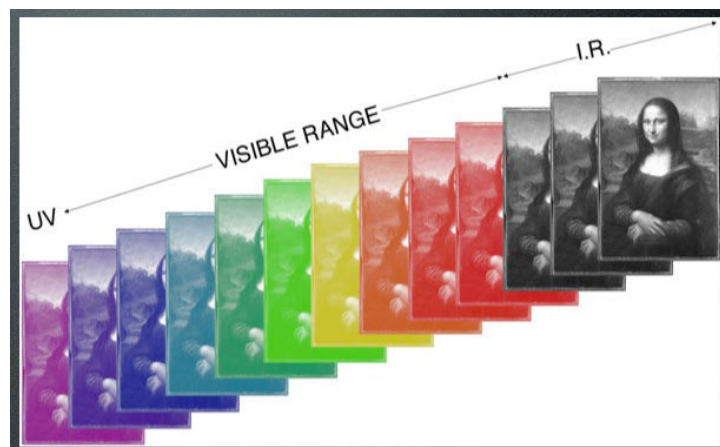
Une image couleur contient trois plans de couleur, le rouge, vert et le bleu (R, V, B) chaque plan est codé comme une image en niveau de gris, avec des valeurs allant de 0 à 255. Lorsque  $R=V=B$ , la valeur associée est un niveau de gris.



**Figure I.4** : Image couleur

### I.2.4 Image multi spectrale

En imageries multi-spectrale, on utilise plus d'une image chaque image correspondant à une bande très étroite du spectre. On a ainsi une définition beaucoup plus précise de la lumière réfléchiée par une surface et on peut ainsi accéder à des caractéristiques non visibles à l'œil nu. Les bandes spectrales peuvent être choisies en fonction des longueurs d'ondes caractéristiques des matières ou des produits à analyser.



**Figure I.5** : Image multi spectrale

## I.3 La vision omnidirectionnelle

les systèmes de vision classiques sont basés sur l'utilisation d'une caméra CCD (Charge Coupled Device). L'utilisation d'une seule camera ne peut fournir qu'une seule information 2D qu'on appelle vision monoculaire. Pour obtenir des informations 3D, il existe

différentes techniques qui sont généralement liées à l'adjonction d'un autre capteur; par exemple: l'utilisation de deux caméra nous permettent de reconstruire la structure 3D des objets avec deux images, on l'appelle: la stéréovision. On peut aussi ajouter un système de réflexion de type miroir donnant ainsi la vision omnidirectionnelle.

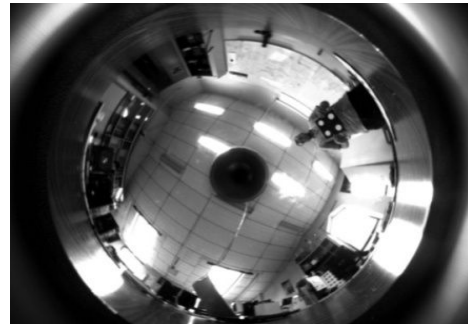
Donc, nous pouvons dire qu'on peut identifier trois techniques: La vision monoculaire, la stéréovision et La vision omnidirectionnelle.

Dans le cadre de ce mémoire, nous nous intéressons à la vision omnidirectionnelle.

Le terme de la vision omnidirectionnelle se réfère au capteur de vision avec un très large champ de vue, c'est-à-dire un capteur de champ de vue de 360°. Comme il existe un autre terme pour un champ de vue large c'est la vision panoramique.



**Figure I.6** : Image panoramique



**Figure I.7** : image omnidirectionnelle

Plusieurs techniques ont été développées pour augmenter le champ de vue panoramique et omnidirectionnel que l'on peut classer en trois catégories

- l'utilisation de plusieurs images pour former un panorama.
- l'utilisation d'objectifs grands angles (fish-eye).
- l'utilisation d'un miroir (catadioptrique).

### **I.3.1 Fusion de plusieurs images**

La première catégorie est basée sur l'utilisation de plusieurs images pour former une image panoramique. Cependant ce système pose plusieurs problèmes en pratique.

Deux techniques peuvent être mises en œuvre pour l'acquisition des images:

- La rotation d'une caméra autour d'un axe
- Utilisation de plusieurs caméras.

**a/ Systèmes rotatifs**

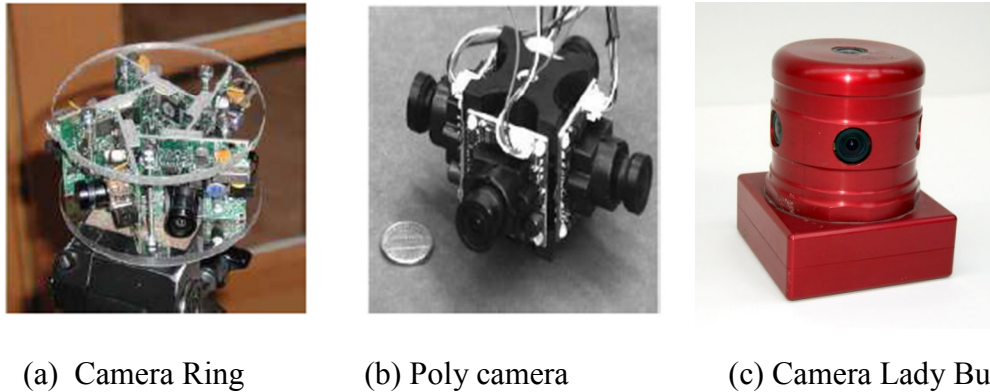
La première technique consiste à effectuer une acquisition d'une image par rotation d'une caméra autour d'un axe vertical fixe à une vitesse angulaire constante pour acquérir plusieurs images de la scène qui sera ensuite assembler en image panoramique cette technique de reconstruction est appelée mosaïque d'images. L'avantage d'utiliser cette technique est de fournir une très grande résolution à l'image. L'inconvénient réside dans l'impossibilité de l'utilisée pour des scènes en mouvement. La scène doit donc rester fixe durant toute la phase d'acquisition. En effet si une personne change de place lors de la prise de vue consécutive, elle risque d'apparaître plusieurs fois sur l'image à la place qu'elle a occupée successivement.



**Figure I.8** : Exemple de caméra en rotation autour d'un axe vertical fixe

**b/ Systèmes de caméras multiples**

Une autre manière d'agrandir le champ d'observation d'un capteur consiste en l'augmentation du nombre de capteurs le constituant. L'acquisition des images est synchronisée pour toutes les caméras, ce qui permet de réduire le temps d'acquisition



(a) Camera Ring

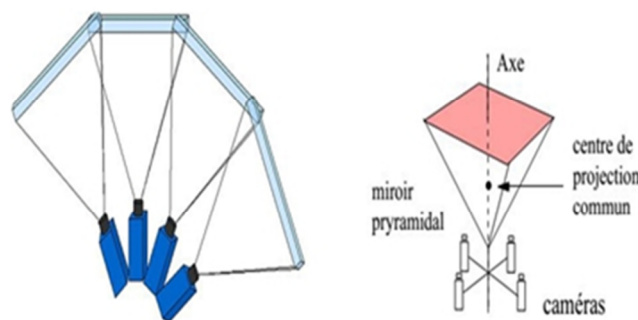
(b) Poly camera

(c) Camera Lady Bug

**Figure I.9** : Systèmes de caméras multiples

Ces caméras permettent d'atteindre des résolutions raisonnables, mais l'inconvénient réside d'une part dans les angles morts, c'est-à-dire les parties de l'environnement qui ne sont pas acquises par le système, et d'autre part dans les zones de l'environnement qui sont dupliquées à cause de l'intersection de l'angle de vue entre les caméras adjacentes. Son prix qui reste très cher. La difficulté de synchroniser toutes les caméras et l'importance du flux de données à traiter.

Une autre technique complètement différente peut être trouvée dans le travail de [Nalwa 1996] appelée fullview en plus des quatre caméras l'auteur utilise quatre miroirs triangulaires pour acquérir quatre images. Ces systèmes s'affranchissent pas aux problèmes liée aux différentes expositions, à moins que le contrôle d'exposition des différentes caméras soit centralisé.

**Figure I.10** : Système Fullview

### I.3.2 Objectifs grand angle

Les objectifs grands angles font souvent référence à l'objectif « fish-eye » (œil de poisson) qui se fixe sur les caméras standards à la place des objectifs classique (figure I.11). Cette lentille dévie les rayons lumineux et offre un angle d'ouverture important ( $>180^\circ$ ). Les

images acquises par ce genre d'objectif sont déformées, et présente une faible résolution principalement sur la périphérie. Elles ne peuvent pas être transformées en images perspectives, car les objectifs fish-eyes n'ont pas un centre de projection unique, et ils sont difficiles à modéliser.



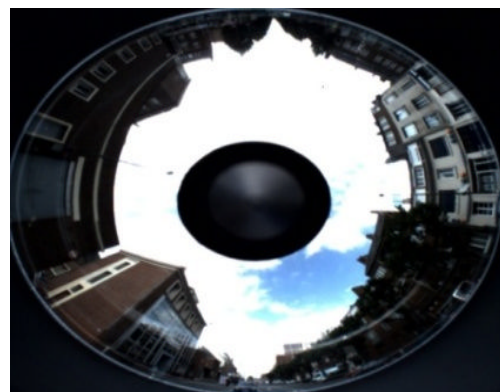
**Figure I.11** : Image acquise par une caméra fish-eye

### I.3.4 Système catadioptrique

Les caméras qui utilisent uniquement l'effet de réfraction de la lentille de refléter la lumière sont appelés caméras dioptriques (la dioptrie est la science des éléments de réfraction). Les caméras qui utilisent les effets combinés de la réflexion sur un miroir et de réfraction d'une lentille sont appelées les caméras catadioptriques (à partir de catoptrique, c'est à dire la science des surfaces de réflexion et dioptrique). Un capteur catadioptrique (figure I.12) est composé d'une caméra et d'un miroir ayant en général un axe de révolution.



**Figure I.12** : système catadioptrique



**Figure I.13** : exemple d'image omnidirectionnelle

Nous pouvons distinguer plusieurs types de capteurs catadioptriques selon la nature de la surface du miroir convexe utilisée. Selon [Naya et Bakerles] différents types de capteurs

catadioptriques sont partagés en deux catégories : systèmes catadioptriques centraux et non centraux

### I.3.4.1 Caméras catadioptriques centrales

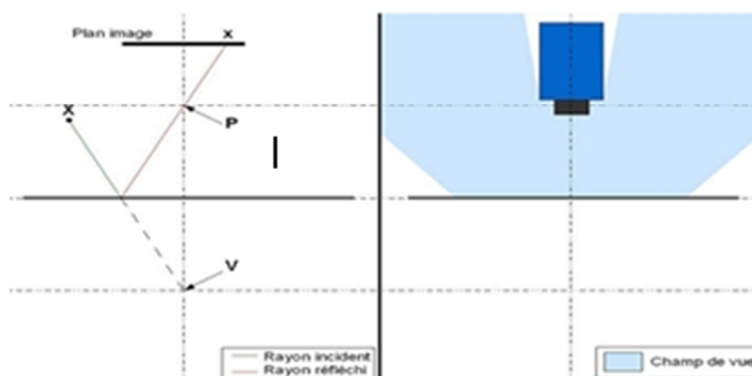
Les caméras catadioptriques centrales possèdent un Point de Vue Unique (PVU) dans lequel chaque rayon lumineux de la scène observée passe dans une seule direction à travers ce point de vue unique.

Les capteurs catadioptriques admettant un point de vue unique sont les capteurs disposant des miroirs plans, miroirs coniques, miroirs hyperboloïdes convexes et miroirs paraboloides.

Nous allons donner un bref aperçu sur les types de miroir omnidirectionnels les plus utilisés dans la pratique. Seules certaines formes de miroir de révolution permettent de conserver la contrainte de point de vue unique, à savoir les miroirs sphériques, ellipsoïdes et hyperboloïdes avec une caméra perspective et le miroir paraboloides avec une caméra orthographique [Baker99]. Ces formes ont été largement utilisées dans les robots mobiles qu'elles peuvent être appelées les formes classiques de miroir. Chacune de ces formes de miroir réalisent une cartographie différente entre les points du monde et les points d'image.

#### a- Le miroir plan

Le point de vue unique du miroir plan est situé sur l'axe optique à une distance égale à celle séparant le miroir du centre optique de la caméra



**Figure I.14:** Caméras catadioptriques : Miroir plan.

L'utilisation d'un seul miroir plan et une caméra ne permet pas d'augmenter l'angle de champ de vue puisque celui-ci sera identique à celui de la caméra. Pour augmenter l'angle de champ de vue avec la conservation d'un point de vue unique il faut multiplier le nombre de miroir et le nombre des caméras à condition d'associer une caméra à chaque miroir. [Nalwa96] a réalisé un capteur panoramique à l'aide de quatre miroirs plans disposés en pyramide et de quatre caméras CDD. Il a réussi à obtenir un point de vue unique et un champ de  $360^\circ \times 50^\circ$ .

### b- Le miroir conique

Le miroir conique possède un point de vue unique confondu avec le centre optique de la caméra P au sommet du cône.

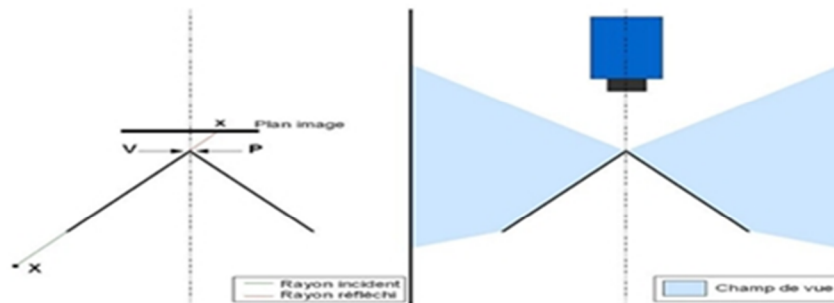


Figure I.15: Caméras catadioptriques : Miroir conique.

### c- Le miroir ellipsoïde

Le miroir ellipsoïde respecte la contrainte du point de vue unique lorsque le centre Optique P de la caméra est situé sur le second foyer de l'ellipsoïde, le premier étant le Point de vue V

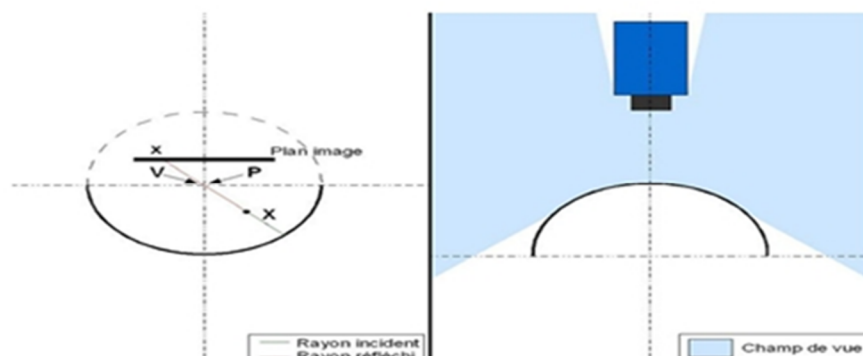


Figure I.16: Caméras catadioptriques : Miroir ellipsoïde.

#### d- Le miroir sphérique

Ce type de vision présente la particularité d'inclure les deux points dans le centre de la sphère. Donc, le point de vue unique  $V$  et le centre optique  $P$  sont confondus. L'image obtenue par ce système a une bonne résolution dans la région centrale et une faible résolution dans la région périphérique.

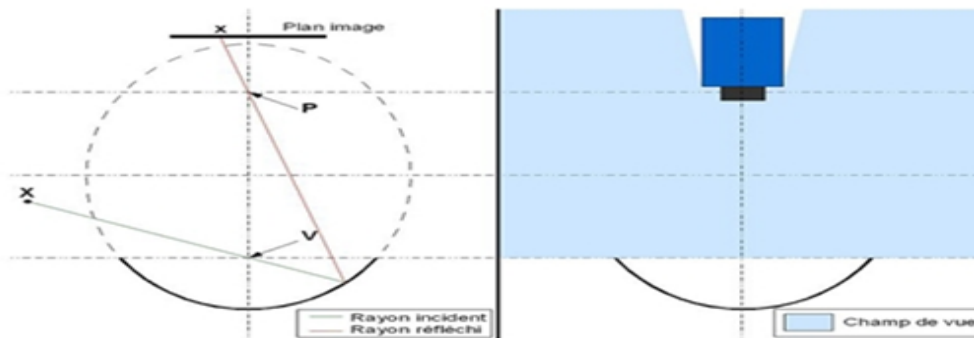


Figure I.17: Caméras catadioptriques : Miroir sphérique.

#### e- Le miroir hyperboloïde

L'avantage de ce miroir "miroir hyperboloïde" est le point de vue unique  $V$  coïncide avec le centre optique  $P$  (figure I.18) Cet avantage permet de reconstruire une image libre de distorsion et elle donne une bonne résolution dans la zone centrale.

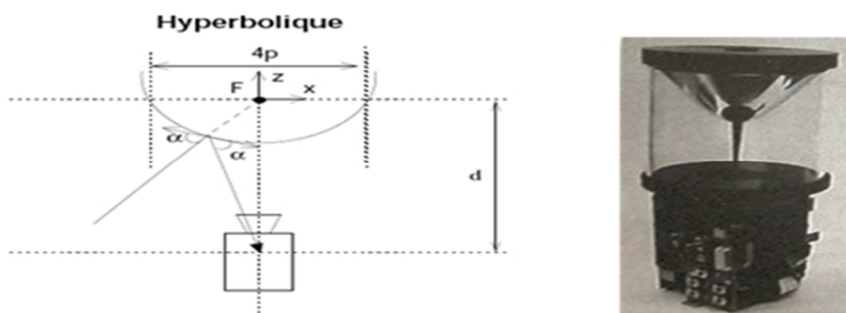


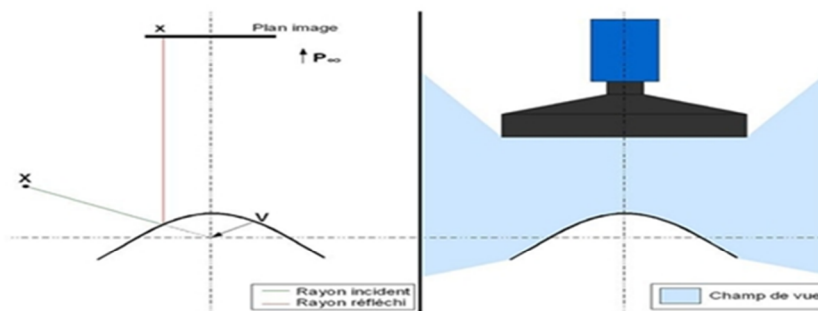
Figure I.18: Caméras catadioptriques : Miroir hyperboloïde.

A partir de l'image d'un miroir hyperboloïde, il est possible de reconstituer un panoramique

cylindre ou des images perspectives à des angles désirés comme l'illustre la figure ci-dessus

#### f- Le miroir parabolique

Le miroir parabolique présente un seul point de vue lorsqu'il est utilisé avec un appareil photo de projection orthographique, c'est-à-dire les rayons qui passent par le point de vue  $V$  sont réfléchis parallèlement à l'axe optique. Il est facile de réaliser le calibrage et le calcul des images perspectives est simple.

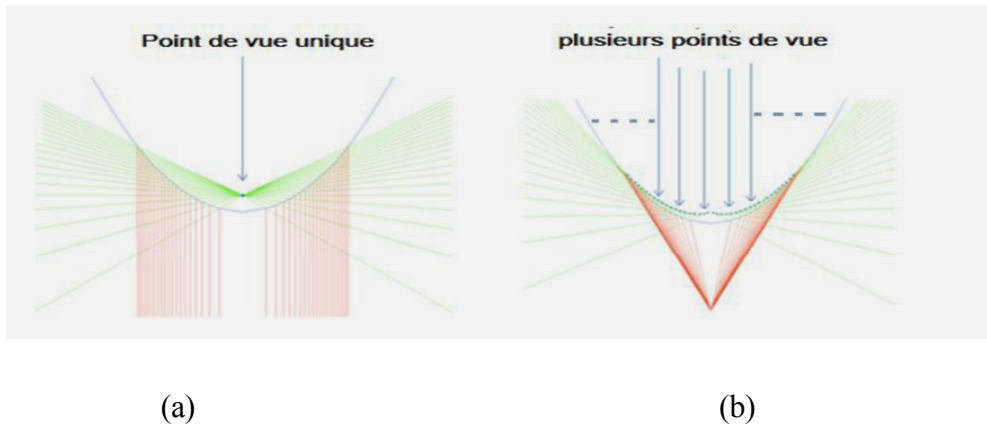


**Figure I.19:** Caméras catadioptriques : Miroir parabolique.

#### I.3.4.2 Capteurs catadioptriques non centraux

Dans le cas des capteurs catadioptriques non centraux, la réflexion des points sur le miroir (ou sur le dioptre) forme une ligne ou une surface qu'on appelle une caustique (voir figure I.20). Les deux miroirs qui peuvent être classés dans cette catégorie sont le miroir conique et sphérique.

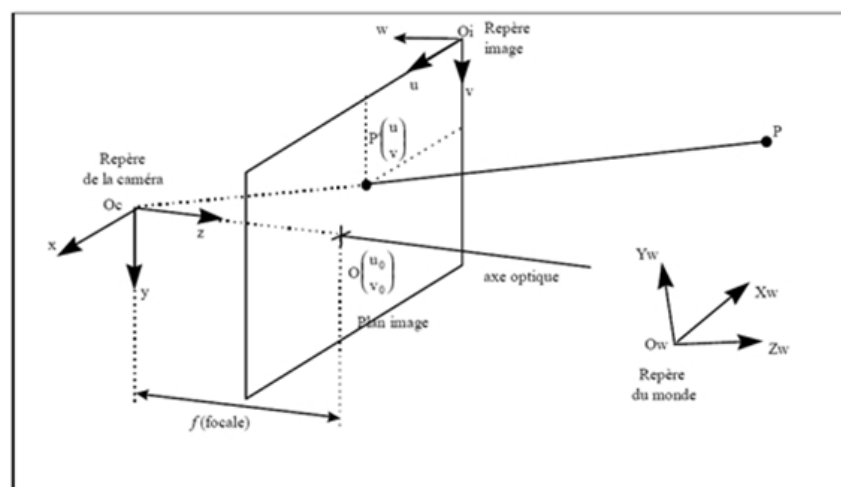
Une fois combiné avec une caméra perspective, le miroir conique produit un système non central avec une caustique ayant la forme d'un cercle. Si la focale de la caméra coïncide avec le sommet du cône, alors le capteur composé d'un miroir conique et une caméra perspective satisfait la contrainte du PVU. Cette solution ne peut être employée pour élargir le champ visuel. La solution communément utilisée consiste à éloigner la caméra du cône. Le point de vue n'est plus unique : il s'agit d'un cercle dont le rayon dépend de la distance entre le point de vue et le sommet du cône, ainsi que de l'angle au sommet du cône.



**Figure I.20 :** (a) Système avec un PVU constitué d'un miroir parabolique avec une caméra orthographique (b)- système avec un point de vue non unique composé d'un miroir parabolique et d'une caméra perspective.

#### I.4 Modélisation d'une caméra

Un modèle géométrique de caméra modélise le processus de formation de l'image au niveau géométrique et permet d'exprimer les relations existantes entre les points 3 D d'une scène et leurs projections sur l'image de la caméra. Plusieurs modèles décrivant le processus de formation des images existent. Le plus simple et couramment utilisé est le modèle du sténopé qui correspond au centre de la caméra par lequel passe tous les rayons lumineux de façon rectiligne. Le modèle dépend de deux paramètres : intrinsèques l'ensemble des paramètres modélisent la géométrie interne et les caractéristique optiques du capteur et extrinsèques l'ensemble associé à la scène où se trouvent les objets de référence utiles pour la calibration.



**Figure I.21 :** représentation géométrique des repères utilisés pour la détermination d'un modèle de caméra

La simplicité du modèle réside dans sa construction géométrique un point 3D ( $P$ ) se projette sur le plan image en un point 2D ( $p$ ) par une projection centrale de centre  $O$ . Dont les coordonnées sont exprimées dans le repère du monde ( $P_xw, P_yw, P_zw$ ) et un second repère, ayant pour origine  $O_c$  et comme axes ( $X, Y, Z$ ), placé sur le centre optique de la caméra comme le montre la (figure I.21).

Les paramètres employés dans la modélisation sont usuellement les paramètres intrinsèques qui sont internes à la caméra, et les paramètres extrinsèques qui peuvent varier suivant la position de la caméra dans l'espace de travail. Parmi les paramètres intrinsèques, on compte :

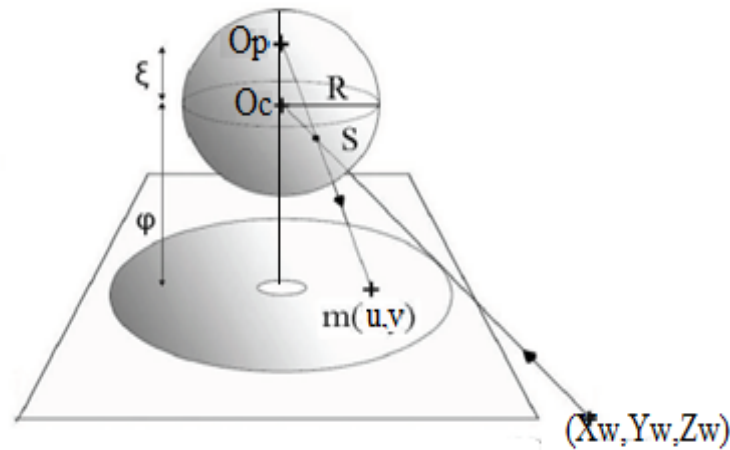
- $f$  : la distance focale ;
- $K_u$  et  $K_v$  : les facteurs d'agrandissement de l'image ;
- $u_0$  et  $v_0$  : les coordonnées de la projection du centre optique de la caméra sur le plan image ;

#### **I.4.1 Modélisation d'une caméra catadioptrique**

La modélisation d'une caméra catadioptrique relève de la même démarche que celle des caméras perspectives, excepté les rayons lumineux qui sont réfléchis par le miroir de révolution. Cette réflexion introduit une transformation supplémentaire dont il faut tenir compte. La transformation dépend de la surface du miroir.

#### **I.4.2 Modèle de projection unifié**

Tous les miroirs catadioptriques centraux satisfont la contrainte du point de vue unique. (Geyer et Daniilidis) ont introduit un modèle de projection unifié de traitement d'images omnidirectionnelles pour l'ensemble des capteurs catadioptriques centraux. Ils ont démontré que la projection catadioptrique centrale est équivalente à une projection centrale à une sphère virtuelle.



**Figure I.22** : Equivalence entre la projection catadioptrique centrale et la projection sur la sphère unité

Deux paramètres sont nécessaires pour associer un point 3D  $S(X_s ; Y_s ; Z_s)$  avec sa projection sur l'image plan  $m(x,y)$  (équation I.3 et I.4). Le paramètre  $\varphi$  représente la distance entre l'image omnidirectionnelle et le centre  $O_c$  de la sphère alors que le paramètre  $\varepsilon$  représente la distance  $O_p - O_c$ . Par exemple, pour un miroir parabolique le point  $O_p$  se trouve sur le pôle nord de la sphère.

$$\frac{X_s}{\varepsilon - Z_s} = \frac{x}{\varepsilon + \varphi} \quad \text{I.1}$$

$$\frac{Y_s}{\varepsilon - Z_s} = \frac{y}{\varepsilon + \varphi} \quad \text{I.2}$$

Où  $(x,y)$  sont les coordonnées de  $P_i$  dans le système métrique,  $\varepsilon$  et  $\varphi$  définissent la forme du miroir (voir Annexe B)

$$x = \frac{(\varepsilon + \varphi)X_s}{\varepsilon\sqrt{X_s^2 + Y_s^2 + Z_s^2} - Z_s} \quad \text{I.3}$$

$$y = \frac{(\varepsilon + \varphi)Y_s}{\varepsilon\sqrt{X_s^2 + Y_s^2 + Z_s^2} - Z_s} \quad \text{I.4}$$

Si on considère les paramètres intrinsèques de la caméra ( $K_u$ ;  $K_v$ ;  $u_0$  ;  $v_0$ ) On obtient alors les coordonnées des points de l'image exprimées en pixels

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{pmatrix} k_u & 0 \\ 0 & k_v \end{pmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} \quad \text{I.5}$$

$$u = k_u \cdot x + u_0 = \frac{k_u (\varepsilon + \varphi) Xs}{\varepsilon \sqrt{X^2 s + Y^2 s + Z^2 s} - Zs} + u_0 \quad \text{I.6}$$

$$v = k_v \cdot y + v_0 = \frac{k_v (\varepsilon + \varphi) Ys}{\varepsilon \sqrt{X^2 s + Y^2 s + Z^2 s} - Zs} + v_0 \quad \text{I.7}$$

Les paramètres du modèle sont :

- $\xi$ , qui dépend de l'excentricité (voir annexe B)
- $\varphi$  qui est fonction de l'excentricité et de l'échelle (voir annexe B)
- $k_u$ ,  $k_v$ ,  $u_0$  et  $v_0$  sont les paramètres intrinsèques de la caméra.

### I.5 Calibrage des caméras catadioptriques

Le calibrage est une opération qui consiste à estimer les paramètres intrinsèques de la caméra. Dans le cas des caméras catadioptriques, ces paramètres englobent les paramètres du miroir, ceux du dispositif optique (objectif), ceux du capteur CCD (la rétine) et de la carte de numérisation du signal vidéo. Afin de pouvoir réaliser le calibrage, il est indispensable de modéliser tout le processus de formation de l'image. Le modèle doit faire l'objet d'un compromis entre simplicité et précision. La simplicité est indispensable à la résolution analytique ou numérique des équations et la précision est garante de la fidélité du comportement du modèle. Le calibrage en vision omnidirectionnelle est l'aspect le plus étudié (abouti) à l'heure actuelle. Nous proposons la classification suivante pour espérer une lecture plus aisée.

### I.5.1 Le calibrage avec des mires

Une mire de calibrage est généralement constituée de points de référence dont on connaît parfaitement les coordonnées 3D dans un repère lié à l'étalon. En générale les mires utilisées sont composées de plusieurs cercles ou carrés noirs sur fond blanc ou l'inverse afin d'obtenir le meilleur gradient possible. A partir des mires composées de formes circulaires, on calcule les centres de gravité de la projection de chaque cercle. Lorsque la mire est composée de carrés, ce sont alors généralement des segments de droites qui seront analysés ou les points des coins de chaque carré [Mouaddib 2005 ]

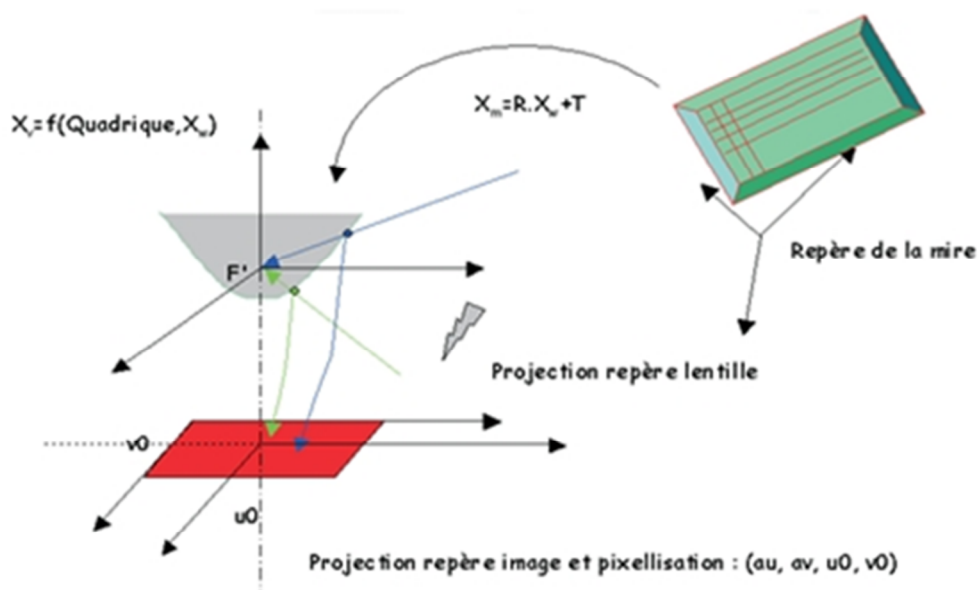
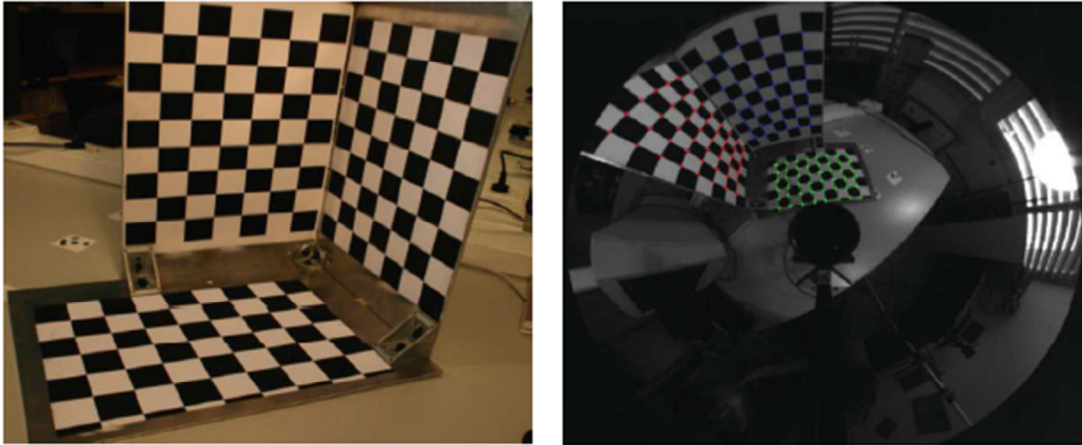


Figure I.23 : calibrage avec des mires

### I.5.2 Calibrage avec des points

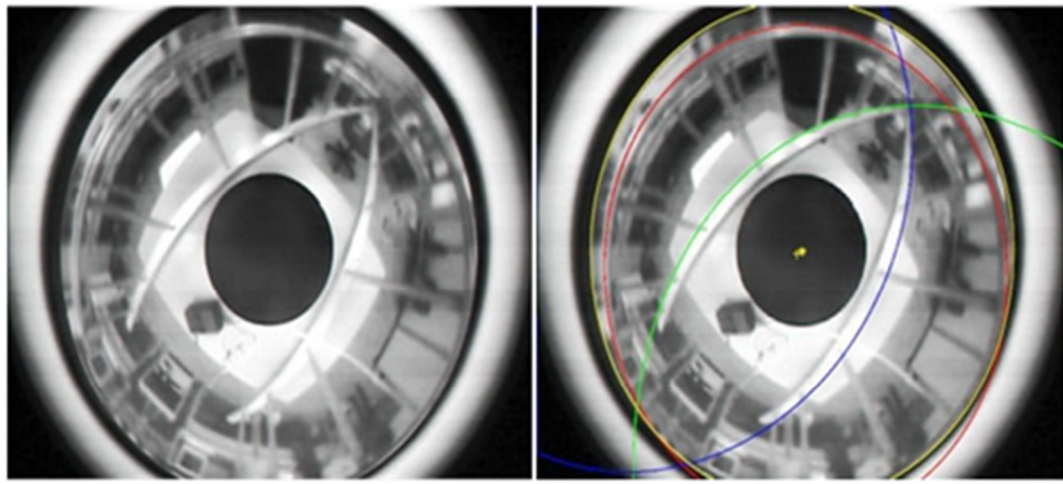
On utilise ici des mires comprenant des points connus dans un repère local [Cauchois et al.1999]. En utilisant les points images correspondants (voir la figure I.24), on peut résoudre numériquement le modèle, par une technique d'optimisation. L'avantage est que l'approche est utilisable pour tout type de miroir et de caméra et permet de traiter des modèles complets.



**Figure I.24** : calibrages avec des points

### I.5.3 Calibrage avec des droites

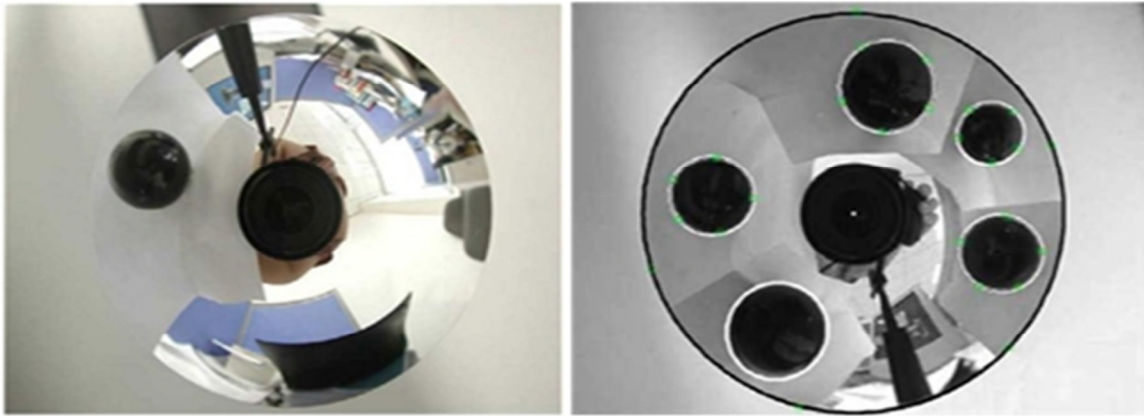
Contrairement aux méthodes utilisant les points, les méthodes qui ont recours aux droites cherchent des propriétés intéressantes des capteurs catadioptriques pour simplifier le calibrage (voir la figure 25). Ainsi dans [Geyer et al.1999], les auteurs décrivent un algorithme de calibrage de caméras para-catadioptriques (miroir parabololoïde) basé sur l'utilisation d'ensembles de droites parallèles. Les propriétés géométriques et l'hypothèse faites par les auteurs, d'alignement entre l'axe du miroir et celui de la caméra, rendent possible l'estimation des paramètres intrinsèques



**Figure I.25** : Calibration avec des droites [Barreto 2002]

### I.5.4 Calibrage avec des sphères

Les Techniques de calibration basées sur les sphères détectent d'abord des formes sphériques et puis leurs contours sont exploités pour calibrer le capteur. Pour la calibration des images omnidirectionnelles en utilisant les forme sphériques, Ying et Hu [YH03] ont analysé la relation entre les paramètres intrinsèques de la caméra et les contours des sphères dans les images omnidirectionnelles.



**Figure I.26** : Calibration avec des sphères

## **I.6 Discussion**

Dans ce chapitre, nous avons donné quelques définitions de l'image omnidirectionnelle et les différentes solutions pour atteindre une vision large ainsi les méthodes de modélisation.

L'utilisation des images omnidirectionnelles est de plus en plus importante en vue qu'elles fournissent des images couvrant un champ de vision augmenté de 360°.

Cette augmentation du champ de vision est accompagnée par un volume de donnée important causant ainsi des problèmes de stockage et de transmission, ce qui a rendu l'étude de la compression des données omnidirectionnelles un sujet de recherche toujours en évolution.

La compression des images omnidirectionnelles sera l'objectif de ce travail, le chapitre suivant nous renseignera sur quelques méthodes et normes de compression d'images utilisées en compression d'images perspectives.

# *Chapitre II*

## *Généralités sur la compression d'image*

## II.1 Préambule

De nos jours l'image constitue l'un des moyens les plus importants qu'utilise l'homme pour communiquer avec autrui. C'est un outil de communication universel, dont la richesse du contenu permet aux êtres humains de tout âge et de toutes cultures de se comprendre. Manipuler une image dans un quelconque domaine revient à son analyse avec différents logiciels de traitement d'images qui permettent sa numérisation afin d'en extraire le plus d'information.

La quantité d'informations récupérées des images a considérablement augmenté grâce au développement des techniques et des instruments de leurs acquisitions.

L'évolution de l'image allant d'une image deux dimensions 2D à niveau de gris à une image couleur, et même de nouvelles extensions nous ont donné des images avec une troisième dimension omnidirectionnelle 3D acquises par des capteurs catadioptriques a provoqué une augmentation de leur taille mémoire. De ce fait le stockage et la transmission de ces images constitue le principal enjeu et pour cela la compression de ces données s'impose comme étape indispensable.

## II.2 Définition de la compression

La compression consiste à réduire la taille physique des blocs d'informations, elle s'appuie sur l'analyse du contenu de l'image et on tire profit de son organisation interne, afin d'en éliminer les données redondantes qu'elles soient temporelles, spatiales ou statistiques.

Dans la pratique, on distingue deux classes de compressions : ceux qui peuvent reconstituer l'information exacte (méthodes réversibles) et celles qui tolèrent une perte d'information (méthodes irréversibles).

## II.3 Principe général de la compression d'image

Le processus de compression d'une image nécessite trois étapes principales (figure 1), à savoir, l'étape de transformation ou de décorrélation, l'étape de quantification et l'étape de codage entropique

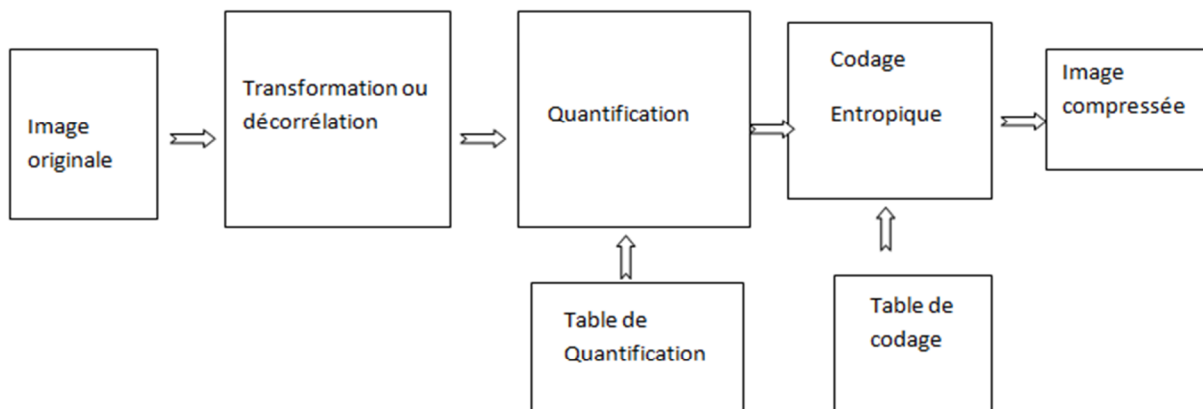


Figure II.1 : Schéma synoptique de compression d'images fixe

### -Etape de transformation ou de décorrélation

Cette étape a pour objectif de réduire la redondance d'informations contenues dans l'image, afin d'obtenir une représentation de l'image qui se prête à la quantification et au codage. De plus, elle nous offre la possibilité d'ajuster les erreurs de quantification selon la sensibilité du système visuel humain. Cette transformation est obtenue en exploitant la forte corrélation spatiale entre les pixels de l'image. C'est une opération réversible qui a pour effet de redistribuer l'énergie de l'image originale en un nombre restreint de coefficients transformés.

**-Etape de quantification**

Cette étape permet de réduire le nombre de bits nécessaires à la représentation des coefficients issus de l'étape de transformation. Les valeurs de ces coefficients sont approximées par un ensemble fini d'éléments (scalaire ou vecteur) qui forme le dictionnaire de quantification. Ainsi, un élément quelconque d'entrée est remplacé par l'élément du dictionnaire le plus approprié.

**-Etape de codage entropique**

C'est la phase finale du processus de compression, qui permet de générer le flux binaire à stocker ou à transmettre. Notons que le codage entropique est un codage sans pertes.

**II.4 Les méthodes de compression sans perte (réversible)**

La compression est dite sans perte lorsqu'il n'y a aucune perte de données sur l'information d'origine. Il y a autant d'informations après la compression qu'avant elle est seulement réécrite d'une manière plus concise. La compression sans perte est dite aussi compactage ou réversible.

Les méthodes de compression sans pertes d'informations permettent de faible taux de compression. Elles se résument en l'élimination de la redondance, qui permet de réduire la taille de l'image sans altération de la qualité de celle-ci. Ce groupe de méthodes peut être divisé en deux catégories. Les méthodes qui codent les pixels ou les coefficients issus des transformations individuellement. Parmi ces méthodes nous distinguons le codage de Huffman et Shannon-Fano. La seconde partie des méthodes sont celles qui codent des séquences de symboles tels que le codage arithmétique, le codage par dictionnaires LZW et le RLE.

**II.4.1 Codage par symboles individuels**

C'est des techniques désignées aussi par le nom « code à longueur variable », chaque symbole est codé par un code de longueur inversement proportionnel à sa probabilité d'apparition. Les valeurs fréquentes prennent moins de place. Les codages les plus utilisés sont ceux de « Shannon-Fano », « Huffman », « prédictifs » et « par plan de bit ».

### II.4.1.1 Codage de Shannon-Fano

Le codage de Shannon-Fano est la première méthode de codage entropique efficace, développé en même temps par Claude Shannon et Robert Fano en 1949. Elle consiste à calculer les fréquences d'apparition de chaque symbole contenu dans la scène à coder puis, construire un arbre descendant à partir de la racine par divisions successives. Le classement des fréquences se fait par ordre croissant. A chaque division deux sous-ensembles sont obtenus et ils doivent être de probabilité aussi proche que possible. On affecte ensuite un « 0 » au sous-ensemble gauche et un « 1 » au sous-ensemble droit. Le processus est itéré jusqu'à ce que chaque sous-ensemble ne contienne qu'un seul élément. Pour décoder les symboles, il suffit de lire les codes à partir de la racine et descendre vers les feuilles. Le codage de Shannon-Fano est un codage entropique dont le but est de s'approcher le plus possible de l'entropie  $H_0$  de l'image initiale. Il permet de déterminer des codes VLC (Variable length coding) préfixés [Courellemont, 2000 ; Guillois, 1996 ; Schanon, 1948].

Ce type de codage peut être illustré par l'exemple suivant :

Soit une source de symbole  $\{S_1 S_2 S_3 S_4 S_5 S_6 S_7 S_8\}$  dont les probabilités sont respectivement : 0.40, 0.15, 0.15, 0.10, 0.10, 0.06, 0.02, 0.02.

L'application des étapes de codage de Shannon-Fano à cette source est illustrée dans le tableau 1 :

Symbole source	probabilité	1 <sup>er</sup> Partition	2 <sup>em</sup> partition	3 <sup>em</sup> partition	4 <sup>em</sup> partition	5 <sup>em</sup> Partition	codage	
S <sub>1</sub>	0.40	0.55	0.40				00	
S <sub>2</sub>	0.15		0.15				01	
S <sub>3</sub>	0.15	0.45	0.25				100	
S <sub>4</sub>	0.10						101	
S <sub>5</sub>	0.10		0.20					110
S <sub>6</sub>	0.06				0.06			1110
S <sub>7</sub>	0.02			0.04	0.02		11110	
S <sub>8</sub>	0.02				0.02		11111	

**Tableau II.1** : Exemple de codage de Shannon-Fano

### II.4.1.2 Codage de Huffman

En exploitant le fait que certains symboles soient susceptibles d'apparaître plus souvent que d'autre dans un fichier 'David Huffman', publie une autre technique qui permet de coder l'image sur un nombre restreints de bits [Huffman 1952].

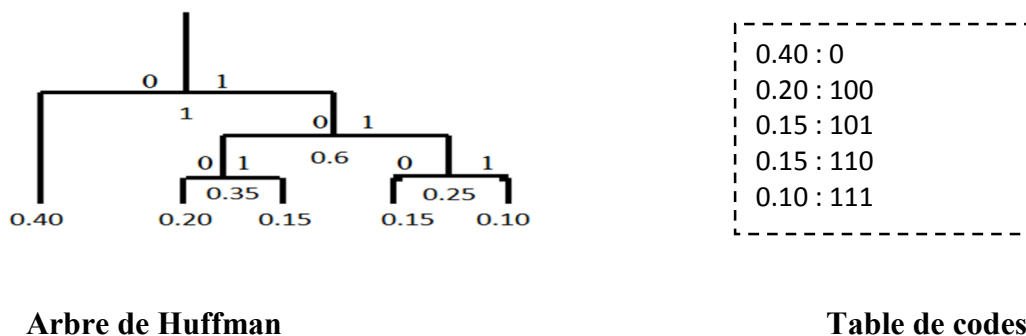
Le codage consiste en la construction d'un arbre en partant des feuilles vers racine contrairement au codage de Shannon-Fano.

Sa mise en œuvre est décrite par l'algorithme suivant :

- 1- Le tri des symboles par ordre de probabilités décroissant (par exemple de gauche vers la droite).
- 2- Successivement, les deux symboles de plus faible probabilité d'apparition sont rattachés à un nœud dont le poids vaut la somme des probabilités des deux symboles. A ces derniers sont affectés respectivement les chiffres binaires « 0 » pour le plus fréquent et « 1 » pour le plus rare.
- 3- Le même procédé est répété en considérant les nœuds comme des symboles jusqu'à obtenir un seul nœud parent appelé racine.
- 4- Le code de chaque symbole s'obtient en écrivant de gauche vers la droite les chiffres binaires rencontrés en parcourant l'arbre de la racine vers le symbole. Ainsi, plus le symbole est rare plus il est profond dans l'arbre, plus son mot de code est long.

Dans le codage de Huffman, la table des codes est enregistrée dans le fichier compressé pour être utilisé dans le décodage.

Exemple : soit une source de symboles ayant les probabilités suivantes : {0.40, 0.20, 0.15, 0.15, 0.10} ; l'arbre de Huffman est la table de codes correspondant sont :



**Arbre de Huffman**

**Table de codes**

**Figure II.2** : Exemple d'un codage de Huffman

### II.4.1.3 Codage prédictif

Les méthodes décrites ci-dessus ont pour effet de réduire la redondance entre les pixels voisins. Le codage prédictif est une autre manière de réduire cette redondance en prédisant les valeurs des pixels à partir de leurs voisins. Les pixels voisins du pixel 'y' considéré forment un vecteur  $x$  à partir duquel est calculée une prédiction  $\hat{y} = f(x)$ . une fois le prédicteur défini, le codage prédictif consiste à remplacer les valeurs  $y$  par l'erreur de prédiction  $e = y - \hat{y}$ . Le codage prédictif est performant lorsque les variations de l'erreur de prédiction ne sont pas importantes. Au niveau du décodeur les valeurs sont reconstruite en calculant la valeur de chaque pixel  $y$  selon la relation suivante :  $y = \hat{y} + e$ .

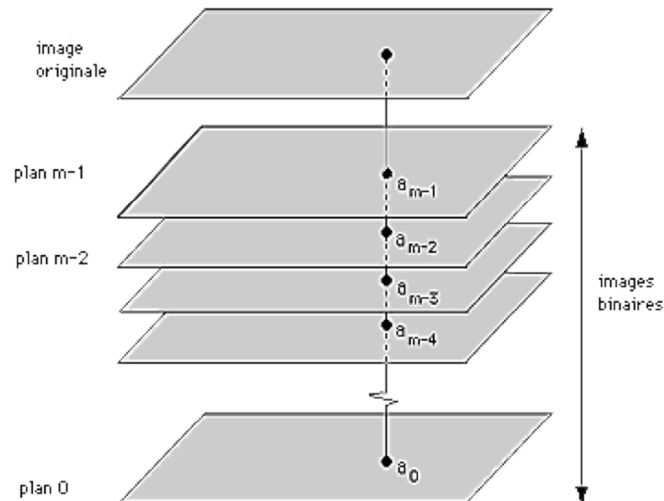
Remarquons que le codage prédictif est notamment utilisé dans le codage Differential Pulse Code Modulation (DPCM). [Totozafiny, 2007 ; Davoine, 1995]. De plus, cet algorithme offre l'avantage d'être simple à implémenter et s'exécute en un temps réduit, ce qui est très utile pour des applications en temps réel.

### II 4.1.4 Codage par plan de bit

Une image codée sur  $n$  niveaux de gris est  $m$  bits par pixel peut être considérée comme une superposition de  $m$  plans. Ces plans sont obtenus en isolant à chaque fois un bit de même poids pour chaque pixel. Chaque plan de bits (image binaire) peut être codé séparément en utilisant la méthode RLE.

Le codage réversible par plans de bits permet des taux de compression compris entre 15 et 20 mais présente l'inconvénient d'être sensible vis-à-vis des erreurs de transmission. Les plans composés des bits de poids fort contiennent la majeure partie de l'information visuelle de l'image, ils sont transmis en premier, puis l'image est raffinée en ajoutant les autres plans jusqu'au plan de bits le moins significatif. [Davoine, 1995 ; Waku Kouomo, 1992 ; Cziho, 1999]

La figure ci-dessous illustre les étapes du codage par plan de bits du bit le plus fort à gauche au bit le plus faible à droite.



**Figure II.3** : schéma de décomposition d'une image en plans de bits.

## II .4.2 Codage de séquences de symboles

La redondance des pixels, qu'elle soit dans le domaine spatial ou fréquentiel apparaît lorsque des séquences de symboles se répètent. Cette redondance peut être éliminée en codant directement les séquences. Parmi les types de codage les plus utilisés nous pouvons citer : le codage RLE, la méthode par dictionnaires (LZW) et le codage arithmétique.

### II.4.2.1 Codage Run Length Encoding (RLE)

Le Codage Run Length Encoding est très utilisé dans les méthodes par transformation et son efficacité dépend de la complexité de l'image. En effet, plus les transitions des niveaux de gris sont faibles plus le codage est meilleur et plus on atteint des taux de compression élevés. De plus, il est simple à implémenter et rapide d'exécution [Lahdir,1999 ;Guillois 1996].

Remarquons toutefois, que ce type de codage est moins efficace que les précédents pour le codage des images naturelles, car la possibilité de trouver deux niveaux de gris ayant la même valeur est faible.

Très sommairement, ce codage consiste à coder une plage de pixels ayant la même valeur.

Chaque plage définit de façon unique une paire de nombre  $P = (\text{plage}, n)$  où :

- Plage est le nombre de points voisins possédant le même niveau de gris,

- $n$  est le niveau de gris commun à ces points.

Dans un bloc de l'image la direction de balayage pour définition d'un code RLE se fait soit horizontalement, verticalement ou bien par balayage zig zag comme est le cas pour le standard JPEG.

Ce type de codage peut être illustré par l'exemple suivant :

Soit la suite suivante prélevée dans une ligne d'image :

11, 11, 15, 16,16, 16, 16, 25, 25, 25, 31, 31, 31,31, 31, 8.

Le codage par plage d'une telle suite donne la suite des paires suivantes :

(2, 11), (1, 15), (4, 16), (3, 25), (5, 31), (1, 8), Fournissant le nouveau code de la ligne :

2 11 15 4 16 3 25 5 31 18.

#### **II .4.2.2 Méthode par dictionnaire (Lemple-Ziv-Welch)**

Le code LZW, qui doit ses travaux à A.Lampel, J.Ziv et T.Welch, est un codage par dictionnaire, est utilisé lorsqu'on n'admet aucune perte d'information. Il est une amélioration des algorithmes LZ77, LZ78 et LZSS [Ziv et al, 1977 ; Zhang, 2004 ; Wyner et al, 1994 ;

Courtellemont, 2000].cette technique est utilisée pour la compression de toute sorte de document.

Le principe du codeur LZW consiste à coder une suite de pixels à l'aide d'un dictionnaire construit au fur et à mesure de la lecture des données. Le codage d'une image s'effectue selon les étapes suivantes :

- Découpage de l'image en chaînes de pixels (création du dictionnaire).
- Attribution d'un code à chaque chaîne.
- Stockage ou transmission du couple (code, chaîne).

Le décodeur fonctionne de manière symétrique : après initialisation du dictionnaire avec les symboles, il doit conserver en mémoire le contenu de 2 motifs : le motif courant et le précédent. Ainsi, en recevant le code suivant, il lui est toujours possible d'en extraire le 1<sup>er</sup> caractère et de le concaténer à droite du motif précédent pour former un nouveau motif qui sera intégré au dictionnaire qui se reconstruit dynamiquement.

En revanche, même si certains le considèrent comme un des algorithmes simples, il est loin derrière d'autres comme Huffman ou RLE au niveau de la simplicité d'utilisation. De plus, ce n'est pas un des algorithmes les plus rapides tant au niveau de la compression que de la décompression. Le taux de compression est directement proportionnel à la taille du dictionnaire choisie, il faut donc s'assurer de faire le bon choix dépendamment du fichier à compresser.

### II.4.2.3 Codage arithmétique

Le codage arithmétique est un codage récent utilisant un modèle statistique, tout comme le codeur de Huffman. Dans les formats de compression classique le codage arithmétique remplace de plus en plus le codage de Huffman. Pour comprendre le principe de cette technique, il peut être utile de considérer le flux binaire final comme la mesure de la suite des symboles, qu'il code, sur le segment des réels  $[0,1]$  [Guillois, 1996 ; Courtellemont, 2000 ; Messmoui et *al*, 2006 ; Witten et *al*, 1987, Rao, 2001 ; Tinku et *al*, 2000, Glen et *al*, 1984]. A chaque symbole, est alors associé un sous intervalle de  $[0, 1[$  de longueur égale à la probabilité  $P_i$  du symbole  $i$ .

Les intervalles sont construits de la manière suivante :

$$I_0 = [0; P_0[, I_1 = [P_0; P_1[, \dots, I_n = [P_{n-1}; P_{n+1}[$$

Le codage consiste à générer la suite d'intervalles emboîtés convergeant vers un réel  $R$  de  $]0,1[$  qui codera la source complète.

Le codeur arithmétique est plus performant que le codeur de Huffman vis-à-vis du taux de compression, mais il présente l'inconvénient d'être plus complexe en implémentation et plus lourd en temps de calcul.

Nous donnons ci-après un tableau récapitulatif résumant les caractéristiques des codages étudié dans cette partie.

Caractéristiques type de codage	Taux de compression (%)	Principaux points forts	Principaux points faibles
Shanon-Fanon	~27	-Codage adaptatif	-Non adapté aux chaînes aléatoires et de longueur variable
Huffman	25~35	-Codage à longueur variable	-Non adapté aux chaînes aléatoires et de longueur variable
RLE	30~40	-Simplicité d'implémentation. -Efficace dans le cas des images à zone uniformes.	-Inadaptée aux images complexes. -Nécessite un caractère spécial
LZW	30~40	-codage dynamique. -codage d'image à faible entropie.	-Temps d'exécution lent. -Sensible aux erreurs de transmission. -Inefficace dans le cas des images naturelles.
Arithmétique	~40	-Codage en virgule flottante	-Algorithme complexe. -Temps d'exécution très élevé.
Codage par plan de bits	15~20	-Simple à implémenter	-Sensible aux erreurs de transmission

**Tableau II.2** : caractéristiques des principales méthodes de compression réversibles.

Les méthodes décrites ci-dessus, permettent de coder les pixels individuellement ou par séquences. L'image restituée est une copie fidèle de l'image originale. Le tableau 2 montre que le codage sans perte le plus performant reste le codage arithmétique en terme de taux de compression.

## **II.5 Les méthodes de compression avec perte (irréversibles)**

A fin d'atteindre des taux de compression assez importantes avec des images complexes, une compression irréversible est nécessaire. Il s'agit alors de réaliser le meilleur compromis possible entre la quantité de l'image restitué et le taux de compression. Ces méthodes engendrent des pertes d'informations, même si elles sont indécélabes à l'œil nu. Elles sont utilisées dans le domaine spatial et fréquentiel (transformé). On peut les classer en deux classes à savoir, les méthodes par transformation et les méthodes hybrides.

### **II.5.1 Méthodes par transformation**

Les méthodes par transformation nécessite passage dans le domaine fréquentiel ou spatial pour mieux concentrer l'énergie dans un nombre restreint de coefficients et pour mieux adapter l'image à la quantification et au codage. L'application d'une transformation à l'image revient donc à lui changer de représentation.

En effet les pixels de l'image passent d'un espace où ils sont fortement corrélés à un espace où cette corrélation est minimisée. Nous allons détailler dans cette section les transformations qui ont connues un essor très important en compression d'images. Parmi ses transformations nous citons : la DFT (Discret Fourier Transform) la DCT (Transformation en Cosinus Discrète), la KLT (Transformation de Karhnnun Loève), la transformation en ondelettes .

#### **II.5.1.1 La transformée de Fourier Discrète (TFD)**

La transformée de Fourier discrète est un outil mathématique de traitement de signal, qui est l'équivalent discret de la transformée de Fourier. Sans atteindre les performances de la KLT, elle permet de compacter grandement l'énergie moyenne sur un petit nombre de composantes [Rao et al, 2001 ; Aneur, 2007]. Sa définition mathématique pour un signal  $s$  de  $N$  échantillons est donnée par l'équation (II.1):

$$S(k) = \frac{1}{N} \sum_{n=0}^{N-1} s(n) \cdot e^{-2\pi k \frac{n}{N}} \quad \text{II.1}$$

La transformée inverse est donnée par (l'équation II.2)

$$S(n) = \frac{1}{N} \sum_{k=0}^{N-1} s(k) \cdot e^{2\pi k \frac{n}{N}} \quad \text{II.2}$$

On obtient ainsi un signal discret renseignant sur le contenu fréquentiel du signal  $S(n)$ , correspondant au spectre échantillonné.

Le noyau de projection de la TFD est complexe, car il est représenté par une exponentielle, ce qui rend son utilisation plus complexe.

### II.5.1.2 La transformation De Karhunen Loeve

La KLT (Karhunen-Loeve Transformation) est une transformation optimale en termes de compactage d'énergie. Dans un processus de compression si l'on décide de ne garder qu'un nombre limité de coefficients transformés, les coefficients KLT sont ceux qui contiennent la plus grande fraction de l'énergie totale par rapport à toutes les autres transformations possible. Malheureusement la matrice de transformation de la KLT est une fonction de l'image et il faut calculer la matrice de covariance de l'image pour obtenir la matrice de transformation T. De plus il n'existe pas d'algorithme de calcul rapide de la KLT. Pour ces raisons, la KLT est très peu utilisée malgré sa supériorité théorique [Rao et al, 2001 ; Ameer, 2007].

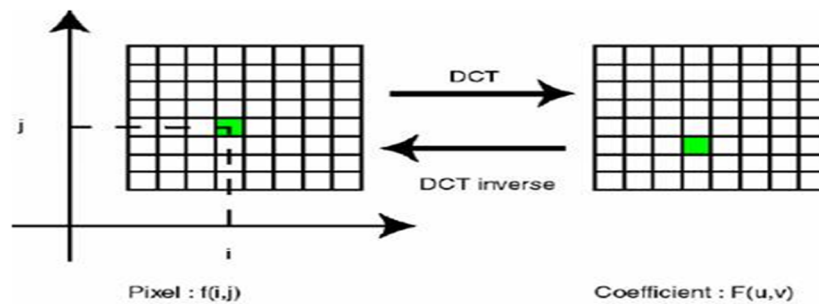
Parmi les inconvénients majeurs de la KLT, citons :

- La connaissance a priori des images à coder,
- La matrice est généralement non séparable, d'où l'absence d'algorithme rapide.
- Le calcul de la matrice de covariance s'appuie sur des blocs dont les statistiques doivent être préalablement connues. Or une modélisation rigoureuse des blocs d'images est impossible.

Des solutions existent pour plier ces inconvénients mais elles font appel à des compromis et s'écartent de la solution optimale.

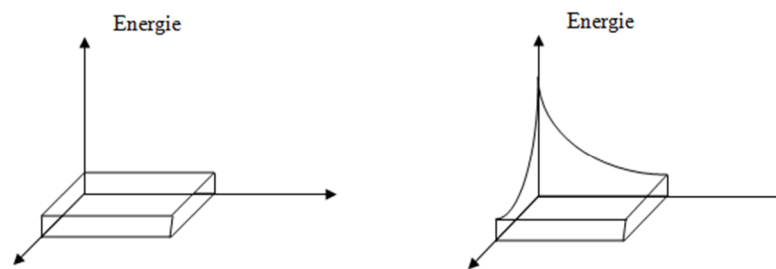
### II.5.1.3 La transformation en cosinus Discrète DCT

La transformation en cosinus discrète DCT est une technique qui a été proposée par Ahmed Natarajan et Rao en 1974, elle a été utilisée dans l'algorithme de compression (schéma hybride) MICD. La DCT est la transformation qui garde l'optimalité de la KLT en termes de compactage d'énergie. En effet, elle fournit pour les signaux images une approximation convenable qui permet de la représenter en un minimum de composantes [Rao et *al*, 2001 ; Cavaro-Menard, 2004]. La figure (4) montre l'application de la DCT sur un bloc de l'image.



**Figure II.4 :** application de la DCT sur un bloc de pixels de 8\*8

La figure (II.5) Montre l'évolution de l'énergie après application de la DCT sur un bloc de l'image. En effet nous remarquons que l'énergie du bloc est concentrée dans un nombre réduit de coefficients. Le maximum de cette énergie se situe en haut à gauche. Ce dernier correspond à la moyenne des niveaux de gris de l'image « basses fréquences », les autres coefficients correspondent aux détails.



**(a) Dans le domaine spatial**

**(b) Dans le domaine transformé**

**Figure II.5 :** Répartition de l'énergie d'un bloc d'image

La DCT appliquée au traitement des images possède les propriétés suivantes :

- La décorrélation des coefficients dans le domaine transformé.
- La concentration de l'information (énergie) autour des basses fréquences (à l'origine du plan transformé).

#### II.5.1.4 Transformée en ondelettes

L'analyse par ondelettes a été introduite au début des années 1980, son invention est attribuée au géophysicien Jean Morlet dans le contexte de l'analyse du signal et l'exploitation pétrolière. Par ailleurs, les ondelettes permettent de caractériser à la fois en temps et en fréquence un signal.

Le domaine d'application des ondelettes varié. L'analyse des signaux, compression d'images, traitement du son, géologie en son des exemples de champ d'application. Les ondelettes est une technique de compression de données, bien adaptée à la compression d'images.

En effet, la transformée en ondelettes présente l'avantage de faire ressortir à la fois les grandes variations et les détails de l'image analysé, ce que ne permet pas la transformée DCT.

#### Que'est-ce que c'est une ondelette ?

Une ondelette est une fonction mathématique à valeur moyenne nulle et limitées dans le temps. Les ondelettes sont déduites d'une ondelette mère ayant été dilatée ou compressé et translatée afin de couvrir tout le domaine de la fonction décomposée. Le caractère de localisation de l'ondelette s'exprime par le fait que la fonction est non nulle sur un intervalle fini et nulle partout ailleurs.

Principe : la transformation par ondelettes est une technique qui consiste à décomposer une image en une myriade de sous-bandes, c'est-à-dire des images de résolution inférieure. La transformation en ondelettes provient d'une analyse multirésolution de l'image. On considère des espaces d'approximations de plus en plus grossiers noté  $c_j$  et des espaces capturant les détails perdus entre chaque niveau d'approximation notés  $d_j$ . Les coefficients d'ondelettes dans les espaces  $d_j$  mettent en évidence des informations sur les contours, les textures, leur localisation et leur orientation. Le choix de l'ondelette mère est très important et fait toujours l'objet d'expérimentation pour adapter l'analyse du signal image au système de perception visuelle de l'homme.

Les coefficients des espaces précédemment citées sont obtenus à base d'une ondelette mère

$$\Psi_{u,s}(t) = \frac{1}{\sqrt{s}} \Psi\left(\frac{t-u}{s}\right) \quad \text{II.3}$$

Avec  $s > 0$  et  $u \in \mathbb{R}$

L'ondelette mère doit satisfaire les deux conditions suivantes pour que le signal admet une inverse de la transformée en ondelettes :

Condition d'admissibilité :

$$C_{\Psi} = \int_{-\infty}^{+\infty} \frac{|\widehat{\Psi}(w)|^2}{w} dw < +\infty \quad \text{II.4}$$

Avec  $\widehat{\Psi}$  est la transformée de Fourier de  $\Psi$ .

Condition d'oscillation :

$$\int_{-\infty}^{+\infty} \Psi(t) dt = 0 \equiv \Psi \text{ est de moyenne nulle} \quad \text{II.5}$$

Le facteur d'échelle comporte à la fois un aspect translation dans le temps amené par le terme  $u$ , mais aussi un aspect dilatation à la fois en temps et en amplitude apporté par les termes  $s$  et  $\sqrt{s}$ . La dilatation en amplitude permet de conserver une norme constante pour tous les éléments de la base (énergie de l'ondelette).

On distingue deux types de transformées en ondelettes : la transformée en continue TOC et la transformée en ondelette discrète TOD.

On définit la TOC telle que :

$$\text{TOC} = \{\Psi_{u,s} f(t)\} = \frac{1}{\sqrt{s}} \Psi^* \left( \frac{t-u}{s} \right) f(t) \quad \text{II.6}$$

$$\text{TOC inverse} \text{ TOC}^{-1} = \frac{1}{c_{\Psi}} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \{\Psi_{u,s} f(t)\} \Psi_{u,s}(t) \frac{1}{s^2} ds du \quad \text{II.7}$$

$\Psi^*$  : représente le conjugué de  $\Psi$ , et  $f(t)$  une fonction de  $L^2(\mathbb{R})$ .

Pour rendre la fonction  $\Psi$  sous une forme discrète, on choisit les valeurs des paramètres  $u$  et  $s$  tel qu'on fixe un pas de dilatation  $s_0 \geq 1$  et un pas de translation  $u_0 \neq 0$ .

Les paramètres  $u$  et  $s$  sont déduits comme suit :

$$s = s_0^j$$

$$u = kb_0 s_0^j$$

Avec:  $(j, k) \in \mathbb{Z}^2$

Et :  $s > 1$  (pas de dilatation) ;  $u \neq 0$  (pas de translation).

Nous obtenons ainsi une famille d'ondelettes discrète dont l'expression générale est :

$$\Psi_{j,k}(t) = \frac{1}{\sqrt{s_0^j}} \Psi(s_0^{-j} t - kb_0) \quad \mathbf{II.8}$$

L'expression de la transformée en ondelettes discrète associée à un signal  $f(t)$  est donnée par :

$$\text{DWT} = C_{j,k} = \langle f, \Psi_{j,k} \rangle = \frac{1}{\sqrt{s}} \int_{-\infty}^{+\infty} \Psi(s_0^{-j} t - kb_0) f(t) dt. \quad \mathbf{II.9}$$

avec :  $(j, k) \in \mathbb{Z}^2$  tel que :  $j$  : paramètre de dilatation et  $k$  : paramètre de translation.

N.B le critère le plus important pour le choix d'une ondelette est de présenter, pour elle et sa transformée de Fourier, des oscillations les plus faibles possibles : c'est ce qui permettra d'assurer une bonne résolution temporelle et fréquentielle

La transformée en cosinus discrète et la transformée en ondelettes sont deux méthodes de compression proche de la transformée de Fourier mais plus performante que celle-ci.

Caractéristiques Transformations	Points forts	Points faibles
DFT	-concentration de l'énergie sur un petit nombre de composantes.	-noyau de projection complexe. -temps de calcul élevé
KLT	-optimale en termes de compactage de l'énergie.	-non séparable -absence d'algorithmes rapides -nécessité d'une connaissance à priori des images à coder.
DCT	-décorrélation des coefficients des blocs de 8*8. -concentration de l'énergie dans un nombre réduit de coefficients -coefficients réels -séparabilité.	-effet de blocs à des taux très élevés. -temps d'exécution long
Ondelettes	-décorrélation des coefficients de toute l'image. -localisation spatio-temporelle. -analyse multi-résolution. -rapide en exécution. -transmission progressive.	-effet de lissage.

**Tableau II.3** : tableau récapitulatif des caractéristiques de certaines transformations

### **II.5.2 Les fractales**

La compression fractale est une méthode de compression d'image encore peu utilisée aujourd'hui. Elle repose sur la détection de la récurrence des motifs, et tend à éliminer la redondance d'informations dans l'image. C'est une méthode destructive puisque l'ensemble des données de départ ne se retrouve pas dans l'image finale.

#### **Principe de l'algorithme de compression**

Contrairement aux autres techniques de compression habituelles, la compression fractale ne tente pas de réduire le nombre de couleurs (format gif) ou de compresser de manière classique les octets composant l'image. Le principe ici est de remplacer l'image par des formules mathématiques.

La compression fractale a pour principe qu'une image n'est qu'un ensemble de motifs identiques en nombre limité, auxquels on applique des transformations géométriques (rotations, symétries, agrandissements et réduction). Évidemment, plus l'image possède cette propriété, meilleur sera le résultat.

On parle de transformations fractales basées sur un opérateur contractant. Ces transformations décrivent l'image de plus en plus finement. A la fin de ce processus, on ne stocke pas le contenu d'un bloc autant de fois qu'il a été vu dans l'image mais seulement les équations mathématiques permettant de représenter le contenu de ces carrés.

Au final on obtient une structure présentant des caractéristiques similaires à des échelles différentes. Pour retrouver l'image il suffira de décrire les transformations qui ont été appliquées aux blocs initiaux. Ce processus rend la compression indépendante de la taille de l'image. De plus, l'image produite est vectorisée et ne subit pas les effets de ma pixellisation. Ce phénomène est surtout visible lors d'un zoom par exemple, l'image fractale peut devenir floue mais ne pixellise pas. Ceci est dû au fait que lors de l'agrandissement, ce ne sont pas les pixels qui sont élargis, mais toute l'image qui est recalculée mathématiquement.

Le problème lié à cette technique est la lenteur du procédé de compression, la décompression quant à elle est aussi rapide que pour les autres formats.

### II.5.3 Codage par quantification

En traitement du signal, la quantification est le procédé qui permet d'approximer un signal continu (ou à valeurs dans un ensemble discret de grande taille) par des valeurs d'un ensemble discret d'assez petite taille appelé dictionnaire « codeblock ».

#### II.5.3.1 La quantification scalaire

On parle d'une quantification scalaire lorsque le dictionnaire est de dimension 1. C'est-à-dire ses valeurs sont des scalaires. La quantification scalaire est la forme la plus simple de quantification, le cas où 1 est appelé la quantification vectorielle.

##### Définition

Un quantifieur scalaire de taille  $N$  est une application  $Q$  de  $\mathbb{R}$  dans un ensemble discret fini  $F$  de dimension 1 et de taille  $n$ ,  $F = \{\hat{x}_1 \dots \hat{x}_n\}$ ,  $Q : \mathbb{R} \rightarrow F$  on notera :  $\hat{x} = Q(x)$

Un quantifieur peut se définir comme un ensemble d'intervalles de l'espace de départ,  $[x_k, x_{k+1}]$ , ces intervalles sont appelés niveaux (intervalles) de décision. A chaque intervalle  $[x_k, x_{k+1}]$ , on fait correspondre une seule valeur de l'espace d'arrivée,  $y_k$ , appelée niveau de reconstruction, c'est-à-dire chaque symbole appartenant à l'un de ces intervalles de décision, sa valeur codée correspond au numéro de son intervalle d'arrivée. La forme typique d'un quantifieur est donc une fonction en escalier.

D'une manière générale, la largeur d'un intervalle n'est pas constante :

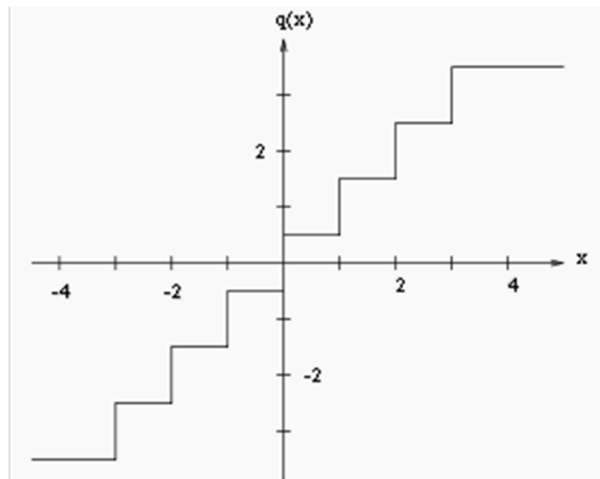
$Q_k = x_k - x_{k+1} \neq \text{cste}$ . Cette largeur est appelée le pas de quantification.

En générale, les niveaux de reconstruction ne sont pas non plus uniformément répartis :

$y_{k+1} - y_k \neq \text{cste}$ .

Tous les symboles d'entrées appartenant à même intervalle sont codés et reconstruits avec la même valeur.

### II.5.3.2 Quantification scalaire uniforme



**Figure II.6:** quantification uniforme avec un pas de quantification =1

C'est le type de quantifieur le plus simple, où les intervalles sont de longueur constante. Le pas de quantification est donc fixe :  $t_{k+1} - t_k = q$  pour tout  $k$ .

Les niveaux de reconstruction sont aussi uniformément répartis. Il est parfois appelé quantifieur scalaire symétrique.

### II.5.3.3 Quantification uniforme à zone morte

Il s'agit d'un quantificateur uniforme avec une légère modification. Les intervalles du quantificateur scalaire sont, comme nous venons de le voir, tous égaux à  $\Delta$ . C'est la définition même de l'uniformité. Dans le cas du quantificateur avec zone morte, on impose d'avoir un intervalle supérieur à  $\Delta$  autour de la valeur  $x = 0$ . En générale, la longueur choisie correspond à un multiple de  $\Delta$  (généralement  $2\Delta$ ).

Ce qui a motivé la mise en place de cette zone morte est directement lié aux caractéristiques, déjà évoquées précédemment, du système visuel humain. Son utilisation trouve tout son sens lorsque la distribution du signal source est centrée en zéro. Généralement les coefficients de hautes fréquences sont de faible amplitude. On décide ainsi de négliger automatiquement ces fréquences, trop pénalisantes pour le processus, sachant que leur retrait n'induit pas de dégradation perceptible pour l'œil humain.

La norme JPEG 2000 utilise un quantifieur scalaire uniforme à zone morte.

### II.5.3.4 La quantification vectorielle

Les techniques de compression d'images exploitent généralement la redondance statistique présente dans l'image. La quantification scalaire qui associe à une variable continue une variable discrète pouvant prendre un nombre plus faible, et fini de valeurs.

Ces valeurs ne sont jamais totalement décorrélées, ou indépendantes. Shannon a montré qu'il était toujours possible d'améliorer la compression de données en codant des vecteurs plutôt que des scalaires.

Dans la suite, l'abréviation (QV) sera utilisée pour distinguer la quantification vectorielle.

La QV, développée par Gersho et Gray a pris une place très importante dans le domaine de la compression d'image que ce soit dans le but de transmission ou d'archivage.

La QV dans son sens le plus général est l'approximation d'un signal d'amplitude continue par un signal d'amplitude discrète. Elle peut être vue comme une application  $Q$  associant à chaque vecteur d'entrée  $x$  de dimension  $K$  un vecteur  $y = Q(x)$  de même dimension appartenant à un ensemble fini  $Y$  appelé dictionnaire de taille finie  $N$ .

Avec  $Y = (y_j, j=1 \dots N)$ .

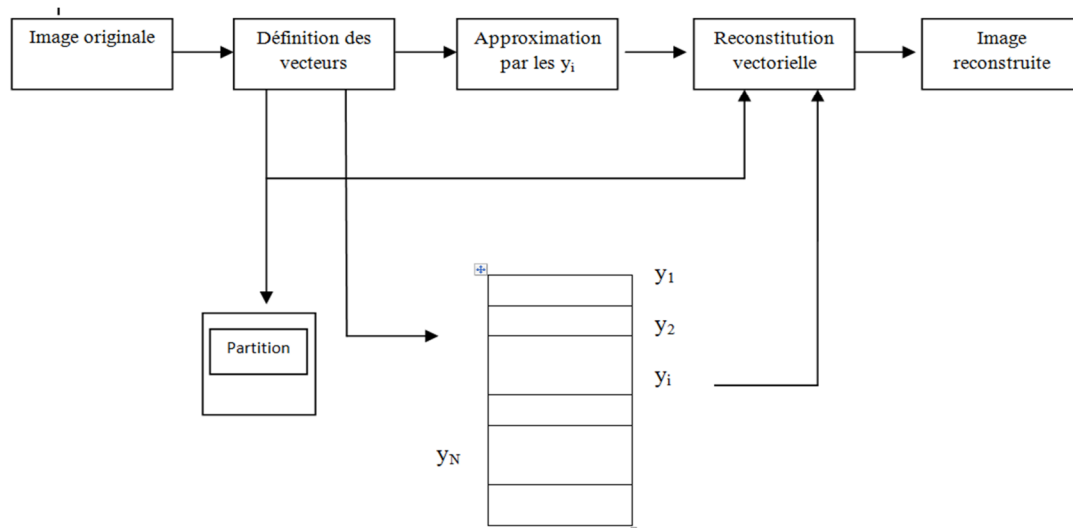
Elle se décompose en deux applications : codeur, décodeur.

– **Codeur :**

Le rôle du codeur consiste, pour tout vecteur  $x_1$  du signal en entrée à rechercher dans le dictionnaire  $Y$  le code vecteur  $y_j$  le plus proche du vecteur source  $x$ . c'est uniquement l'adresse du code vecteur  $y_j$  ainsi sélectionnée qui sera transmise ou stockée. C'est à ce niveau donc que s'effectue la compression.

– **Décodeur :**

Il dispose d'une réplique du dictionnaire et consulte celui-ci pour fournir le code vecteur d'indice correspondant à l'adresse reçue. Le décodeur réalise l'opération de décompression.



**Figure II.7:** synoptique général du codage d'image par quantification vectorielle.

### II.5.4 Méthode hybrides

Les méthodes de compression hybrides sont une combinaison de plusieurs techniques de compression visant à jumeler les avantages de celle-ci, afin d'avoir un taux de compression élevé, une meilleure qualité de restitution de l'image et un temps de calcul minimal lors du processus de codage.

### II.6 Les normes de compression d'image

Jusqu'au début des années 80, les recherches ont essentiellement porté sur des algorithmes de compression et ont donné naissance à des normes qui permettaient des économies de l'ordre de 10 à 90 %, mais qui ont été très vite insuffisantes devant les problèmes que posaient le stockage de milliers d'images (banques d'images) ou l'utilisation de séquences vidéo sur ordinateur, ce qui a rendu nécessaire la mise en place sur le plan international de groupes de coordination et d'étude, chargés de mettre au point des standards adaptés à ces applications afin de rendre cohérents et compatibles les échanges d'informations sur les canaux de communication connus ou futurs.

Les normes JPEG et JPEG 2000 pour la compression d'images sont des méthodes de compression sans perte et avec perte utilisant les codages cités avant.

### II.6.1 La norme JPEG

L'acronyme JPEG (Joint Photographic Expert Group) provient de la réunion en 1982 d'un groupe d'experts de la photographie, dont le principal souci était de travailler sur les façons de transmettre des informations (image fixes ou animées).

Contrairement à la compression LZW, la compression JPEG est une compression avec pertes, ce qui lui permet, en dépit d'une perte de qualité, un des meilleurs taux de compression (sans perte notable de qualité).

Cette méthode de compression est beaucoup plus efficace sur les images photographiques (comportant de nombreux pixels de couleurs différentes) et non sur des images géométriques (à la différence de la compression LZW) car sur ces dernières les différences de nuances dues à la compression sont très visibles.

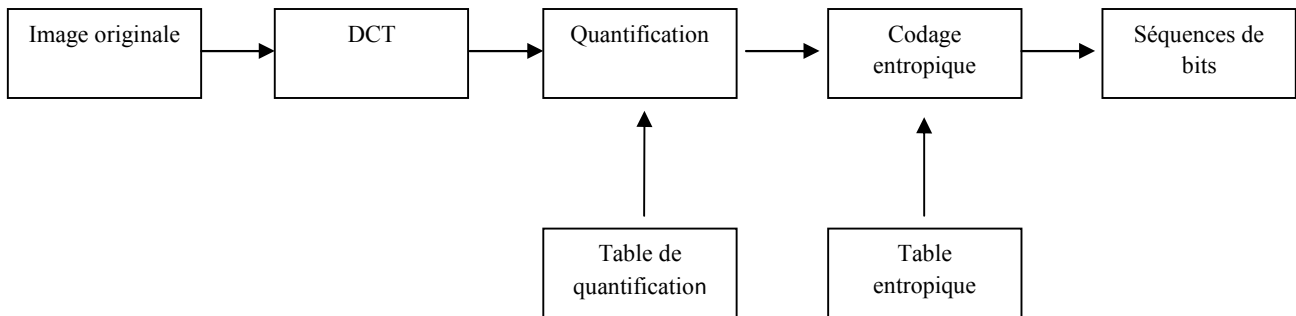
Les étapes de la compression JPEG sont les suivantes :

1. Ré échantillonnage de la chrominance, car l'œil ne peut discerner de différences de chrominance au sein d'un carré de 2x2 points.
2. Découpage de l'image en blocs de 8x8 points, puis l'application de la fonction DCT (Discrete Cosinus Transform, transformation discrète en cosinus) qui décompose l'image en somme de fréquences.
3. Quantification de chaque bloc, c'est-à-dire qu'il applique un coefficient de perte (qui permet de déterminer le ratio taille/qualité) 'annulera' ou diminuera des valeurs de hautes fréquences, afin d'atténuer les détails en parcourant le bloc intelligemment avec un codage RLE (pour enlever un maximum de valeurs nulles).
4. Encodage de l'image puis compression avec un codage entropique (Huffman).

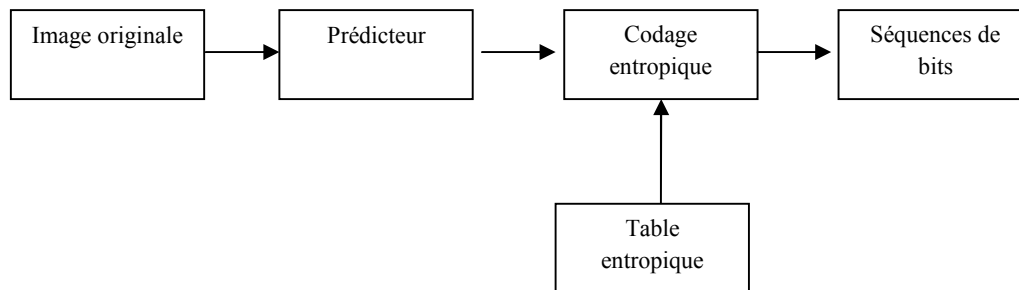
Le format de fichier embarquant un flux codé en JPEG est en réalité appelés JFIF (JPEG File Interchange Format, soit en français Format d'échange de fichier JPEG), mais par déformation le terme de « fichier JPEG » est couramment utilisé.

Il est à noter qu'il existe une forme de codage JPEG sans perte. Bien que peu utilisé par la communauté informatique en général, il sert surtout pour la transmission d'images médicales pour éviter de confondre des artefacts (purement liés à l'image et à sa numérisation) avec de réels signes pathologiques. La compression sans pertes ou compression réversible est alors

beaucoup moins efficace. Les gains en termes de compression sont alors plus modestes, avec un taux de compression de l'ordre de 2.



**Figure II.8:** Principe de compression JPEG avec perte.



**Figure II.9:** Principe de compression JPEG sans perte.

### II.6.2 La norme de compression JPEG 2000

La norme élaborée par le groupe mixte ISO (organisation internationale de la normalisation (ISO)/CEI (Commission électronique internationale) pour le codage d'image référencée JPEG 2000 définit une compression de l'ordre de 50 à 100 fois plus élevée qu'en JPEG, par transformation en ondelettes, et non en DCT. Elle s'applique aux différents types d'images fixes (à deux niveaux, niveaux de gris, la couleur, multi, hyper-composante et omnidirectionnelles), le codage est robuste et convient à la transmission radio, l'affichage de l'image s'effectue de façon graduelle, en améliorant la résolution transmise en première transmission. Parmi les avantages qu'offre ce nouveau standard nous pouvons citer :

- Obtention de performances de compression supérieures à JPEG, notamment pour des débits très faibles.

- Organisation du fichier compressé en fonction de la résolution désirée ou de la qualité de reconstruction (décompression progressive).
- Possibilité de coder des parties d'une image avec une qualité supérieure à d'autres parties (compression en régions d'intérêts).

## II.7 Discussion

Ce chapitre a été consacré à l'étude des méthodes de compression d'images. Ces méthodes ont été réparties en deux types à savoir, les méthodes sans perte d'informations (réversible) et les méthodes avec pertes d'informations (irréversibles). Ainsi, les avantages et les inconvénients de ces techniques.

Les méthodes utilisant les ondelettes sont les plus performantes du fait qu'elles offrent des taux de compression satisfaisants et apportent une souplesse d'exploitation de données.

Les ondelettes sont appréciées surtout pour leurs avantages on cite à titre d'exemple :

- La TOD est adaptée aux types signaux non périodiques car elle permet une analyse temps-fréquence que ne permet pas la transformée de Fourier.
- Le traitement de l'image n'est pas uniforme dans l'espace, puisque on conserve une information spatiale.
- Calculs simple et rapide
- On peut supprimer des détails en négligeant des valeurs inférieures à un certain seuil. Cela n'altérera pas la reconnaissance des composantes de l'image
- L'affichage de l'image peut se faire de manière progressive. Cela peut être intéressant sur des supports de transmission faible débit
- Pour ces critères, les transformées en ondelettes en particulier la transformée en ondelettes discrète est bien appréciée et utilisée par les nouveaux standards de compression d'images tel que le standard JPEG 2000 qui fera l'objectif du chapitre suivant.

## *Chapitre III*

# *Présentation et application du standard JPEG 2000 pour les images omnidirectionnelles*

### III.1 Préambule

La compression est un processus qui crée une représentation de données compactes pour des buts de stockage et de transmission. La norme JPEG 2000 définit le codage des images numériques et qui est destinée à supplanter la norme JPEG car cette dernière présente un certain nombre d'inconvénients :

- L'efficacité de codage est limitée.
- Le codage par blocs de 8X8 pixels génère un effet de mosaïque à bas débit très gênant visuellement.
- Les applications liées à l'image sont de plus en plus spécifiques et nécessitent de nouvelles fonctionnalités non résolues par JPEG.
- La transmission d'images codées est très peu robuste en environnement bruité.

La norme JPEG 2000 est la nouvelle norme internationale pour la compression d'image fixe développée conjointement par l'organisation internationale de normalisation (ISO) et la Commission Electrotechnique Internationale (CEI). La norme est également recommandée par l'Union Internationale des Télécommunications(UIT). Introduite en mars 1997, elle est devenue standard en décembre 2000.

JPEG 2000 est adapté à tous les types d'images photographiques, image scientifiques, clichés médicaux, vues de télésurveillance, et cela avec des applications allant de la transmission d'images sur internet à la photographie numérique en passant par l'impression, la télécopie, l'imagerie médicale et satellite ainsi que les télécommunications mobiles. De plus, JPEG 2000 est prévu pour fonctionner dans un contexte de transmission en environnement bruité (marqueurs de resynchronisation pour lutter contre les pertes de paquets), caractéristique fondamentale pour le domaine des communications mobiles.

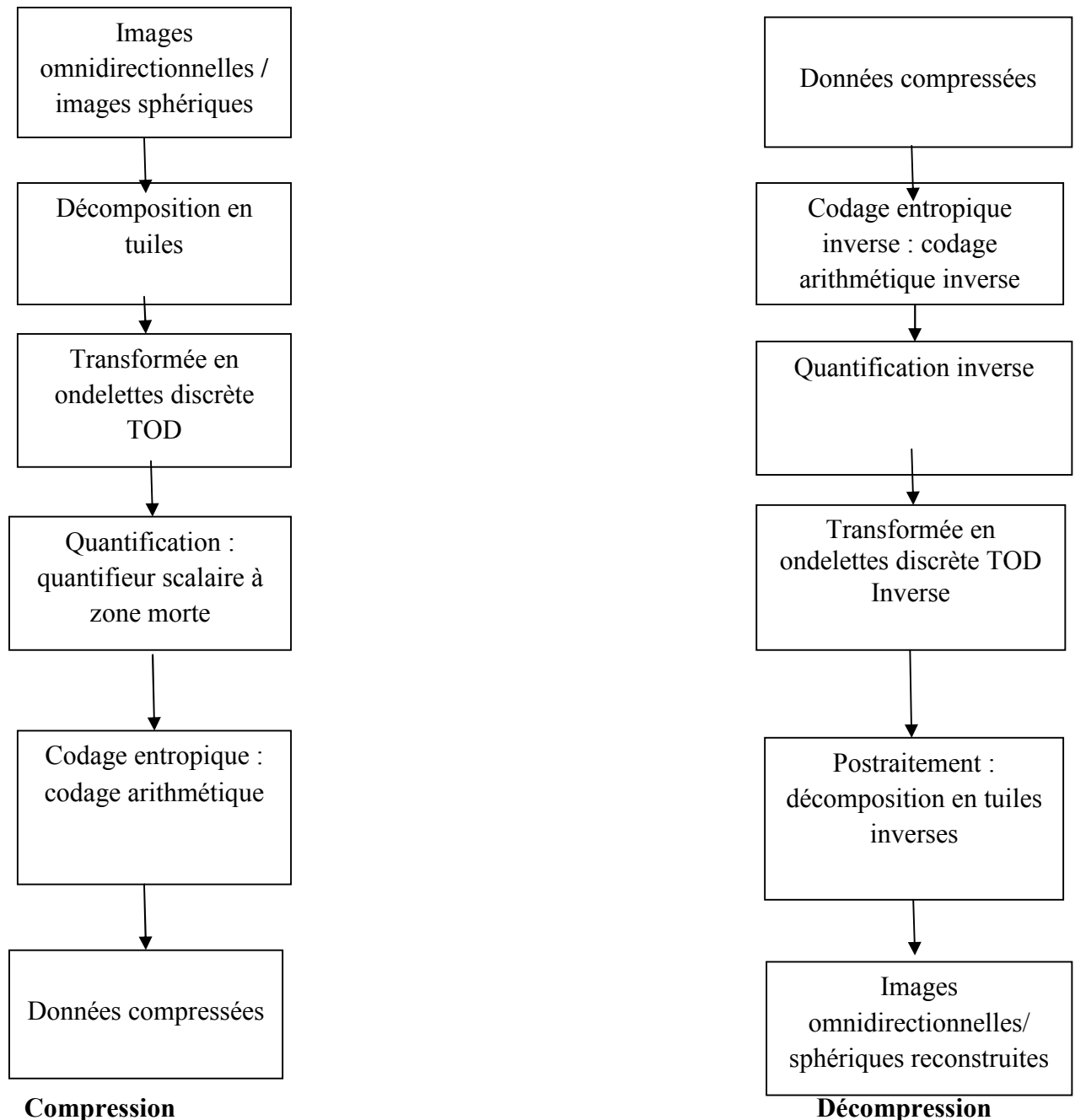
La norme JPEG 2000 propose un certain nombre de fonctionnalités telle que :

- Compression avec et sans pertes et protection contre les erreurs.
- Excellente qualité d'image et traitement d'une large variété d'images.
- Ordonnancement du bitstream (données compressées), accès et décompression aléatoire de ces données.
- Région d'intérêt.
- Protection contre les erreurs.

Cette norme comprend 12 parties sous la même dénomination JPEG 2000, c'est cependant à la première partie que nous allons nous intéresser dans ce chapitre, afin de l'appliquer pour la compression des images omnidirectionnelles.

### III.2 Organisation de l'algorithme JPEG 2000

Les étapes de l'algorithme de compression et décompression des images catadioptriques sont résumées dans le schéma synoptique suivant :



**Figure III.1:** blocs constituant la chaîne de codage et décodage JPEG 2000

### III.2.1 Décomposition en tuiles

Dans quelques situations, une image peut être tout à fait grande par rapport à la quantité de mémoire disponible de code. En conséquence, elle n'est pas toujours faisable pour coder l'image entière comme unité simple, pour simplifier son traitement l'image est divisée en plusieurs régions rectangulaires séparées appelées les tuiles (tiles). Les tuiles seront ensuite chacune codées séparément, avec ses propres paramètres. Ce partitionnement est particulièrement utile dans les applications qui possèdent des ressources mémoire limitées.

L'étape qui consiste à effectuer la transformation en couleur est supprimée, parce que les images utilisées sont en niveau de gris et ne nécessite aucune transformation en couleur.

### III.2.2 Transformée en ondelettes discrète(TOD)

La transformée DCT de JPEG a été remplacée par la transformée en ondelettes TOD dans JPEG 2000.

L'idée de base de la TOD est de séparer les basses et hautes fréquences d'une image. Les basses fréquences correspondent à une version grossière de l'image originale dans laquelle les valeurs des pixels ont été moyennées et où aucune variation brusque n'est observée d'un pixel à l'autre. Les hautes fréquences quant à elles contiennent toute l'information sur les détails de l'image. On comprend aisément que plus d'information est contenue dans la version basse fréquences que dans celle qui ne fournissant que les détails de l'image.

L'objectif de la TOD est donc de concentrer l'information de l'image en une zone très localisée de manière à pouvoir ensuite compresser fortement les zones ne contenant que peu d'information.

En pratique, La TOD consiste à appliquer sur l'image deux filtres successivement à réponse impulsionnelles finie, un filtre passe bas (h) et un filtre passe haut(g) suivie d'une décimation par deux.

Les étapes du filtrage de l'image sont les suivantes :

- Dans un premier temps, chaque ligne de l'image est décomposée en utilisant horizontalement les filtres h et g suivie par une décimation. Cela produit deux sous bandes, l'une correspond aux basses fréquences horizontales L disposée à gauche, l'autre aux hautes fréquences horizontales H, disposée à droite.

— Dans un second temps, les mêmes filtres sont appliqués aux colonnes de ces deux bandes. Chacune des deux bandes de départ est alors décomposée en deux nouvelles sous-bandes, l'une correspond aux basses fréquences verticales (LL, LH) disposée dans la partie supérieure de la matrice, l'autre aux hautes fréquences verticales (HL, HH) disposée dans la partie inférieure.

Nous obtenons à la fin quatre sous bandes :

- Une sous-bande issue des deux filtres passe-bas représentant la réduction de l'image source (approximation LL).
- Une sous-bande issue d'un double filtrage passe-haut (détails diagonaux HH).
- Une sous-bande issue d'un filtrage passe-haut horizontal et un filtrage passe-bas vertical (détails horizontaux HL).
- Une sous-bande issue d'un filtrage passe-bas horizontal et un filtrage passe-haut vertical (détails verticaux LH).

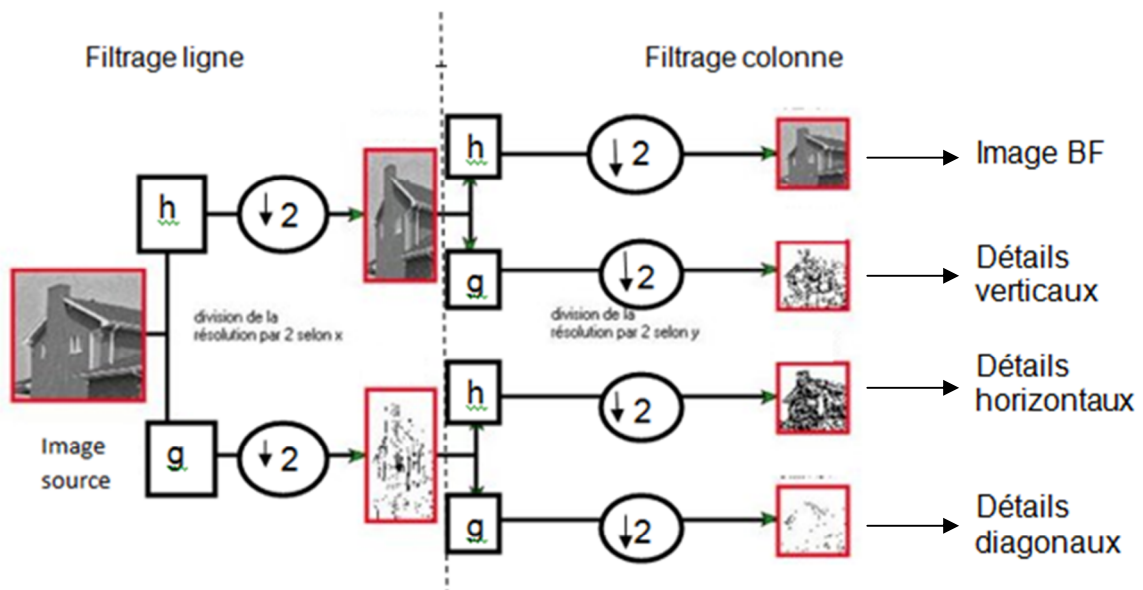
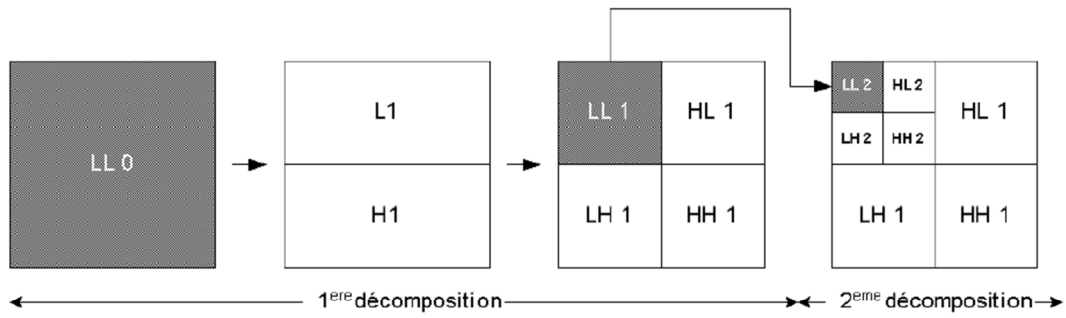
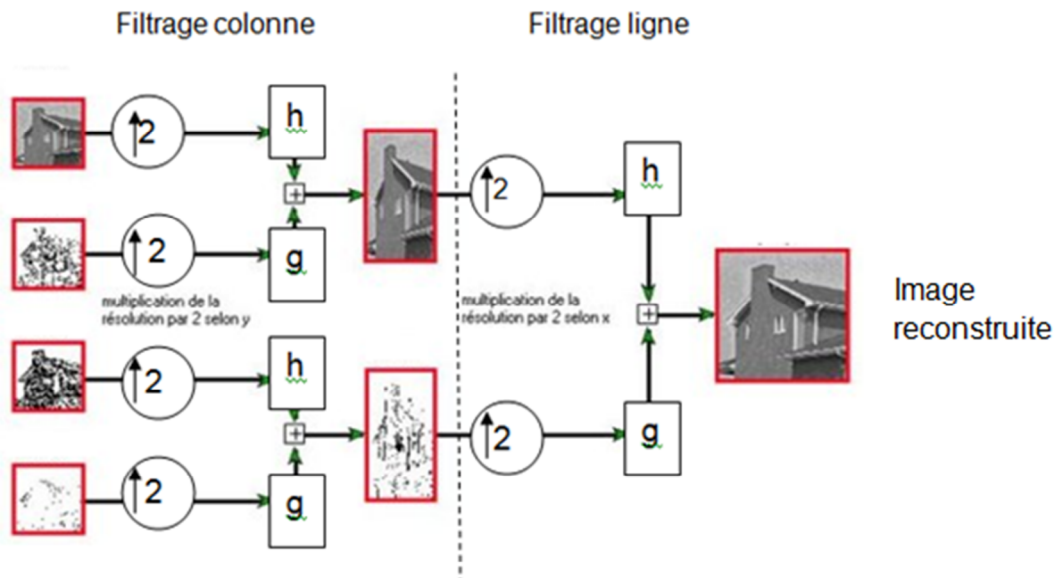


Figure III.2 : Décomposition multi-résolutions d'une image



**Figure III.3:** Décomposition en ondelettes à deux niveaux

Pour la transformée inverse elle consiste à appliquer une autre paire de filtres passe-bas et passe-haut, précédés par un sur-échantillonnage de facteur 2



**Figure III.4 :** Reconstruction multi-résolutions d'une image

Cette étape représente l'opération inverse de la transformée en ondelettes. La transformée inverse a pour but de reconstruire l'image originale à partir des sous bandes détails et de l'image de basse résolution. La reconstruction est donc obtenue par la somme du filtrage h (filtre passe-bas) des coefficients d'approximations et du filtrage g (filtre passe-haut) des coefficients de détails auxquels sont intercalés des zéros.

### III.2.3.1 Quantification

La transformée en ondelettes est suivie d'une quantification scalaire uniforme avec dead-zone, permettant de réduire la dynamique des données. Chaque coefficient de la transformation  $c_b(u, v)$ , d'une la sous-bande  $b$  est quantifié à valeur  $q_b(u, v)$  suivant l'équation :

$$q_b(u, v) = \text{sign}(c_b(u, v)) * \left\lceil \frac{c_b(u, v)}{\Delta_b} \right\rceil \quad \text{III.1}$$

Où  $\Delta_b$  est le pas de quantification en fonction de la dynamique et du type de la sous-bande, du nombre de niveaux de décomposition. La dimension du pas quantification.

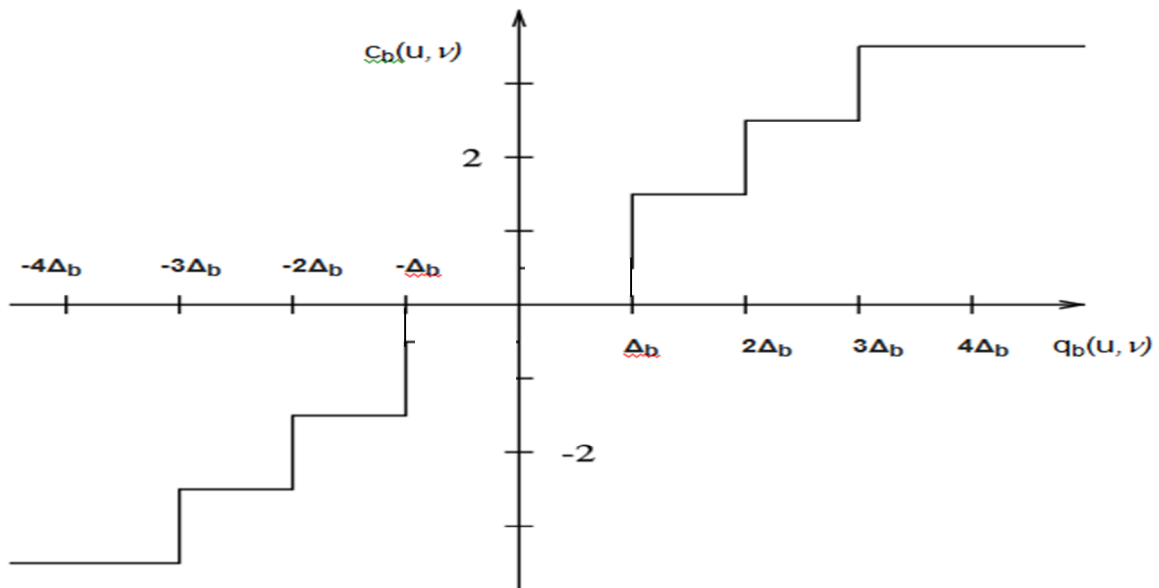


Figure III.5 : Quantification scalaire uniforme avec zone morte (Dead zone).

**III.2.3.2 Règle de la quantification inverse (déquantification)**

Pour reconstruire les sous bands à partir des séquences générées par le décodeur entropique on utilise l'équation suivante :

$$c_b(u, v) = \begin{cases} [q_b + r \operatorname{sign}(q_b)] \Delta_b & \text{si } q_b \neq 0 \\ 0 & \text{autrement} \end{cases} \quad \text{III.2}$$

Où  $q_b$  est l'index de quantification,  $\Delta_b$  est le pas de quantification,  $c_b(u, v)$  est la valeur reconstruite du signal, le  $\operatorname{sign}(q_b)$  dénote le signe de  $q_b$ , et  $r$  est la polarisation de reconstruction une valeur fixée par le développeur du codec qui est souvent égale à 0.5.

**III.2.4 Codage entropique :**

Le codeur utilisé dans JPEG 2000 est constitué de deux parties bien distinctes mais interagissant l'une avec l'autre :

L'EBCOT (Embedded Bloc Coding Optimized Truncation) : il s'agit de l'entité qui parcourt les bits d'un code-block et qui envoie au MQ une succession de couples (symbole à encoder ; contexte du symbole) : il s'agit du codeur proprement dit qui reçoit une séquence de  $\langle 0 \rangle$  et de  $\langle 1 \rangle$  et qui en produit une autre plus petite.

Après la quantification scalaire, les coefficients issus des différentes sous-bandes sont rangés en blocs, appelés code blocs(CB), de forme rectangulaires, la taille de blocs est un nombre entier de puissance de deux et le produit de largeur et hauteur ne dépasse pas 4096 avec une hauteur minimale de 4. Chaque code-bloc est codé indépendamment des autres, les coefficients dans chaque code-bloc sont rangés en bit-planes (plans de bits) comme illustré dans les figures (7) et (8) chaque plan de bits est divisé en bandes horizontales dites « stripes », chaque bande a une hauteur de 4. Si la hauteur de code bloc n'est pas un multiple de 4, la hauteur de dernière bande est inférieure à 4. Au sein d'une stripe, les bits à coder sont parcourus selon l'ordre prédéfini figure(9) en commençant par le plan de bits de poids fort (MSB), tous ces plans sont codés en fonction de leur signifiante et de leur contexte par trois passes successives **signifiante pass** (Passe de signifiante), **refinement pass** (Pass d'affinage), **cleanup pass** (Passe de nettoyage)). Le premier bit-plane signifiant rencontré est codé seulement avec cleanup pass. Les bits-planes suivants sont codés en trois passes. Chaque bit d'un plan est codé une seule fois par l'une des trois passes. Les bits issus de ces différentes

passes, ainsi que les contextes associés (un contexte est une façon de tenir en compte des voisins et de chaque passe), sont ensuite envoyés à un codeur arithmétique MQ

### III.2.5 Le codage arithmétique

La nouveauté apportée par le codage arithmétique vient du traitement vectoriel des symboles de la source. Cette vectorisation n'est cependant pas à confondre avec un groupement des symboles par deux ou trois qu'on appelle codage par blocs. Il n'y a qu'un seul vecteur dans un codeur arithmétique, celui constitué de l'ensemble des symboles successifs à encoder. Cette technique permet d'obtenir un débit très proche de la limite théorique de l'entropie. Cela est dû au fait qu'il n'y a plus, contrairement au codeur de Huffman, de correspondance univoque entre symbole et un mot-code. Le vecteur entier à encoder est représenté par un seul mot-code. Ainsi, il est possible d'avoir un codage d'un symbole qui utilise moins d'un bit. Les étapes à suivre pour encoder une suite de symboles sont les suivantes :

1/ Créer une table de probabilité de différents symbole possibles.

2/ Associer de manière arbitraire une partie de l'intervalle probabiliste [0,1] à chaque symbole. La largeur de ces sous-intervalles est choisie proportionnellement aux probabilités des symboles.

3/ Prendre comme intervalle de départ celui associé au premier symbole à encoder.

4/ Réduire la largeur de cet intervalle à chaque nouveau symbole à encoder, de la manière suivante :

$$A_{i+1} = A_i * P_x$$

$$C_{i+1} = C_i + A_i * Q_x$$

Où

$A_i$  est la largeur de l'intervalle suite à l'encodage du  $i^{\text{ème}}$  symbole.

$P_x$  est la probabilité du  $(i+1)^{\text{ème}}$  symbole 'x' à encoder.

$C_i$  est la borne inférieure de l'intervalle suite à l'encodage du  $i^{\text{ème}}$  symbole.

$Q_x$  est la borne inférieure de l'intervalle associé au symbole 'x', appelé aussi intervalle cumulé de probabilité.

La sortie du QM est une séquence de 0 et de 1 qui peut être découpée en un certain nombre de segment. Un segment est constitué d'un nombre fini d'octets et représente l'entité de données élémentaires produite par le QM. Le nombre de segment produit lors du codage d'un code-block dépend de la taille du code-bloc.

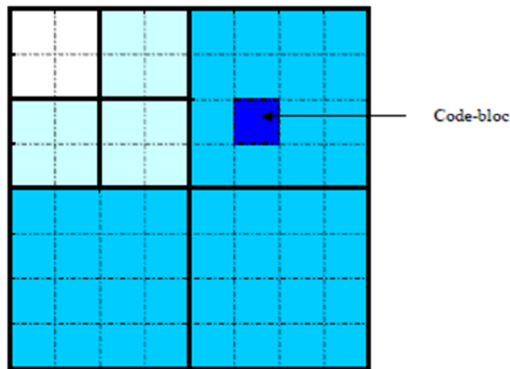


Figure 6 : Partitionnement en code-blocks

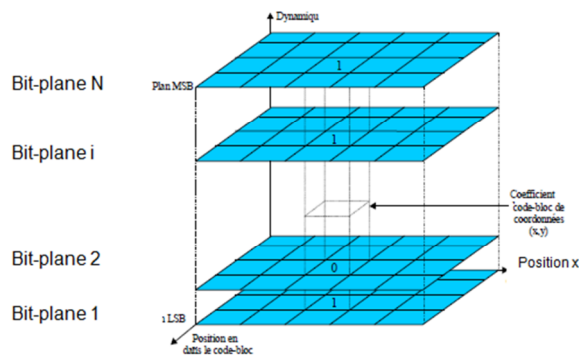


Figure 7: Représentation en bit-plane d'un code-block.

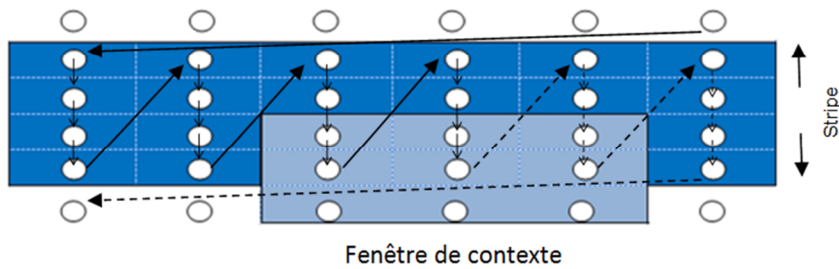


Figure 8: l'ordre de parcourt d'une stripe.

### **III.3 Allocation de débit**

La séquence binaire issue du MQ est divisée en un certain nombre de paquets. Chacun d'eux contient le bitstream correspondant à une même composante, un même niveau de résolution, une même couche de qualité et une même zone spatiale du niveau de résolution. Les zones spatiales de chaque niveau de résolution sont appelées precincts. chacun des paquets est précédé d'un en-tête contenant des informations permettant d'identifier très précisément les données véhiculées par ce paquet. L'ordre dans lequel les paquets apparaissent dans le codestream est appelé ordre de progression. Quatre ordres de progression différents sont définis. Ils permettent lors du décodage d'obtenir en priorité soit les données d'une même composante, soit celles d'une même zone spatiale de l'image.

### **III.4 Organisation de bistream**

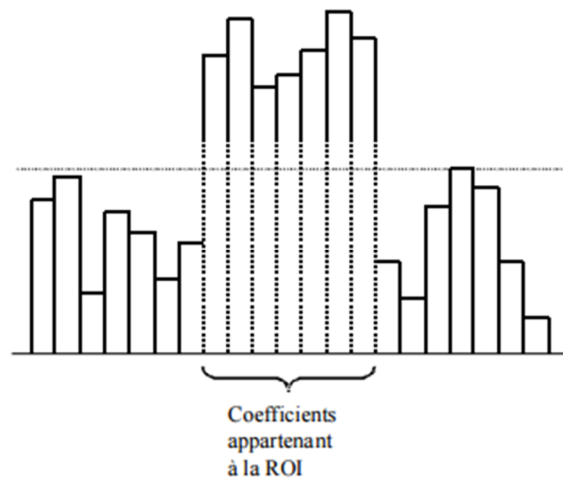
La séquence binaire issue du MQ est divisée en un certain nombre de paquets. Chacun d'eux contient le bitsream correspond à une même composante, un même niveau de résolution, une même couche de qualité et une même zone spatiale du niveau de résolution.

NB.une organisation de JPEG 2000 est d'inclure des outils de résistance aux erreurs de transmission. Par défaut, le codage arithmétique agit en effet sur un code-bloc. En cas d'erreur (un seul bit erroné suffit) c'est la totalité du code-bloc qui est perdu. Pour limiter les effets de ces erreurs, la norme propose plusieurs outils, afin d'éviter la propagation des erreurs ou de limiter leurs effets.

### **III.5 Région d'intérêt**

L'encodeur JPEG 2000 permet de coder certaines régions de l'image avec plus de quantité que d'autres appelées Région d'Intérêt (ROI). Le codage de régions d'intérêt s'opère entre la quantification scalaire et le codage entropique. Les coefficients appartenant à la ROI sont multipliés par une puissance de 2. Cette multiplication a pour conséquence de déplacer chaque bit constituant le coefficient ROI dans des plans supérieurs (au-delà de la dynamique maximum des coefficients) par rapport aux autres coefficients, comme montré dans la (figure 9).

La forme de la région d'intérêt n'est pas signalée au décodeur, seulement la présence d'une région d'intérêt ainsi que la valeur du décalage opéré sont signalées. Au décodage, les données seront redécalées de façon à retrouver la dynamique d'origine.



**Figure III.9:** Localisation de région d'intérêt.

### III.6 Le contrôle de taux

Il s'agit ici de fixer un taux de bits que le codeur ne doit pas dépasser en respectant la contrainte d'une distorsion minimale. Dans le codeur, le contrôle du taux est accompli par deux mécanismes distincts :

- Choix des pas de quantification : il fixe des pas de quantification plus ou moins grands afin d'arriver au débit binaire voulu. Cependant cette méthode n'est pas la meilleure parce qu'à chaque fois que l'on modifie le pas de quantification il faut refaire intégralement l'étage de codage.
- Sélection des passes de codage qui seront inclus dans le flux de données final (codestream): cette méthode est la meilleure car elle permet d'enlever rapidement des données superflues et elle n'affecte que le deuxième étage de codage. Le standard ne spécifie pas comment ces mécanismes devraient être employés, il est possible d'utiliser l'un ou l'autre ou même tous les deux ensembles.

### III.7 Résistance aux erreurs

Une originalité de JPEG 2000 est d'inclure des outils de résistance aux erreurs de transmission. Le problème vient essentiellement du codeur arithmétique car un seul bit erroné entraîne le décodage d'une mauvaise séquence. Par défaut, le codage arithmétique agit en effet sur un code-bloc. En cas d'erreur (un seul bit erroné suffit) c'est l'ensemble du code-bloc qui est perdu. Afin de limiter les effets de ces erreurs, la norme propose plusieurs outils, dont la

philosophie est essentiellement de compartimenter les mots de codes ou de réduire leur longueur afin d'éviter la propagation des erreurs ou de limiter leurs effets. Les outils proposés sont les suivants :

– **Marqueur de segment**

Ces marqueurs sont insérés après chaque plan de bits et sont codés arithmétiquement. Leur bon décodage indique que le plan de bit courant a été correctement décodé. Inversement, si le marqueur n'est pas trouvé, le plan de bits sera considéré comme erroné et donc supprimé.

– **Terminaison à chaque passe**

C'est un moyen de limiter la propagation des erreurs, en compartimentant les données de façon fine (à chaque passe). Le décodeur arithmétique peut ainsi continuer le décodage en cas d'erreur.

– **Marqueur de resynchronisation (SOP/EPH)**

Ces deux marqueurs indiquent le début et la fin de chaque paquet et permettent au décodeur de se synchroniser grâce au numéro de paquet inclus dans ces marqueurs. Il s'agit ici de marqueurs de syntaxe (non codés arithmétiquement) et dont la gestion est propre à chaque décodeur.

### **III.8 Compression d'image omnidirectionnelle**

Afin d'appliquer correctement le standard JPEG 2000 aux images omnidirectionnelles centrales, nous proposons dans notre travail d'utiliser la méthode JPEG 2000 dans les deux espaces de projections, sphérique et plan de l'image omnidirectionnelle issue des caméras para-catoptriques.

Les caméras para-catadioptriques sont réputées pour leur facilité de mise en œuvre par rapport aux autres caméras (miroir sphérique, hyperbolique) car seul le parallélisme entre l'axe optique et l'axe de révolution du miroir pour assurer le point de projection unique.

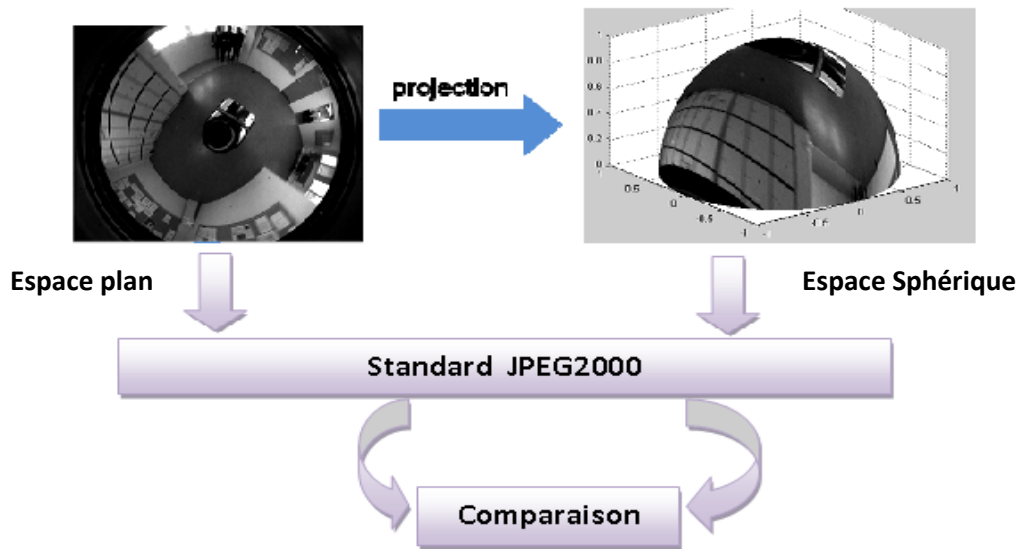


Figure III.10 : Compression d'images omnidirectionnelles

### III.8. Compression sur la sphère

Pour appliquer la méthode de compression d'image JPEG2000 aux images paracatadioptriques sphériques. Nous avons choisi un schéma de traitement des images omnidirectionnelles qui utilise la sphère équivalente.

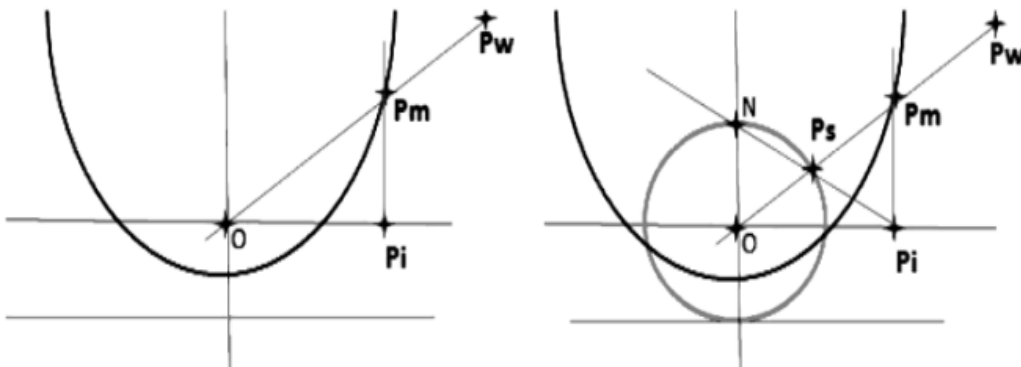


Figure III.11 : Equivalence entre un modèle de projection parabolique

La définition de la projection stéréographique à partir du pôle Nord vers le plan équatorial est la suivante (figure (10)) :

Tout d'abord le point 3D est projeté dans la sphère du centre de la sphère "O"; la seconde étape consiste à projeter le point  $P_s$  sur la sphère pour le plan d'image d'un point "N" placé sur l'axe optique (figure 10). Ces points de projection sont obtenus par étalonnage du capteur.

Soit  $(X_s, Y_s, Z_s)$ , les coordonnées cartésiennes de points  $(\theta, \phi)$  les coordonnées sphériques équivalents de point  $P_s$ .

$$P_s = \begin{cases} X_s = \sin \theta \cos \phi \\ Y_s = \sin \theta \sin \phi \\ Z_s = \cos \theta \end{cases} \quad \text{III.3}$$

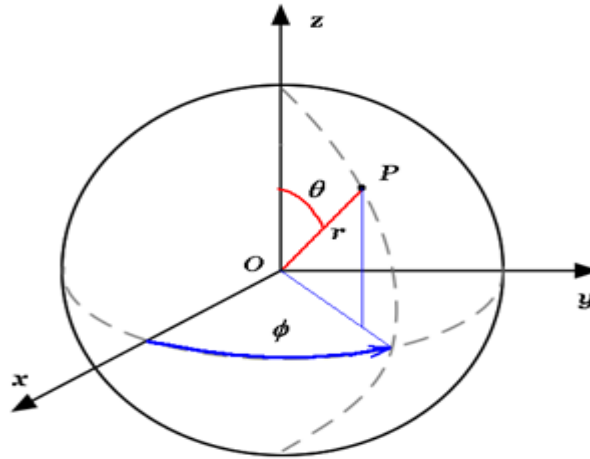


Figure III.12: les coordonnées sphériques

Avec  $\phi \in [0, 2\pi [$ , l'angle de longitude et  $\theta \in [0, \pi / 2]$ , l'angle de latitude, la projection stéréographique de  $P_s$  sur le plan d'image dans le point  $P_i(x, y)$  donnée par:

$$\begin{cases} x = X_s / (1 - Z_s) \\ y = Y_s / (1 - Z_s) \end{cases} \quad \text{III.4}$$

En combinant les équations (4) et (5), on obtient les coordonnées sphériques du point  $P_i(x(\theta, \phi), y(\theta, \phi))$  :

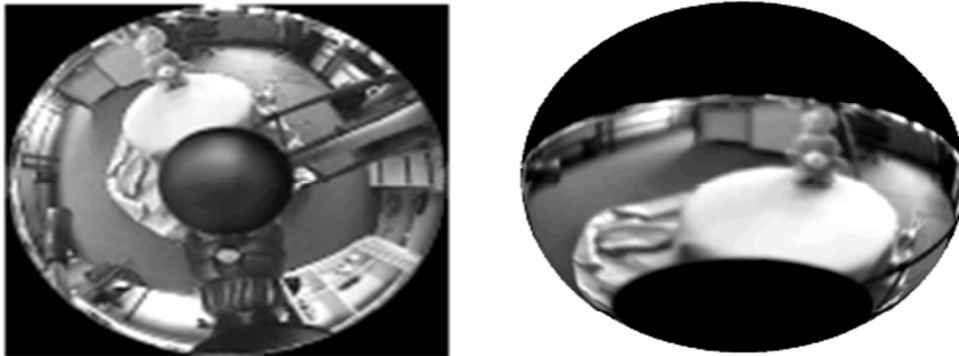
$$\begin{cases} x = \cot \frac{\theta}{2} \cos \phi \\ y = \cot \frac{\theta}{2} \sin \phi \end{cases} \quad \text{III.5}$$

Pour la projection stéréographique inverse, qui à un point  $P(x, y)$  du plan équatorial associe le point  $P(X_s, Y_s, Z_s)$  (ou  $P(\theta, \phi)$  en coordonnées sphériques) de la sphère unité, les formules sont les suivantes :

$$\begin{cases} X_s = \frac{2x}{1+x^2+y^2} \\ Y_s = \frac{2y}{1+x^2+y^2} \\ Z_s = \frac{-1+x^2+y^2}{1+x^2+y^2} \end{cases} \quad \text{III.6}$$

Notons que  $X_s^2+Y_s^2+Z_s^2=1$  puisque le point  $P_s$  appartient à la sphère unité.

$$\begin{cases} \theta = \arccos\left(\frac{-1+x^2+y^2}{1+x^2+y^2}\right) \\ \varphi = \arctan\left(\frac{x}{y}\right) \end{cases} \quad \text{III.7}$$



**Figure III.13** : image para-catadioptrique et l'image sphérique équivalente

### III.9 Discussion

Dans ce chapitre nous avons défini les différentes étapes de la norme JPEG 2000 qui seront appliquées pour la compression des images omnidirectionnelles et aux images sphériques équivalentes acquises par les capteurs catadioptriques à miroir parabolique.

Les résultats du standard JPEG 2000 sur ces images seront exposés et interprétés dans le chapitre suivant.

*Chapitre IV*

*Tests et résultats*

## IV.1 Préambule

Ce chapitre fera l'objet d'évaluation des résultats d'application du standard JPEG 2000 pour la comparaison d'images omnidirectionnelles para-catadioptriques et leurs images sphériques équivalentes. Nous avons testé les deux espaces, sphérique et plan, sur des séquences de synthèses dont les paramètres de calibrations sont connus et sur les séquences réelles dont les paramètres calibrations sont estimés. Nous avons cherché à comparer les deux espaces, en utilisant les mêmes critères d'évaluation afin de prouver et valider leur efficacité sur les images omnidirectionnelles.

Des images réelles ont été également utilisées pour vérifier et comparer les deux approches dans les conditions d'utilisation habituelles. Support de programmation utilisé est le MATLAB.

## IV.2 Critères d'évaluation

On évalue généralement la performance d'une compression effectuée par le JPEG 2000 par le calcul de certains critères tel que :

- Taux de compression.
- La qualité de reconstitution de l'image (mesure de distorsion/mesure de fidélité).

La notion de qualité image est indispensable pour caractériser le besoin selon les applications. Les critères de qualité image qualifie toute la chaîne image, depuis la scène observée jusqu'au produit final.

### IV.2.1 Taux de compression

Rappelant que, l'objectif d'une compression est de minimiser la quantité d'informations nécessaires à la représentation d'une image, on définit le rapport de compression  $R_C$  par :

$$R_C = \frac{\text{Nb de bits de l'image originale}}{\text{Nb de bits de l'image compressée}} \quad \text{IV.1}$$

Par conséquent, on peut définir la quantité  $T_C$  appelée taux de compression il sert à mesurer l'efficacité d'une méthode de compression par :

$$T_C = \left(1 - \frac{1}{R_C}\right) \times 100 \quad \text{IV.2}$$

L'objectif de la compression d'image est donc d'avoir un taux de compression le plus élevé possible. Toutefois, la théorie de l'information donne une limite théorique au  $R_c$  maximale qu'il est possible d'atteindre sans distorsion pour toute méthode de compression sur une image donnée.

### IV.2.2 Mesure de distorsion

Pour évaluer numériquement la qualité de l'image reconstruite, il est nécessaire de contrôler sa distorsion, donc de les mesurer. Ces mesures se traduisent par des critères objectifs qui ne sont qu'une approche de la qualité d'une image. Ces critères sont :

- L'erreur quadratique moyenne (MSE) :

$$MSE = \frac{1}{N} \sum_{ni=0}^{N-1} (\hat{n}_i - n_i)^2 \quad \text{IV.3}$$

Avec  $n_i$  : le niveau gris du l' $i^{\text{ème}}$  point de l'image originale,  $\hat{n}_i$  : le niveau de gris du l' $i^{\text{ème}}$  point de l'image transformée (reconstruite),  $N$  : le nombre total de points constituant chacune des images .

Ce calcul nous renseigne de l'erreur quadratique moyenne d'une image classique.

Le rapport signal sur bruit crête PSNR (Peak Signal to Noise Ration) en dB pour une image 2D numérique:

$$PSNR = 10 \log_{10} \frac{N_{dG \max}^2}{MSE} \text{ dB} \quad \text{IV.4}$$

Avec  $N_{dG \max}$  : le niveau de gris maximum et l'exemple couramment utilisé en télévision numérique est  $N_{dG \max} = 255$  donc :

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \quad \text{IV.5}$$

Etant donné que, l'image reconstruite doit s'apprécier visuellement, on peut l'améliorer en calculant l'image différence 'I<sub>Diff</sub>' entre l'image originale et l'image reconstituée qui est définie par :

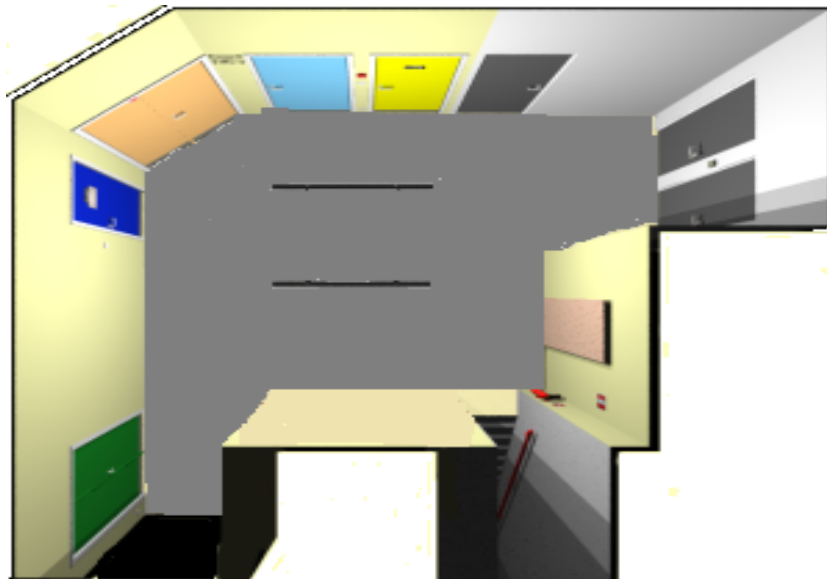
$$I_{\text{Diff}} = 2(n_i - \hat{n}_i) + 128 \quad \text{IV.6}$$

Où le facteur 2 est adopté pour rehausser la dynamique de l'image, et 128 pour rendre l'erreur I<sub>Diff</sub> positive (pour des raisons de visualisation). Cette image différence devrait être uniforme pour une image parfaitement reconstruite. Lorsque la reconstitution de l'image est parfaite, Donc pas de distorsion. On a:

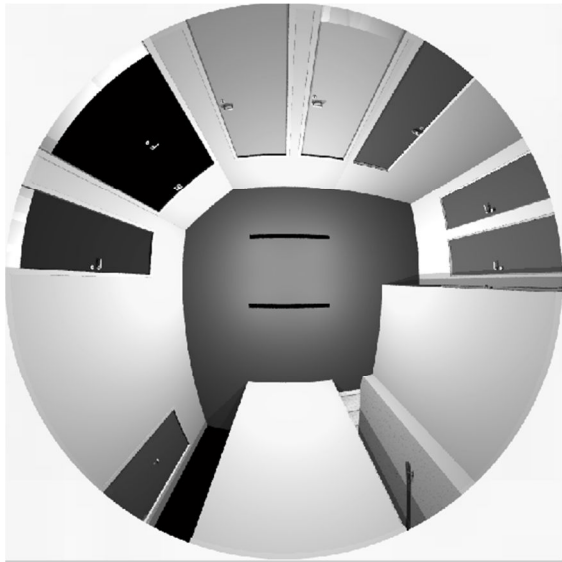
$$\begin{cases} MSE & \rightarrow 0 \\ PSNR & \rightarrow +\infty \end{cases}$$

### IV 3 Séquences de synthèses

Pour comparer les deux approches élaborées, nous avons utilisé une séquence de synthèse. Pour l'application des images de synthèses, nous avons utilisé le logiciel POVRay (voir Annexe A) et un simulateur développé au sein du laboratoire MIS (Modélisation Information et Systèmes) de l'université Picardie Jules Verne, qui donne la possibilité de créer des prises de vue d'un environnement 3D (voir figure IV.1).



**Figure IV.1** : Exemple d'image de l'environnement virtuel (3D) générée par POV-Ray.

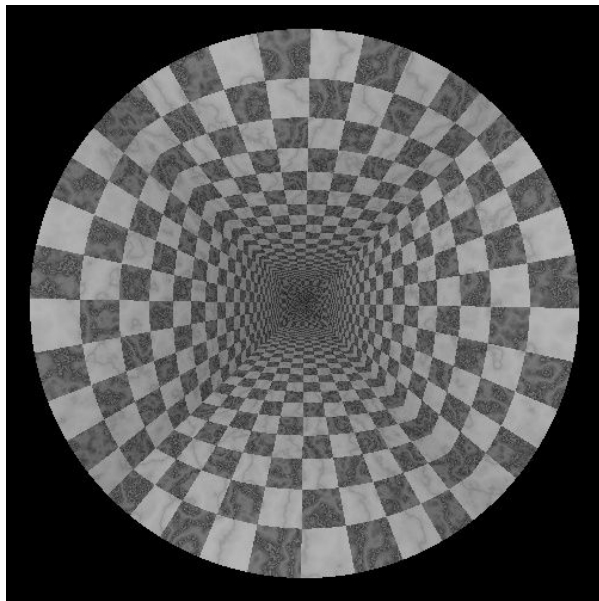


a) Image omnidirectionnelle correspondante

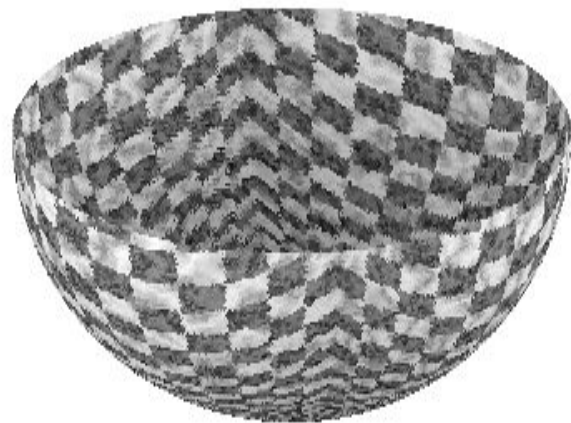


b) image sphérique équivalente

**Figure IV.2** : image para-catadioptrique de synthèse (moins texturée) générée par le simulateur



a) Image omnidirectionnelle correspondante



b) image sphérique équivalente

**Figure IV.3** : image para-catadioptrique de synthèse 2 (fortement texturée)

Ces séquences ont été générées avec POVRay. Une caméra parabolique virtuelle observe quatre plans texturés séparés par 100 cm. Le logiciel POVRay génère des images de 500x500 pixels. Les paramètres intrinsèques sont fixés avec  $k_v=k_u=100$  et  $u_0=v_0=250$  pour toutes les séquences. Les figures IV.2 et IV.3 montrent les images catadioptriques de synthèse utilisées ainsi que leurs représentations sur une demi-sphère unité.

#### IV.4 images réelles

Les séquences sont obtenues à l'aide d'une caméra catadioptrique et un miroir parabolique placé sur un plan perpendiculaire à son axe optique. Nous avons choisi trois types de séquences qui sont: séquence.1 fortement texturée (Figure.IV.3 -), avec une rotation pure de la caméra placée en bas de la scène, séquence.2 moyennement texturée (Figureiv.4), avec une caméra fixe placée en haut de la scène et des objets se trouvant dans la scène en mouvements et séquence.3 (FigureIV.5), avec une caméra fixe dans un couloir qui comporte des textures homogènes.

Les caméras sont calibrées en utilisant le CatPack toolbox [barreto 2000] Nous obtenons pour chaque caméra les paramètres intrinsèques suivants :

Séquence 1 :

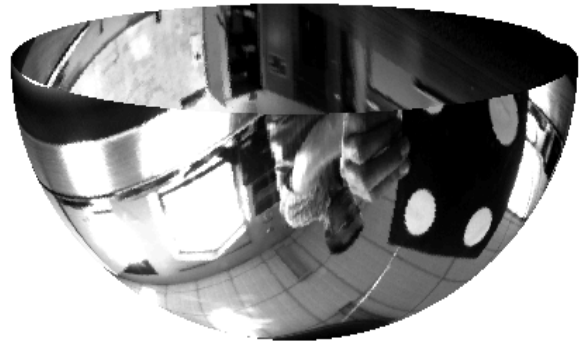
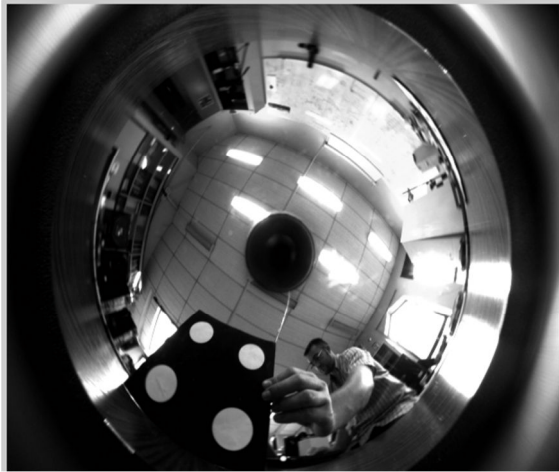
$U_0=634.008867$ ,  $V_0=506.4652661$ ,  $k_v=484.4948585$ ,  $k_u$  483.8251696 et la taille de l'image est 960\*1280.

Séquence 2 :

$U_0=514.5713753392.6985726$   $k_v=459.2830198$   $k_u$  458.1876516 et la taille de l'image est 768\*1024

Séquence 3 :

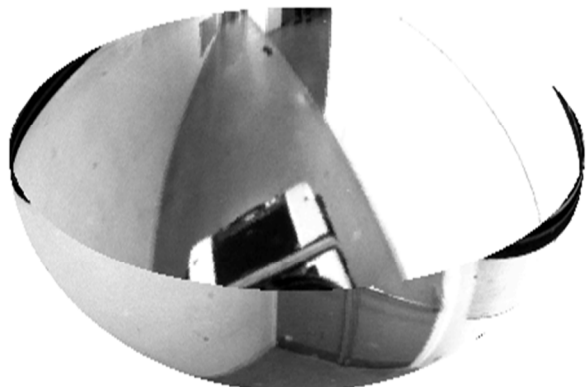
$U_0=627.39934$   $V_0=$  463.62001  $k_v=457.83703$   $k_u$  457.83703 et la taille de l'image est 960\*1280.



**Figure IV.4 :** Image omnidirectionnelle et l'image sphérique équivalente de la séquence 1



**Figure IV.5 :** Image omnidirectionnelle et image sphérique équivalente de la séquence 2

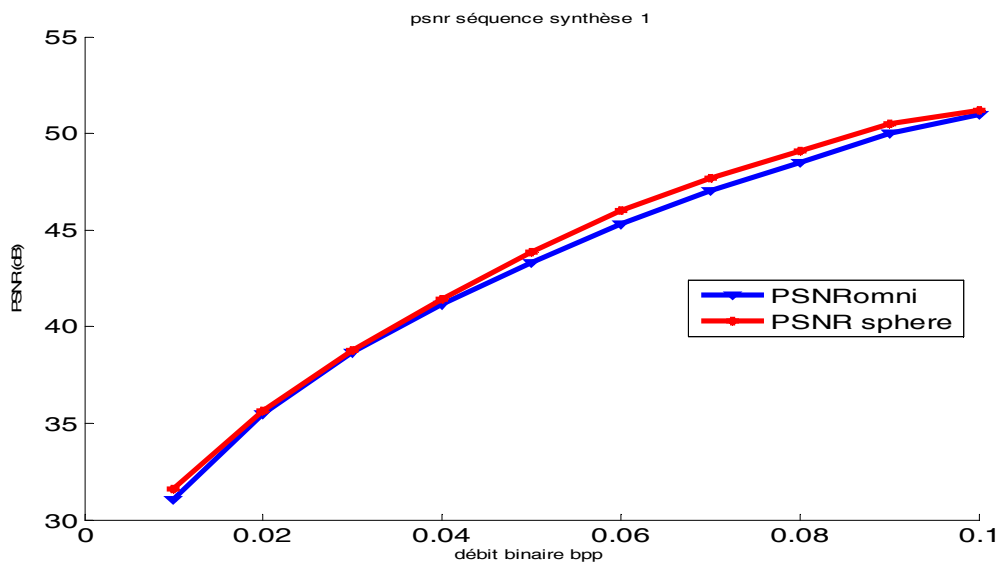


**Figure IV.6 :** Image omnidirectionnelle et image sphérique équivalente de la séquence 3

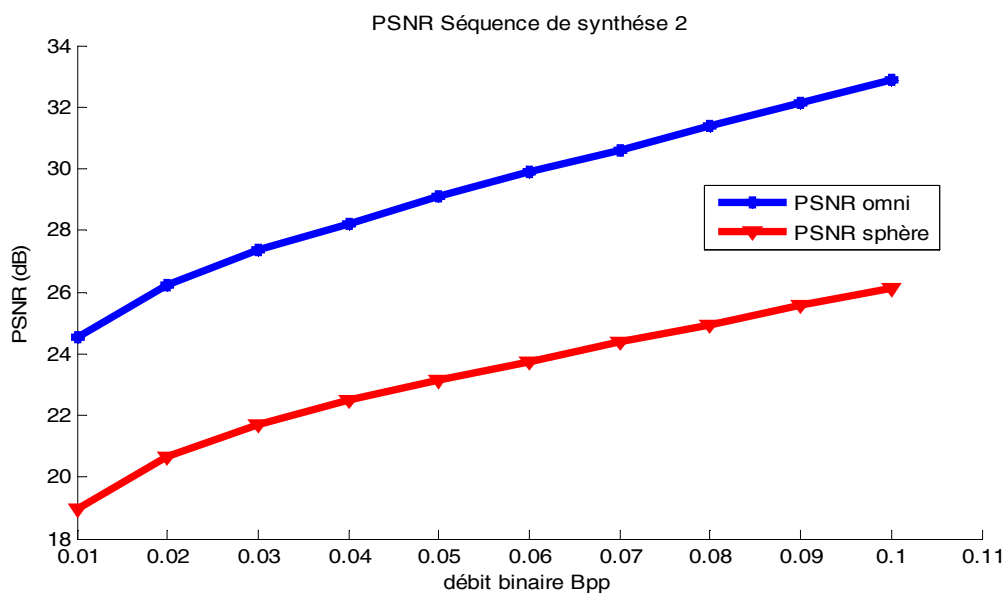
## IV 5 Résultats obtenus et évaluation

### IV 5 1 Résultats obtenus : Séquences de synthèses

Dans cette section nous exposerons les différents résultats comparatifs obtenus dans le cas des images de synthèse en utilisant le standard JPEG 2000. Les résultats obtenus sont illustrés par les figures ci-dessous ainsi que les valeurs moyennes des PSNR sont données par le tableau IV.1 et le tableau IV.2.



**Figure IV.7 :** Courbe comparative de l'évolution du PSNR en fonction du débit binaire. (Séquence de synthèse 1)



**Figure IV.8:** Courbe comparative de l'évolution du PSNR en fonction du débit binaire. (Séquence de synthèse 2)

Les courbes ci-dessus IV.6 et IV .7, ressortent que le PSNR moyens varie lentement en fonction du débit binaire. Nous remarquons que la compression appliquée aux images synthèses (1), les valeurs moyennes du PSNR obtenues sur les images sphériques sont plus élevées que celles obtenues sur les images omnidirectionnelles.

Aussi pour la compression appliquée aux images de synthèses (2) les valeurs du PSNR obtenues sur les images sphériques sont inférieures à celles obtenues pour les images omnidirectionnelles. Notons aussi que cette méthode donne de bons taux de compression.

Débit-bpp (bit/pixel)	Taux de compression (Tc%)	PSNR moyen (dB) OMNI	PSNR moyen (Db) Sphère
0.01	99.87	31.07	31.57
0.02	99.75	35.50	35.63
0.03	99.62	38.67	38.77
0.04	99.5	41.17	41.44
0.05	99.37	43.32	43.83
0.06	99.24	45.33	46.02
0.07	99.12	47.01	47.70
0.08	99	48.47	49.10
0.09	98.87	49.98	50.51
0.1	98.75	50.97	51.21

**Tableau IV.1** : PSNR omni moyen et PSNR sphérique de la séquence de synthèse 1

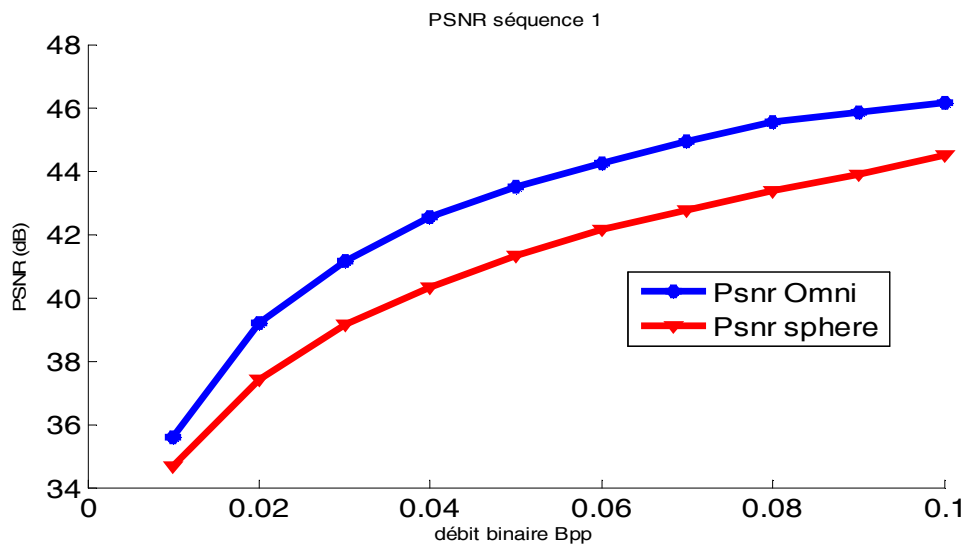
Débit-bpp (bit/pixel)	Taux de compression (Tc%)	PSNR moyen (dB) OMNI	PSNR moyen (Db) Sphère
0.01	99.87	24.54	18.97
0.02	99.75	26.24	20.64
0.03	99.62	27.36	21.69
0.04	99.5	28.22	22.49
0.05	99.37	29.09	23.13
0.06	99.24	29.89	23.75
0.07	99.12	30.61	24.36
0.08	99	31.38	24.92
0.09	98.87	32.12	25.56
0.1	98.75	32.90	26.14

**Tableau IV.2** : PSNR omni moyen et PSNR sphérique de la séquence de synthèse 2

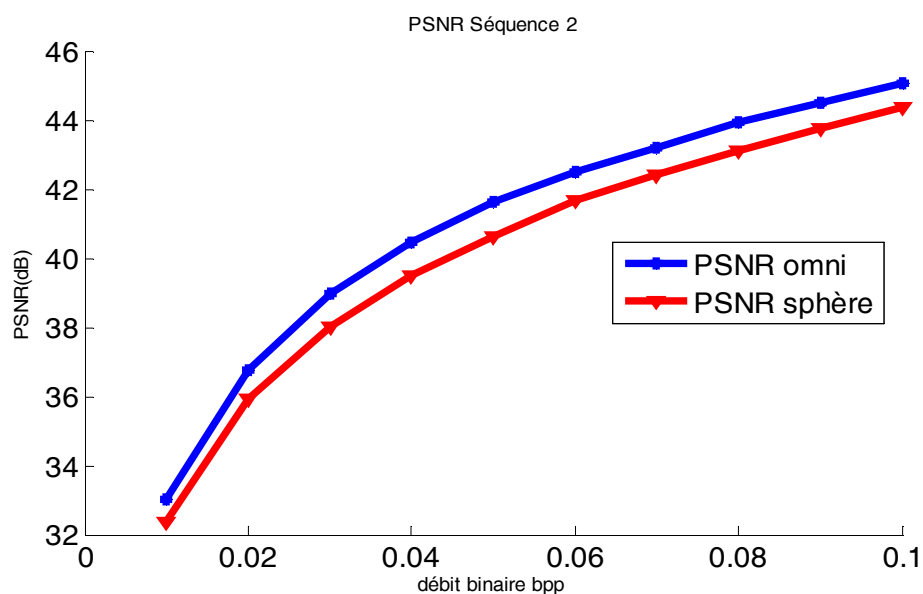
Le tableau IV.1 et le tableau IV.2 nous montrent que les meilleures valeurs du PSNR moyens de la séquence de synthèse (1) sont pour le PSNR moyens Sphérique qui varie entre [31.57 ; 51.21] .Alors que pour la séquence de synthèse (2) les meilleures valeurs du PSNR moyens sont pour les images Omnidirectionnelles qui varie entre [24.54 ; 32.90] pour un débit de compression de 0.01 Bpp.

### IV 5 2 Résultats obtenus : Séquences réelles

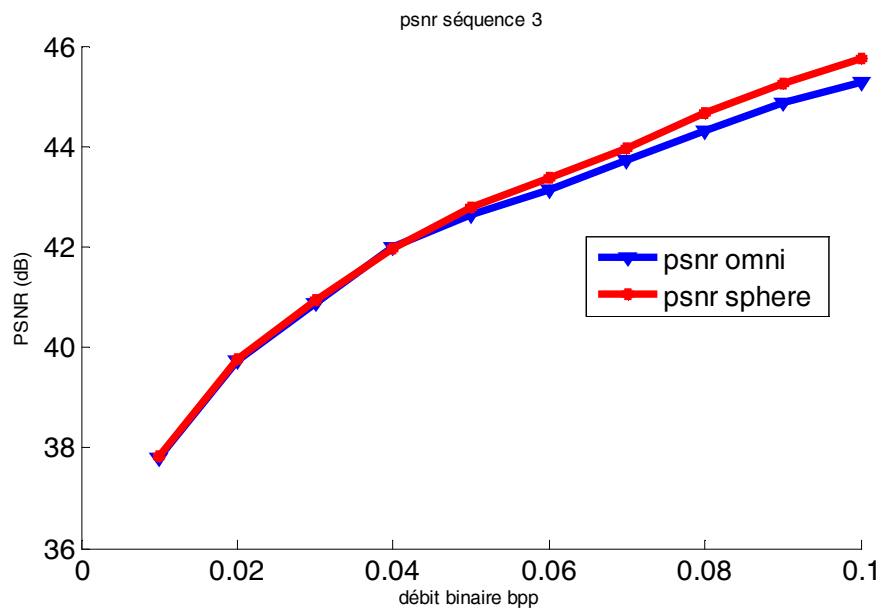
Dans cette partie nous présenterons les différents résultats comparatifs obtenues dans le cas des images réelles en utilisant le standard JPEG 2000. Les résultats obtenus sont illustrés par les figures ci-dessous ainsi que les valeurs moyennes des PSNR enregistrées sont données par les tableaux IV.3, IV.4 et IV.5.



**Figure IV.9 :** Courbe comparative de l'évolution du PSNR en fonction du débit binaire (Séquence 1).



**Figure IV.10 :** Courbe comparative de l'évolution du PSNR en fonction du débit binaire (Séquence 2).



**Figure IV.11:** Courbe comparative de l'évolution du PSNR en fonction du débit binaire (Séquence 3).

Les courbes ci-dessus (Figures IV.9, IV .10 et IV.11), ressortent que le PSNR moyens varie lentement en fonction du débit binaire. Nous remarquons que la compression appliquée aux images des séquences réelles (1) et (2) les valeurs du PSNR obtenues sur le plan image omnidirectionnelles sont plus élevées que celles obtenues sur les images sphériques équivalentes.

Aussi pour la compression appliquée aux images de la séquence réelle (3) les valeurs du PSNR obtenues sur le plan image omnidirectionnelles sont inférieures à celles du PSNR obtenues pour les images sphériques. Notons aussi que cette méthode sphérique donne de bons taux de compression.

Débit-bpp (bit/pixel)	Taux de compression (Tc%)	PSNR moyen (dB) OMNI	PSNR moyen (Db) Sphère
0.01	99.87	35.59	34.65
0.02	99.75	39.21	37.40
0.03	99.62	41.15	39.15
0.04	99.5	42.54	40.32
0.05	99.37	43.48	41.33
0.06	99.24	44.22	42.13
0.07	99.12	44.93	42.76
0.08	99	45.52	43.34
0.09	98.87	45.84	43.87
0.1	98.75	46.14	44.49

**Tableau IV.3 :** PSNR omni moyen et PSNR sphérique de la séquence réelle 1

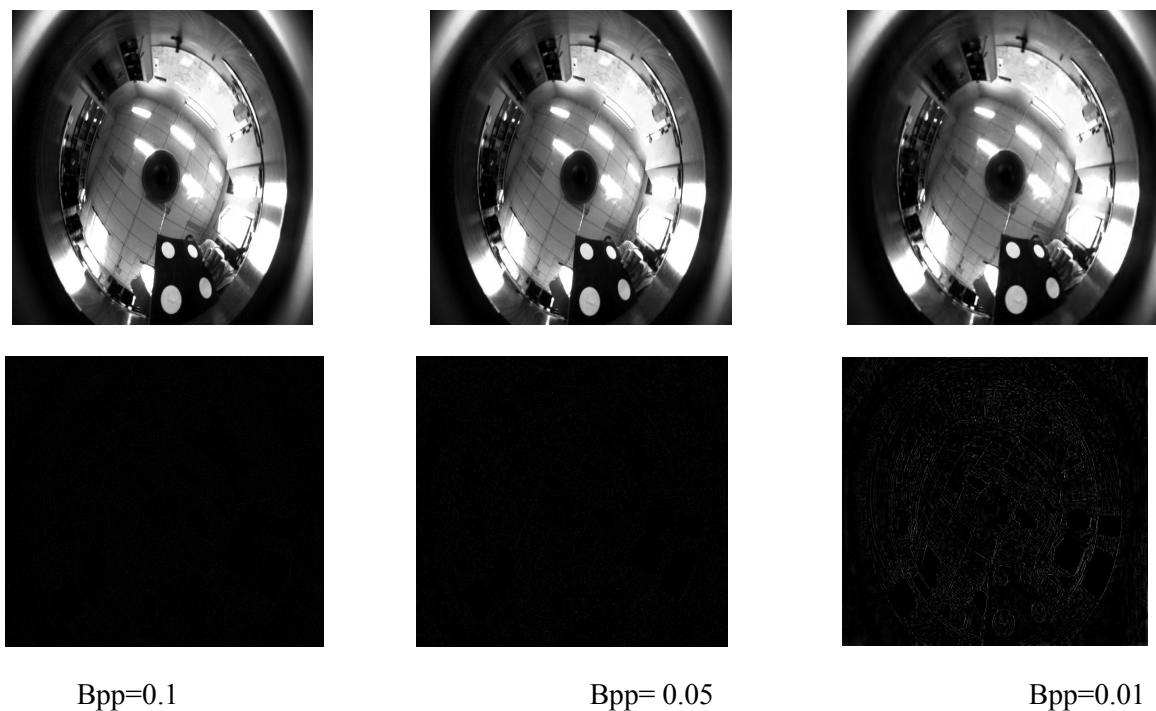
Débit-bpp (bit/pixel)	Taux de compression (Tc%)	PSNR moyen (dB) OMNI	PSNR moyen (Db) Sphère
0.01	99.87	33.00	32.38
0.02	99.75	36.77	35.93
0.03	99.62	38.96	38.00
0.04	99.5	40.44	39.48
0.05	99.37	41.64	40.62
0.06	99.24	42.48	41.65
0.07	99.12	43.18	42.41
0.08	99	43.93	43.10
0.09	98.87	44.51	43.74
0.1	98.75	45.04	44.37

**Tableau IV.4:** PSNR omni moyen et PSNR sphérique de la séquence réelle 2

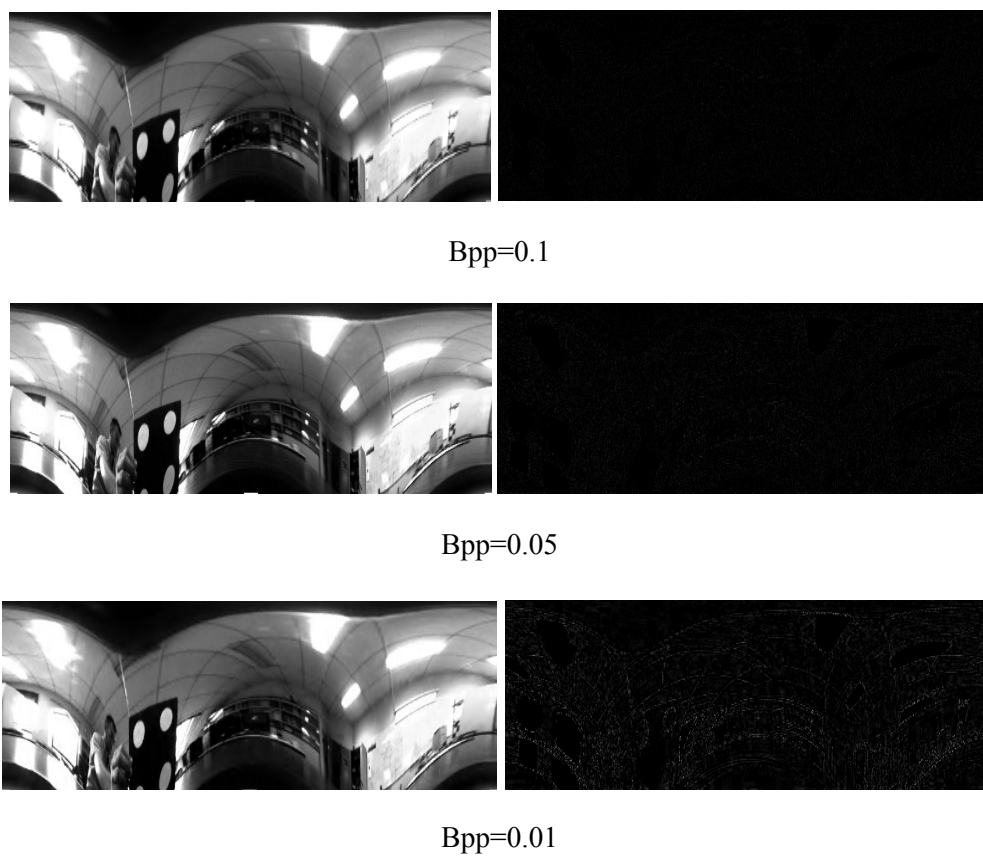
Débit-bpp (bit/pixel)	Taux de compression (Tc%)	PSNR moyen (dB) OMNI	PSNR moyen (Db) Sphère
0.01	99.87	37.80	37.82
0.02	99.75	39.75	39.75
0.03	99.62	40.83	40.94
0.04	99.5	41.99	41.97
0.05	99.37	42.65	42.78
0.06	99.24	43.15	43.37
0.07	99.12	43.73	43.97
0.08	99	44.32	44.66
0.09	98.87	44.88	45.26
0.1	98.75	45.28	45.73

**Tableau IV.5:** PSNR omni moyen et PSNR sphérique de la séquence réelle 3

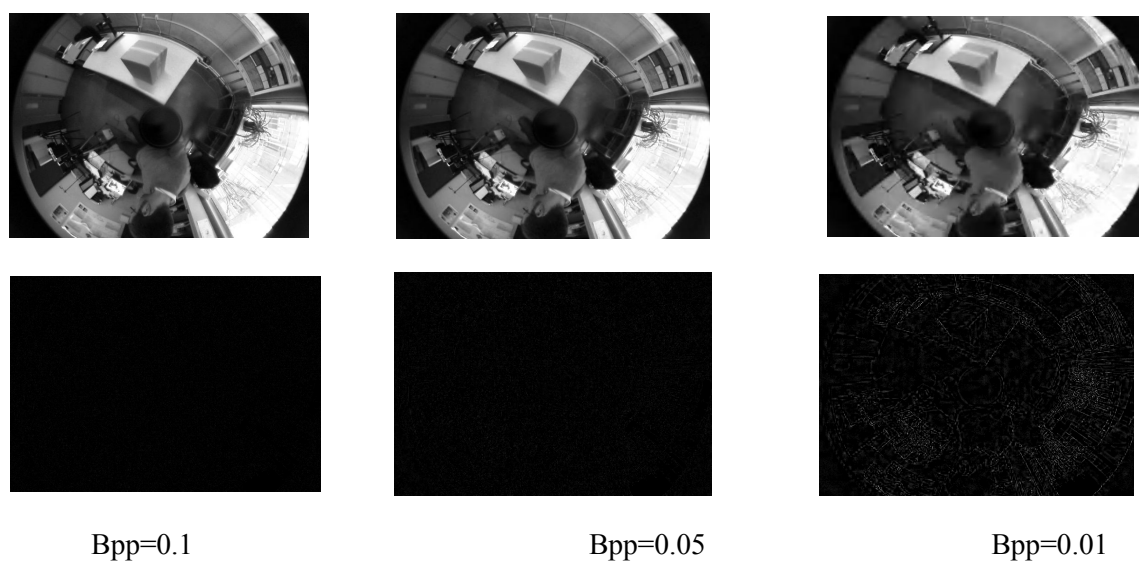
Le tableau IV.3 et le tableau IV.4. nous montrent que les meilleures valeurs du PSNR moyens de la séquence réelles (1) et (2) sont pour le PSNR omnidirectionnelles qui varie entre [35.59 ; 46.17] pour le tableau IV.3 et de [33 ; 45.04] pour le tableau IV.4. Alors que le tableau IV.5 pour la séquence réelle (3) les meilleures valeurs du PSNR moyens sont pour les images sphériques qui varie entre [37.82 ; 45.73 ] pour un débit de compression de 0.01 Bpp.



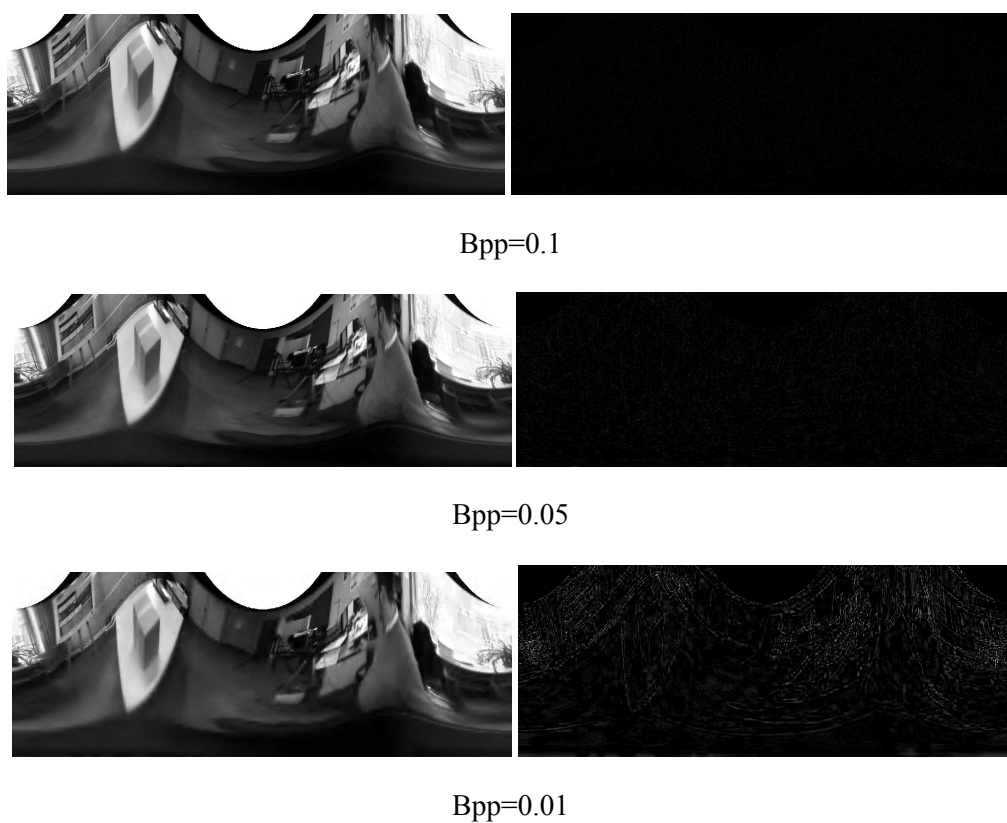
**Figure IV.11 :** Images restituées et les images erreurs de la 1<sup>ère</sup> séquence réelle dans le cas plan image



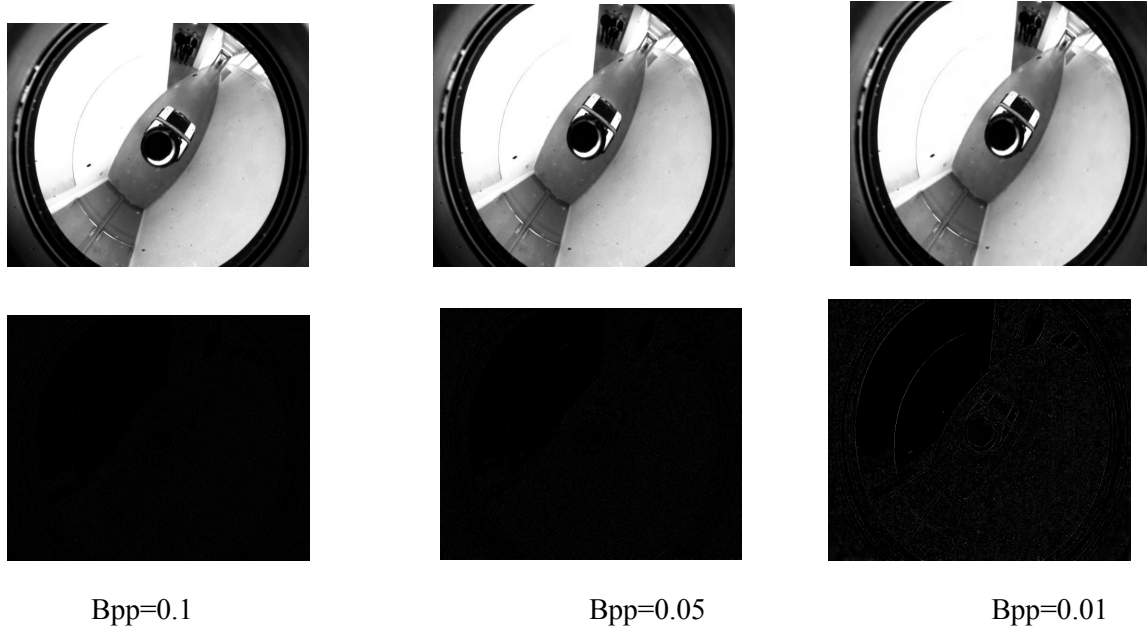
**Figure IV.12 :** Images restituées et les images erreurs de la 1<sup>ère</sup> séquence réelles dans le cas sphérique



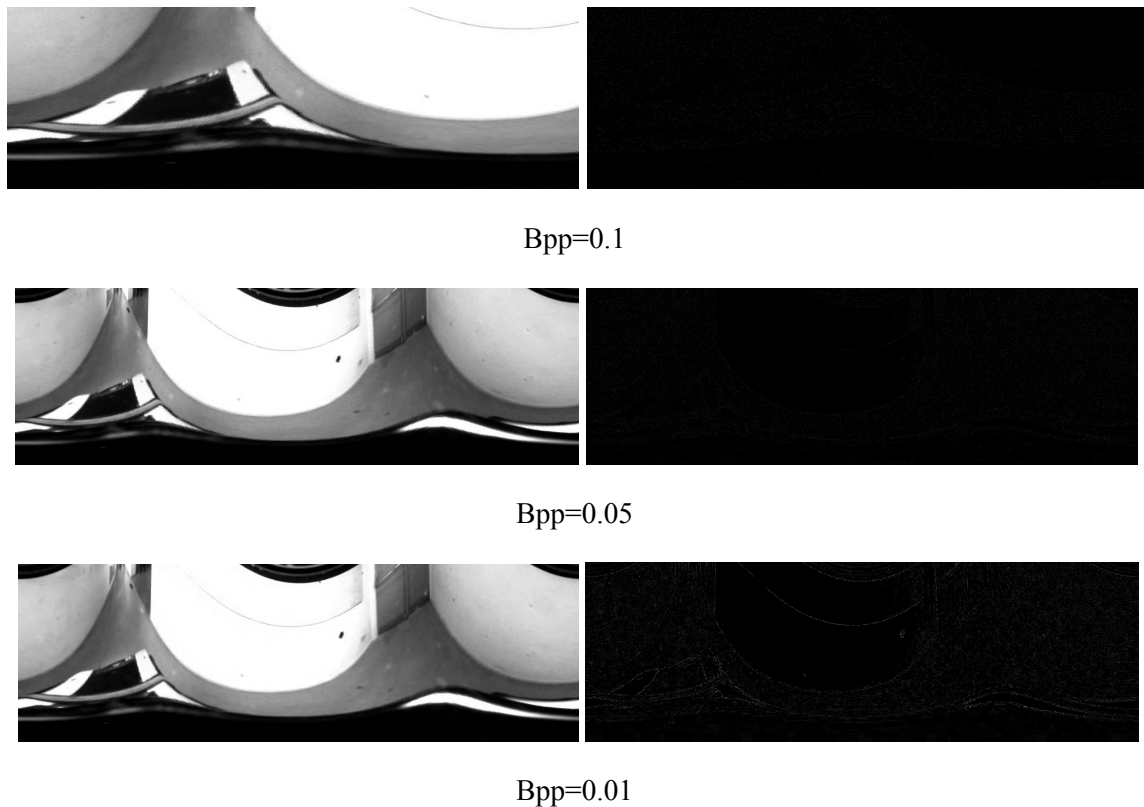
**Figure IV.13 :** Images restituées et les images erreurs de la 2<sup>ème</sup> séquence réelle dans le cas plan image



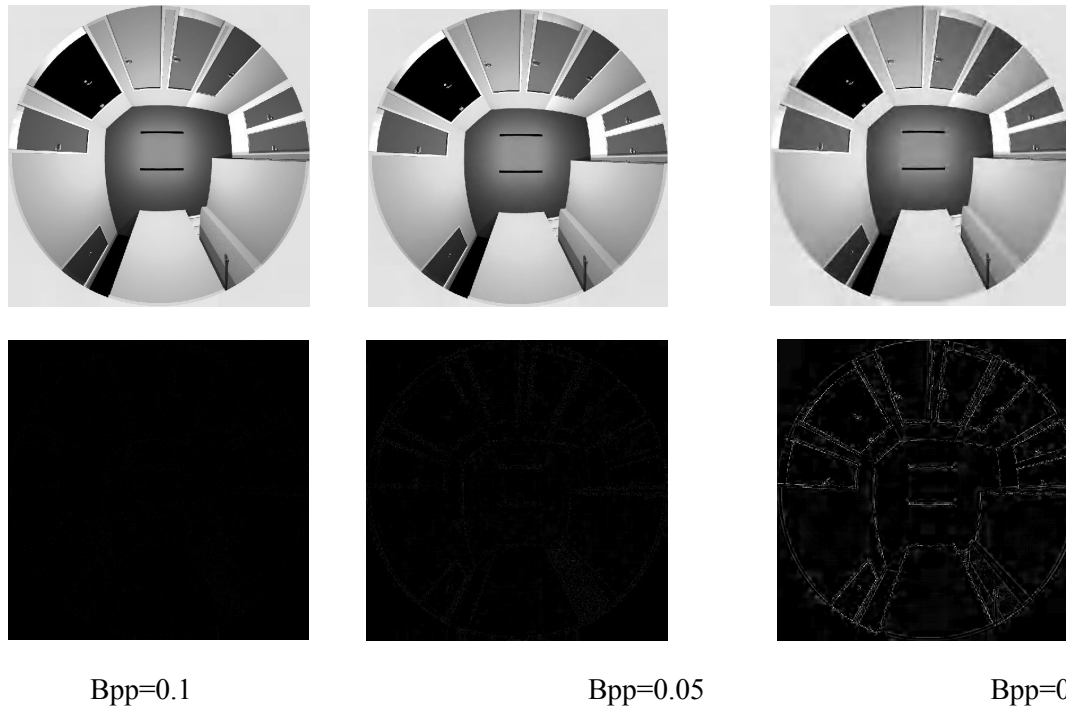
**Figure IV.14 :** Images restituées et les images erreurs de la 2<sup>ème</sup> séquence réelle dans le cas sphérique



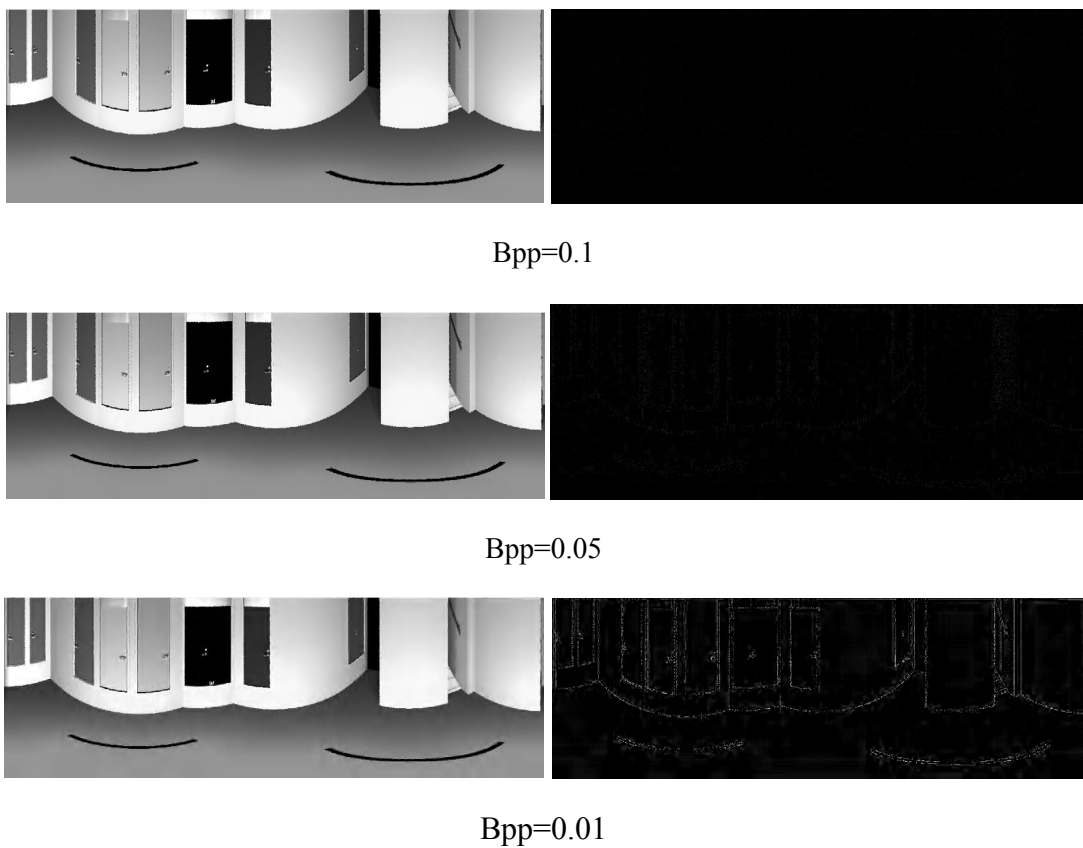
**Figure IV.15 :** Images restituées et les images erreurs de la 3<sup>ème</sup> séquence réelle dans le cas plan image



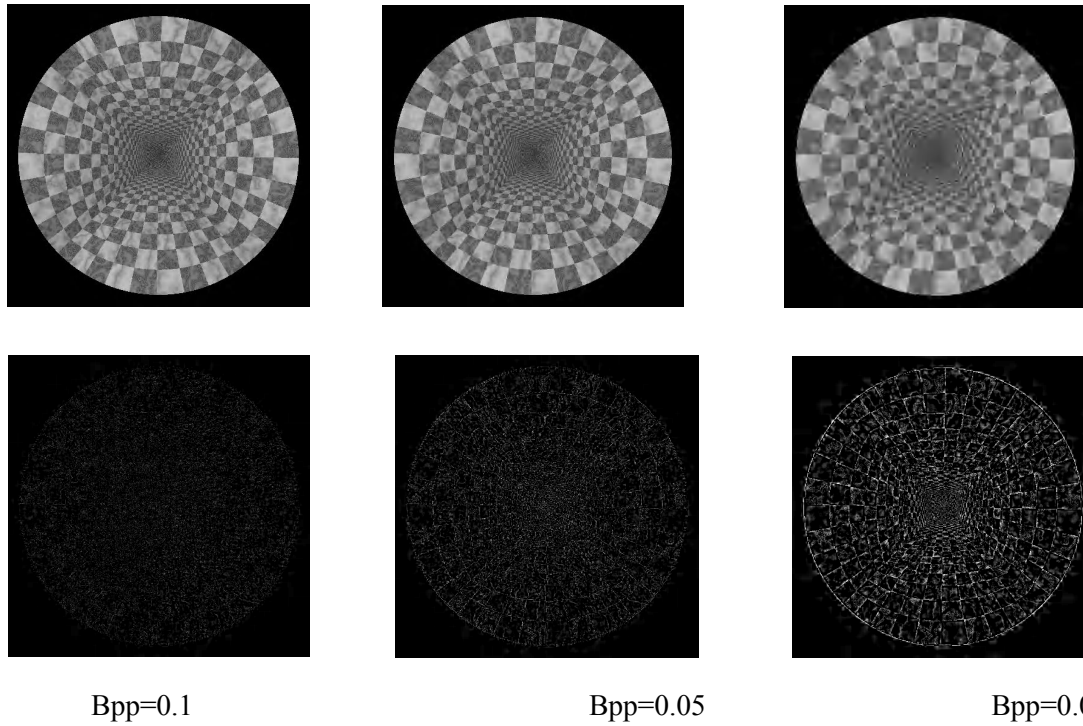
**Figure IV.16 :** Images restituées et les images erreurs de la 3<sup>ème</sup> séquence réelle dans le cas sphérique



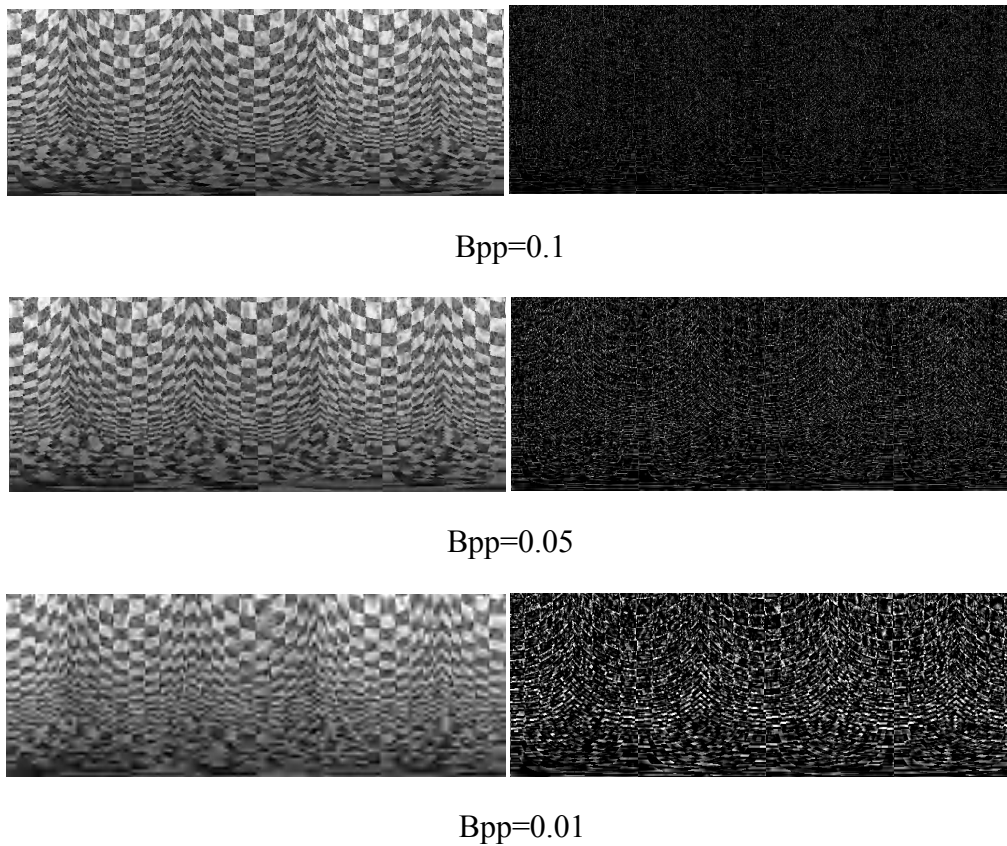
**Figure IV.17 :** Images restituées et les images erreurs de la 1<sup>ère</sup> séquence synthèse dans le cas plan image



**Figure IV.18 :** Images restituées et les images erreurs de la 1<sup>ère</sup> séquence synthèse dans le cas sphérique



**Figure IV.19** : Images restituées et les images erreurs de la 2<sup>ème</sup> séquence synthèse dans le cas plan image



**Figure IV.20** : Images restituées et les images erreurs de la 2<sup>ème</sup> séquence synthèse dans le cas sphérique

Les différents résultats de compression obtenus avec le standard JPEG 2000, appliqué dans le cas des images plan homogènes, montrent bien l'insuffisance de cette démarche dans les images omnidirectionnelles centrales. En effet, les résultats comparatifs obtenus par exemple dans les figures ci-dessus montrent bien cette insuffisance de reconstruction des différents contours existants dans ces images restituées. Nous remarquons d'après les résultats donnés dans les images erreurs qu'il y'a un effet de bloc et de pixellisation dans celles-ci.

Les images reconstruites représentées dans les figures ci-dessus sont restituées à partir de décodeur JPEG 2000 sont presque confondues avec les images originales surtout pour les débit binaire [0.04- 0.1] (voir les courbes PSNR). En revanche, celles des figures des images fortement texturées (synthèse 2 séquence 1) restituées à partir du standard JPEG 2000 avec la méthode utilisant l'espace sphérique présentent des artefacts au niveau des régions contenant des contours et un effet de pixellisation.

#### **IV.6 Discussion**

L'interprétation des différents résultats obtenus avec JPEG 2000 s'est faite en référant aux valeurs du PSNR (dB) obtenues dans l'intervalle du débit binaire [0.01 ; 0.01;0.1] pour la compression de la séquence omnidirectionnelle brute et la compression de la séquence sphérique équivalente.

Nous avons constaté que pour la compression des images qui représentent une texture élevées le travail sur le plan image est plus adéquat alors que pour les images omnidirectionnelles homogènes le travail sur les images sphériques équivalentes est meilleur.

Ces résultats sont due au fait que les ondelettes classiques, qui sont à la base de standard JPEG 2000, sont par construction plus adaptées à la représentation des zones homogènes et lisses qui représentent la plus grande partie de l'image.

## Conclusion

Dans ce travail, nous avons fait une étude comparative de deux méthodes de compression d'images omnidirectionnelles. Cette étude nous a permis de tirer les avantages et les inconvénients ainsi que les domaines d'applications de ces méthodes.

Nous avons réalisé des tests sur des séquences réelles et des séquences de synthèse afin de prouver l'efficacité des deux approches appliquées aux images omnidirectionnelles centrales (à point de vue unique). Les deux approches proposées ont été validées sur des séquences de synthèse dont les paramètres de calibration intrinsèques sont connus.

Nous avons constaté que pour la compression des images omnidirectionnelles qui présentent une texture élevées le travail sur plan image est plus adéquat alors que pour les images homogènes, le travail sur l'espace sphérique équivalent est meilleur.

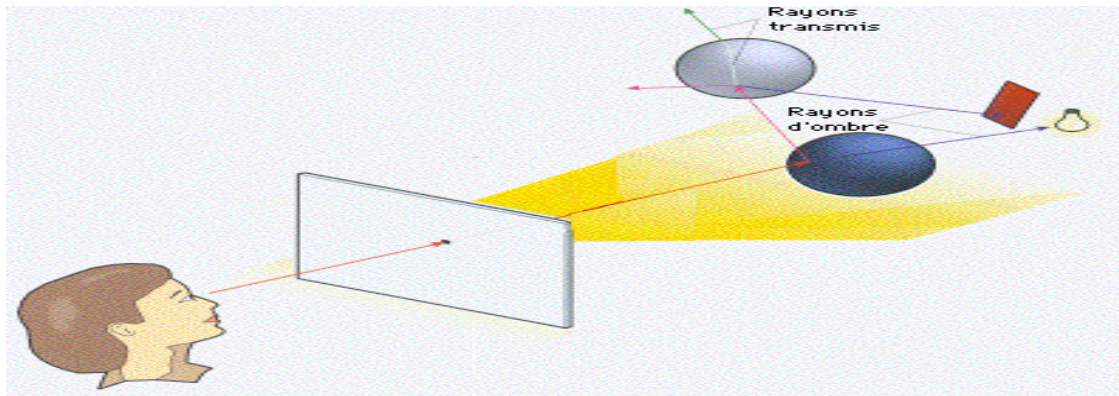
Diverses améliorations peuvent être apportées à notre travail la perspective qui nous semble intéressante pour continuer ce travail est l'adaptation du JPEG 2000 pour les images sphériques en utilisant les ondelettes sphérique 3D cette dernière serait plus adéquate pour une bonne qualité de restitution et surtout pour les images catadioptriques fortement texturées.

### Logiciel POV-Ray :

Le projet (Persistence of Vision Raytracer), ou plus simplement POV-Ray est un logiciel libre, a été initié en 1989 et se base sur le logiciel DKBTrace développé par David K. Buck en 1987 [HIS]. POV-Ray est un outil permettant d'interpréter un code source écrit dans le langage du même nom. Ce langage permet de décrire une scène tridimensionnelle 3D et l'interprétation de ce code va fournir l'image correspondante. POV-Ray est donc à voir comme un compilateur qui ne fournira pas un binaire, mais une image de synthèse. Les images sont générées via un algorithme de raytracing (lancer de rayons).

Le raytracing c'est quoi ?

C'est une méthode de rendu d'image dans laquelle la couleur de chaque pixel est le résultat du trajet d'un rayon lumineux. Les rayons partent de la caméra et subissent tous les effets créés par l'environnement décrit (réflexion, réfraction, diffraction). Cette technique permet d'obtenir des images très réalistes. Il exploite cette technique jusque dans les objets de la scène : en effet, ils sont gérés comme des objets mathématiques parfaits et non comme des ensemble de polygones. Il devient donc possible de rendre une image dans une résolution très élevée sans avoir d'effet de facettes.



Algorithme de lancer de rayons

Création d'un exemple d'une scène basique :

POV-Ray est un logiciel en ligne de commandes. Vous n'aurez donc besoin pour travailler que d'une fenêtre de terminal et de votre éditeur de texte préféré (Kate, gedit, Vim...). Une scène contient une caméra, une source lumineuse et des objets. La première étape pour générer une image avec POV-Ray est de décrire la scène dans un fichier texte ayant pour extension **.pov**. Nous allons travailler à partir de l'exemple suivant que nous sauvegarderons sous le nom **basique.pov**

```

1: #version 3.7;
2: #include "colors.inc"
3: global_settings { assumed_gamma 1.0 }
4:
5: camera { location <-20, 8, 20>
6: right x*image_width/image_height
7: look_at <-2, 4, 0> }
8:
9: light_source { <10, 75, -40> color rgb <1, 1, 1> }
10:
11: // Paysage :
12: sky_sphere { pigment { color LightBlue } }
13:
14: plane { <0, 1, 0>, 0 pigment { color YellowGreen } }
15:
16: // Objets :
17: sphere { <0, 5, 0>, 5 pigment { color OrangeRed } }
18:
19: box { <-10, 0, 0>, <-5, 5, 5> pigment { color Red } }
20:
21: cylinder { <-5, 13, -7>, <-5, 10, +5>, 1 pigment { color White } }
22:
23: // Trièdre :
24: cylinder { <0, 0, 0>, <3, 0, 0>, 0.1
25: pigment { color Red }
26: translate <-1, 0, +7> }
27: cylinder { <0, 0, 0>, <0, 3, 0>, 0.1
28: pigment { color Green }
29: translate <-1, 0, +7> }
30: cylinder { <0, 0, 0>, <0, 0, 3>, 0.1
31: pigment { color Blue }
32: translate <-1, 0, +7> }

```

Au premier coup d'œil, la syntaxe de POV-Ray semble clairement inspirée de celle du langage C : un **#include** en ligne 2, des commentaires introduits par //, un grand nombre d'accolades... À noter qu'il ne s'agit pas ici d'un langage algorithmique, mais d'un langage de description de scène contenant essentiellement des déclarations d'objets géométriques. Contrairement à ce qui se passe dans un algorithme, l'ordre dans lequel sont déclarés les objets n'aura donc ici pas d'importance.

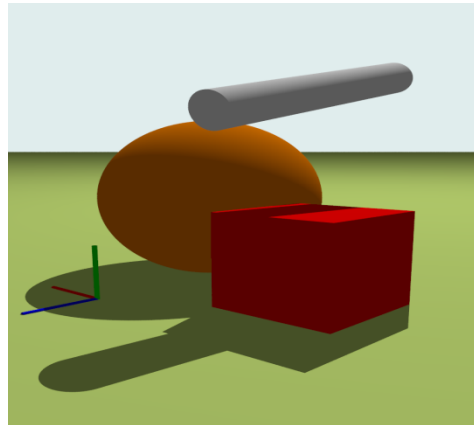


Fig : Rendu de la scène

Attention, le système d'axes de POV-Ray est un peu particulier puisqu'il ne s'agit pas d'un trièdre direct (la règle de la main droite, vous vous souvenez ?). Le trièdre étant indirect, vous pouvez utiliser votre main gauche : l'axe x est représenté par votre pouce pointant vers la droite, l'axe y par votre index pointant vers le haut, et l'axe z par le majeur pointant vers l'écran. La coordonnée y correspond donc à l'altitude et z à la profondeur de l'image. Dans l'image de la figure 1, les axes x, y, z ont été représentés respectivement en rouge, vert, bleu (voir les lignes 23 à 32 du fichier).

Ceci posé, nous pouvons commencer à analyser le fichier source. Première ligne, la directive **#version** permet de spécifier une version du logiciel, ce qui peut être intéressant pour accéder à une fonctionnalité désormais obsolète. À noter que si vous voulez bénéficier de toutes les nouveautés apportées par la version 3.7, vous devez obligatoirement spécifier cette version. N'oubliez pas le point virgule final !

L'instruction **global\_settings** permet de définir des paramètres globaux, comme ici le gamma de l'image, paramètre qu'il est obligatoire de déclarer à partir de la version 3.7. Nous ne rentrerons pas dans les détails de ce paramètre compliqué qui sert à corriger les couleurs apparaissant à l'écran.

Nous n'avons abordé qu'une petite partie des instructions disponibles, chacune de ces instructions possédant de plus de nombreux paramètres permettant de modifier son comportement. Mais vous pouvez déjà créer de belles images de synthèse et même de courts films. Enfin, vous trouverez facilement sur le web tout un tas de tutoriels et d'exemples dans toutes les langues.

**B-1 paramètres intrinsèques du miroir**

Pour la Modélisation d'une caméra catadioptrique à point de vue unique on a pour chaque miroir ces paramètres  $\xi$ ,  $\varphi$  et la surface qui change comme le montre le tableau suivant :

caméra	Surface miroir	$\xi$	$\varphi$
parabolique	$z = \frac{x^2 + y^2}{4p} + p$	1	$1+2p$
Hyperbolique	$\frac{(z+\frac{d}{2})^2}{a_h^2} - \frac{x^2+y^2}{b_h^2} = 1$	$\frac{d}{\sqrt{d^2 + 4p^2}}$	$\frac{d + 2p}{\sqrt{d^2 + 4p^2}}$
Elliptique	$\frac{(z+\frac{d}{2})^2}{a_e^2} - \frac{x^2+y^2}{b_e^2} = 1$	$\frac{d}{\sqrt{d^2 + 4p^2}}$	$\frac{d - 2p}{\sqrt{d^2 + 4p^2}}$
Planaire	$z = \frac{d}{2}$	0	1
conventionnel	Aucun	0	1

**Tableau B.1** : Description des caméras à point central unique :  $a_h$ ,  $b_h$ ,  $a_e$ ,  $b_e$  dépendent uniquement des paramètres intrinsèques du miroir  $d$  et  $p$ .

$4p$  et  $d$  sont respectivement latus rectum et la distance entre les deux foyers dans le cas des miroirs hyperbolique, elliptique et plan.

$a_h = \frac{1}{2} \left( \sqrt{d^2 + 4p^2} - 2p \right)$
$b_h = \sqrt{p \sqrt{d^2 + 4p^2} - 2p^2}$
$a_e = \frac{1}{2} \left( \sqrt{d^2 + 4p^2} + 2p \right)$
$b_e = \sqrt{p \sqrt{d^2 + 4p^2} + 2p^2}$

## Bibliographiques

### Références bibliographiques

- [1] [Nalwa .1996] V Nalwa. *A true Omnidirectional Viewer*. Rapport technique, Bell Laboratoires, Holmdel NJ 07733, USA, February 1996.
- [2] [Nayar .1999] S.K. Nayar S. Baker. A theory of single-viewpoint catadioptric image formation. *International Journal of Computer Vision*, 35(2) :175–196, 1999.
- [3] [Nayar .2001] S.K. Nayar S. Baker. *Single Viewpoint Catadioptric Cameras*. Springer-Verlag, 2001.
- [4] [Geyer .2001] C. Geyer et K. Daniilidis. Catadioptric projective geometry. *International Journal of Computer Vision*, vol. 43, pages 223–243, 2001 .
- [5] [Mouaddib .2005] E. Mouaddib. *Introduction à la Vision Panoramique Catadioptrique. Traitement du Signal*, vol. 22, 2005
- [6] [Cauchois .1999] C. CAUCHOIS, E. BRASSART, C. PÉGARD and C. CLÉRENTIN, Technique for calibrating an omnidirectional sensor. In *IEEE/RSJ int. Conf. on Int. Robots and Systems-IROS 99*, 1999.
- [7] [Geyer .1999] C. GEYER and K. DANIILIDIS, Catadioptric camera calibration. In *ICCV99*, pages 398-404, 1999.
- [8] [Barreto . 2002] J.P. BARRETO and H. ARAUJO, Geometric properties of central catadioptric line images. In *ECCV02*, page IV: 237 ff., 2002.
- [9] [Ying .2003] X. YING and Z. HU, Catadioptric camera calibration using geometric invariants. In *ICCV03*, pages 1351-1358, 2003
- [10] [Guillois, J.P. 1996] Guillois, J.P *Techniques de compression des images*.Edition Hermes ,1996.
- [11] [Huffman. D.A 1952] Huffman. D.A A methode for the construction of minimum redundancy codes *IRE Proc*, vol 40, pages 1098-1101, 1952
- [12] [Davoine, F. 1995] Davoine, F *Compression d'images par fractales basée sur la triangulation de Delaunay*. Institut national polytechnique de Grenoble, France, page 180,1995
- [13] [Totozafiny.T. 2007] Totozafiny.T *Compression d'images couleur pour l'application à la télésurveillance routière par transmission vidéo à très bas débit*. Thèse de doctorat. Ecole doctorale des sciences exactes et de leurs applications. Page 170, 2007
- [14] [Waku Kouomi, J. 1992] Waku Kouomi, J *Ondelettes et applications en imagerie et en calcul de surface*. thèse de doctorat. University Joseph Fourier-Grenoble 1.page 189,1992

## Bibliographiques

- [15] [Wyner.A.D ,J.Ziv, 1994] Wyner.A.D ,J.Ziv, The sliding window Lempel-Ziv Algorithm is asymptotically optimal, Proceeding of the IEEE, Vol. 82,pages. 872-877, june 1994
- [16] [Zhang, W.2004] Zhang, Tutorial on Lempel-Ziv Data compression algorithm, Dept. Of electrical Engineering University of Notre Dame, 2004
- [17] [Ziv, J., Lempel, A.1997] Ziv, J., Lempel A universal algorithm for sequential data compression. IEEE trans. Inform theory, vol.23 , pages.337-334, May 1977
- [18] [Lahdir. M, S.Ameur, A.Adane,2007] Lahdir. M et M, S.Ameur Algorithme non itératif, basé sur les ondelettes biorthogonales et les fractales, pour la compression d'images satellitaires 'télétection, vol.6 n°4, pages.345-360,2007
- [19] [Rao, K.R, yip, P.C 2001] Rao, K.R, yip The transform end data compression handbook CRC press LLC. page 391, 2001
- [20] [Cavaro-Menard.C 2004] Cavaro-Menard.C Mise en place de paramètres quantitatifs caractérisant la dégradation engendrée par une chaîne de compression en ondelettes (JPEG 2000). Groupe signal et image –LISA CNRS-FRE 2656- centre hospitalier Universitaire d'Angers 2004
- [21][Messaoui .O 2008] Messaoui.O « étude comparative des méthodes de compression d'images : application aux images Msg »,mémoire de magister 2008
- [22] [alouache .D.2014] “An Adapted Block-Matching method for optical flow estimation in catadioptric images”, Proceeding of the IEEE;pages 69-74;april 2014

<http://connect.ed-diamond.com/Linux-Pratique/LP-088/Creez-vos-images-de-synthese-avec-POV-Ray>

[http://www.taformation.net/ntics/informatique/lzw/compression\\_lzw.php](http://www.taformation.net/ntics/informatique/lzw/compression_lzw.php)

<http://www.calvat.com/tutorial-compression-RLE.html>

<http://www.public.int-evry.fr/~uro/cours-web/compression-htm>

[http://www.ulb.ac.be/cours/acohen/travaux\\_2006\\_infodoc/compressionNumerique/TypeDonneeImageJPEG2000.htm](http://www.ulb.ac.be/cours/acohen/travaux_2006_infodoc/compressionNumerique/TypeDonneeImageJPEG2000.htm)

<http://home.isr.uc.pt/~jpbar/CatPack/main.htm>

## **Résumé**

Les images catadioptriques, développées récemment, fournissent des images avec un grand champ visuel très large 360°. Néanmoins, ces images contiennent des distorsions radiales importantes. Les méthodes de compressions dans ces images catadioptriques ne sont pas un problème résolu. L'objectif de ce travail est de faire une étude comparative entre JPEG 2000 appliqué directement aux images omnidirectionnelles sur l'espace plan image et JPEG 2000 appliqué aux images sphérique équivalentes.

## **Mots clé**

Image omnidirectionnelle, JPEG 2000, catadioptrique, miroir, parabolique, compression, sphérique.