

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

UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT AUTOMATIQUE

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Domaine : Sciences et Technologies

Filière : Génie Electrique

Spécialité : **Commandes des Systèmes**

Présenté par
Kahina IMMOUN
Zakia HESSAS

Thème

Filtrage des images de profondeur du capteur Kinect

Mémoire soutenu publiquement le 06/07/2017 Devant le jury composé de :

M.me ALKAMA Sadia	MC A,	UMMTO,	Présidente
M.me ADJEMOUT Ouiza	MA A,	UMMTO,	Encadreur
M.me NAIT BELAID Ouiza	MA A,	UMMTO,	Examineur
M.elle Chillali Ouardia	MA A,	UMMTO,	Examineur

Remerciements

Tout d'abord, nous remercions le bon dieu qui nous avoir donné la santé, la volonté pour finir ce travail.

Nous tenons à adresser nos vifs remerciements à notre promotrice M^{me} ADJEMOUT Ouiza, maitre assistante classe A à l'UMMTO pour nous avoir proposé le thème de ce mémoire et de nous avoir dirigés, orientés et conseillés tout le long de notre travail.

Nous remercions également les membres du jury qui nous ferons l'honneur de juger notre travail.

Sans oublier nos chères familles, amies(s) et tous ceux de près ou de loin qui ont contribués à l'élaboration de ce mémoire.

Dédicaces

Je dédie ce modeste travail à :

- * *L'être le plus cher au monde « ma ravissante mère » et pour mon cher père.*
- * *A tous les gens qui m'aime.*

IMMOUN Kahina

Dédicaces

Je dédie ce modeste travail a :

- * Mes très chers parents pour leur aide et leur soutien tout au long de mes études, et qui ont fait de moi ce que je suis aujourd'hui.*
- * Mes chers frères Hilal, Hafidh, Kamel et sa femme Nassima et leur enfants Fares et Ayan, et CHerif et sa femme Ojinia « maro » et leur fils Liam.*
- * Mes très chères sœurs : Khira et ces enfants Adam et Nihal, et Naima qui ma soutenu tout au long de mes études.*
- * Mon cher mari Aghilas et ma belle-famille.*
- * Mes chères amies, et mon binôme Kahina ainsi que sa famille.*
- * Toute la promotion 2016/2017.*

HESSAS Zakia

LISTE DES FIGURES ET TABLEAUX

Figure 1.1: La Kinect	3
Figure 1.2: Spectre de la lumière	5
Figure 1.3: Exemple de motif infrarouge projeté par la Kinect	5
Figure 1.4: Les composantes de la Kinect	6
Figure 1.5: Angle d'inclinaison du capteur	7
Figure 1.6: Squelettes « trackés » et « non-trackés »	7
Tableau 1.1: Les spécifications techniques de la Kinect	8
Figure 1.7: Mesure de la valeur de disparité d . Mise en évidence de la corrélation entre le motif observé (Object plane) et le motif de référence (Reference Plane)	10
Figure 2.1: Le pavage	15
Figure 2.2: Conventions de notation	16
Figure 2.3: La composition d'une image couleur	17
Figure 2.4: Les différents types d'images	17
Figure 2.5: Des exemples de quelques types de bruit dans les images	19
Figure 2.6: Exemple d'application du filtre moyenneur	22
Figure 2.7: Fonction gaussienne en 3D	23
Figure 2.8: Exemple d'application du filtre gaussien	23
Figure 2.9: Exemple de calcul pour le masque médian	25
Figure 2.10 Exemple d'application du filtre gaussien et du filtre médian	26
Figure 2.11: Le filtre bilatéral lisse l'image d'entrée tout en préservant ses arêtes.....	28
Figure 2.12: Les différentes variances des deux paramètres d'un filtre bilatéral sur une image	28
Figure 3.1: (a) Image RGB, (b) Image de profondeur, (c) Image de profondeur avec les fausses couleurs	32

Figure 3.2: (a) Image RVB (couleur), (b) Image de profondeur	33
Figure 3.3: (a) Image de profondeur, (b) Le 1er Filtrage avec un masque de taille 5×5, (c) Le 2ème Filtrage avec un masque de taille 7×7	34
Figure 3.4: L'image de profondeur résultante avec les différentes valeurs d'écart type	35
Figure 3.5: L'image de profondeur résultante avec les différentes valeurs d'écart type	36
Figure 3.6: (a) Image RVB (couleur), (b) Image de profondeur	37
Figure 3.7: (a) Image de profondeur, (b) Le 1er filtrage avec un masque de taille 11×11, (c) Le 2ème filtrage avec un masque de taille 21×21, (d) Le 3ème filtrage avec un masque de taille 39×39.....	38
Figure 3.8: L'image de profondeur résultante avec différentes valeurs d'écart type de voisinage et d'intensité	39
Figure 3.9: L'image de profondeur résultante avec différentes valeurs d'écart type de voisinage et d'intensité	40
Figure 3.10: Remplissage des trous	42
Figure 3.11: (a) Image de profondeur, (b) Image de profondeur en fausse couleurs, (c) Image filtrée	42
Figure 3.12: (a) Image de profondeur, (b) Image de profondeur en fausse couleurs, (c) Image filtrée	43
Figure 3.13: (a) Image de profondeur, (b) Image de profondeur en fausse couleurs, (c) Image filtrée	43
Figure 3.14: (a) Image de profondeur, (b) Image filtrée	43
Figure 3.15: (a) Image de profondeur, (b) Image filtrée	44

SOMMAIRE

Introduction générale.....	1
----------------------------	---

Chapitre 1 : Le capteur RGB-D Kinect

1. 1. Introduction	3
1. 2. Présentation de la caméra Kinect	3
1. 3. Les caractéristiques techniques de la Kinect	4
1. 3. 1. La caméra couleur (RVB)	4
1. 3. 2. L'émetteur laser IR	5
1. 3. 3. Le récepteur IR (caméra IR)	6
1. 4. Installation de Kinect sur PC	8
1. 5. Détection de la profondeur	9
1. 6. Modèle mathématique	10
1. 7. Les limitations de l'image de profondeur	12
1. 8. Calibrage du capteur Kinect	13
1. 9. Les applications de la Kinect en vision par ordinateur	14
1. 10. Conclusion.....	14

Chapitre 2 : Filtrage des images

2. 1. Introduction	15
2. 2. Définition d'une image numérique	15
2. 2. 1. Les différents types d'images	16
2. 3. Le bruit	18
2. 3. 1. Définition de bruit et rapport signal sur bruit	18
2. 3. 2. Les types de bruit	18
2. 3. 2. 1. Bruit linéaire ou bruit additif	18

2. 3. 2. 2. Bruit non linéaire ou bruit multiplicatif	19
2. 4. Débruitage par filtrage	20
2. 4. 1 Définition du filtrage	20
2. 4. 1. 1. Le filtrage linéaire	20
2. 4. 1. 1. 1. Le filtre moyenneur	21
2. 4. 1. 1. 2. Le filtre gaussien	22
2. 4. 1. 2. Le filtrage non linéaire	24
2. 4. 1. 1. 1. Le filtre médian	25
2. 4. 1. 1 .2. Le filtre bilatéral	26
2. 5. Conclusion.....	29

Chapitre 3 : Débruitage des images RGB-D Kinect

3. 1. Introduction	30
3. 2. Définition de l'image de profondeur	30
3. 3. Les bruits dans les images profondeur Kinect	31
3. 4. Acquisition des images RGB-D Kinect	31
3. 5. Filtrages des images de profondeur	32
3. 5. 1. Base de données RGB-D Kinect	32
3. 5. 2. Tests et Résultats	33
3. 5. 2. 1. Par un filtrage linéaire	33
3. 5. 2. 2. Par un filtrage non linéaire	37
3. 6. Stratégie de remplissage des trous des images profondeur	41
3. 7. Conclusion	44
Conclusion générale	45
BIBLIOGRAPHIE	46

***Introduction
générale***

Introduction Générale

Les caméras 3D, ou caméras de profondeur, forment une famille de capteurs ayant connu un essor important au cours des dernières décennies. Du fait de leur coût limité et de leur grande transportabilité, les caméras de profondeur sont très utilisées dans la robotique, ainsi que dans le domaine de la vision par ordinateur.

L'un des principaux atouts de l'acquisition 3D du monde réel est la possibilité d'avoir une information de profondeur (distance) indépendante de la luminosité et une information sur les dimensions réelles de la scène. Cette acquisition peut se faire par différents capteurs comme les scanners 3D, les systèmes de stéréo, les scanner lidar, les caméras temps de vol, ...etc.

Les techniques de mesure de profondeurs (distance) sont multiples. Nous citons à titre d'exemple, le scanner par temps de vol qui utilise un faisceau laser pour la mesure de profondeur. La technique de temps de vol (TOF) consiste à calculer la distance à la surface de l'objet étudié en mesurant le temps nécessaire au trajet d'aller- retour de l'impulsion du faisceau laser réfléchi. Puisque la vitesse de la lumière est connue, le temps de retour permet de déterminer la distance parcourue par la lumière, qui est deux fois la distance entre le scanner et la surface de l'objet.

La technologie capteur de profondeur 3D à base de la caméra infrarouge est un autre exemple et elle permet d'obtenir une image représentant les dégagements thermiques émis par l'objet observé. Parmi ces capteurs notre intérêt est porté sur la caméra Kinect.

Ce capteur est dévoilé en octobre 2010 par Microsoft comme console de jeu Xbox 360 qui permet de contrôler des jeux vidéo sans utiliser de manette. Avec 10 millions d'unités vendus, Kinect est entré le 11 mars 2011 dans le livre Guinness des records comme étant l'accessoire high-tech le plus vendu dans un court laps de temps. Sa deuxième version « Kinect v2 » est présentée en 2014. Ces capteurs sont souvent appelés « capteur RGB-D », car ils donnent à la fois une image de profondeur (Depth) et une image couleur (RGB), les deux étant superposés. Le système Kinect a donné naissance à une nouvelle ère de jeu vidéo qui fournit une interaction directe et temps-réel entre le joueur et la scène de jeu 3D.

Récemment, la communauté de la vision par ordinateur a découvert que la technologie d'estimation de profondeur de Kinect pouvait être étendue bien au-delà des jeux et à un coût

bien inférieur à celui des caméras 3D traditionnels. Ce qui a rendu ce capteur très populaire dans la communauté de vision par ordinateur.

Dans ce travail, nous nous sommes intéressés dans un premier temps à la caméra Kinect, ses caractéristiques sont son fonctionnement et la technique utilisée pour l'estimation de profondeur. Par la suite, nous avons étudié les images de profondeur qui sont caractérisées par différents types de bruit, ainsi, l'objectif principal de notre travail est le prétraitement et le débruitage de ce type de données, cette étape est très importante pour l'exploitation des images de profondeur dans diverses applications.

Le mémoire est principalement divisé en trois chapitres. Que nous avons organisé comme suit :

Le premier chapitre est consacré à la description de la caméra Kinect ses caractéristiques et son fonctionnement.

Dans le deuxième chapitre nous avons abordé les approches de filtrage linéaire et non linéaire sur des images.

Le troisième chapitre est consacré aux techniques de filtrage des images de profondeur Kinect. Et les résultats obtenus sont également représentés.

Chapitre 1

Le capteur RGB-D

Kinect

1.1. Introduction

La Kinect a été conçue à partir du rachat de technologies d'une société appelée « Prime Sense » qui développe des caméras 3D capable de détecter les mouvements. Sur cette base; Microsoft a mobilisé les compétences de ses chercheurs pour intégrer les composants nécessaires à la production d'un périphérique grand public capable non seulement de suivre les mouvements mais aussi d'interagir via la commande vocale.

Le but de ce chapitre est de faire un descriptif de la caméra Kinect, appelé aussi caméra RGB-D, donner ces caractéristiques techniques d'une manière approfondie et aussi son principe de fonctionnement, ces limitations et les domaines de son application.

1.2. Présentation de la caméra Kinect

Le capteur Kinect de Microsoft a été introduit sur le marché en Novembre 2010 comme périphérique d'entrée pour la console de jeu Xbox 360. Ce produit a fait une grande réussite avec plus de 10 millions d'appareils vendus en Mars 2011. La Kinect est un exemple des capteurs les plus exploités pour l'acquisition non intrusive, [1] qui a rencontré assez vite un franc succès à la fois commercial mais aussi chez les développeurs. En juin 2011, Microsoft a publié un kit de développement logiciel (SDK) pour la Kinect, lui permettant ainsi d'être utilisé pour des applications non commerciales ce qui a incité plus d'intérêt pour ce capteur.



Figure 1.1: La Kinect

La Kinect se présente sous forme d'un boîtier noir non fermé (ventilé) (figure (1.1)). Cette interface incorpore essentiellement deux capteurs, l'un délivrant une image vidéo

RVB standard, et l'autre une image de profondeur construite à partir des déformations d'un éclairage infrarouge structuré éclairant la scène à imager.

La première fonction de ce capteur RGB-D est de permettre le contrôle, à partir des gestes du corps, d'une console de jeu vidéo. Ce capteur à faible coût ouvre en même temps de nombreuses autres possibilités intéressantes pour l'imagerie 3D.

1.3. Les caractéristiques techniques de la Kinect

Ce périphérique de Microsoft combine et embarque plusieurs composants (figure (1.4) et Tableau (1.1)) qui sont représentés comme suit:

1.3.1. La caméra couleur (RVB)

Ce système se situe au centre de la barre horizontale (la Kinect) qui combine les trois couleurs primaires tels que le Rouge, le Vert et le Blue. Avec l'association de ces trois couleurs on peut obtenir une image blanche ou un panel infini de couleur qui se situe dans le spectre visible de l'œil humain (figure (1.2)) [2].

Pour la prise d'image cette caméra couleur embarque une caméra couleur RGB (standard) avec un capteur photographique de type CMOS, qui permettra la capture de 30 images par seconde avec une résolution VGA (Video Graphics Array) de 640×480 pixels codé sur 8 bits [3], et une résolution maximale de 1280 x 960 pixels jusqu'à 12 images par seconde [4].

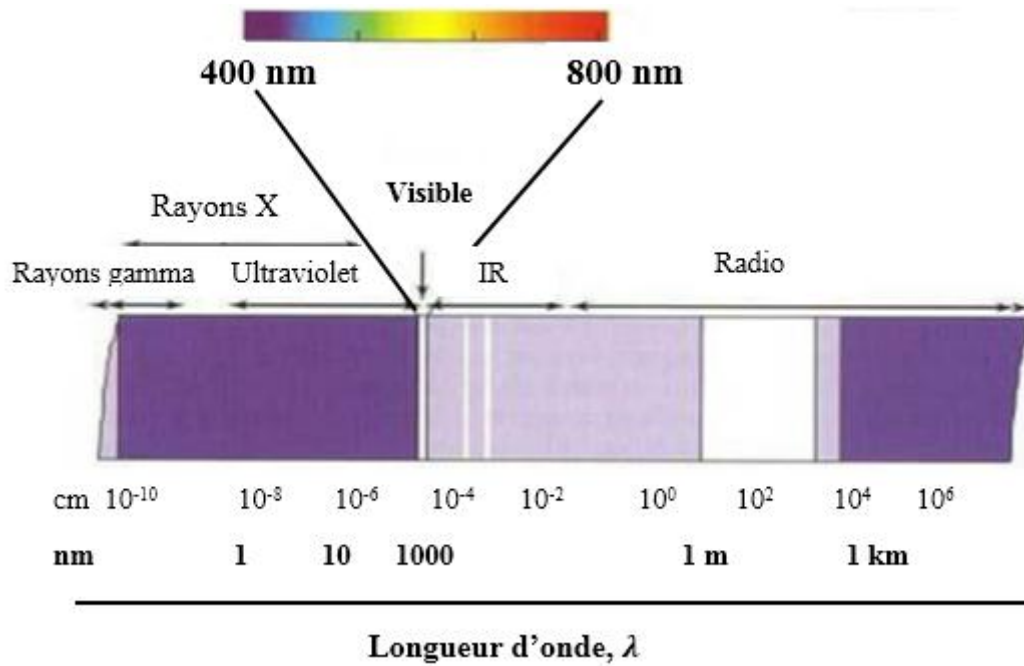


Figure 1.2 : Spectre de la lumière

1.3.2. L'émetteur laser IR

L'émetteur infrarouge est un système d'illumination qui réfléchit une partie de sa lumière reçue vers le centre optique de la caméra infrarouge [5], qui est invisible à l'œil nu, compris entre 0.7 et $1.6 \mu\text{m}$ avec une définition de l'ordre de 5 millions de points (figure (1.3)) [6].



Figure 1.3 : Exemple de motif infrarouge projeté par la Kinect

1.3.3. Le récepteur IR (La caméra IR)

Le récepteur infrarouge fonctionne de la même manière qu'un capteur CMOS (RVB), mais CMOS IR laisse uniquement le passage des infrarouges grâce à un filtre [1].

Il permettra d'obtenir des images représentant les dégagements thermiques émis par l'objet observé, cependant les images obtenues ne sont pas colorées, ce sont les utilisateurs qui décident de rajouter les niveaux de couleurs en fonction de la température mesurée.

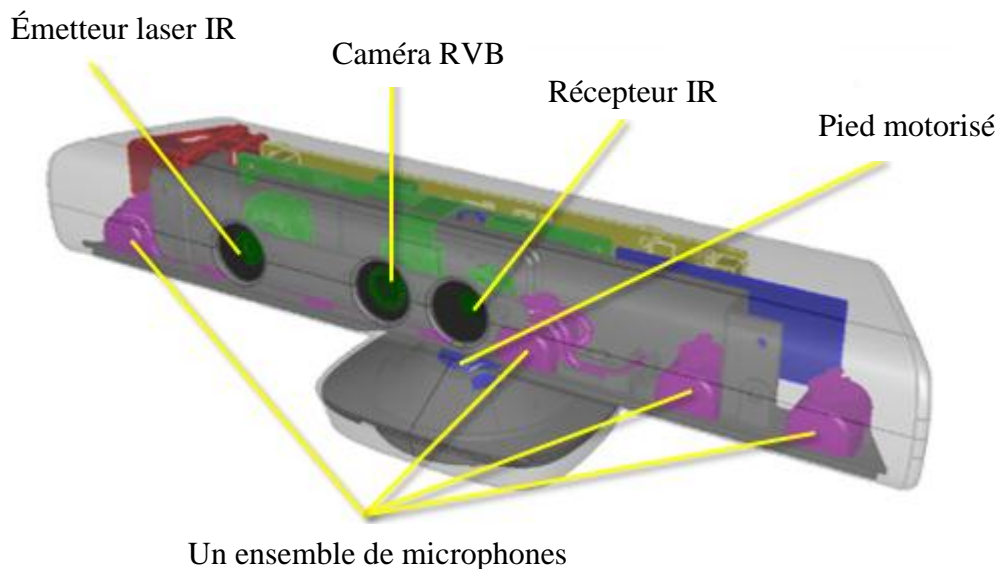


Figure 1.4 : Les composantes de la Kinect

Ces capteurs ainsi que le socle sont placés sur un pied motorisé qui permet à la caméra d'effectuer des petits mouvements vers le haut ou vers le bas (l'inclinaison verticale, jusqu'à un maximum de ± 27 degrés), cela pour suivre les mouvements et les déplacements des joueurs dans une pièce (figure (1.5)) [4].

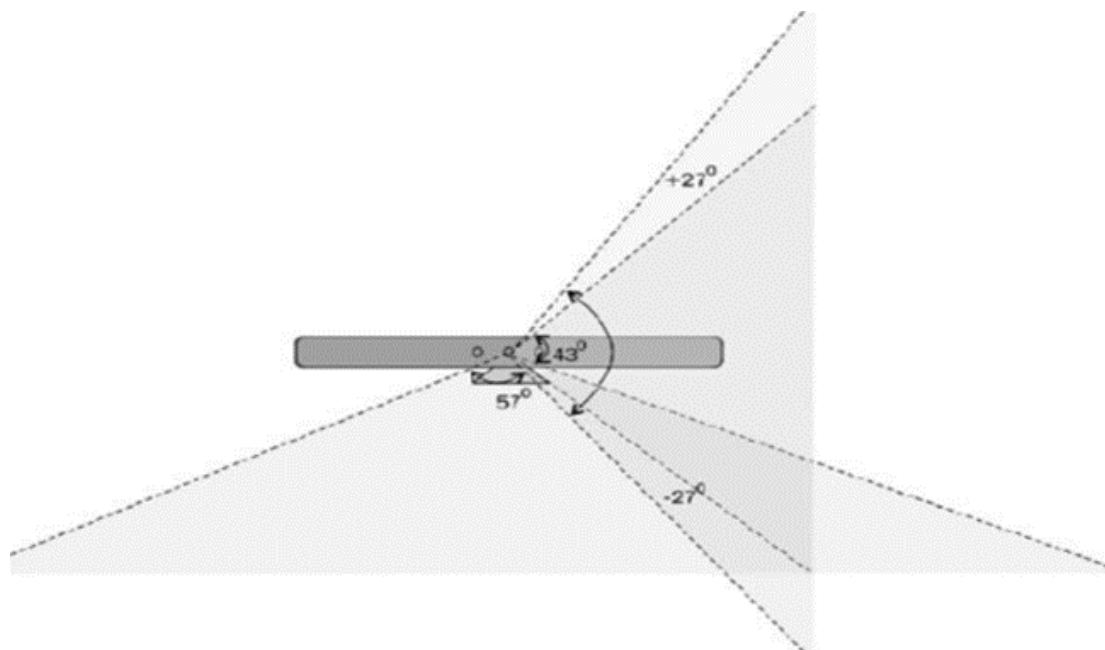


Figure 1.5 : Angles d'inclinaison du capteur et de son champ visuelle

On cite aussi parmi ces caractéristiques, le champ visuel qui est limité à 57 degrés horizontal, 43 degrés vertical et la portée du capteur 80 cm à 4 m comme le montre la figure (1.5).

Son système de reconnaissance physique admet jusqu'à 6 individus et 2 uniquement actifs, suivre les squelettes en mode debout avec 20 articulations et également suivre les squelettes en mode assis avec 10 points de jointeurs (figure (1.6)) [4].

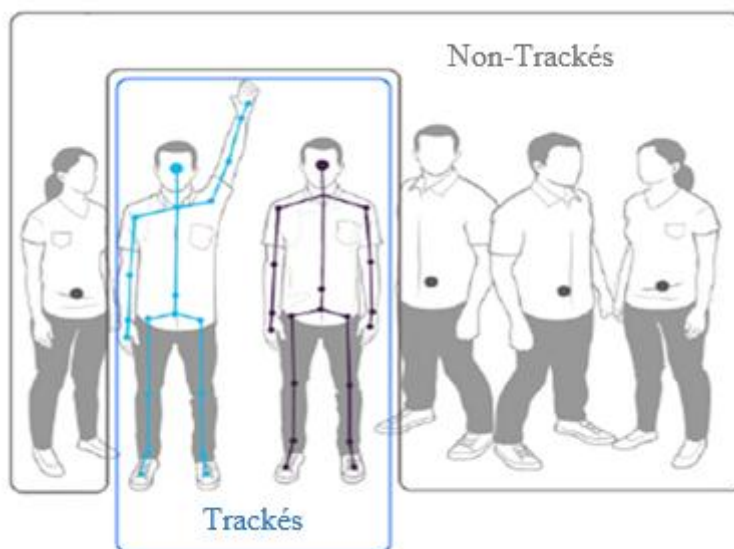


Figure 1.6 : Squelettes « trackés » et « non-trackés »

Comme elle est composée d'un système audio de quatre microphones qui sont disposés à des endroits différents dans un ordre linéaire qui fonctionnent de la même façon qu'une oreille humaine. Ces microphones permettent l'identification de la personne qui parle ainsi que la reconnaissance vocal en trois langues en 2011 (anglais, espagnol et japonais) d'où chaque signal audio est échantillonné avec une fréquence de 16 kHz et codé sur 16 bits [4,22].

Un voyant d'activité (LED) placé sur la façade qui a une couleur verte ou rouge, qui est utilisé pour indiquer l'état de l'appareil c'est-à-dire, les pilotes de ce périphérique sont chargés correctement ou sont pas chargés correctement [4,20].

On peut résumer tout ça dans un tableau qui inclut tous les spécifications matériel (hardware) de la Kinect.

Paramètres	Valeurs	
Résolution spatiale	RVB/Profondeur /IR	640pix ×480pix
	X	1.70 mm/pixel /mètre
	Y	1.64 mm/pixel /mètre
La portée de la profondeur	Défaut	0.8 m-4.0 m
	Proche	0.4 m-3.0 m
Résolution de profondeur	2 mm a 40 mm (dépend de la profondeur)	
Taux d'images	30 fps (image /second)	
Champs de vision	43° Vertical et 57° Horizontal	
Pied motorise	±27° Vertical	
Distance focale	Profondeur	5.453±0.012 mm
	RVB	4.884±0.006 mm
Longueur d'onde IR	830 nm	

Tableau 1.1 : Les spécifications techniques de la Kinect

1.4. Installation de Kinect sur PC

La Kinect est munie d'un port USB classique qui permet de récupérer les images en couleur et les images de profondeur sur l'ordinateur. L'acquisition de ces images nécessite plusieurs logiciels fournis gratuitement sur le web, qui fonctionnent sous Windows 7, Windows 8 ainsi que sous Linux.

Fin 2010, PrimeSense, a publié Open NI (Open Natural Interaction) qui est un framework open source permettant de développer des applications utilisant des interactions naturelles (voix, mouvements du corps, etc.). Ce framework est accompagné d'un driver open source pour la Kinect et d'un middleware propriétaire appelé NITE qui s'occupe à proprement parler du traitement des images de profondeur. Le middleware NITE permet de segmenter les différentes personnes en face de la caméra, d'obtenir leur pose ou encore de reconnaître certains gestes. Le framework Open NI est conçu pour être indépendant des capteurs utilisés et des algorithmes de traitement [9].

Ainsi que Kinect SDK qui est un kit de développement logiciel adapté. Le SDK offre une multitude de nouveaux outils et de fonctionnalités pour permettre aux développeurs de simplifier le développement d'applications et de créer des applications plus intelligentes. En effet les applications utilisent la voix humaine et les gestes de la même manière que les gens communiquent les uns avec les autres [10].

1.5. Détection de la profondeur

La mesure de la profondeur des objet dans une scène se fait par le principe de la triangulation, ce dernier est une technique qui est basé sur l'émission de la lumière structuré dans sa forme la plus simple est un rayon qui se manifeste par un point de surbrillance sur l'objet visé (figure (1.3)), la détection de cette lumière se fait grâce à un angle spécifique qui existe entre l'émetteur et le récepteur pour faciliter la mise en correspondance, le motif projeté est codé d'une manière à pouvoir le reconnaître facilement dans l'image.

La lumière infrarouge projetée est en fait matérialisée par cette multitude des points clairs qui forme un motif non aléatoire dont la différence avec le motif projeté connu permet de calculer la disparité, avec cette mise en correspondance il suffit ensuite de trianguler les point afin d'obtenir l'information de profondeur ou bien la relation entre la disparité et la profondeur [11, 2] (équation 1):

$$d = \frac{b \times f}{Z} \quad (1.1)$$

Où :

d : La disparité (La différence entre deux images d'un même objet).

b : La Baseline (La distance entre le centre de la source et le centre du capteur).

f : La focale.

Z : La profondeur.

1.6. Modèle mathématique

Le schéma dans la figure (1.7) illustre la relation entre la distance d'un objet (point k) et de capteur par rapport à un plan de référence pour mesurer la disparité (d). Pour exprimer les coordonnées 3D des points de l'objet, l'en considérons comme un système de coordonnées de profondeur avec son origine au centre optique de la caméra IR. L'axe Z est orthogonal au plan de l'image vers l'objet, l'axe X perpendiculaire à l'axe Z en direction de la ligne de base (b) entre le centre de la caméra IR et du projecteur laser, et l'axe Y perpendiculaire à X et Z donnant un système de coordonnées.

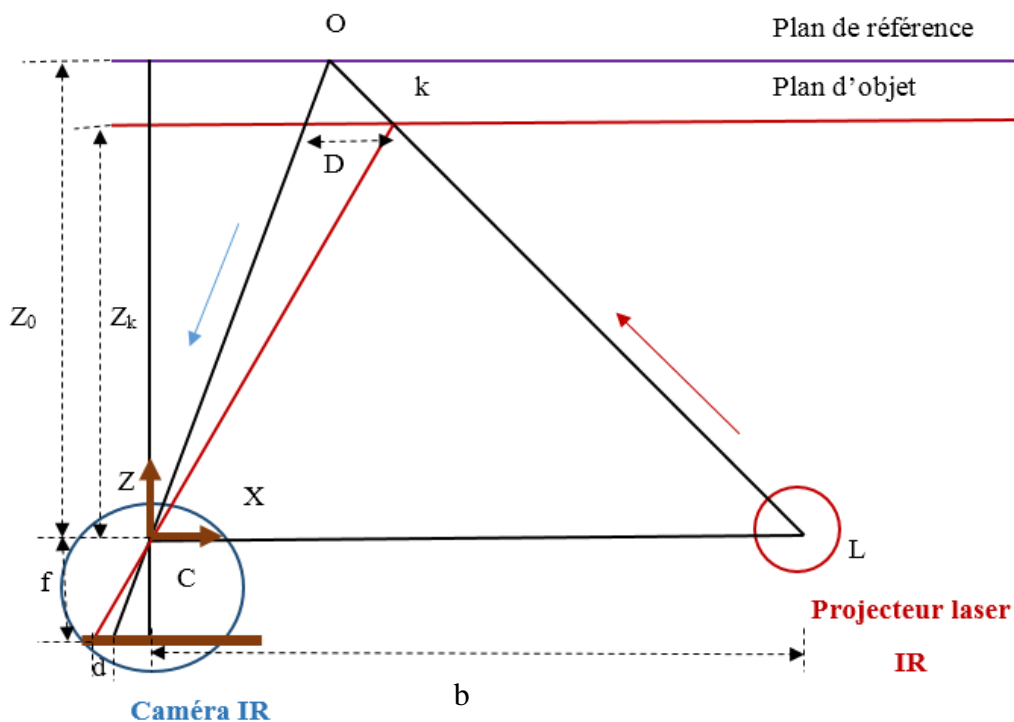


Figure 1.7: Mesure de la valeur de disparité d . Mise en évidence de la corrélation entre motif observé (Plan d'objet) et le motif de référence (Plan de référence)

Supposons qu'un objet se trouve sur le plan de référence à une distance de Z_0 du capteur, la nuée de points projetée sur l'objet est capturée sur le plan image de la caméra infrarouge. Si l'objet est décalé plus près (ou plus loin) du capteur alors l'emplacement de la nuée de points sur le plan de l'image sera déplacée dans la direction X et sera mesuré en fonction de la disparité et des coordonnées du point k. De Thalès (De la similitude des triangles), nous avons :

$$\frac{d}{f} = \frac{D}{Z_k} \quad (1.2)$$

$$\frac{D}{b} = \frac{Z_0 - Z_k}{Z_0} \quad (1.3)$$

Où :

Z_k : La distance du point k de l'objet de l'espace. (Représentatif de la profondeur d'un objet en mètre).

Z_0 : La distance entre le plan de référence et la ligne de base (en mètre).

b : La longueur de base (distance entre l'émetteur et le récepteur, en mètre).

D : Le déplacement du point k par rapport à sa position dans le plan de référence (en mètre).

f : La distance focale de la caméra infrarouge (en mètre).

d : Le déplacement (disparité) observée dans l'espace image (en mètre).

La substitution de D de l'équation (1.3) dans l'équation (1.2) permet d'obtenir une expression de Z_k (l'équation (1.4)).

$$Z_k = \frac{Z_0}{\frac{Z_0}{fb}d + 1} \quad (1.4)$$

En divisant l'équation (3) par Z_0 , elle devienne :

$$Z_k = \frac{1}{\frac{1}{Z_0} + \frac{d}{fb}} \quad (1.5)$$

Cependant Z_0 , f et b sont des constantes connues (en usine) issues de la calibration de la caméra Kinect. Les coordonnées X_k et Y_k peuvent être obtenues par les formules suivantes :

$$\begin{aligned} X_k &= -\frac{Z_k}{f} (x_k - x_0 + \delta x) \\ Y_k &= -\frac{Z_k}{f} (y_k - y_0 + \delta y) \end{aligned} \quad (1.6)$$

Où x_k et y_k sont les coordonnées image du point k , x_0 et y_0 sont les coordonnées du point principal; δx et δy sont les corrections de distorsion de l'objectif, pour lesquelles différents modèles avec différents coefficients existent.

Remarque : il suppose que le système de coordonnées image est parallèle à la ligne de base et donc avec le système de coordonnées de profondeur.

Pour la normalisation de l'équation (1.5), d doit être remplacé par $m\rho+n$. Avec ρ l'écart normalisé de la disparité, et m , n les paramètres d'une normalisation linéaire. Les inclure dans l'équation (1.5) donne :

$$Z_k = \frac{Z_0}{\left(\frac{m}{fb}\right)\rho + \left(\frac{n}{Z_0^{-1} + \frac{n}{fb}}\right)} \quad (1.7)$$

L'équation (1.7) exprime une relation inversement proportionnel entre la profondeur d'un point et sa disparité normalisée correspondante. Pour observer la disparité normalisée d'un certain nombre de points de l'objet (ou plan) à des distances connues du capteur, les coefficients de cette relation linéaire peuvent être estimés de façon moindre au carré. Toutefois, l'inclusion des paramètres de normalisation ne permet pas de déterminer b et Z_0 séparément [7, 12].

1.7. Les limitations de l'image de profondeur

- La portée de la Kinect est d'environ 80 cm à 4 mètre et la portée optimale est d'environ 1.2 à 3.5 mètre. Au-delà de 5 mètres, le rayonnement IR réfléchi devient trop faible pour être mesurable avec précision. Pour tout objet dont la distance est inférieure à 1 mètre, le phénomène inverse est observé et le signal devient totalement

saturé. Les points qui sont les plus proches de la caméra et les plus éloignées de la caméra sont marqués en noir, i.e. non détectés [20].

- Comme pour les images en couleur, les zones qui ne sont pas éclairées par la grille IR ne peuvent pas être mesurées, c'est le problème des zones ombragées et les zones d'occlusion [2, 21].
- Le désalignement entre l'image en couleur et l'image de profondeur, ce décalage entre les images est causé par le décalage entre la caméra RVB et la caméra IR estimé par 2.5 cm [19].
- Des erreurs systématiques dans le calcul de coordonnées des objets sur des points particuliers (les erreurs des capteurs), qui viennent principalement d'un calibrage inadéquat et de mesures inexactes de disparités [2].
- Les erreurs de mesure causé par la mise en place sont principalement liée aux conditions d'éclairage externe qui influe sur la corrélation et la mesure de disparité aussi en fonction de la géométrie de l'image des parties de la scène peuvent être occulté (la surface est plus grande que le champ de vue de la Kinect) [2].
- Bruit sur les contours des objets : ce bruit apparait comme des trous dans les données de la Kinect (pixels noirs). On rappelle que la mesure de distance se fait uniquement si la lumière des points émis par le projecteur IR est réfléchi vers le récepteur IR, or que la réflexion de la lumière IR aux frontières s'effectue suivant des angles qui ne permettent pas le retour vers le récepteur IR.
- Réflexions sur les surfaces lisses (miroirs, vitres, ...etc.), pour ce type de surface il n'y a pas de diffusion de la lumière IR qui permet le retour de celle-ci vers le récepteur IR [21].

1.8. Calibrage du capteur Kinect

L'étalonnage ou calibrage de caméra revient à modéliser le processus de formation des images, c'est-à-dire trouver la relation entre les coordonnées spatiales d'un point de l'espace avec le point associé dans l'image prise par la caméra.

La Kinect a été très utilisée en vision par ordinateur. La plupart de ces applications nécessitent que la paire de caméras soit étalonnée à la fois de manière intrinsèque et extrinsèque. Et ce pour éliminer les distorsions dans le nuage de points et les désalignements

entre les données de couleur et de profondeur, en raison de la distance entre la caméra IR et la caméra RVB. Plusieurs travaux ont été réalisés pour le calibrage du capteur Kinect [13].

1.9. Les applications de la Kinect en vision par ordinateur

Depuis que capteur Kinect de Microsoft a été introduit sur le marché en Novembre 2010 en tant que périphérique d'entrée pour la console de jeu Xbox 360, la communauté de vision par ordinateur a exploité la technologie de détection de profondeur a d'autres applications que les jeux, et avec un coût beaucoup plus bas que les caméras 3D traditionnelle. En effet, La complémentarité de l'information de profondeur et les informations visuelles fournies par le capteur Kinect notamment sa caméra RGB a ouvert de nouvelles possibilités pour résoudre des problèmes fondamentaux en vision par ordinateur. Des applications récentes en vision par ordinateur sont classées en fonction du type de problèmes de vision telle que la localisation et la cartographie simultanées en 3D (SLAM) [12,13], le suivi des personnes [16], la reconnaissance d'objets [17] et l'analyse de l'activité humaine [18] Etc.

Récemment, on trouve dans la littérature beaucoup d'état de l'art sur les applications du capteur Kinect en vision par ordinateur [3].

1.10. Conclusion

Dans ce chapitre nous avons donné une présentation de la caméra Kinect, ces caractéristiques techniques et le principe utilisé par ce capteur pour récupérer l'information de profondeur ainsi que les limitations des données de profondeurs obtenues.

Le capteur RGB-D Kinect de Microsoft a attiré une attention considérable dans les domaines de la vision par ordinateur et la reconnaissance de formes, en plus de son faible cout, Kinect est capable de générer des images profondeur et des images couleur simultanément. Cependant, limité par la mesure de la profondeur et les propriétés de la surface de l'objet, l'image en profondeur de Kinect contient inévitablement du bruit optique ainsi que d'autres types de bruits, ce qui la rend inadéquat dans de nombreuses applications.

Chapitre 2

Filtrage des images

2.1. Introduction

L'étude d'une image numérique acquise par différent type d'appareils photos nécessite de supprimer ou d'éliminer le maximum de bruit parasite dû aux conditions d'acquisition. De ce fait, nous nous intéressons aux méthodes de filtrage des images numériques, pour obtenir une image qui soit la plus proche possible de l'image idéale.

Toutefois, notre objectif dans ce chapitre n'est pas de présenter les filtres de manière exhaustive, mais de présenter quelques techniques de lissage pour réduire le bruit, afin d'obtenir une meilleur image pour son analyse.

2.2. Définition d'une image numérique

Une image numérique est définie comme un signal fini échantillonné à valeurs quantifiées dans un certain espace de couleurs [23]. Le signal est multidimensionnel, en général 2D ou 3D donné par une mesure physique. Une image numérique est associée à un pavage de l'espace, en général rectangulaire, chaque élément du pavage, appelé pixel, est désigné par ses coordonnées entières [24] (figure (2.1)).

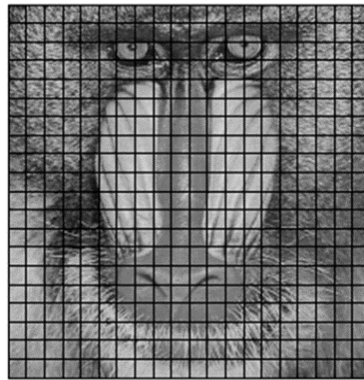


Figure 2.1 : Le pavage

Une image numérique 2D est représentée par un tableau I (matrice I) de h lignes et w colonnes. Le pixel est désigné par un couple (i, j) .

Où i est l'indice de colonne $i \in \{0, w-1\}$, et j l'indice de ligne $j \in \{0, h-1\}$, w est la largeur, h la hauteur de l'image I . Par convention, le pixel origine $(0,0)$ est en général en haut à gauche de l'image (voir figure (2.2)).

Le nombre $I(i,j)$ est la valeur (le niveau de gris) du pixel (i,j) , $I(i,j) \in \{0, N_{max} - 1\}$. N_{max} est le nombre de niveaux de gris [24].

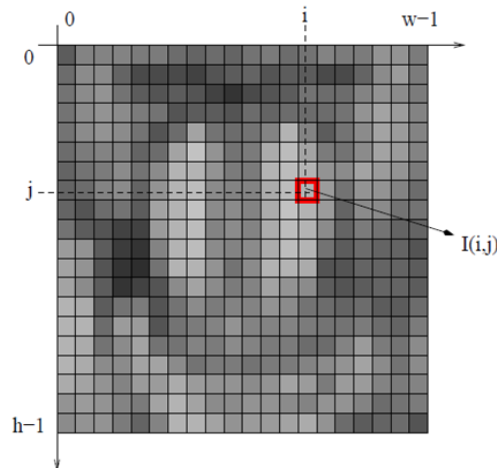


Figure 2.2 : Conventions de notation

2.2.1. Les différents types d'images

Il existe plusieurs type d'images (figure (2.4)) qui sont définit comme suit :

- **Image binaire**

Une image binaire est une image $h \times w$ où chaque point peut prendre uniquement la valeur 0 (noir) ou 1 (blanc). Le niveau de gris est codé sur un bit, donc la relation sur les niveaux de gris devienne: $I(i,j) = 0$ ou $I(i,j) = 1$ [23].

- **Image en niveaux de gris**

Une image en niveaux de gris autorise un dégradé de gris entre le noir (0) et le blanc (255). En général, le niveau de gris est codé sur un octet (8 bits) soit 256 nuances de dégradé. L'expression de la valeur du niveau de gris avec $N_g = 256$ devienne: $I(i,j) \in [0, 255]$.

- **Image couleur**

Une image couleur est la composition de trois (ou plus) images en niveaux de gris sur trois (ou plus) composantes. On définit donc trois plans de niveaux de gris, un rouge, un vert et

un bleu. La couleur finale est obtenue par une synthèse additive de ces trois composantes (ou plus) (figure (2.3)).

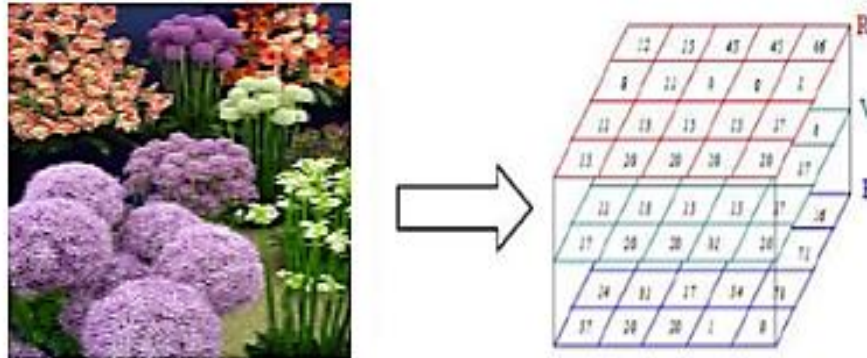


Figure 2.3 : La composition d'une image couleur

On a les relations sur les niveaux de gris: « $I_r(i,j)$, $I_v(i,j)$, $I_b(i,j)$ » $\in [0,255]$. On voit bien sur la figure qu'une image couleur est en fait l'association de trois plans de niveau de gris, chacun d'eux étant une couleur de base.



Image en niveaux de gris

$$I(i, j) \in [0,255]$$

Image binaire

$$I(i, j) \in \{0, 1\}$$

Image couleur

$$I_r(i, j), I_v(i, j), I_b(i, j)$$

Figure 2.4: Les différents types d'images

2.3. Le bruit

2.3.1. Définition de bruit et rapport signal sur bruit

Le bruit (grain) est un signal parasite aléatoire de nature différente qui altère l'image, qui peut être modélisé d'une façon déterministe.

Il est considéré comme un phénomène de brusque variations de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage, des dispositifs optiques et électroniques du capteur (mauvaise mise au point, distorsion géométrique,...), par les processus d'acquisition (bougé, flou, dérive lumineuse,...) et de transmission ou de stockage (codage, quantification,...)

Il apparaît comme une structure granuleuse plus ou moins fine et/ou un moutonnement coloré. Les fabricants s'efforcent de le minimiser mais il est impossible de l'éliminer complètement car c'est une composante même de la matière. La figure (2.5) montre des exemples de quelque type de bruit.

2.3.2. Les types de bruit

Il existe deux principaux types de bruits:

2.3.2.1. Bruit linéaire ou bruit additif

Ce genre de bruit est lié à l'électronique du capteur qui affecte tous les pixels de l'image par l'ajout d'une valeur aléatoire, on parle du bruit uniforme « bruit gaussien ». Ce bruit de type gaussien est plus difficile à supprimer et le bruit généré par les appareils d'acquisition se rapproche généralement du bruit gaussien, donc pour cela il est nécessaire de concevoir des algorithmes de suppression de bruits efficaces, pour objectif de la réduction ou suppression du bruit dans une image on fait plusieurs filtrages.

2.3.2.2. Bruit non linéaire ou bruit multiplicatif

Quelques pixels de l'image prennent des valeurs complètement aléatoires, on appelle aussi un bruit impulsif, lié aussi au parasitage des lignes de transmissions entre les composantes électroniques.

Ici on s'intéressera au bruit « Poivre et Sel » qui modélise assez bien les poussières sur un objectif ou sur un scanner, des petits objets sur l'image (par exemple un objet clair sur un fond foncé et dont la taille dans l'image serait proche du pixel: il pourrait apparaître ou disparaître lors d'une séquence vidéo créant ainsi du bruit) ainsi que des pertes de données.

Pour ajouter ce bruit de $p\%$ à une image de dimension $w \times h$, il suffit de colorier $w*h*p\%/2$ pixels en noir et $w*h*p\%/2$ pixels en blanc, ces pixels étant choisis aléatoirement. La figure (2.(5.d)) présente un exemple d'un tel bruit [25].



(a) bruit blanc gaussien additif



(b) flou de bougé



(c) compression JPEG très forte



(d) bruit poivre et sel

Figure 2.5 : Exemples de quelques types de bruit dans les images

2.4. Débruitage par filtrage

Afin de pouvoir améliorer la qualité visuelle de l'image, on doit éliminer ou au moins réduire les effets des bruits cités ci-dessus, en lui faisant subir un traitement bien précis « le filtrage ».

2.4.1. Définition du filtrage

Le filtrage est un traitement qui s'applique pour chaque pixel de l'image, Le principe du filtrage est de modifier la valeur des pixels d'une image, généralement dans le but d'améliorer son aspect. En pratique, il s'agit de créer une nouvelle image en se servant des valeurs des pixels de l'image d'origine.

Parmi les différents types de filtrage, certains sont linéaires et certains d'autres sont non linéaires. Les filtres peuvent effectuer plusieurs types d'opération comme le lissage ou le rehaussement de contours.

2.4.1.1. Le filtrage linéaire

Un filtre linéaire transforme un ensemble de données d'entrée en un ensemble de données de sortie selon une opération mathématique appelée convolution. Lorsqu'il s'agit de données numérisées comme dans le cas du traitement d'image, la relation entre les valeurs des pixels de sortie et celle des pixels d'entrée est décrite par un tableau de nombres, généralement carré, appelé matrice de convolution.

Un opérateur h de traitement d'image est dit linéaire si : $h(a.I + b.J) = h(a.I) + h(b.J)$. Autrement dit, l'application du filtre à l'image originale I consiste à déplacer le masque h sur chaque pixel de l'image, et à remplacer la valeur de ce pixel par une combinaison des valeurs de ses voisins.

- Un produit de convolution, est un opérateur mathématique utilisé pour multiplier des matrices entre elles. Dans notre cas, la convolution d'une image par un filtre est donnée par les relations suivantes :

$$I'(i,j) = I(i,j) * h(i,j) \quad (2.1)$$

$$I'(i,j) = \sum_x \sum_y I(i-x, j-y) \cdot h(x,y) \quad (2.2)$$

I' est l'image résultante ; I est l'image originale ; h est le cœur de tous les changements qui vont affecter l'image I , appelé masque de convolution, noyau de convolution, filtre, fenêtre, kernel,...

Cette convolution numérique d'image se fera par une sommation de multiplications, d'où la taille de l'image résultante est la même que celle de l'image originale [26].

Pour mieux comprendre ce filtre comment ça marche, on a choisi deux filtre « le filtre moyeneur et le filtre gaussien ».

2.4.1.1.1. Le filtre moyeneur

Le principe de ce filtre est très simple, un pixel est remplacé par la moyenne de lui-même et de ses voisins, c'est dans la définition du voisinage que les filtres vont différer. On peut considérer un voisinage en connexité 4 ou 8, ou même encore plus large.

$$\frac{1}{5} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

4 connexités 8 connexités

Pour une implémentation plus rapide, de préférence utiliser des filtres avec des coefficients entiers puis diviser ensuite le résultat par la somme des coefficients (cela permet d'effectuer des opérations sur des entiers et sur les réelles).

Le filtre moyeneur est un filtre passe bas permettant ainsi d'éliminer les hautes fréquences, correspondant au bruit. Son inconvénient est qu'il élimine également les hautes fréquences correspondantes aux détails de l'image : il rend ainsi l'image moins bruitée mais plus floue [25] (figure (2.6)).



originale (poivre et sel a 2%) filtre moyennneur 5×5 filtre moyennneur 11×11

Figure 2.6 : Exemple d'application du filtre moyennneur

On observe que le lissage permet d'éliminer une partie de bruit, mais les détails sont altérer

Ces effets :

- Permet de lisser l'image.
- Réduit le bruit et les détails non-important.
- Brouille ou rend floue l'image.
- Filtre dont tous les coefficients sont égaux (chaque pixel est remplacé par la moyenne de ses voisins).
- Plus le filtre grossit, plus le lissage devient important et plus le flou s'accroît [26].

2.4.1.1.2. Le filtre gaussien

Le filtre gaussien est un filtre isotrope (c'est-à-dire que les propriétés de leur noyau de convolution sont identiques dans toutes les directions), spécial avec des propriétés mathématiques bien précises, son principe est une convolution avec une gaussienne en deux dimension $G(x, y)$ [26] :

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (2.3)$$

σ : L'écart type.

x : L'indice de colonne, y : L'indice de ligne.

La fonction gaussienne est très commune dans la nature, elle est souvent utilisée dans les distributions statistiques (figure (2.7)).

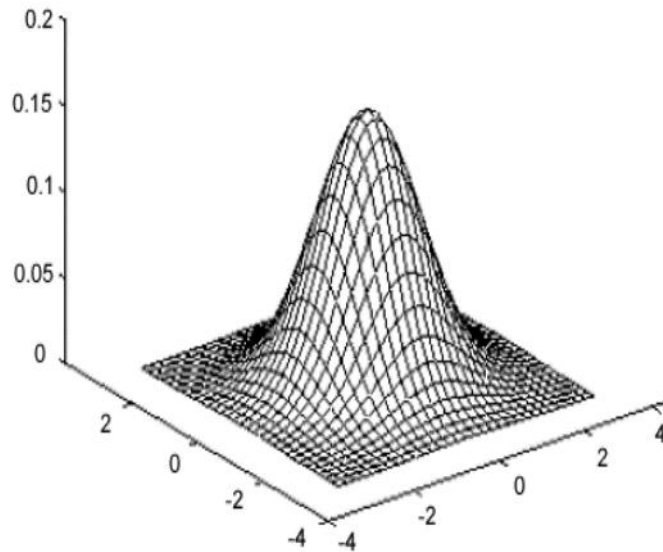
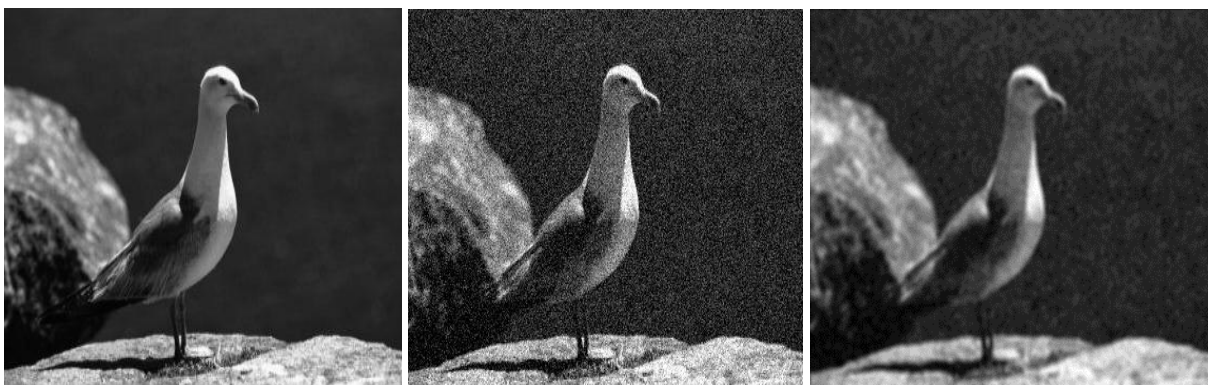


Figure 2.7: Fonction gaussienne en 3D

Le filtre gaussien utilise un paramètre sigma « σ » s'appelle l'écart-type ou la déviation standard, il détermine la largeur de la cloche gaussienne (l'intensité du lissage).

Le lissage gaussien permet une mise en évidence plus marquée des régions à forte variation de luminosité (de corriger le bruit dans les parties homogènes des images) mais moins efficace que le lissage moyenneur. Cependant, il dégrade moins les détails que le lissage moyenneur [25]. Autrement dit il donnera un meilleur lissage et une meilleure réduction du bruit que le filtre moyenneur (figure (2.8)).



originale

bruit gaussien

filtre gaussienne 5×5

Figure 2.8: Exemple d'application du filtre gaussien

On observe que le lissage gaussien (avec une taille 5×5 et un $\sigma = 2.5$) permet de réduire le bruit gaussien surtout dans la partie de la mer mais les détails sont altérés. Pour un filtre 5×5 , il faut 5×5 multiplications et 24 additions par pixel.

Remarque:

En général un filtre gaussien avec un $\sigma < 1$ est utilisé pour réduire le bruit, et si $\sigma > 1$ c'est dans le but de fabriquer une image qu'on va utiliser pour faire un "masque flou" personnalisé qu'on applique souvent aux images satellitaires. Il faut noter que plus σ est grand, plus la cloche gaussienne est large et plus le flou appliqué à l'image sera marqué [26].

- Le filtrage linéaire montre des limites dans deux aspects particuliers de l'amélioration des images numériques :
 - Le bruit impulsionnel : Lorsque l'image fait apparaître des valeurs parasites aberrantes qui ne dépendent pas des valeurs initiales de l'image, le filtrage linéaire diffuse ces valeurs aberrantes sur les pixels voisins [27].
 - Le respect des frontières : Le filtrage linéaire lisse les transitions abruptes, rendant ainsi les images floues (la mauvaise conservation des contours).

2.4.1.2. Le filtrage non linéaire

On désigne par filtre non linéaire un opérateur qui remplace la valeur de chaque pixel par une combinaison non linéaire des valeurs de ses pixels voisins. L'objectif des filtres lisseurs non linéaires est d'apporter une solution aux limites exprimées ci-dessus, par l'utilisation d'une (ou de plusieurs) techniques :

- Statistiques d'ordre : Opérateurs basés sur un classement des valeurs des pixels voisins (minimum, maximum, médiane, . . .).
- Moyennes robustes : Calculs de moyennes locales - éventuellement pondérées- en excluant du calcul les valeurs marginales.
- Anisotropie : Opérateurs où le support de calcul du filtre effectivement calculé varie localement en fonction du contenu de l'image [27].

Il existe un grand nombre de filtres non linéaires exploitant une ou plusieurs de ces propriétés. Nous présentons dans ce travail le filtre médian et le filtre bilatéral.

2.4.1.2.1. Le filtre médian

Le filtre médian est un filtre non linéaire, son principe consiste à remplacer un pixel par la médiane de ses voisins. Ainsi, même si plusieurs pixels voisins sont bruités, le pixel courant peut se corriger. Ce filtre induit cependant un lissage puisque même des pixels corrects peuvent être modifiés. De plus, ce filtrage est plus coûteux car nécessite d'effectuer un tri des voisins pour chaque pixel par ordre croissant. Plus le voisinage considéré est plus, plus l'algorithme sera coûteux [25].

Les étapes de calcul (figure (2.9)) :

- Choisir un voisinage contenant un nombre impair ($2w+1, 2w+1$) de pixels.
- Trier les niveaux par ordre croissant.
- Donner au pixel d'analyse le niveau situé au milieu des niveaux triés.

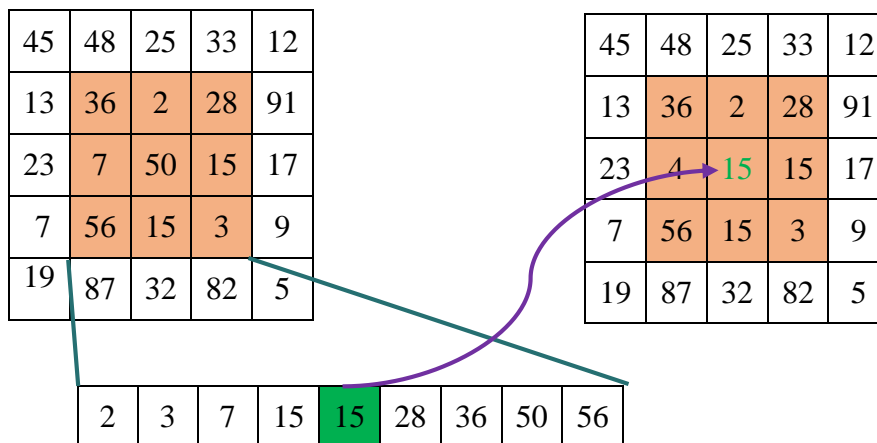


Figure 2.9 : Exemple de calcul pour le masque médian

Ces effets :

- Le filtre médian permet d'obtenir de bons résultats sur un bruit poivre et sel (figure (2.10)). Mais si le bruit P&S est supérieur à la moitié de la taille du filtre, le filtrage est inefficace.

- Préserve mieux les contours sans altérer le fond, aussi il réduit le bruit additif uniforme ou gaussien (lissage de l'image) [28].
- Il supprime les détails.
- Il enlève les points isolés.
- Il est coûteux en temps de calcul.



Image originale



Bruit poivre et sel (20%)



Filtre gaussien 7×7



Filtre médian 7×7

Figure 2.10: Exemple d'application du filtre gaussien et du filtre médian

On observe que le filtre médian est bien plus efficace sur le bruit P&S par rapport au filtre gaussien.

2.4.1.2.2. Le filtre bilatéral

Pour lisser une image et en même temps conserver les bordures, Tomasi et Manduchi [29] ont introduit un filtre qui d'une part renforce la situation à la fois géométrique et photométrique et d'autre part remplace le pixel par une moyenne des pixels voisins similaires.

Le filtre bilatéral est une technique simple de combinaison non linéaire, il est défini comme une moyenne pondérée des pixels proches, d'une manière très semblable à la convolution gaussienne. La différence c'est que le filtre bilatéral prend en compte la différence de valeur avec les voisins pour conserver les contours (les bords) tout en lissant l'image. L'idée clé de ce filtre est que pour qu'un pixel influence sur un autre pixel, il devrait non seulement occuper un lieu voisin mais aussi avoir une valeur similaire [30]. Donc, les coefficients de ce filtre dépendent non seulement de la distance euclidienne au point à lisser, mais aussi des similitudes photométriques (la ressemblance des intensités entre le pixel central et le pixel du masque) [31]. L'expression de ce filtre est définie comme suit :

$$\hat{u}(x) = \frac{1}{C(x)} \sum_{\beta_p} \exp - \frac{|x-y|^2}{2\rho^2} \exp - \frac{|Y(x)-Y(y)|^2}{2h^2} u(y) \quad (2.4)$$

Avec:

$$C(x) = \sum_{\beta_p} \exp - \frac{|x-y|^2}{2\rho^2} \exp - \frac{|Y(x)-Y(y)|^2}{2h^2} \quad (2.5)$$

Où :

$\hat{u}(x)$: la valeur estimée du pixel à la position x .

$C(x)$: Le coefficient de normalisation qui normalise la somme des poids des pixels (égale à 1).

β_p : Représente la fenêtre de convolution.

x : la position 2D d'un pixel dans le plan de l'image.

y : est un ensemble de coordonnées 2D dans le masque.

$Y(x)$: représente l'intensité du pixel à la position x dans l'image bruitée.

$Y(y)$: représente l'intensité du pixel à la position y dans l'image bruitée.

ρ et h sont respectivement la déviation standard (écart type) de la distribution gaussienne des poids géométriques (voisinage) et la déviation standard de la distribution gaussienne des poids d'intensités.

Dans la pratique, le paramètre ρ est fixé en fonction de la taille du masque et le paramètre h permet de régler le paramètre de similarité photométrique du filtre, le réglage de ce paramètre doit prendre en compte le type d'application pour laquelle le filtre est utilisé, dans le cas particulier de la restauration des images bruitées, ce paramètre doit varier en fonction de la quantité de bruit présent dans l'image, sachant que le filtre bilatéral agit essentiellement en tant

qu'un filtre de domaine standard avec des petites différences faible entre les valeurs de pixel [32].

La figure (2.11) illustre le comportement du ce filtre sur une image quelconque et la figure (2.12) montre un exemple d'une image filtrer par ce filtre a des déviations standard différentes.

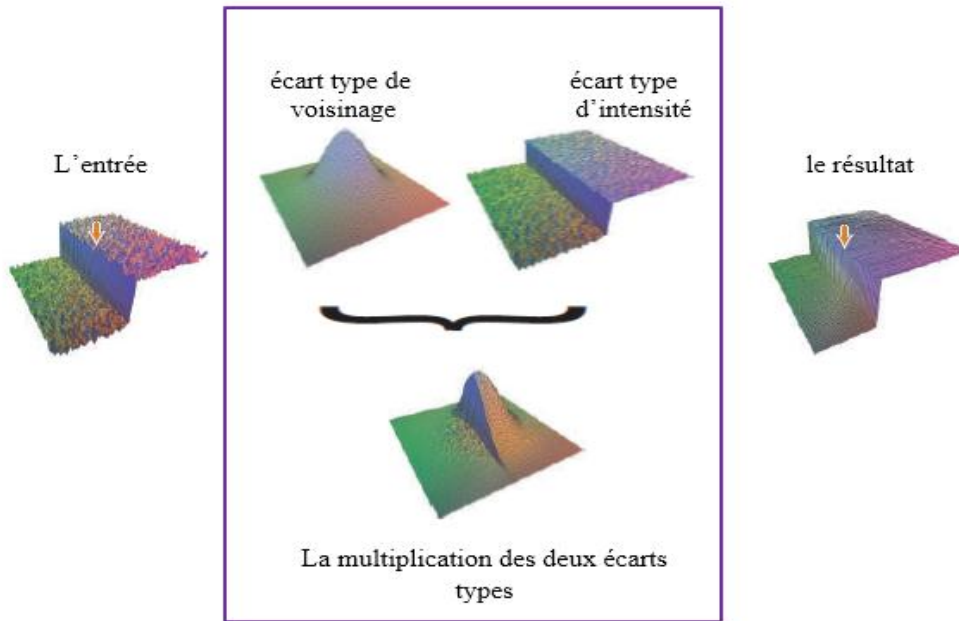


Figure 2.11: Le filtre bilatéral lisse l'image d'entrée tout en préservant ses arêtes

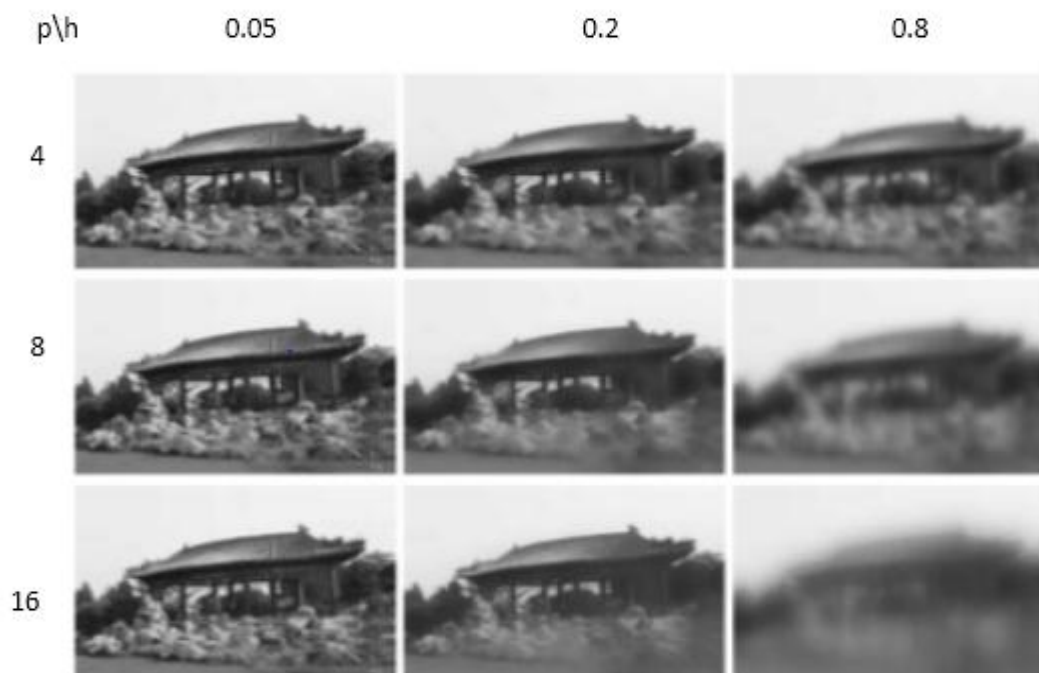


Figure 2.12: Les différentes variances des deux paramètres d'un filtre bilatéral sur une image

Les effets de filtre bilatéral :

- S'adapte en fonction de l'information locale de l'image.
 - Lisse où c'est possible (lissage non uniforme).
 - Préserve les détails quand il le faut.
- Le filtre bilatéral est un filtre qui conserve mieux les arêtes que le filtre gaussien.

Remarques :

- Le filtre bilatéral peut diviser une image en deux parties: l'image filtrée et l'image résiduelle. L'image filtrée elle contient la partie traitée par le filtre bilatéral (lissé les variations locales sans qu'il affecte ses discontinuités) et l'image résiduelle réalisée en soustrayant l'image filtrée de l'original, ne contient que les parties d'image que le filtre a été retiré. Cela est basé selon les paramètres et l'application [30].
- Il peut être calculé à une vitesse interactive, même sur de grandes images, grâce à des systèmes numériques efficaces et en même temps réel si le matériel graphique est disponible.
- Le filtrage bilatéral est une solution relativement simple et non-itérative pour préserver les discontinuités tout en filtrant les zones semblables.

2.5. Conclusion

Dans ce chapitre, nous avons essayé de présenter quelques techniques de filtrage sur les images numérique tel que le filtrage linéaire (moyenneur et gaussien) et le filtrage non linéaire (médian et bilatéral). Ces filtres permettent de réduire le bruit dans l'image, cependant chacun d'eux a ces avantages et ces inconvénients, le choix du type de filtre dépend alors de l'application.

Chapitre 3

*Débruitage des
images RGB-D*

Kinect

3.1. Introduction

Les données de la Kinect sont principalement utilisées pour la reconstruction 3D, comme on trouve des algorithmes de segmentation, reconnaissance d'objet, suivi d'objet, etc. Or, le bruit présent dans ces images peut perturber et détériorer les performances de ces algorithmes. Beaucoup de méthodes pour supprimer ou réduire l'influence du bruit ont été présentées dans la littérature. Ce qui fait de la qualité des images de profondeur l'un des problèmes les plus importants du capteur Kinect, en effet, les données fournies présentent différents problèmes liés au bruit qui doivent être résolus afin d'améliorer la précision des mesures et permettre ainsi d'élargir les applications de la Kinect en vision par ordinateur.

Dans un premier temps nous avons adopté une approche simple qui consiste à appliquer des méthodes de filtrage sur les images de profondeur, par la suite nous avons appliqué une stratégie de remplissage de trous qui vise à améliorer la qualité des images de profondeur obtenues avec le capteur Kinect.

3.2. Définition de l'image de profondeur

Une image de profondeur est un tableau bidimensionnel de position 3D, vérifiant la propriété de cohérence spatiale, chaque composante de cette matrice représente la distance entre le capteur et l'objet de la scène, elle est l'équivalent d'une image vidéo dans laquelle le niveau de gris de chaque pixel (x,y) serait remplacé par une altitude z .

Dans l'image de profondeur Kinect, les pixels sont codés sur 11 bits permettant ainsi 2048 niveaux de luminosité pour la captation de profondeur. Ces images en profondeurs délivrées par la Kinect enregistrent la distance entre les différentes parties de la scène. L'image de profondeur est interprétée comme une image en niveaux de gris distordue. La couleur des différentes parties de l'image n'indique pas à quel point les objets sont lumineux, mais à quelle distance ils sont. Les parties les plus claires indiquent les objets proches, les plus sombres les plus lointains. Ce type d'image facilite l'interprétation d'une scène par un ordinateur [33].

3.3. Les bruits dans les images profondeur Kinect

Le capteur de profondeur Kinect utilise le principe de la lumière structurée, où une source de lumière infrarouge projette des motifs lumineux: la lumière réfléchiée est reçue par une caméra infrarouge et traitée pour extraire des informations géométriques sur les surfaces d'objet. Les mesures des capteurs de lumière structurée sont affectées par le bruit dû aux réflexions multiples, aux objets transparents ou à la diffusion dans des surfaces particulières (comme les tissus humains et les cheveux, surface lisse, ...etc.). L'un des effets de bruit les plus importants sur la précision des images de profondeur Kinect est dû à la présence de régions pour lesquelles la caméra n'est pas en mesure d'estimer correctement la profondeur. Ces régions proviennent principalement de la présence d'objets occultés et d'objets concaves, mais elles se retrouvent également dans des régions homogènes.

Ce type de bruit se manifeste par la présence des trous dans l'image de profondeur, ces trous apparaissent généralement sur les frontières des objets (bordures), dans les surfaces unies et brillantes, et dans d'autres endroits épars cela appelé bruit spatial. De plus, la profondeur d'un pixel particulier change souvent d'une trame à une autre, même lorsque la scène est statique (bruit temporel) et un bruit d'interférence (un bruit spatial et un bruit temporel) qui est causé par l'utilisation de deux ou plusieurs Kinect qui introduisent un chevauchement des motifs infrarouge [34].

Ainsi la réduction de ces bruits est une tâche très importante pour améliorer la qualité de l'image de profondeur dans le but de l'utiliser par la suite dans d'autres applications en vision par ordinateur.

3.4. Acquisition des images RGB-D Kinect

Pour l'acquisition de nos propre images couleur et profondeur, nous avons installé la Kinect sur ordinateur. Pour l'installation de ce capteur sur PC Windows, il faut d'abord installé les pilotes de la Kinect qui sont fournis gratuitement sur l'internet [9,10], puis la brancher par son port USB classique, une fois la Kinect est branchée, une LED s'allume soit en vert dans la cas ou les pilotes sont chargés correctement, ou en rouge (si les pilotes ne sont pas chargés correctement).

Un exemple des images couleur et profondeur obtenue avec Kinect est illustré à la figure (3.1). Comme on peut le constater, l'erreur la plus importante dans l'image de profondeur Kinect est l'absence de mesures de profondeur dans certaines zones d'image. Ces pixels de profondeur non mesurés correspondent à des concavités dans les éléments de la scène (entre l'onduleur et l'unité centrale), ils sont également présentés au hasard dans des régions homogènes des images et sur les bordures (écran du PC), ainsi que les objets transparents (les vitres). Comme on peut représenter l'image de profondeur avec des fausses couleurs (un dégradé de bleu) (figure (3.1 (c))), pour interpréter les distances dans lequel cette image a été captée, par exemple dans cette image le bleu foncé représente l'absence de mesure de profondeur (les trous) et du bleu clair jusqu'au jaune qui correspond aux objets les plus éloignés.

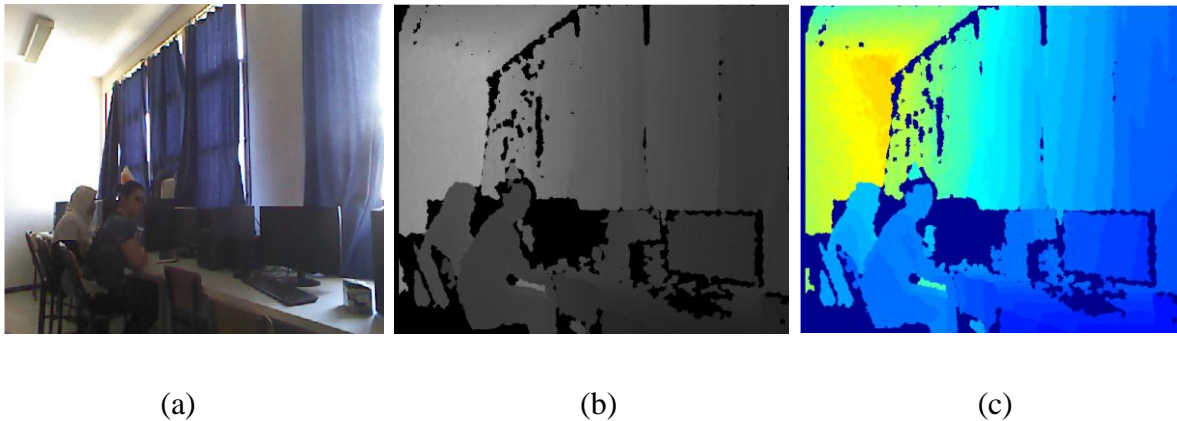


Figure 3.1 : (a) Image RGB, (b) Image de profondeur, (c) Image de profondeur avec les fausses couleurs

3.5. Filtrages des images de profondeur

Dans le 2^{ème} chapitre nous avons étudié quelques techniques de filtrage linéaire (moyenneur, gaussienne) et non linéaire (médian, bilatéral) sur des images quelconques. Dans ce chapitre nous allons tester ces différents filtres sur des images de profondeur Kinect.

Les images test que nous avons utilisées sont extraites de deux bases de données (RGB-D Scenes Dataset v.2 [35] et RGB-D Kinect [36]).

3.5.1. Base de données RGB-D Kinect

Il existe plusieurs bases de données RGB-D qui ont été créées pour évaluer les différents algorithmes de vision par ordinateur, depuis que le capteur Kinect a été lancé. Ces bases de

données sont utilisées dans différents domaines tels que la détection et le suivi des objets, la segmentation, la reconnaissance d'objets et de scène, l'analyse gestuelle. Cependant, chaque base de données peut ne pas être limitée à une seule application spécifique.

Pour l'ensemble des tests que nous avons effectués, nous avons choisi deux bases de données, la première est composée de 14 scènes contenant des meubles (chaise, table basse, canapé, table) et la deuxième est un grand ensemble de données de 300 objets ménagers communs, ces objets sont organisés en 51 catégories. Cette base de données comprend également 22 séquences vidéo annotées de scènes naturelles contenant des scènes couvrant les environnements intérieurs communs, qui comprennent les espaces de travail du bureau, les salles de réunion et les zones de cuisine. Les objets sont visibles à partir de différents points de vue et distances et peuvent être partiellement ou totalement occlus dans certaines trames.

3.5.2. Tests et Résultats

3.5.2.1. Par un filtrage linéaire

- Les images à tester pour un filtrage Moyenneur et pour un filtrage Gaussien:

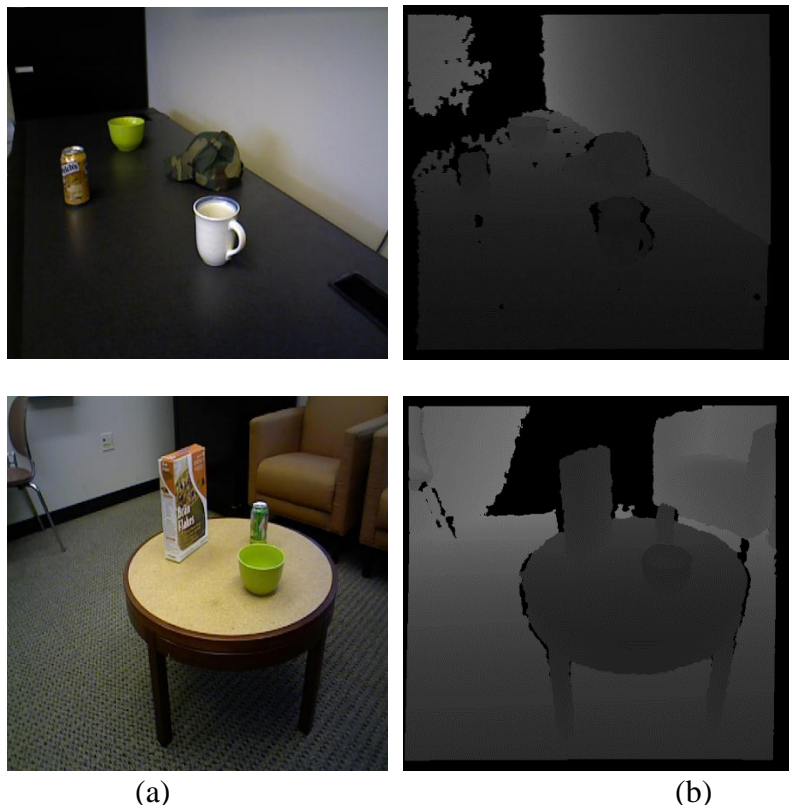


Figure 3.2: (a) Image RVB (couleur), (b) Image de profondeur

- Le 1^{er} teste avec le filtre moyenneur :

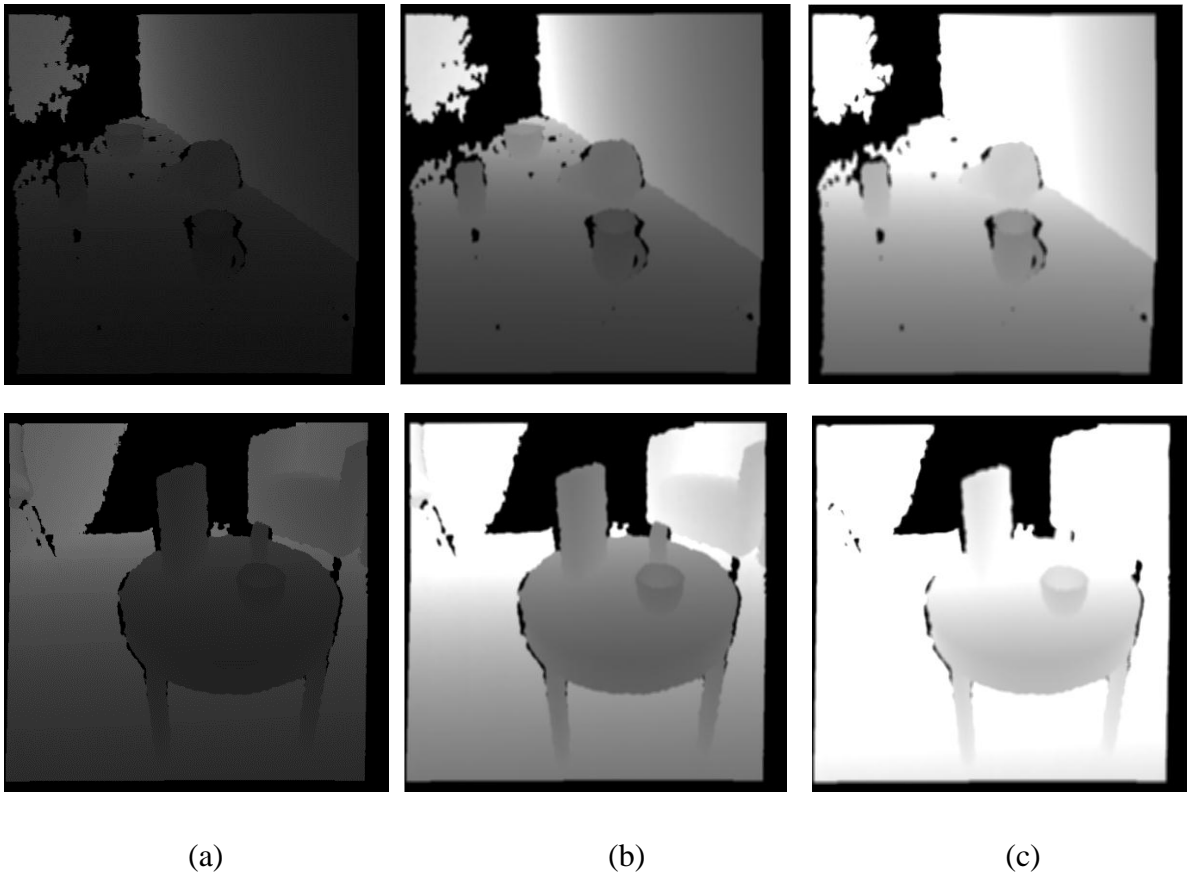


Figure 3.3 : (a) Image de profondeur, (b) Le 1^{er} Filtrage avec un masque de taille 5×5 , (c) Le 2^{ème} Filtrage avec un masque de taille 7×7

Interprétations

1^{er} Filtrage : pour le filtre moyenneur de taille (5×5) testé sur les images de profondeur (figures (3.3(b)) et (3.3(c))), les résultats obtenus sont les suivants:

- Les résultats obtenus sont insatisfaisants car la plupart des pixels noirs ne sont pas filtrés.
- Les contours sont pour l'instant préservés.

2^{ème} Filtrage : on a utilisé les même images du 1^{er} teste mais avec un masque de taille (7×7) on a obtenus les résultats suivants :

- perte du détail (le bol vert, la canette ainsi que le fauteuil n'apparaissent pas dans les images résultantes), et les contours ne sont pas préservés.
- Comme on remarque le remplissage de quelque trous (pixels noirs), mais le résultat reste toujours insuffisant.
- Le 2^{ème} teste avec le filtre gaussien :

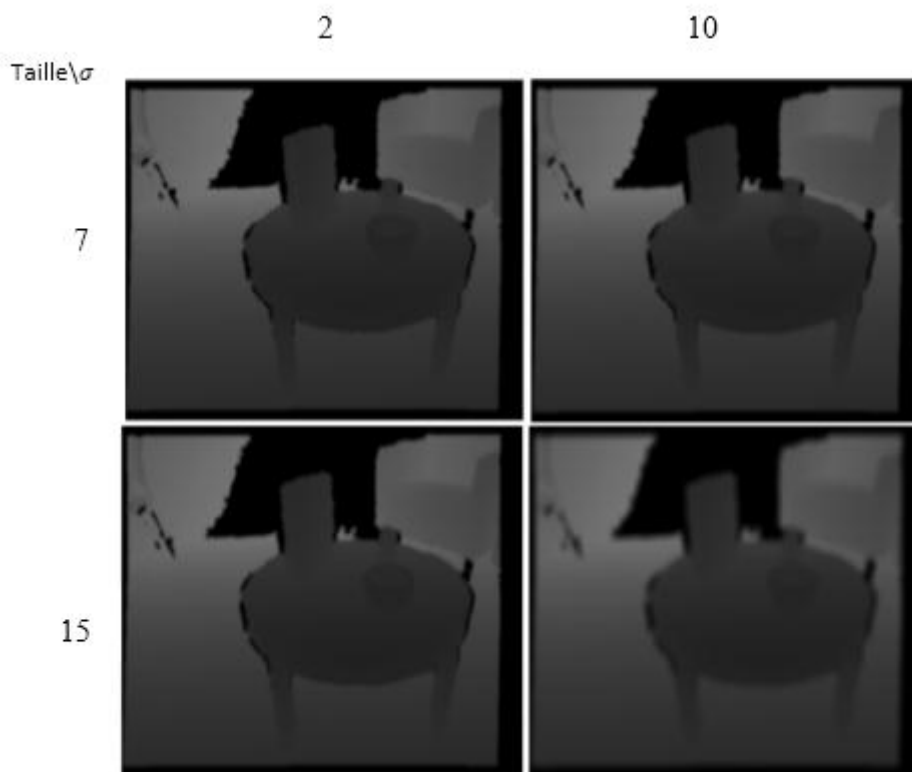


Figure 3.4 : L'image de profondeur résultante avec les différentes valeurs d'écart type

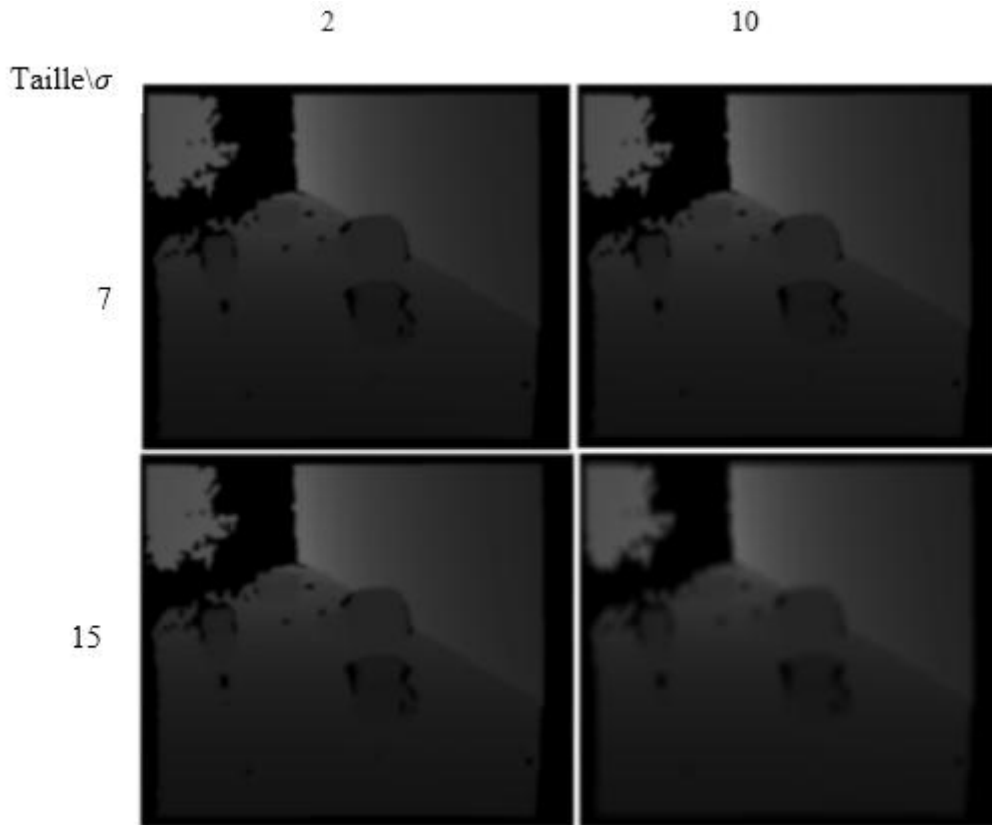


Figure 3.5 : L'image de profondeur résultante avec les différentes valeurs d'écart type

Interprétations

Dans les deux testés nous avons utilisés un masque de taille (7×7) et un masque de taille (15×15) et nous avons variés l'écart type ($\sigma = 2$ et $\sigma = 10$).

Après plusieurs tests sur les images de profondeur, on fait varier la taille du filtre gaussien et son écart type, nous avons constatés qu'à chaque fois qu'on augmente la taille, le lissage sur cette image est visible, et à chaque fois qu'on augmente sigma un flou sur la même image est remarquable. Donc :

- Plus sigma est grand, plus on réduit le bruit, mais plus l'image filtrée est floue.
- Si sigma est choisi trop grand, tous les détails de l'image seront perdus.

3.5.2.2. Par un filtrage non linéaire

- Les images à testés pour un filtrage Médian et pour un filtrage Bilatéral:

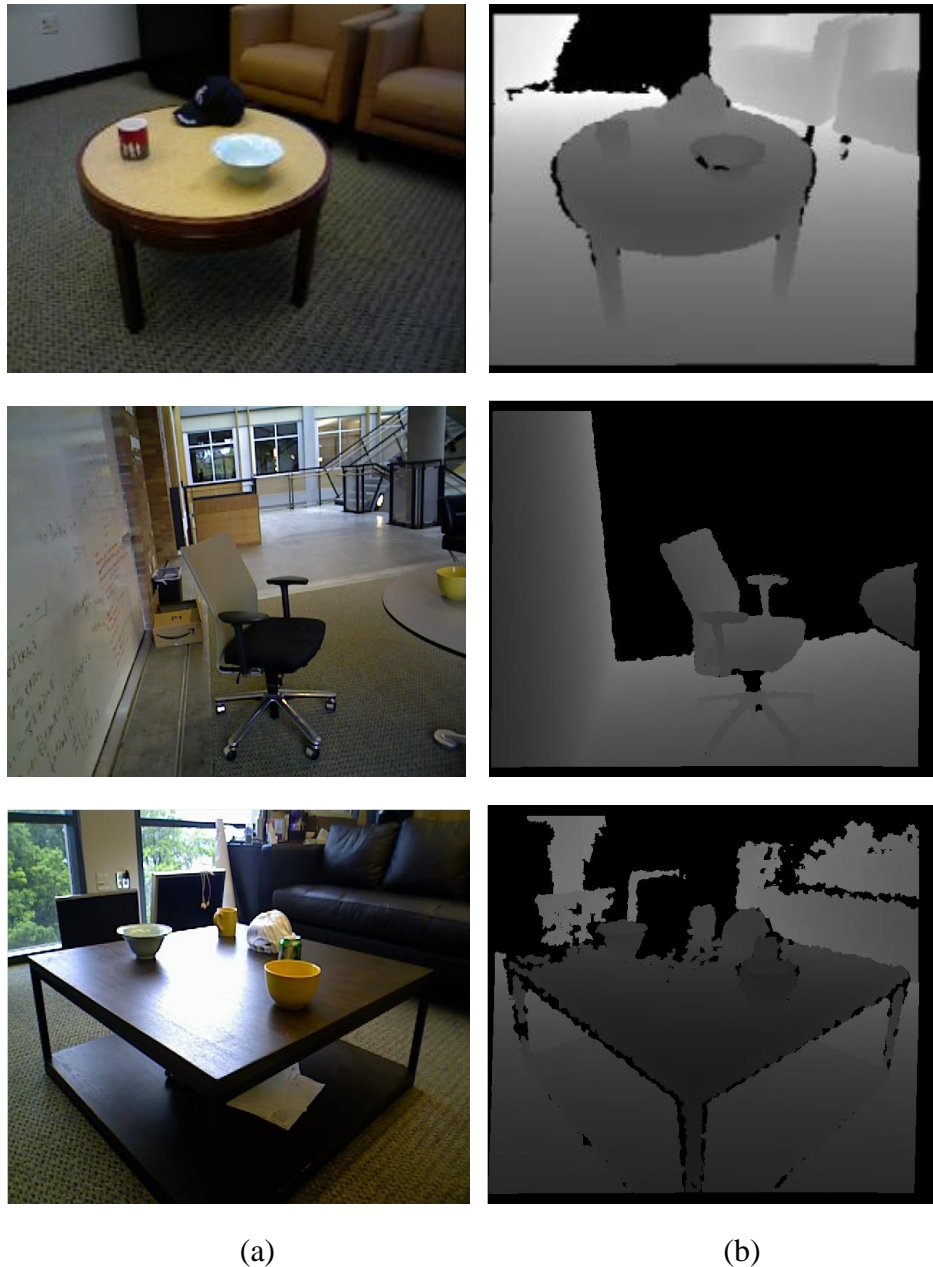


Figure 3.6: (a) Image RVB (couleur), (b) Image de profondeur

- Le 1^{er} teste avec le filtre médian :

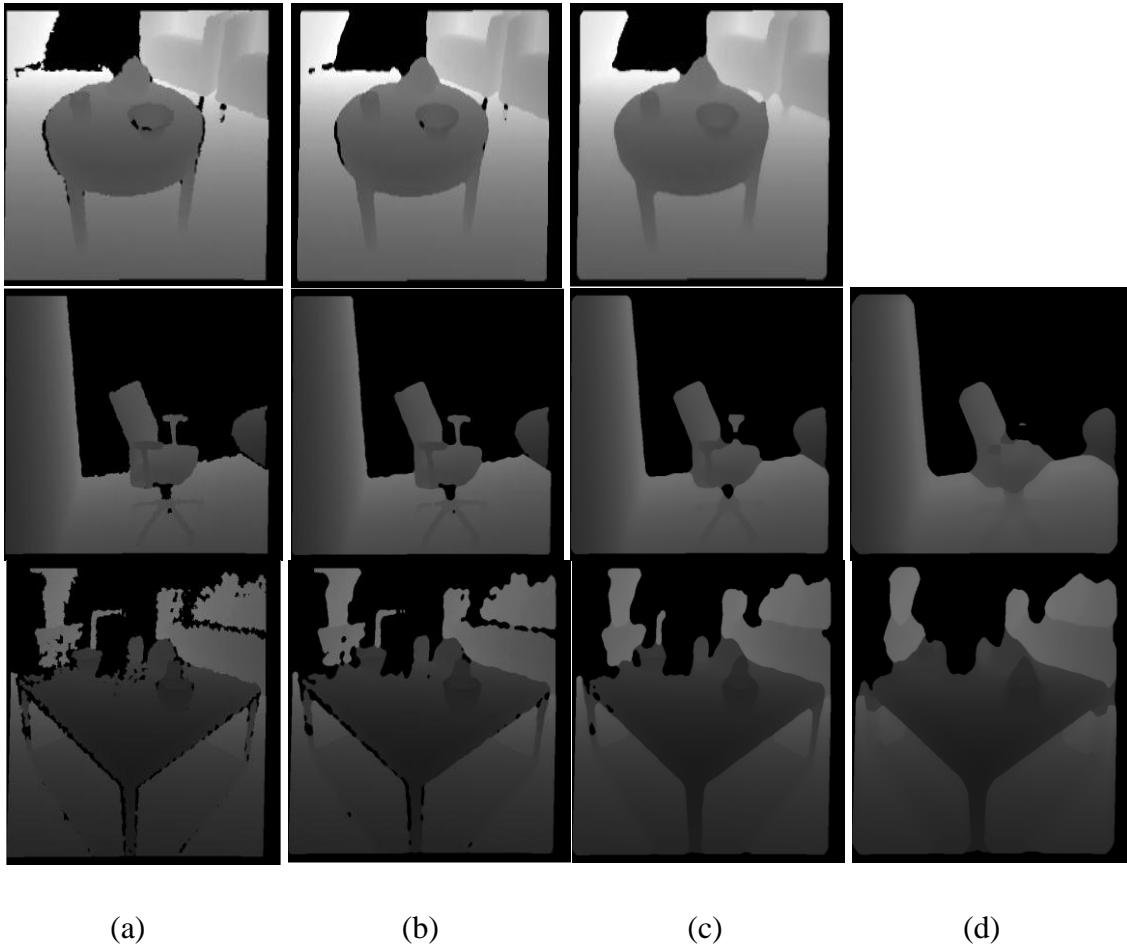


Figure 3.7: (a) Image de profondeur, (b) Le 1^{er} filtrage avec un masque de taille 11×11 , (c) Le 2^{ème} filtrage avec un masque de taille 21×21 , (d) Le 3^{ème} filtrage avec un masque de taille 39×39

Interprétations

Sur la figure (3.7), on voit bien que dans le 1^{er} filtrage le bruit est réduit (sur tous les pixels isolés), et pour le 2^{ème} filtrage les pixels noir isolé sont éliminés, même chose avec les deux autre images (voir la chaise) et (voir la table carrée) donc on constate que ce filtre élimine d'une façon significative les trous, mais avec l'augmentation de la taille de filtre (39×39) une perte de détails sera bien marqué (voir les pieds de la chaise et de la table carrée), donc ce filtre devient presque inefficace sur ces images puisque même l'information importante dans l'image sera supprimée pour les tailles importantes.

- Le 2^{ème} teste avec le filtre bilatéral :

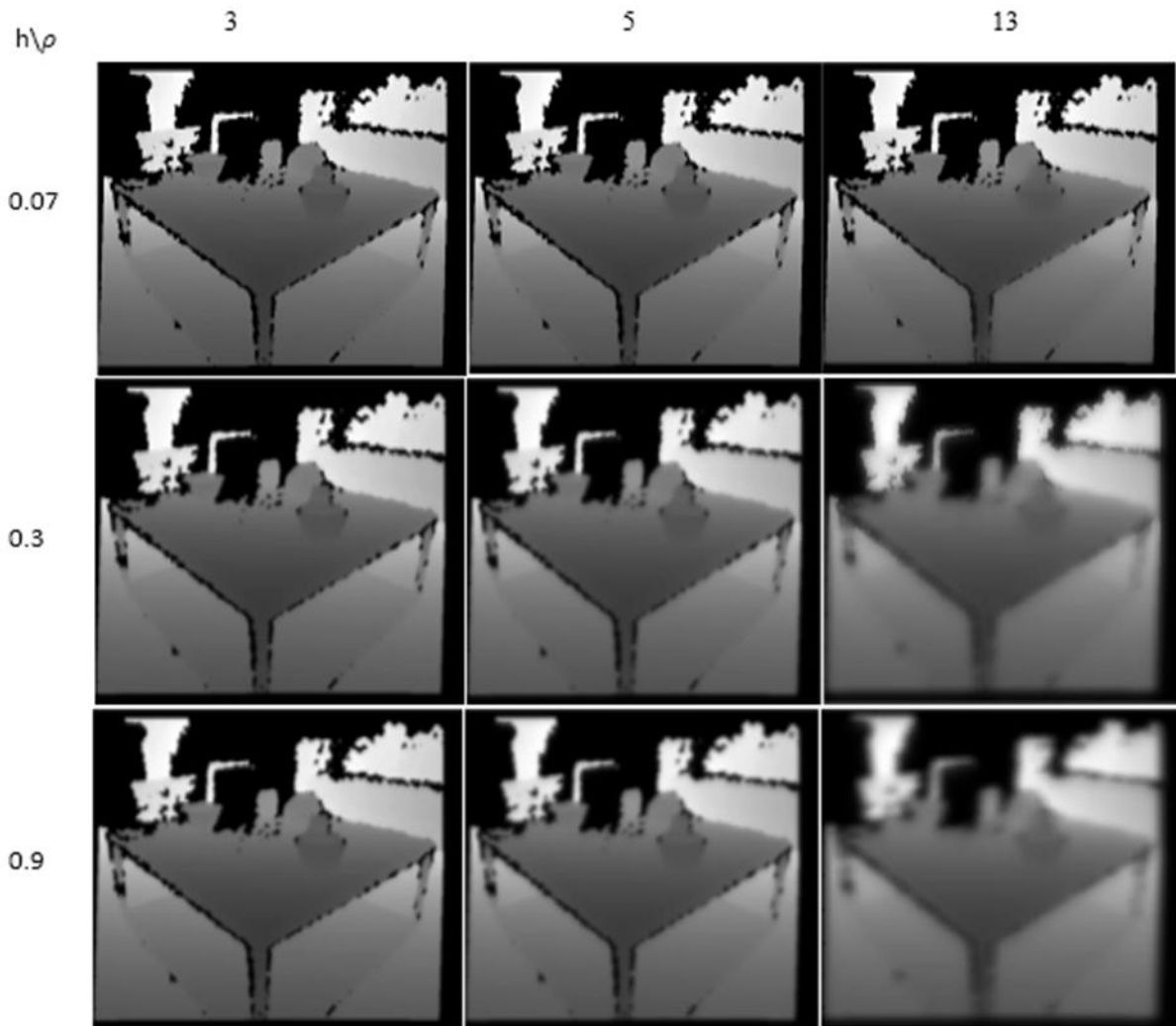


Figure 3.8: L'image de profondeur résultante avec différentes valeurs d'écart type de voisinage et d'intensité

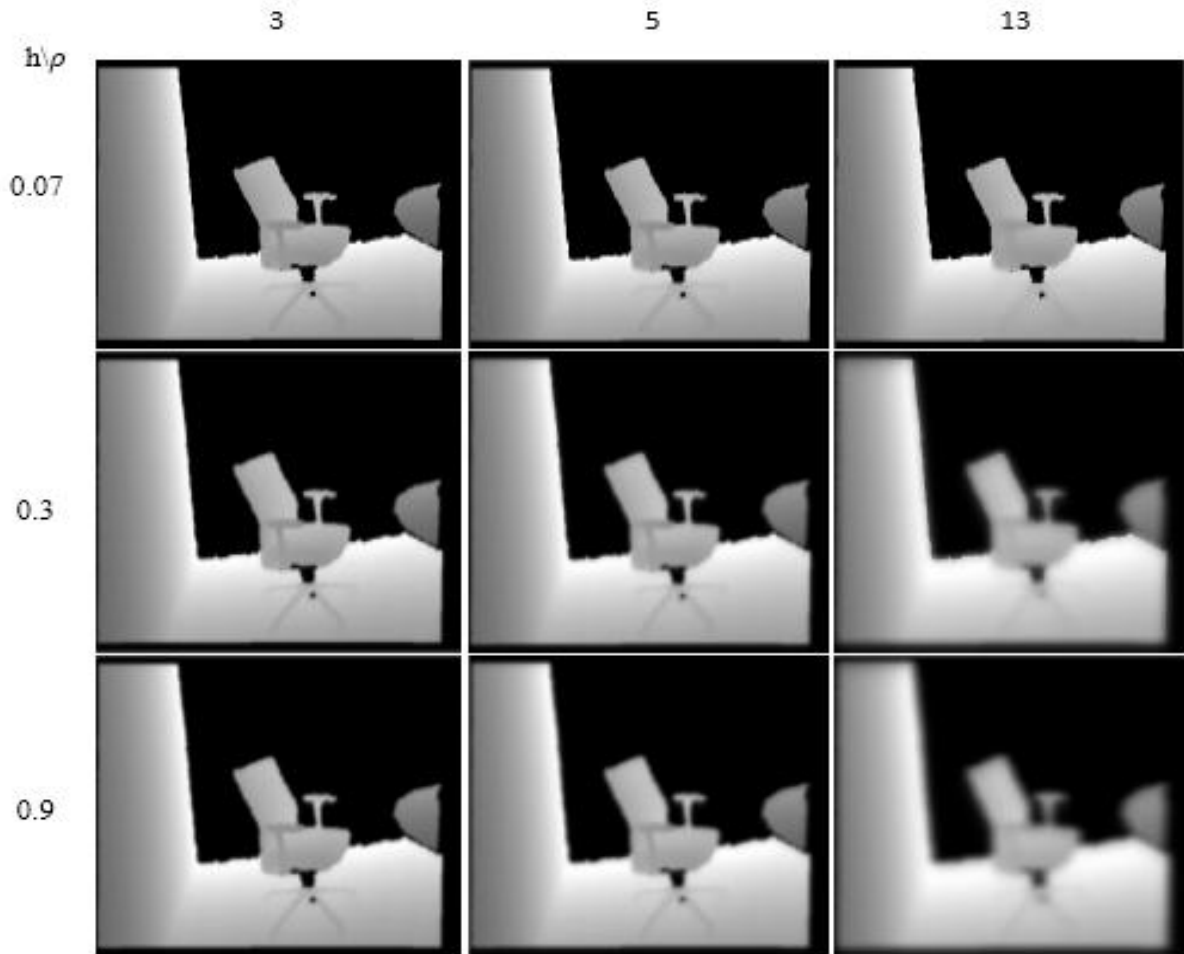


Figure 3.9: L'image de profondeur résultante avec différentes valeurs d'écart type de voisinage et d'intensité

Interprétations

Les figures ((3.8) et (3.9)) montrent l'effet de différentes valeurs des paramètres ρ et h sur l'image résultante, d'où les colonnes correspondent aux différentes valeurs de l'écart type de voisinage ' ρ ' et les lignes correspondent aux différentes valeurs de l'écart type d'intensité ' h '.

- Lorsque la valeur de ρ est grande (13) par rapport aux petites valeurs de h (0.3, 0.9), le paramètre h du filtre bilatéral a peu d'effet sur l'image résultante, donc la déviation standard de voisinage agit comme un filtre gaussien, c'est-à-dire que l'image de profondeur devienne floue (un fort lissage), cela signifie d'une certaine manière la suppression ou la réduction du bruit (les trous) dans les régions isolées), puisque toutes les valeurs des pixels de n'importe quel voisinage ont environ le même poids

pour le filtrage d'intensité, cet effet peut être vu dans la dernière colonne les deux derniers lignes des deux figures.

- Pour la plus petite valeur de l'écart type de la distribution gaussienne des poids d'intensités σ (0.07), h domine perceptuellement parce qu'il préserve les contours (voir la table dans la figure (3.8) et la chaise dans la figure (3.9)).

3.6. Stratégie de remplissage des trous des images profondeur

L'image de profondeur de la caméra RGB-D Kinect contient des régions avec des informations manquantes (des valeurs nulles). Dans les paragraphes précédents, nous avons décrit les problèmes les plus importants qui affectent la précision de l'image de profondeur en particulier la présence de trous, l'objectif de notre approche est le remplissage des trous. Cette approche consiste à prendre un voisinage autour du pixel de profondeur 0, ce pixel est le candidat au filtrage, nous avons pris un voisinage 3×3 , 5×5 et 11×11 . Pour les tests nous avons effectués, une fenêtre de voisinage 5×5 a été retenue.

Cette dernière est une approche de filtrage par pixel qui consiste à remplacer le pixel centrale qui est un pixel zéro par la valeur la plus répétée dans ces 25 pixels voisins. Autrement dit, lorsque on trouve une valeur zéro dans une matrice, il est considéré comme un candidat pour le filtrage, en tenant compte de ces pixels voisins. Par la suite on cherche toutes les valeurs non nulles des pixels voisins, la valeur la plus répéter parmi les pixels voisins sera affectée au pixel noir [37].

Dans la figure (3.10), la valeur la plus répéter est 2000, donc elle sera affecté au pixel candidat.

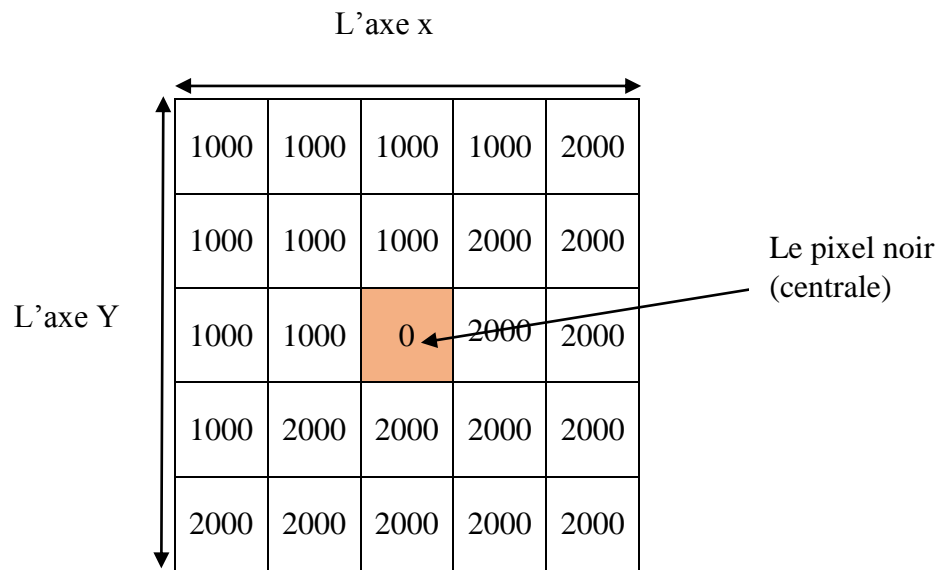


Figure 3.10: Remplissage des trous

Après l'application de cette approche sur nos images test, on voit bien que la méthode de filtrage des pixels fonctionne parfaitement (voir les figures (3.11), (3.12), (3.13), (3.14), (3.15)), elle donne de très bon résultats puisque elle enlève carrément le bruit (les trous de profondeur qu'on a pas pu enlever avec les approches de filtrage classique). Mais selon la figure (3.11) on remarque bien une perte de détail (le bras de la chaise) qui sont des informations importante dans cette image.

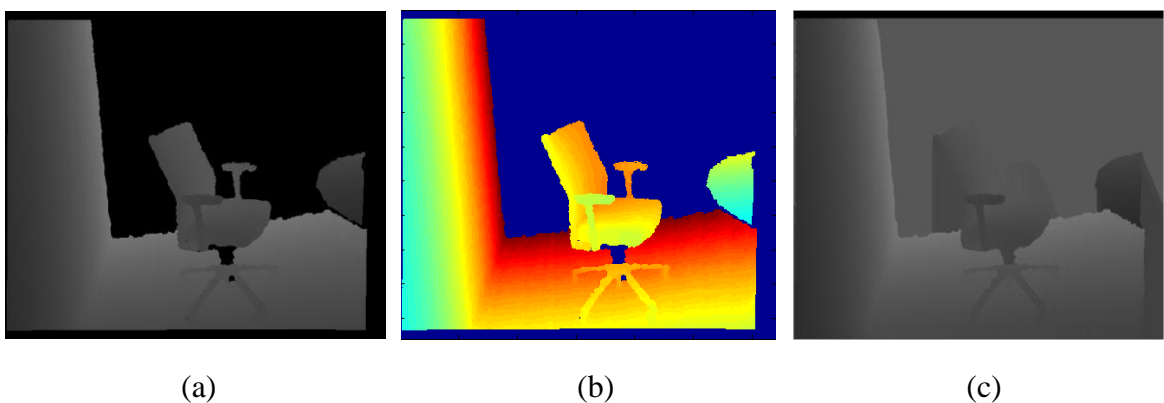


Figure 3.11: (a) Image de profondeur, (b) Image de profondeur en fausse couleurs, (c) Image filtrée

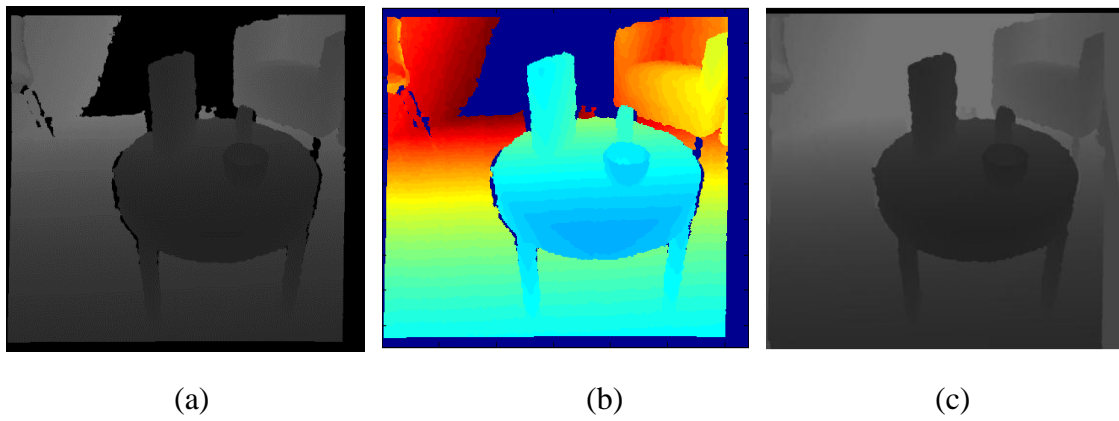


Figure 3.12: (a) Image de profondeur, (b) Image de profondeur en fausse couleurs, (c) Image filtrée

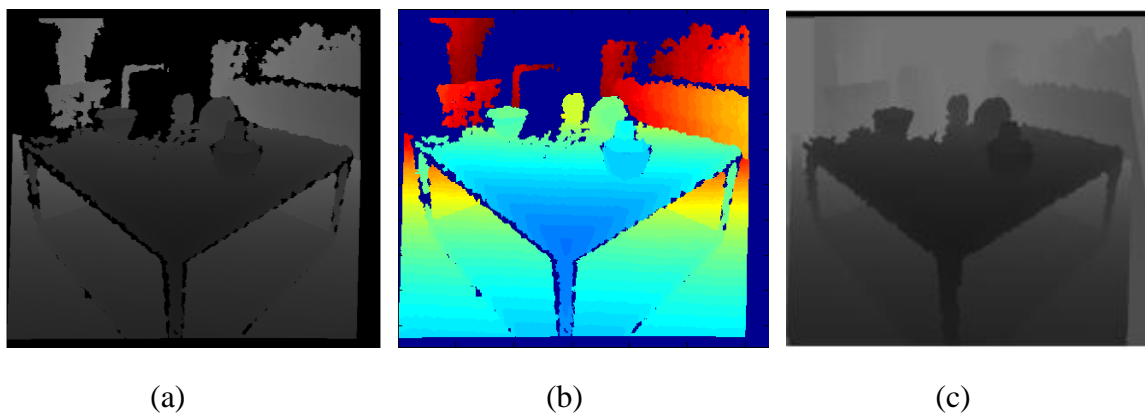


Figure 3.13: (a) Image de profondeur, (b) Image de profondeur en fausse couleurs, (c) Image filtrée

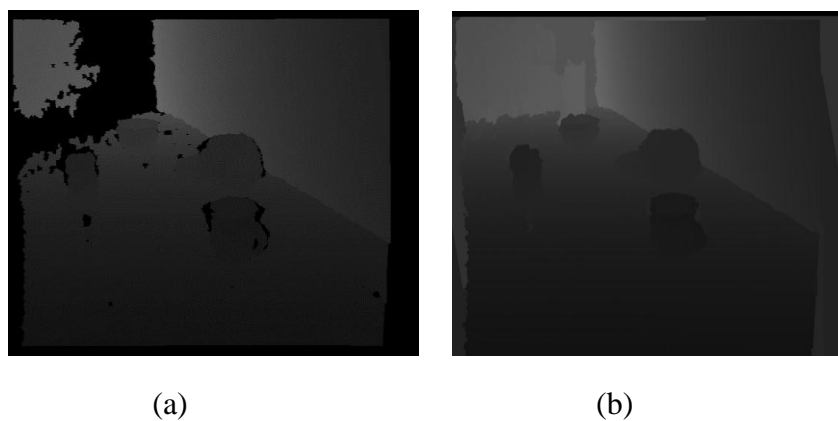


Figure 3.14: (a) Image de profondeur, (b) Image filtrée

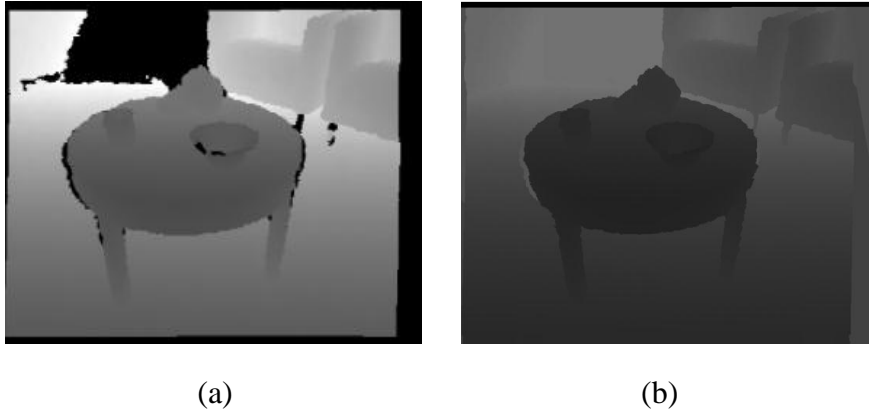


Figure 3.15: (a) Image de profondeur, (b) Image filtrée

3.7. Conclusion

Dans ce chapitre nous avons choisi d'appliquer des approches simples de filtrage classique linéaires et non linéaires.

Nous avons constaté qu'avec ce type de filtrage sur des images profondeur, on arrive à supprimer les bruits présents dans l'image mais le bruit qui affecte le plus ce type d'image est la présence de trous qui représente l'information perdue, n'a pas pu être complètement supprimé avec ce type de filtrage. Tandis que le filtrage par pixel appliqué sur ces images a permis de supprimer tous les pixels noirs et le remplissage des trous.

Conclusion générale

Conclusion générale

Dans ce travail, nous avons étudié le capteur RGB-D Kinect v1 de Microsoft, dans un premier temps nous avons présenté la caméra Kinect, ses caractéristiques et son principe de fonctionnement. Dans le but d'exploiter les images de profondeur et couleur fournies par ce capteur, notre intérêt est porté sur le problème de bruits qui altère ces images, pour le débruitage de ce type de données nous avons appliqué quelques techniques de filtrage 2D classiques, soit un filtrage linéaire par un filtre moyenneur et un filtre gaussien et un filtrage non linéaire par un filtre médian et un filtre bilatéral, ce dernier manifeste un bon comportement à la fois aux frontières et à la bordure. Par la suite, nous avons montré avec les multiples tests réalisés l'effet de chaque filtre sur ce genre d'images qui ont été insuffisants, en revanche une méthode simple qui consiste à filtrer pixel par pixel donne de très bons résultats a été effectuée avec succès.

Puis nous avons constaté que ce bruit est dû essentiellement à la portée limitée de la Kinect et d'autres facteurs extérieurs, qui impose un énorme impact sur des recherches récentes en vision par ordinateur.

Enfin, dans notre travail nous avons suivi une démarche simple de filtrage pour éliminer les points indésirables afin de limiter l'influence provenant du bruit, on s'est intéressé particulièrement à l'image de profondeur, cependant, l'utilisation des images couleur en même temps peut nous fournir de meilleurs résultats.

A l'issue de ce travail plusieurs perspectives s'ouvrent, il serait alors intéressant d'utiliser des techniques de filtrage en tenant compte des images couleur et profondeur.

Pour le filtrage bilatéral qui est très utilisé en stéréovision, nous avons constaté qu'il existe plusieurs variantes de ce filtre dans la littérature, il sera alors intéressant d'étudier d'autres types de filtres en tenant compte de l'information complémentaire des images couleur (RGB) et profondeur (Depth) fournies par Kinect.

BIBLIOGRAPHIE

- [1] Djamel Eddine Goumidi, Mémoire de Stage de Fin d'Etude, « Dynamique de la gestuelle ». Université de Caen Basse-Normandie, 2013.
- [2] Bertrand PECUCHET. Kinect for Xbox360. Ecole d'ingénieurs télécom Lille 1. Université Lille 1. 2012/2013.
- [3] Han, J.; Shao, L.; Xu, D.; Shotton, J. « Enhanced Computer Vision with Microsoft Kinect Sensor: A Review ». IEEE trans. Cybern. 2013, 43, 1318–1334
- [4] Romain Paccolat. « Ki-Mouse ». Haute Ecole Spécialisée de Suisse occidentale, 26 juillet 2013.
- [5] Antoine Manzanera. ENSTA-ParisTech/U2IS. Cours M2 IMA-UE VISION, UPMC-Paris6.
- [6] Philippe FABRE. 19 Septembre 2013. Mémoire de Soutenance du Diplôme d'Ingénieur INSA, Spécialité TOPOGRAPHIE. « Exploitation de la caméra 3D (Kinect) pour la réalisation de nuages de points en temps réel dans le cadre de relevés d'intérieur de grande ampleur »
- [7] Sébastien Poulmane, Guillaume Dauster, Marc Amberg. 2012/1013. Projet scientifique de Master 2 Image, Vision, Interaction. « Reconstruction de scène réaliste via Kinect (Amélioration du rendu des textures plaquées dans une scène observée avec une Kinect et reconstruite avec PCL) »
- [8] Jean-Christophe Lapayre, « Projet kinect détection de mouvements intempestifs dans un bloc opératoire ». UNIVERSITE DE FRANCHE-COMTE, 2012/2013.
- [9] A. LEJEUNE, S. PIÉRARD, M. VAN DROOGENBROECK et J. VERLY. Utilisation de la kinect. Linux magazine France, n°151, pages 16-29, juillet-août, 2012.
- [10] <https://www.microsoft.com/fr-fr/download/details.aspx?id=19988>.
- [11] Jean-clément DEVAUX, thèse de doctorat, « Perception multicapteur : Etalonnage extrinsèque de caméra 3D, télémètre laser et caméra conventionnelle. Application au déplacement autonome et téléopéré d'un robot mobile ». UNIVERSITÉ D'ÉVRY VAL-D'ESSONNE, 24 octobre 2014.

- [12] Khoshelham, K. Accuracy Analysis of Kinect Depth Data. In Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science, Calgary, Alberta, CA, 29–31 August 2011; pp. 133–138.
- [13] Herrera C., J. Kannala, and J. Heikkila. Joint depth and color camera calibration with distortion correction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*,34(10), 2012.
- [14] Hu G, Huang S, Zhao L, Alempijevic A, Dissanayake G (2012) A robust rgb-d slam algorithm. In: International Conference on Intelligent Robots and Systems, pp 1714–1719.
- [15] Lee TK, Lim S, Lee S, An S, Oh SY (2012) Indoor mapping using planes extracted from noisy rgb-d sensors. In: International Conference on Intelligent Robots and Systems, pp 1727–1733.
- [16] Oikonomidis I, Kyriazis N, Argyros AA (2011) Efficient model-based 3d tracking of hand articulations using kinect. In: British Machine Vision Conference, pp 1–11.
- [17] Bo L, Ren X, Fox D (2013) Unsupervised feature learning for rgb-d based object recognition. In: Experimental Robotics, pp 387–402.
- [18] Chen L, Wei H, Ferryman J (2013) A survey of human motion analysis using depth imagery. *Pattern Recogn Lett* 34(15):1995–2006.
- [19] Jan Smisek, Michal Jancosek and Tomas Pajdla “3D with Kinect”.CMP, Dept. of Cybernetics, FEE, Czech Technical University in Prague.
- [20] Ryane Alla. Détection de chute à l'aide d'une caméra de profondeur. Département d'informatique et de recherche opérationnelle, Faculté des arts et des sciences, Université de Montréal. Avril 2013.
- [21] <http://makingthingssee.com/>
- [22] <https://labo-microsoft.supinfo.com>
- [23] E.N.S.B.A.N.A. 2ème année - TD d'informatique scientifique - 2002/2003. Initiation au traitement d'images avec MATLAB.
- [24] A. Manzanera ENSTA/U2IS. Chapitre 1 LES IMAGES NUMERIQUES.
- [25] Lingrand Diane. Rapport de recherche ISRN I3S/RR-2004-05-FR, Cours de Traitement D'Images. Université Nice Sophia Antipolis. 22 Janvier 2004.

- [26] Mohamed Sahbi Bahroun. Chapitre 3 Filtrage d'image. Université Tunis Elmanar Institut Supérieur d'Informatique. 2011/2012.
- [27] A. Manzanera ENSTA/U2IS. Chapitre 2 Filtrage et restauration.
- [28] Nicolas Thome. Bases du traitement des images, Filtrage d'images.11 octobre 2016.
- [29] C.Tomasi, R. Manduchi. Bilateral Filtering for Gray and Color Images. Proceedings of the 1998 IEEE International Conference on Computer Vision, Bombay, India.
- [30] S. Paris, P. Kornprobst, J. Tumblin and F. Durand. Bilateral Filtering: Theory and Applications. Vol. 4, No. 1 (2008) 1–73.
- [31] TP de traitement d'images. "TP2: Filtrage non linéaire" Grenoble INP ENSIMAG. 2015/2016[13] Patrick Hébert .Traitement de base d'une image (partie 1). Dernière révision octobre 2008.
- [32] Harold Phelippeau. Méthodes et algorithmes de dématricage et de filtrage du bruit pour la photographie numérique.
- [33] <http://lessons.julien-drochon.net/kinect-processing-depthimage/>
- [34] Tanwi Mallick, Partha Pratim Das, and Arun Kumar Majumdar."Characterizations of Noise in Kinect Depth Images: A Review ».IEEE SENSORS JOURNAL, VOL. 14, NO. 6, JUNE 2014.
- [35] <http://rgbd-dataset.cs.washington.edu/dataset/rgbd-scenes-v2/>
- [36] <http://rgbd-dataset.cs.washington.edu/>
- [37] <http://www.codeproject.com/Articles/317974/KinectDepthSmoothing>

Résumé :

Le capteur Kinect est une caméra RGB-D sorti en Novembre 2010 pour la console de jeux vidéo Xbox 360 de Microsoft, qui permet à la fois la capture des images couleurs (RGB) et des images de profondeur (Depth), cette dernière souffre souvent de bruits ; de ce fait, ces images sont subit aux quelques techniques de filtrages 2D classiques afin d'avoir de meilleur résultats pour offrir de nouvelles solutions potentielles aux problèmes de vision par ordinateur.

Mots clés : La Kinect, Le capteur RGB-D, Le filtrage linéaire, Le filtrage non linéaire, Les techniques de filtrage.