



**HAL**  
open science

## Réalité augmentée à partir d'une séquence vidéo en utilisant la stéréoscopie dense.

Nadia Zenati, Karim Achour, Noureddine Zerhouni

► **To cite this version:**

Nadia Zenati, Karim Achour, Noureddine Zerhouni. Réalité augmentée à partir d'une séquence vidéo en utilisant la stéréoscopie dense.. IEEE International Conference Francophone d'Automatique, CIFA'06., May 2006, Bordeaux, France. 6 p. hal-00341108

**HAL Id: hal-00341108**

**<https://hal.science/hal-00341108>**

Submitted on 24 Nov 2008

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

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

# Réalité augmentée à partir d'une séquence vidéo en utilisant la stéréoscopie dense

NADIA ZENATI<sup>1</sup>, KARIM ACHOUR<sup>1</sup>, NOUREDDINE ZERHOUNI<sup>2</sup>

<sup>1</sup> Division Robotique & Productique  
Centre de Développement des Technologies Avancées, Cité 20 Août 1956, Baba Hassen Algérie

<sup>2</sup> Laboratoire d'Automatique de Besançon  
25, rue Alain Savary-25000 Besançon, France

NZENATI@CDTA.DZ, KACHOUR@CDTA.DZ, ZERHOUNI@ENS2M.FR

**Résumé:** — Cet article s'intéresse à la réalité augmentée, c'est à dire à l'intégration d'objets virtuels générés par ordinateur dans une séquence d'images en gérant les occultations.

L'occultation est l'un des problèmes cruciaux en réalité augmentée. Il consiste à tenir compte des interactions entre les éléments virtuels insérés et la scène réelle: les parties occultées de ces éléments doivent être déterminées. La méthode proposée repose sur le calcul des cartes de disparité en utilisant les techniques de mise en correspondance denses.

Afin de retrouver des cartes de disparité denses, nous présentons dans cet article deux techniques d'appariement. La première est basée sur la programmation dynamique. Bien que cette méthode donne des résultats satisfaisants, elle reste néanmoins très gourmande en temps de calcul. Afin d'améliorer le temps de calcul ainsi que la qualité des résultats, nous proposons une autre méthode dite hybride basée sur l'approche multi résolution et la programmation dynamique.

Les cartes de disparité ainsi obtenues sont appliquées en réalité augmentée afin d'intégrer de manière réaliste des objets virtuels générés par ordinateur dans une séquence d'images.

La méthode d'augmentation proposée réduit considérablement l'intervention de l'utilisateur. L'applicabilité de la méthode est démontrée sur de nombreuses séquences d'images.

**Mots clés** — Réalité augmentée, gestion des occultations, mise en correspondance dense, programmation dynamique, approche hybride, cartes de disparité, augmentation.

## I. INTRODUCTION

De nos cinq sens, la vue est sans conteste celle que nous mettons le plus à contribution pour percevoir l'environnement qui nous entoure. Dès le début des années 80, la réalité virtuelle a profité de cette constatation afin de pouvoir immerger l'utilisateur dans un environnement artificiel et virtuel généré par ordinateur. Contrairement à cela, la réalité augmentée, vise à rehausser l'environnement réel par des entités virtuelles tout en laissant l'utilisateur au contact de son environnement réel. Ces nouvelles entités sont des éléments synthétisés par ordinateur et ajoutés aux images du monde réel perçues par les utilisateurs de façon réaliste.

Réaliser un système de réalité augmentée revient à résoudre le principal problème qui est l'intégration d'un ou de plusieurs objets virtuels dont nous connaissons le modèle 3D, la position et les propriétés photométriques. Cela revient à créer une nouvelle image « *image augmentée* » à partir de l'image réelle et du rendu des objets virtuels. La question qui se pose est comment modifier l'image réelle de tel sorte à avoir une intégration réaliste de notre objet virtuel ?

Les systèmes de réalité augmentée sont de nos jours de plus en plus utilisés non seulement par les sociétés de développements

mais aussi par le grand public. Pour ces raisons et d'autres, la réalisation d'un système de réalité augmentée n'est pas facile et a son lot de problèmes que ce soit un système en temps réel ou en post-production. Ces problèmes sont en général: *le point de vue*, *les occultations* et aussi *l'éclairage*. Un grand nombre de travaux s'est intéressé à chacun de ces problèmes en essayant de trouver une solution adéquate. Dans cet article, nous proposons une méthode de gestion des occultations. Celle-ci repose sur le calcul des cartes de disparité par des techniques de mise en correspondance denses. Les cartes de disparité auront le même rôle qu'un masque d'occlusion. En d'autres termes, l'information fournie par les cartes (la disparité) nous permettra de définir quelles sont les parties de la scène qui sont occultées ou non. La méthode que nous proposons est une méthode semi-automatique qui restreint l'intervention de l'être humain au maximum.

## II. MISE EN CORRESPONDANCE STEREOSCOPIQUE DENSE

Le problème de la mise en correspondance est l'un des problèmes les plus difficiles en vision par ordinateur. Des excellentes études sur les techniques de la vision stéréoscopique permettent de faire un tour d'horizon des méthodes publiées jusqu'à présent [10] [20] [22].

Nous présentons dans cette section la technique de mise en correspondance basée sur la programmation dynamique. Afin d'améliorer la qualité des résultats, ainsi que le temps de calcul, nous proposons une technique dite *hiérarchique* combinant entre la programmation dynamique et l'approche multi résolution. Dans ce qui va suivre, nous supposons que les images sont préalablement rectifiées. La rectification des images a été largement étudiée dans la littérature [2] [9] [11] [16] [17] [18]. C'est est une étape intermédiaire entre le calibrage et la mise en correspondance qui permet d'obtenir une géométrie épipolaire simple dans laquelle toutes les droites épipolaires sont parallèles et horizontales et qui correspondent aux lignes des images. Les points (ou pixels) mis en correspondance possèdent obligatoirement la même ordonnée et la recherche d'appariements devient monodimensionnelle le long des lignes de chaque image.

### II.1. Mise en correspondance par programmation dynamique

La programmation dynamique est une technique utile pour apparier deux séquences tout en respectant l'ordre des éléments à l'intérieur de chaque séquence. Son principe [14] [19] consiste donc à retrouver le chemin de correspondance optimal dans un graphe selon un critère de coût. Elle permet en particulier de retrouver le chemin le plus court. C'est

également un moyen d'obtenir la meilleure mise en correspondance entre deux vecteurs. Dans notre cas, les vecteurs sont constitués de deux lignes de l'image, plus précisément, de deux lignes épipolaires homologues, la première appartenant à l'image de référence et la deuxième à l'image objet. Nous définissons un intervalle de disparité  $[d_{\min}, d_{\max}]$  où  $d_{\min}$  et  $d_{\max}$  représentent respectivement la disparité minimale et la disparité maximale. Cet intervalle remplacera la deuxième ligne épipolaire (celle de l'image objet), ce qui correspond au segment de recherche car la programmation dynamique est appliquée sur des images rectifiées. La ligne épipolaire et l'intervalle de disparité constitueront un graphe 2D orienté et valué dans lequel nous chercherons le chemin le plus court. Chaque noeud  $(x, d)$  du graphe contenant la valeur du coût de mise en correspondance  $C(x, x+d)$  (figure.1) entre le pixel  $x$  appartenant à la ligne épipolaire de l'image de référence et le pixel  $x+d$  dans la deuxième lignes épipolaire peut être calculé en utilisant un des critères [1] cités ci-dessous :

La somme des différences absolues

$$SAD = \sum_{\substack{-m \leq i \leq +m \\ -m \leq j \leq +m}} |I_l(x+i, y+j) - I_r(x+i+d, y+j)| \quad (1)$$

2 La somme des différences carrées

$$SSD = \sum_{\substack{-m \leq i \leq +m \\ -m \leq j \leq +m}} (I_l(x+i, y+j) - I_r(x+i+d, y+j))^2 \quad (2)$$

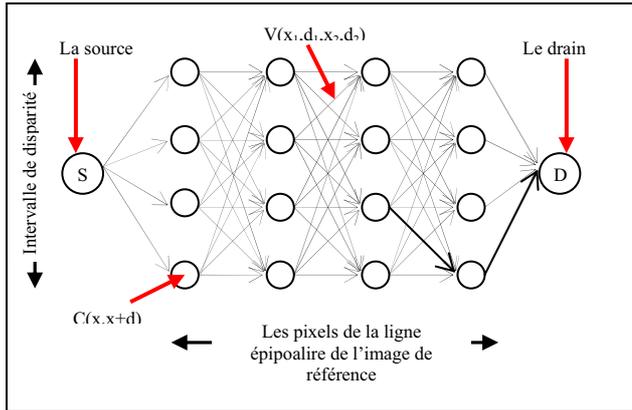
3. La somme des différences carrées centrées

$$ZSSD = \sum_{\substack{-m \leq i \leq +m \\ -m \leq j \leq +m}} ((I_l(x+i, y+j) - \bar{I}_l) - (I_r(x+i+d, y+j) - \bar{I}_r))^2 \quad (3)$$

4. La somme des différences carrées centrées normées

$$ZNSSD = \frac{\sum_{\substack{-m \leq i \leq +m \\ -m \leq j \leq +m}} (I_l(x+i, y+j) - \bar{I}_l) - (I_r(x+i+d, y+j) - \bar{I}_r))^2}{\sqrt{\sum_{\substack{-m \leq i \leq +m \\ -m \leq j \leq +m}} (I_l(x+i, y+j) - \bar{I}_l)^2} * \sqrt{\sum_{\substack{-m \leq i \leq +m \\ -m \leq j \leq +m}} (I_r(x+i+d, y+j) - \bar{I}_r)^2}} \quad (4)$$

$d$  représente une disparité appartenant à  $[d_{\min}, d_{\max}]$ .  $I_l$  and  $I_r$  sont respectivement les valeurs d'intensité des images gauches et droites au pixel  $(x, y)$ .  $(2m+1) \times (2m+1)$  représente la taille de la fenêtre de corrélation



**Figure.1.** Programmation dynamique: Graphe de mise en correspondance

Nous attribuons à chaque arc du graphe une valeur de distance élémentaire  $V(x_1, d_1, x_2, d_2)$  pondérée par une constante, la constante de lissage notée  $Const\_liss$  telle que :

$$v(x_1, d_1, x_2, d_2) = |d_1 - d_2| \times Const\_liss$$

Une fois le graphe construit, le chemin le plus court sera calculé en utilisant l'algorithme de Dijkstra [5] qui représente le meilleur algorithme d'optimisation pour retrouver le chemin de coût minimum dans un graphe. Dans notre cas, nous avons adopté le principe énoncé dans la phase de mise en

correspondance afin de retrouver le chemin le plus court dans le graphe construit en utilisant le principe de la programmation dynamique. Les noeuds appartenant au chemin le plus court correspondent aux couples d'appariement entre les deux lignes épipolaires. La programmation dynamique assure implicitement la contrainte d'unicité et la contrainte d'ordre. Le processus est réitéré pour chaque ligne épipolaire de l'image de référence.

## II.2. Notre approche : la programmation dynamique hiérarchique

La programmation dynamique est considérée comme l'une des méthodes les mieux adaptées pour la mise en correspondance dense mais reste néanmoins coûteuse en temps de calcul lorsque l'image présente un grand déplacement. Afin d'apporter une amélioration quant au temps de calcul et à la qualité des résultats et pour avoir ainsi une meilleure intégration, nous proposons une méthode hybride combinant entre la programmation dynamique et l'approche multi-résolution. L'approche proposée comprend trois phases bien distinctes :

### II.2.1. Construction de la pyramide d'images

L'approche proposée consiste tout d'abord à construire une hiérarchie d'images à différents niveaux de résolutions pour chaque image de la paire en utilisant la méthode quaternaire [21].

### II.2.2. Appariement au niveau grossier

Après la création de ces pyramides, un premier appariement est effectué au niveau le plus grossier (basse résolution) sur des images de plus petite taille en utilisant le principe de la programmation dynamique présenté dans la section II.2.

### II.2.3. Propagation des résultats

Les résultats obtenus lors du premier appariement au niveau grossier (niveau  $n$  de la pyramide) sont propagés au niveau inférieur (niveau  $n-1$ ) (figure.2). Ces résultats permettent de délimiter la zone de recherche associée au pixel  $p_1(x_1, y_1)$  de l'image gauche de niveau  $n-1$ . Cette zone de recherche est constituée de quatre fils du correspondant  $p_2(x_2, y_2)$  trouvé à partir des résultats correspondant au niveau  $n$ . Puisque les images sont rectifiées, la recherche se limitera aux deux fils se trouvant sur la même ligne épipolaire auquel nous ajouterons un fils à gauche et un fils à droite. Le processus est réitéré pour chaque niveau jusqu'au niveau initial. Cette approche permet d'affiner les résultats tout en réduisant le temps de calcul, ce qui la rend plus adaptée aux images de grande taille présentant un grand intervalle de disparité.

L'algorithme peut s'écrire comme suit :

1. Construire une paire de pyramides d'images de  $n$  niveaux (de  $0$  à  $n-1$ ) de résolutions.
2. Initialiser la disparité à zéro pour le niveau  $n$
3. Effectuer l'appariement au niveau grossier, niveau  $n$  en utilisant le principe de la programmation dynamique. décrite dans la section II.1.
4. Si  $n \neq 0$ , propager la carte de disparité obtenue au niveau supérieur de la pyramide, poser  $n=n-1$  et aller à l'étape.3, sinon passer à l'étape 5.
5. Affinement les cartes de disparités obtenues en appliquant un filtre médian.

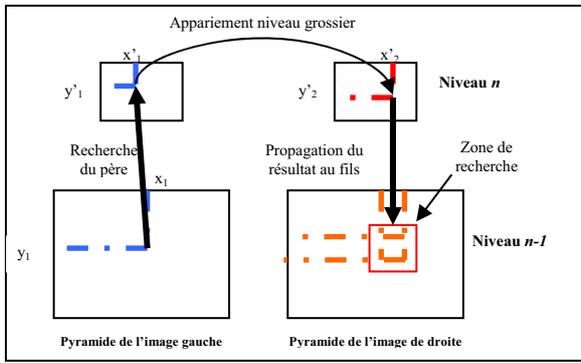


Figure.2. Propagation des résultats

### III. GESTION DES OCCULTATIONS

Réaliser un système de réalité augmentée revient à résoudre le principal problème qui est l'intégration d'un ou de plusieurs objets virtuels dont nous connaissons le modèle 3D, la position et les propriétés photométriques. Cela revient à créer une nouvelle image « *image augmentée* » à partir de l'image réelle et du rendu des objets virtuels. La question qui se pose est comment modifier l'image réelle de tel sorte à avoir une intégration réaliste de notre objet virtuel ?

Les problèmes rencontrés lors de la réalisation d'un tel système sont du nombre de trois : *le point de vue*, *les occultations* et aussi *l'éclairage* qui sont dues principalement au manque ou même à l'absence d'informations sur la scène réelle qui peuvent être calculés à partir de dispositifs spécifiques qui ne sont pas toujours disponibles pour le grand public. Il existe un grand nombre de méthodes traitant du problème des occultations [3] [4] [8] [15]. La plus répandue est la méthode utilisant le masque manuel ou communément le masque d'occlusion [12]. Malgré le fait que cette méthode soit rudimentaire et fastidieuse, elle reste de loin la méthode qui donne les meilleurs résultats. La méthode se résume en quatre étapes : (1) Filmer la séquence originale, (2) Modéliser l'objet virtuel, (3) Dessiner le masque, et (4) Intégrer l'objet virtuel dans la séquence aux endroits qui ne sont pas masqués. La troisième étape de la méthode est la plus importante. Elle exige un travail fastidieux et demande beaucoup de temps.

#### III.1. Etapes de la méthode

Notre méthode de gestion des occultations repose sur le calcul des cartes de disparité en utilisant la mise en correspondance dense. Les cartes de disparité auront le même rôle que le masque d'occlusion. En d'autres termes, l'information fournie par les cartes (la disparité) nous permettra de définir quelles sont les parties de la scène qui sont occultées ou non. La méthode que nous proposons est une méthode semi-automatique qui restreint l'intervention de l'être humain au maximum. Elle se décompose en quatre phases :

##### III.1.1. Filmer la séquence réelle

L'acquisition doit garantir certains critères afin de produire des cartes de disparité de grande qualité. Pour que la carte de profondeur représente bien la profondeur de la scène, seule la caméra doit bouger. Il faut donc filmer à une cadence lente sans qu'il y ait ni de mouvements brusques, ni de retour en arrière. Les séquences d'images utilisées dans ce travail vérifient bien les contraintes imposées.

##### III.1.2. Modéliser l'objet virtuel

Pour modéliser les objets virtuels, nous avons choisi le logiciel **3D Studio Max** pour avoir un meilleur rendu. En effet, le logiciel **3DS MAX** nous permet de créer des modèles 3D de qualité professionnelle. La qualité du rendu est le critère de

mesure sur quoi nous sommes basées vu que notre but majeur est d'intégrer une entité virtuelle dans une séquence de la façon la plus réaliste possible.

#### III.1.3. Calcul des cartes de disparité dans une séquence d'images

Pour calculer les cartes de disparité pour une séquence de  $n$  images, nous considérons que chaque paire d'images successives de la séquence comme étant une paire stéréoscopique.

Le principe suivi appliqué sur une séquence de  $n$  images est décrit comme suit (Figure.3) :

1. Calculer la carte de disparité  $Disp$  entre l'image  $i$ , image de référence, et l'image  $i+1$ , image objet, de la séquence telle que  $i \in [1, n-1]$ .
2. Calculer la carte de disparité pour la paire  $(n, n-1)$  où  $n-1$  est l'avant dernière image et  $n$  est la dernière image de la séquence.

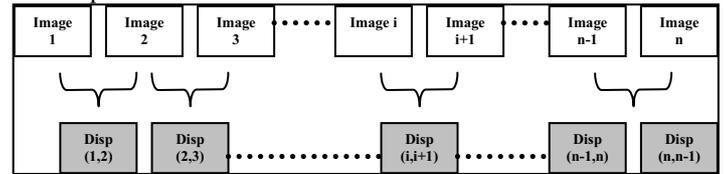


Figure.3. Calcul des cartes de disparité pour une séquence d'images

Un des problèmes majeurs rencontrés dans les applications de réalité augmentée est celui des semi-occlusions causé par les objets apparaissant dans certaines images, et absents dans d'autres images, ce qui rend l'appariement difficile et fausse le calcul. Pour trouver alors la disparité de ces pixels qui n'ont pas de correspondants dans les autres images, nous proposons une méthode qui se base sur la fusion des cartes. Pour pouvoir appliquer cette méthode, il faut que toute la scène soit filmée dans le même sens (de gauche à droite ou de droite à gauche) sans qu'il y ait de retour arrière. Nous pouvons constater que lors des semi-occlusions, les algorithmes de mise en correspondance stéréo dense choisissent des disparités grandes, en d'autres termes, ils choisissent des pixels éloignés. Pour cela, nous proposons de construire deux cartes de disparité, l'une en appariant l'image de référence avec l'image à sa gauche (celle qui la précède) et l'autre en appariant toujours l'image de référence avec l'image à sa droite (celle qui lui succède). Puis il ne reste qu'à fusionner les deux cartes en prenant pour chaque pixel la disparité minimale entre les deux cartes (Figure.4).

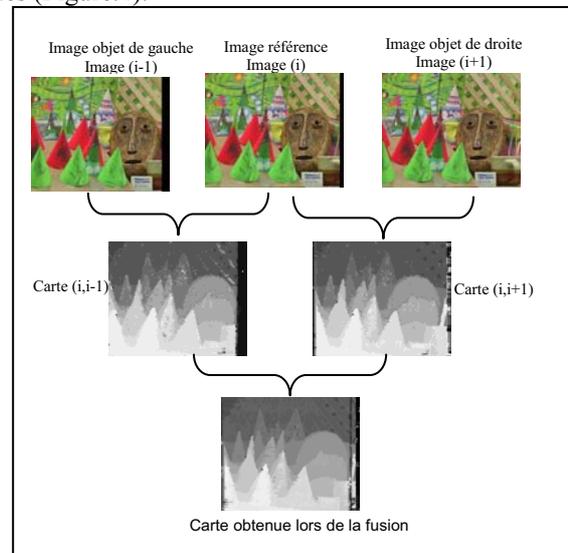


Figure.4. Méthode de fusion des cartes

### III.1.4. Intégrer l'objet virtuel dans la séquence

La technique d'intégration adoptée est une technique qui repose sur la profondeur des objets. Après le calcul des différentes cartes de disparité, un rendu de l'intégration se fera à partir des données fournies par les cartes de disparité de la scène réelle et de la séquence virtuelle.

Le principe de l'intégration se base sur le choix entre l'intensité ou la couleur du pixel dans l'image réelle et celle de l'objet virtuel. Le principe est le suivant :

1. Comparer les valeurs des disparités obtenues de la séquence réelle et virtuelle.
  2. soit  $D_r(x,y)$  et  $D_v(x,y)$  les disparités respectivement d'un pixel  $(x,y)$  obtenu à partir de la scène réelle et celui de la scène virtuelle. Si  $D_r(x,y)$  est plus grande que  $D_v(x,y)$ , alors l'intensité ou la couleur  $I_r(x,y)$  du pixel de la scène réelle sera choisie. Dans le cas contraire, c'est l'intensité ou la couleur  $I_v(x,y)$  du pixel de l'objet virtuel qui sera choisie.
- La figure.5 illustre le principe de l'augmentation décrit.

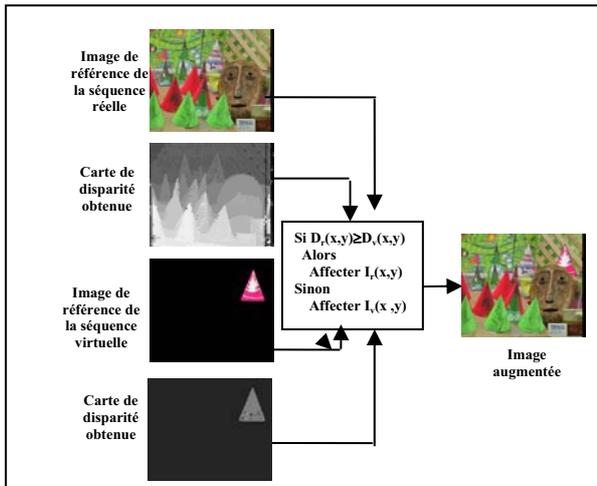


Figure.5. Principe de l'augmentation

## IV.EVALUATION DES RESULTATS

Dans cette section, on se propose de présenter les principaux résultats obtenus de mise en correspondance dense en utilisant les techniques décrites dans cet article. Afin de valider nos résultats, plusieurs paires et séquences d'images réelles sont utilisées. Quelques exemples d'augmentation obtenus sur des séquences d'images en utilisant notre méthode d'intégration seront également présentés. Ces différents tests prennent en compte l'intégration de l'ensemble des modules et outils sous l'environnement C++ Builder 5 sous la plate-forme Windows.

### IV.1. Expérimentation 1 : Résultats d'appariement sur des paires d'images

Les Figures (6c), (7c), (8c) montrent les résultats d'appariement obtenus sur différentes paires d'images en utilisant la technique de la programmation dynamique. Les Figures (6d), (7d), (8d), présentent les différentes cartes de disparité obtenues sur différentes paires d'images en utilisant notre approche hybride. Notons que les points valides apparaissent en blanc sur les différentes cartes de disparité et les points non appariés en noir.

Le tableau.1 montre le temps d'exécution (en secondes) et le pourcentage de points appariés. Notons que TI, ID, FC et également CC et CL représentent respectivement la taille de l'image, l'intervalle de disparités, la taille de fenêtre de corrélation, le critère de corrélation utilisé, et enfin la constante de lissage associée à la technique de programmation dynamique.

(Noms d'images : (A)=Tsukuba, (b) = Hervé, (C)= Cônes)

	T.I	ID.	FC	CC	CL	Temps	%
A	384*288	[0,13]	5*5	SSD	5	2,90s	95,11
B	512*512	[10,60]	11*11	ZSSD	20	164,36s	68,89
C	434*380	[1,7]	9*9	SSD	20	7,91s	98,90

Tableau.1. Temps d'exécution et pourcentage d'appariement sur différentes paires d'images utilisant la programmation dynamique.

La programmation dynamique donne de bons résultats et fournit des cartes de disparités denses. Le pourcentage d'appariement est supérieur à 70%. Cependant dès que l'intervalle de disparité devient grand, comme dans le cas de la paire « Hervé » de l'INRIA, la programmation dynamique n'est plus recommandée.

Une amélioration considérable est observée en utilisant l'approche hybride. Le tableau.2 montre le temps d'exécution ainsi que le pourcentage d'appariements obtenus sur différentes paires d'images en utilisant l'approche hybride.

	Niv_res	ID.	FC	CC	CL	Temps	%
A	2	[0,6]	3*3	SSD	2,5	0,97s	96
B	3	[3,15]	9*9	ZSSD	2,5	27,15s	78
C	2	[0,3]	7*7	SSD	10	4,12s	99

Tableau 2. Temps d'exécution et pourcentage d'appariement sur différentes paires d'images utilisant la technique hybride

A partir de ces résultats, nous pouvons constater que notre méthode permet de réduire considérablement le temps d'exécution surtout lorsque les images sont de grande taille et présentent un grand intervalle de disparité. Nous remarquons que le temps d'exécution a été divisé par 9 par rapport au résultat de la programmation dynamique notamment pour la paire « hervé ». Aussi, la méthode permet de produire des images de disparité de grande qualité. Nos méthodes permettent de préserver les petits détails qui sont négligés par les méthodes d'appariement. Ces détails sont utiles lors de la phase d'augmentation afin d'avoir une intégration de bonne qualité visuelle.

### IV.2. Expérimentation 2 : généralisation sur les séquences d'images et intégration d'objets virtuels dans une séquence

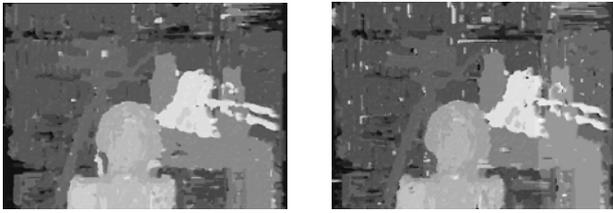
La qualité de l'intégration dépend entièrement de la qualité des cartes de disparités obtenues. Afin d'améliorer les cartes de disparité obtenues sur des séquences d'images, une méthode dite de fusion a été proposée. La figure 9 montre la séquence réelle « Teddy ». La figure 10 montre les images de disparité obtenues en utilisant la méthode dite de fusion. Le nombre de points appariés a augmenté. Cela est visible surtout au niveau des contours qui sont sujets au semi-occlusions. La qualité des augmentations est due essentiellement à la qualité des cartes de disparité et d'autre part au sens créatif de l'utilisateur. La figure 11 montre l'incrustation d'une voiture virtuelle dans la séquence « Teddy ». La figure 13 montre également l'incrustation d'un cône virtuel dans la séquence « cône ». Notons que les augmentations ont été réalisées sur des séquences d'images couleurs. Nous pouvons remarquer que les résultats d'augmentation obtenus sont satisfaisants.



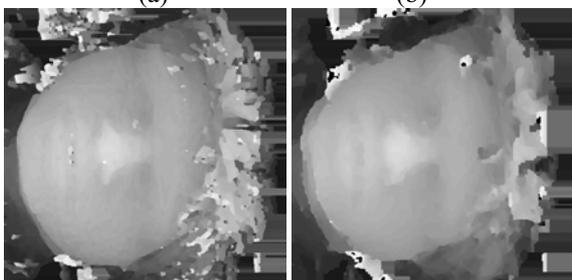
(a)



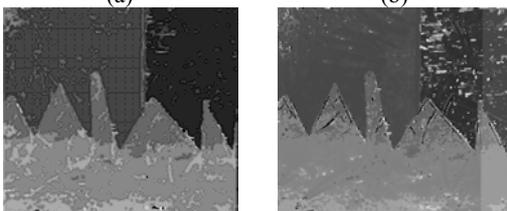
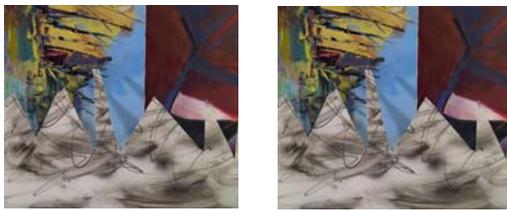
(b)



**Figure.6.** (a, b) Scène réelle avec images rectifiées : Tsukuba  
(c,d) Images de disparité obtenues en utilisant respectivement  
(c) la programmation dynamique et (d) l'approche hybride proposée



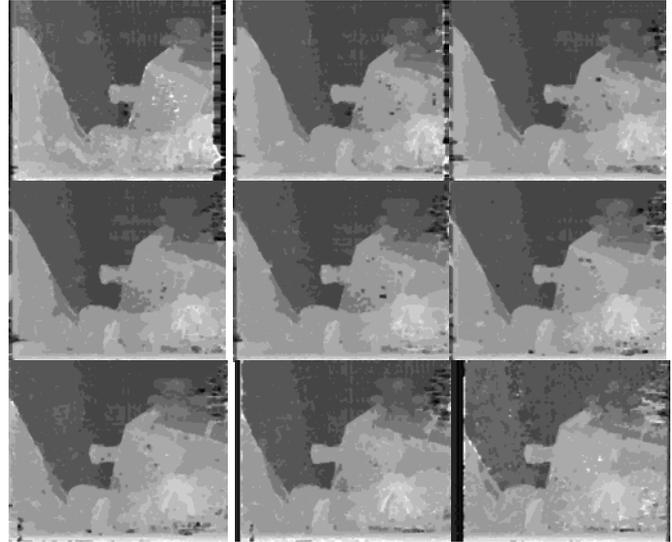
**Figure.7** (a, b) Images gauche et droite de la paire rectifiée "hervé",  
(c,d) cartes de disparité denses obtenues en utilisant respectivement  
la programmation dynamique et l'approche hybride



**Figure.8** (a, b) Scène réelle avec images rectifiées : « Sawtooth. »  
Images de disparité obtenues en utilisant (c) la programmation  
dynamique et (d) l'approche hybride



**Figure.9 :** La séquence réelle « Teddy » avec images rectifiées



**Figure.10.** Cartes de disparité obtenues à partir de la séquence  
« Teddy » en utilisant l'approche hybride avec fusion



**Figure.11.** Incrustation d'une voiture virtuelle dans la séquence  
« Teddy »



**Figure.12.** La séquence réelle « cône » avec images rectifiées



Figure.13. Incrustation d'un cône virtuel dans la séquence « cône »

## V. CONCLUSION

Nous venons de présenter dans cet article une technique qui permet d'augmenter des séquences d'images en gérant les occultations. La méthode que nous avons proposé pour gérer les occultations repose sur l'exploitation des principes et techniques avancés de la stéréovision. Nous avons à cet effet utilisé deux méthodes de mise en correspondance denses afin de calculer les cartes de disparité qui serviront dans la phase d'augmentation. Nous avons en premier lieu implémenté la technique de la programmation dynamique qui donne des résultats très satisfaisants. Cependant une bonne intégration dépend de la qualité des cartes obtenues lors de la phase d'appariement. Pour améliorer la qualité des résultats, nous avons proposé une approche hybride qui combine entre l'approche multi-résolution et la programmation dynamique.

Le problème des semi-occlusions a pu être résolu grâce à la méthode de fusion proposée lors de la généralisation du calcul sur une séquence d'images. La méthode d'augmentation que nous avons proposé réduit considérablement l'intervention de l'utilisateur. Les résultats obtenus sur plusieurs séquences d'images montrent l'efficacité et l'applicabilité de notre méthode. Néanmoins, certaines améliorations peuvent être apportées. Prendre en compte les autres problèmes liés à la réalité augmentée qui sont le point de vue et l'éclairage et envisager d'utiliser le parallélisme afin de réduire le temps d'exécution qui risque d'augmenter avec le nombre d'images composant les séquences utilisées.

## VI. REFERENCES

- [1] P. Aschwander, W. Guggenbul, "Experimental result from a comparative study on correlation-type registration algorithms", In *FORSTNER et RUWIEDEL, Robust computer vision*, Wichmann, pp.268-282, 1992.
- [2] N. Ayache and C. Hansen, "Rectification of images for Binocular and trinocular stereovision", *Proc. Intern. Conf. on Pattern Recognition*, pp. 11-16, 1988.
- [3] R T Azuma, "A Survey of Augmented Reality". *Presence: Tele-operators and Virtual Environments*, Los Angeles 6, pp 355-385, Août 1997
- [4] M.O. Berger, "Resolving Occlusion in Augmented Reality: a Contour-Based Approach without 3D Reconstruction"; *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Puerto Rico, pp91-96, 1997.
- [5] E.W. Dijkstra, "A note on two problems in connection with graphs", *Numer Math*, vol 1 pp 269-271, 1959.
- [6] M. Drumheller, T. Poggio, "On Parallel Stereo"; *Proceeding IEEE Conference on Robotics and Automation*, pp1439-1448, 1986.
- [7] O. Faugeras, "Three-Dimensional Computer Vision: A Geometric Viewpoint", *MIT press*, 1993.
- [8] N. Grammalidis, L.G. Bleris, M.G. Strintzis. "Depth and Occlusion Estimation from Uncalibrated Camera Views using Dynamic Programming along the Epipolar Lines". *Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS'99*, Berlin, June 1999.
- [9] R.I. Hartley, A. Zisserman, "Multiple view geometry in computer vision", Cambridge University Press, 2000.
- [10] A.Koschan., "What is new computational stereo since 1989: a survey of current stereo papers", *Technischer Bericht 93-22*, Technische Univer at Berlin, 1993.
- [11] C. Loop, Z. Zhang, "Computing Rectifying Homographies for Stereo Vision"; *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, juin 1999, Colorado.
- [12] V.Lepetit, "Gestion des occultations en réalité augmentée"; Thèse de doctorat, université de Nancy, 23 mai 2001.
- [13] K. Mühlmann, D. Maier, J. Hesser, R. Männer. "Calculating dense disparity maps from color stereo images, an efficient implementation", *CVPR 2001 Stereo Workshop, IJCV 2002*
- [14] Y. Ohta, T. Kanade, "Stereo by intra- and inter-scanline search using dynamic programming", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ,PAMI-vol 7, pp 39-154, 1985.
- [15] K. C. Ong, H. C. Teh, and T. S. Tan. "Resolving occlusion in image sequence made easy". *The Visual Computer*, Vol14, pp153-165, 1998.
- [16] D. Papadimitriou, T. Dennis, "Epipolar line estimation and rectification for stereo image pairs"; *IEEE Trans. Image Processing*, vol 5, pp.672-676, 1996.
- [17] M. Pollefeys, R. Koch, L. Van Gool, "A simple and efficient rectification method for general motion", *Proc. ICCV*, pp. 496-501, 1999.
- [18] S. Roy, I. Cox, "A maximum-flow formulation of the n-camera stereo correspondence problem"; *In International Conference on Computer Vision (ICCV)*, 1998.
- [19] D.k. Smith, "Dynamic Programming: An Introduction", *Part of the Millennium Mathematics Project* sponsored by the Univeristy of Cambridge, 1997.
- [20] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms", *International Journal of Computer Vision*, pp.7-42, May 2002.
- [21] C. Sun, "Multi-Resolution Stereo Matching Using Maximum-Surface Techniques", *Techniques and Applications*, Perth, Australia, December 7-8, pp.195-200, 1999
- [22] R. Szeliski and R. Zabih. "An experimental comparison of stereo algorithms". In *International Workshop on Vision Algorithms*, pp.1-19, Greece, September 1999.