

**RECONSTRUCTION DE PAIRES STEREO-
SCOPIQUES PAR CHAMPS DENSES DE
DISPARITE**

RECONSTRUCTION DE PAIRES STEREOSCOPIQUES PAR CHAMPS DENSES DE DISPARITE

Thibaud GAUTHEY



Université du Québec

Institut national de la recherche scientifique

INRS-Télécommunications

16, place du Commerce, Verdun

Québec, Canada, H3E 1H6

décembre 1998

Rapport technique de l'INRS-Télécommunications no. 98-15

J'aimerais tout d'abord remercier tous ceux qui ont rendu ce stage possible. M. André Girard, directeur de l'INRS-Télécommunication, M. Janusz Konrad, professeur à l'INRS, qui m'a offert ce stage et m'a dirigé pendant cette période, M. Henri Maitre, à la tête du département images de l'ENST. Je tiens également à remercier Abdol-Réza Mansouri et plus généralement tous les agents de recherches, étudiants et personnels de l'INRS pour leur aide précieuse.

Résumé

Le présent rapport a pour but de définir l'influence de certains paramètres agissant, dans le cadre du traitement d'images stéréoscopiques, sur l'ajustement de parallaxe pour les images fixes. Seule l'approche basée sur les champs de disparités denses est abordée.

Les paramètres peuvent influencer soit sur l'étape d'estimation de la disparité, soit sur la reconstruction de vues intermédiaires proprement dite. Cependant, signalons dès à présent l'importance des images utilisées lors des différentes étapes de la reconstruction. Ces images doivent être de très bonne qualité.

Dans un premier cas, le choix des images à utiliser dans une paire stéréoscopique est abordé. Ensuite le problème de la propagation du champ de disparité est traité, suivi par une description des problèmes dus à l'interpolation, et par une petite discussion sur la précision de la disparité. Pour finir, une nouvelle méthode d'estimation est expliquée.

Même si dans l'ensemble, la reconstruction d'images intermédiaires n'est pas encore parfaite, certaines améliorations ont été apportées. Certains aspects permettant la réduction des fichiers à stocker ou des données à transmettre ont également été abordés.

Table des matières

1	Introduction	1
2	Processus expérimental	1
2.1	calcul de la disparité	1
2.2	reconstruction de l'image	3
3	Utilisation d'une image d'origine	3
4	Propagation du champ de disparité	4
4.1	observations	5
4.2	résultats analytiques	6
5	Influence de la méthode d'interpolation	9
5.1	reconstruction d'une image	9
5.2	reconstruction d'une séquence	10
5.3	gestion des trames impaires	11
6	Influence de la précision sur la disparité	14
7	Interpolation B-spline	14
7.1	introduction aux fonctions de Bessel	14
7.2	estimateur	15
7.3	résultats	16
8	Conclusion	19
A	Annexe – Utilitaires	20

Table des figures

1	disparité	2
2	Un décalage des caméras introduit une rotation.	4
3	Première image de la séquence <i>flower</i> gauche.	5
4	<i>manege</i> reconstruit en $\alpha = 0,5$	6
5	Champs disparité aux positions α_1 et α_2	7
6	surface des erreurs dues à la propagation	8
7	Comparaison des vecteurs de disparité dans une trame paire et dans une trame impaire.	10
8	atténuation linéaire des filtres cubiques et linéaires	12
9	atténuation logarithmique des filtres cubiques et linéaires	13
10	atténuation du filtre B-spline optimal	17
11	atténuations comparées des différentes méthodes d'interpolation.	18
12	Agrandissement par 4 d'un détail de <i>flower</i> par les 3 interpolateurs.	18

1 Introduction

Un film stéréoscopique est obtenu à l'aide de deux caméras identiques séparées par une courte distance. Une caméra joue le rôle de l'œil gauche, tandis que l'autre joue celui de l'œil droit. Une fois le film enregistré, un dispositif approprié¹ permet de projeter sur l'œil du spectateur l'image enregistrée par la caméra correspondante, et ainsi de reproduire une impression de trois dimensions.

On constate que certaines personnes supportent difficilement les films stéréoscopiques. En effet, les individus ne réagissent pas tous de la même façon devant une scène stéréoscopique du fait des différences physiques qui existent entre chacun. De nombreux paramètres influencent ce phénomène, et l'un d'entre eux est l'écart entre les pupilles, qui correspond plus ou moins à l'écart entre les deux caméras.

Les réalisateurs de films stéréoscopiques fixent cet écart lors du tournage et ne peuvent guère le changer. Pour que chaque spectateur ait un confort de vision, il faudrait une série de caméras séparées par des écarts très petits. Comme on ne dispose que des deux vues filmées par les deux caméras, on va créer numériquement des vues intermédiaires et créer ainsi des paires d'images stéréoscopiques avec un écart relatif entre les caméras plus petit (ou plus grand) que l'original. Tout ce qui suit s'appuie sur [1] et le lecteur a grand intérêt à lire cette thèse.

La reconstruction d'images intermédiaires a déjà été effectuée, mais ce qui suit a pour but de démontrer l'influence des différents paramètres utilisés lors du calcul de la disparité ou de la reconstruction.

2 Processus expérimental

On dispose d'un certain nombre de paires stéréoscopiques pour faire nos différents tests. On calcule tout d'abord le champ de disparité entre les images gauche et droite. Ce champ correspond à un ensemble de vecteurs, chaque vecteur représentant la différence de position sur les deux images d'un même point de l'espace filmés par les caméras (cf. figure 1).

Une fois cette disparité calculée, on peut reconstruire une image entre les deux images de référence. On repère la position de l'image reconstruite par α : $\alpha = 0$ représente l'image gauche, $\alpha = 1$ représente l'image droite, les autres positions sont proportionnelles à la distance aux deux images d'origines. L'écart entre deux images peut donc être noté par $\Delta\alpha$ qui est la différence entre les deux positions.

Le processus expérimental se décompose en plusieurs parties, dont les deux plus importantes sont le calcul de la disparité et la reconstruction de l'image intermédiaire.

2.1 calcul de la disparité

Le calcul du champ de disparité doit se faire à partir des images d'origines gauche et droite. On constate cependant qu'il est préférable d'utiliser des images balancées.

1. lunettes à cristaux liquides, lunettes à filtre rouge/bleu, ...

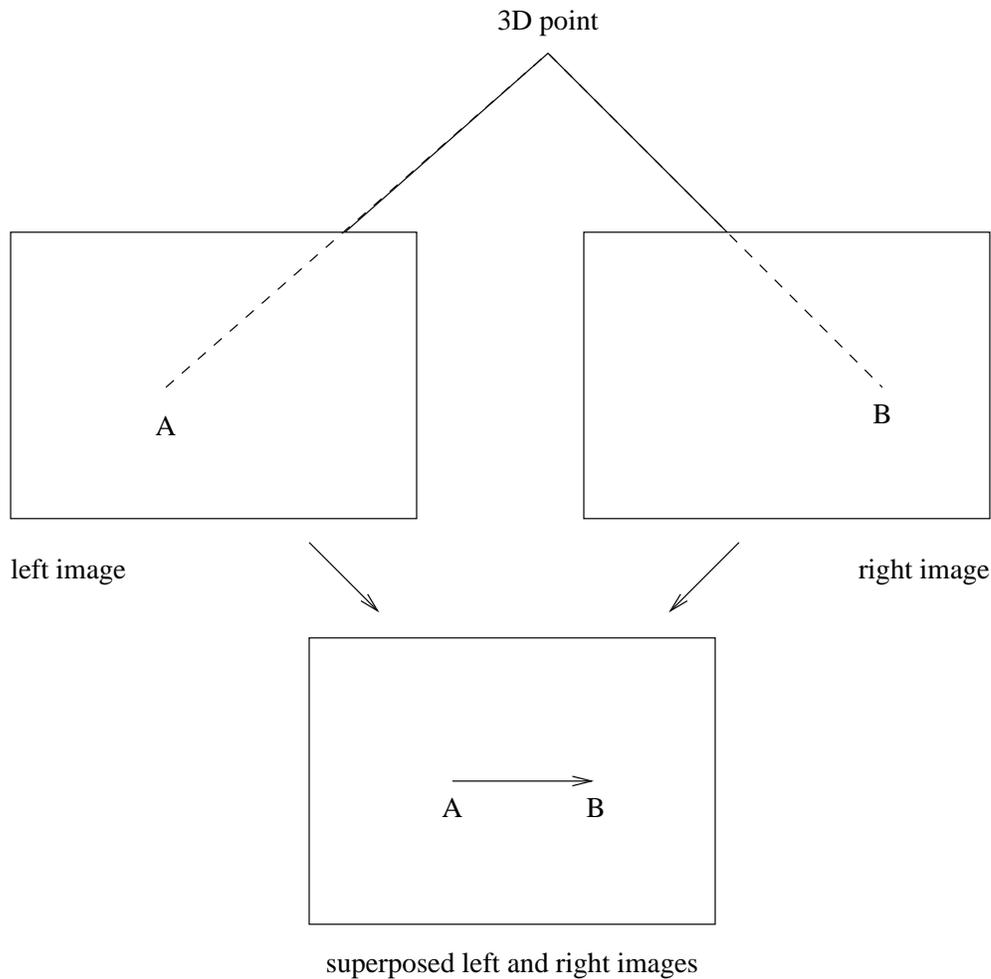


FIG. 1 – la position d'un point spatial est différentes dans chaque image. Cette différence est appelée *disparité*.

Pour ce faire on utilise le programme `balance`.

La méthode de calcul utilise l'approche pyramidale décrite dans [1]. Les images de la pyramide ne doivent pas être simplement sous échantillonnées, mais aussi filtrées. Un programme qui réalise cette étape a été développé. Il s'agit de `Prmd2Dseq` qui crée la pyramide à partir d'une séquence `Vids`. L'utilisateur définit son filtre dans un fichier. Voir la *man page* pour plus de détail.

Les paramètres de lissage utilisés lors de mes expériences ont été : $\lambda = 1500$ (lissage horizontal) et $\gamma = 3000$ (lissage vertical). Bien entendu, pour certaines images (comme *piano*), il est inutile de prendre des contraintes si élevées et des paramètres trois fois plus petits auraient très bien convenu. Pour d'autres images (comme *manege*), il aurait fallu prendre des paramètres encore plus importants. En effet, la facilité du calcul de la disparité dépend de la topologie de l'image (présence d'occlusions, taille des zones non appariées, ...).

Les séries d'expériences ont montré que cette étape est cruciale pour la qualité du

résultat. Les deux contraintes de lissage ont été fixées par tâtonnements successifs, mais il est évident qu'une modification apportée à ces deux valeurs pourrait améliorer le résultat. Il a également été montré que le résultat obtenu est extrêmement sensible à la qualité des images d'origines utilisées pour évaluer la disparité. C'est pourquoi il est recommandé fortement de travailler avec des images balancées et une pyramide qui donne de bons résultats. En effet, plusieurs pyramides ont été utilisées. Ces pyramides ne présentaient pas de différences importantes (l'écart entre la valeur des pixels dans les images des différentes pyramides n'était jamais supérieur à 1 sur une dynamique de 256).

2.2 reconstruction de l'image

Une fois le champ de disparité calculé (création des fichiers “.dspl”), on reconstruit alors l'image avec le programme `reconstruct`. Ce programme permet de créer soit une image fixe, soit une séquence. Une séquence est une suite de N images reconstruites pour un α variant entre deux positions α_1 et α_2 , avec un pas de $\frac{\alpha_2 - \alpha_1}{N}$.

On peut également choisir entre reconstruire une séquence à partir de champs de disparité calculés à chaque position, ou alors propager un champ de disparité et l'utiliser pour reconstruire toutes les positions.

Les séquences permettent de mieux voir certains défauts. Elles correspondent à un mouvement de la tête qu'effectuerait le spectateur. Les positions de reconstruction ne se limitent pas entre les positions gauche et droite des caméras d'origine, mais des images sont également reconstruites à l'extérieur ($\alpha < 0$ et $\alpha > 1$). Ces images permettraient d'augmenter la disparité. Dans ce cas précis, le poids des images d'origine dans l'obtention de l'image reconstruite doit être normalisé. En effet, dans un cas classique (reconstruction pour $0 < \alpha < 1$) l'image gauche est pondérée par $|1 - \alpha|$ et l'image droite par $|\alpha|$. Ces deux poids ont donc été normalisés :

- pour l'image gauche : $\frac{|1-\alpha|}{|1-\alpha|+|\alpha|}$
- pour l'image droite : $\frac{|\alpha|}{|1-\alpha|+|\alpha|}$

3 Utilisation d'une image d'origine

Dans un cas classique, on reconstruirait deux images intermédiaires à des positions symétriques par rapport à $\alpha = 0,5$ pour faire une paire stéréoscopique. Mais qu'obtenons nous si on utilise une image d'origine (i.e. gauche ou droite) dans cette paire. Pour rendre la comparaison plus facile, il faut que les deux paires stéréoscopiques (celle avec l'image d'origine et celle avec les deux images reconstruites) est la même valeur de $\Delta\alpha$. On introduit ainsi une rotation entre les deux vues. (cf figure 2) Or, mis à part cette rotation, on ne s'attend pas à avoir une différence entre les deux paires.

L'image de test utilisée est le premier champ de la séquence `flower`. (cf. figure 3) A première vue, les résultats sont conformes à nos attentes, mais quand on regarde

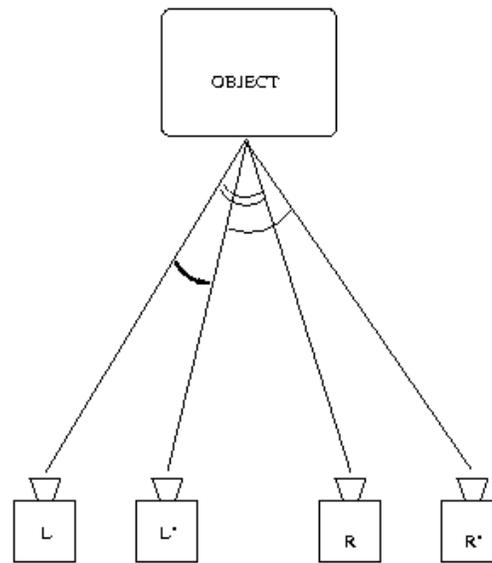


FIG. 2 – *Un décalage des caméras introduit une rotation.*

plus attentivement les images obtenues, on s'aperçoit que la paire stéréoscopique utilisant une image d'origine est un peu plus nette à certains endroits (surtout les zones texturées). En se plaçant dans le cas extrême où on compare les deux paires $\{\alpha = 0; \alpha = 0,5\}$ et $\{\alpha = 0,25; \alpha = 0,75\}$, on constate toujours le même résultat. Pourtant l'image reconstruite en $\alpha = 0,5$ est plus filtrée que les deux autres images reconstruites en $\alpha = 0,25$ et $\alpha = 0,75$.

Sur les autres images de test, on peut observer le même phénomène. En fait, l'utilisation d'une image d'origine améliore le rendu des textures. Les formes elles-mêmes ne sont pas mieux définies, car un défaut de reconstruction comme dans l'image *manege* (cf. figure 4) n'est pas atténué, mais en regardant attentivement, la surface des objets (surtout les surfaces granuleuses), on s'aperçoit qu'elles sont plus nettes.

La différence d'information entre une image d'origine et une image reconstruite se fait surtout au niveau de la netteté de l'image et très peu au niveau de la position des objets. Nos yeux perçoivent la qualité de l'image d'origine et l'autre image ne sert qu'à donner la profondeur de l'image tridimensionnelle.

A priori, l'utilisation d'une image d'origine dans une paire stéréoscopique n'a que des avantages. En effet, cela permet l'économie du calcul d'une image, ce qui est loin d'être négligeable. Et cela améliore aussi la qualité de l'image visualisée.

4 Propagation du champ de disparité

Jusqu'à présent, on a reconstruit une image à la position α à partir du champ de disparité calculé à cette même position. Mais que ce passe-t-il si on utilise le même champ de disparité? (calculé en $\alpha = 0,5$ par exemple). Car ceci a l'énorme avantage



FIG. 3 – Première image de la séquence flower gauche.

de ne faire calculer qu'un seul champ de disparité, ce qui est l'étape la plus longue dans tout le processus.

4.1 observations

En effet, dans le cas précédent, on calculait la disparité pour chaque position α (cf. figure 5). Ainsi les champs de disparité pouvaient présenter des différences entre les positions α_i . Maintenant, c'est le même champ qui est recopié à toutes les positions.

On va donc comparer une paire $[\alpha_1, \alpha_2]$, où les images ont été reconstruites à partir des deux champs de disparités calculés en α_1 et en α_2 , avec la même paire où les images ont été reconstruites à partir du champ de disparité calculé en $\alpha = 0, 5$.

La disparité ne dépend en faite que des images d'origine et devrait être unique pour toutes les positions α . Cependant, la minimisation réalisée lors du calcul de la disparité est fait à partir d'une position α particulière. La position α est utilisée comme pivot et elle influence le choix du vecteur de disparité qui minimise l'erreur entre les images gauche et droite. (cf. [1] pour plus de détails).

Les résultats sont très intéressants. En effet, il n'y a que quelques infimes différences entre les deux paires, et ces différences ne sont que sur quelques points de détail. On a donc un gain en temps important à faire en n'utilisant qu'un seul champ



FIG. 4 – *image manège reconstruite en $\alpha = 0,5$. Les déformations sur les tiges verticales sont importantes car il y a de brusque variations de disparité, ce qui la rend difficile à évaluer.*

de disparité, pour une perte de qualité quasi imperceptible. Ce résultat est particulièrement intéressant si on a à reconstruite plusieurs images intermédiaires à partir de la même paire.

4.2 résultats analytiques

On cherche un moyen de comparer analytiquement la différence entre les images obtenues par propagation d'un champ de disparité et les images ou ce champ est reconstruit à chaque fois. Considérons tout d'abord qu'une image reconstruite à la position où son champ de disparité à été évalué est optimale. C'est hypothèse est justifiable par le fait que tout le monde est d'accord pour trouver que cette image est un peu meilleure que l'image ou le champ à été propagé.

Après avoir effectué la différence des valeurs de luminance sur les deux types d'images, on fait la moyenne quadratique et on compare le résultat obtenue pour

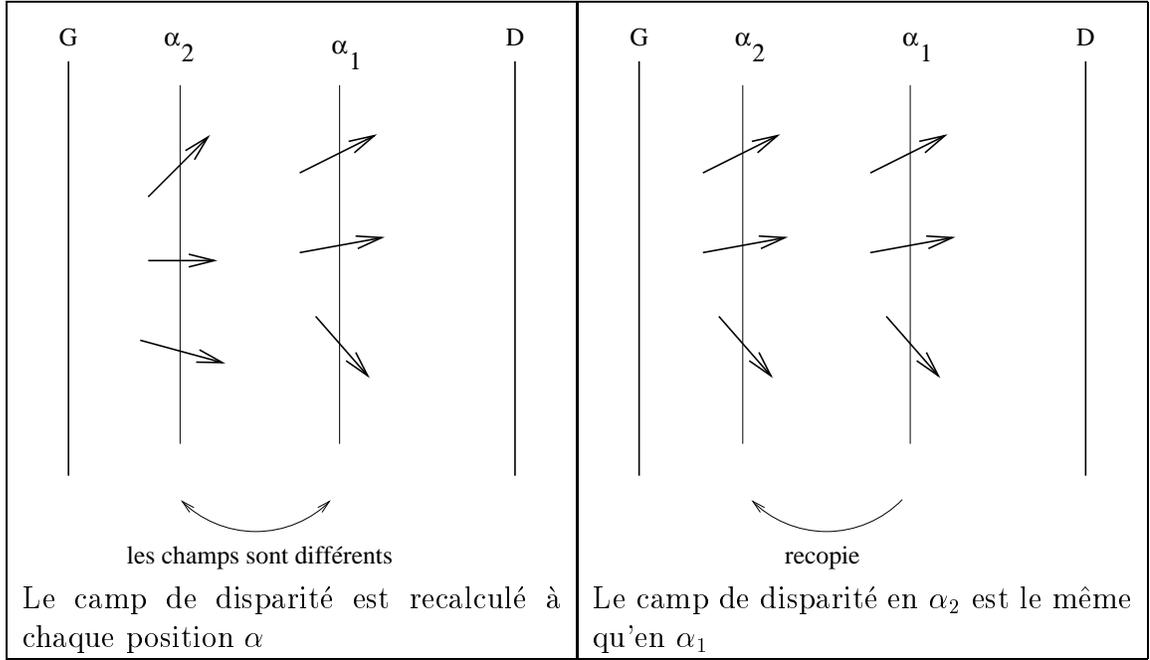


FIG. 5 – Comparaison des champs de disparité pour deux positions α_1 et α_2 . Les figures correspondent à des vues de dessus des images intermédiaires.

différentes positions de reconstruction et d'évaluation de la disparité.

Soit α la position à laquelle l'image est reconstruite et soit γ la position à laquelle le champ de disparité est évalué, on peut faire varier ces deux paramètres et tracer ainsi la surface des erreurs aux différentes positions.

Plus clairement, si $I_{OPT}(\alpha)$ représente la luminance de l'image reconstruite à la position α avec un champ de disparité évalué lui aussi en α , $I_\gamma(\alpha)$ est toujours l'intensité de l'image reconstruite en α , mais à partir d'un champ de disparité évalué en γ . On définit alors l'erreur quadratique moyenne :

$$E_\gamma(\alpha) = \sqrt{\frac{1}{N} \sum_{\{pixels\}} (I_{OPT}(\alpha) - I_\gamma(\alpha))^2}$$

En faisant varier les deux paramètres, on peut alors générer la matrice suivante :

$$\begin{matrix} E_{\gamma_1}(\alpha_1) & E_{\gamma_1}(\alpha_2) & \cdots & E_{\gamma_1}(\alpha_{K-1}) & E_{\gamma_1}(\alpha_K) \\ E_{\gamma_2}(\alpha_1) & E_{\gamma_2}(\alpha_2) & \cdots & E_{\gamma_2}(\alpha_{K-1}) & E_{\gamma_2}(\alpha_K) \\ \vdots & & & & \vdots \end{matrix}$$

La figure 6 montre la surface des erreurs pour l'image **flower**. Le processus introduit un minimum nul sur la diagonale $\alpha = \gamma$ et déforme un peu la surface sur cette diagonale. Les images aux positions $\alpha = 0$ et $\alpha = 1$ ont été ignorées. En effet, comme à ces positions la reconstruction est parfaite (on ne fait que recopier une image d'origine), cela aurait introduit un autre biais. La valeur des erreurs a été augmentée

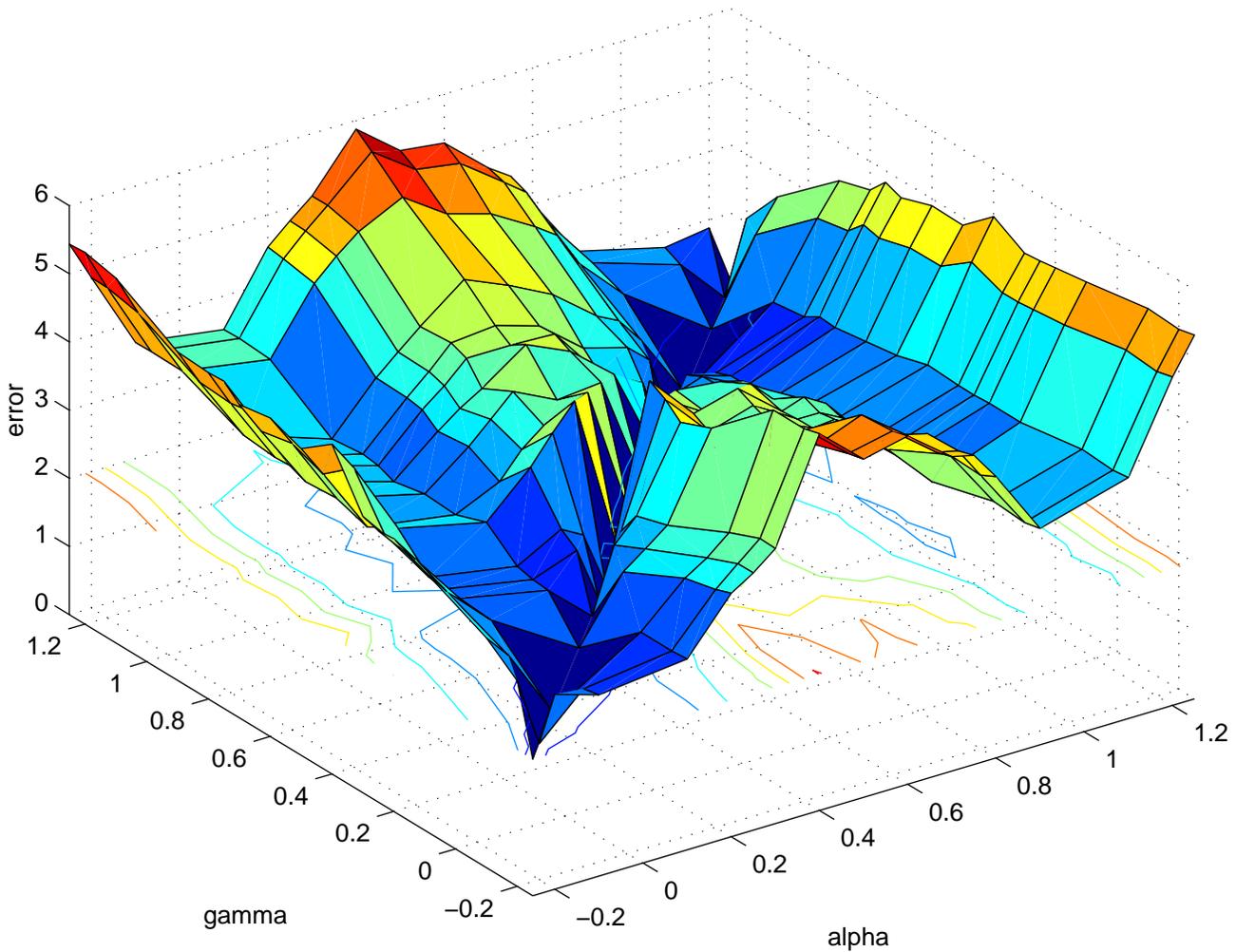


FIG. 6 – surface des erreurs dues à la propagation pour différentes positions. α est la position où l'image est reconstruite. γ est la position où le champ de disparité est calculé.

de 2 pour traduire le fait que les images optimales présentent des erreurs par rapport à ce qu'il faudrait obtenir dans l'idéal.

Sur la figure 6 on constate que pour γ constant (i.e. pour une position d'estimation de disparité fixée), on a des minimums près des positions $\alpha = 0$ et $\alpha = 1$. Ce qui est tout à fait logique étant donné que l'influence de la disparité est moindre quand la reconstruction est proche des images d'origines. On a un maximum pour une position médiane correspondant à $\alpha = 0,5$, ce qui est dû à l'influence de la disparité maximale car l'écart aux images d'origines est maximal.

A l'extérieur des images d'origines, c'est à dire pour $\alpha < 0$ et $\alpha > 1$, on constate une augmentation de l'erreur plus rapide. Cependant, les images reconstruite de façon optimale sont aussi dégradées et leur utilisation comme image de référence est moins justifiée. En effet, si on reconstruit une image pour un α très négatif, le poids des

deux images d'origine est équivalent dans la reconstruction, et une légère erreur dans la disparité introduit des différences plus importantes à l'extérieur qu'entre les deux images gauche et droite. Hors, lors du calcul de la disparité, un champ évalué pour un α négatif présente des erreurs dues à la difficulté plus importante de recherche du minimum de disparité.

Paçons nous maintenant à α constant (i.e. la position de reconstruction est fixée). On constate que l'erreur augmente avec la différence entre α et γ . En fait, plus la reconstruction se fait loin de la position où la disparité à été évaluée, plus l'erreur est grande, ce qui est logique et confirme notre choix d'image optimale. On peut cependant remarquer que cette augmentation est faible. Le fait qu'il est est une brusque discontinuité pour $\alpha = \gamma$ est due au biais introduit, mais si le zéro est ignoré, on peut estimer l'erreur de la reconstruction optimale.

5 Influence de la méthode d'interpolation

Dans toute la série d'expériences précédentes l'interpolation utilisée lors du calcul de la disparité et de la reconstruction était une interpolation bicubique de Keys [2]. Le problème avec cette interpolation est qu'elle est très coûteuse en calcul. On peut se demander si l'utilisation d'une interpolation bilinéaire, beaucoup plus rapide, détériore la qualité.

Etant donné que l'étape d'évaluation de la disparité nécessite une précision importante, l'utilisation de l'interpolation bilinéaire sera sûrement insuffisante. Confirmant cette idée, les résultats dans ce cas montrent de fortes distorsions, dues à la mauvaise évaluation des champs de disparité. Cette série d'expériences sera donc passée sous silence, et on se consacrera dans la suite à l'utilisation de l'interpolation bilinéaire pendant l'étape de reconstruction uniquement.

5.1 reconstruction d'une image

On utilise une interpolation bilinéaire dans le programme de reconstruction. Les résultats sont assez concluant. On ne constate en effet pas de différences entre une image reconstruite avec une interpolation bicubique et la même image reconstruite avec une interpolation bilinéaire.

Un flou dans les zones de transitions (comme les frontières des objets) était attendu, mais il n'a pas été perçu par le panel de testeurs à qui les images ont été montrées.

On peut donc utiliser une interpolation bilinéaire pour la reconstruction sans dégrader la qualité de l'image stéréoscopique. Ceci présente l'intérêt d'une implémentation hardware bien plus simple, et donc moins coûteuse, et d'une plus grande rapidité d'exécution.

5.2 reconstruction d'une séquence

Quand on reconstruit une séquence, on s'aperçoit que les trames impaires sont plus filtrées que les trames paires. En fait, le problème vient de la composante verticale de la disparité. Cette composante est très faible sur toute l'image (inférieure à 0.5). Cela donne des vecteurs de disparité quasi horizontaux. Pour une trame paire, le décalage verticale est nul, donc l'interpolation qui sert à calculer la position du pixel n'a pas beaucoup d'influence. Par contre pour une trame impaire, le décalage verticale est proche de 0,5 et l'interpolation influe beaucoup plus. (cf. figure 7). Le décalage verticale mentionné précédemment est introduit avant l'interpolation pour tenir compte des propriétés de l'affichage d'images entrelacées sur un écran.

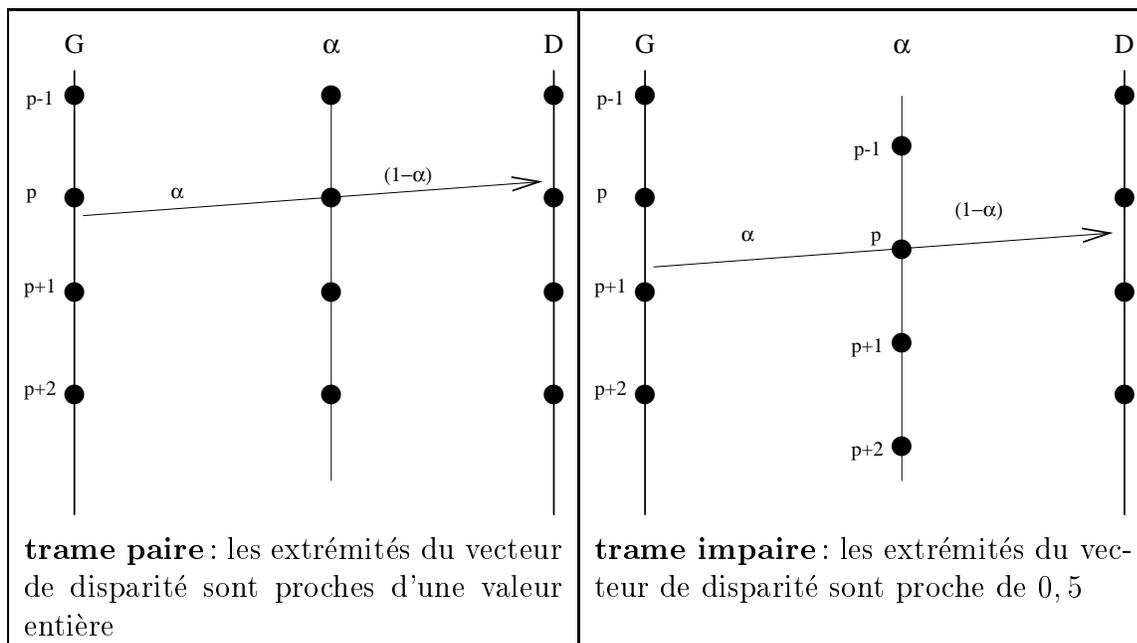


FIG. 7 – Comparaison des vecteurs de disparité dans une trame paire et dans une trame impaire.

L'interpolation effectuée pour déterminer la position des points à utiliser sur les images gauche et droite filtre l'image obtenue. Ce filtre dépend de la méthode d'interpolation et de l'écart entre la position "réelle" d'un point et la position entière la plus proche. Cet écart (noté dx) est donc inférieure à 0,5.

On effectue alors une simulation à l'aide de Matlab pour déterminer l'influence exacte de l'interpolation. Pour l'interpolation cubique de Keys [2], le filtre est de la forme :

$$y(k) = a(\alpha)x(k - \alpha) + a(\beta)x(k - \beta) + a(\gamma)x(k - \gamma) + a(\delta)x(k - \delta)$$

Les $\alpha, \beta, \gamma, \delta$ dépendent de dx . Ce filtre n'est symétrique que pour $dx = 0,5$. Pour l'interpolation linéaire, le filtre n'a que deux coefficients.

On étudie alors la réponse impulsionnelle des filtres correspondant à différentes valeurs de dx . (cf. figure 8 et 9). On observe bien sur les différentes séquences qu'on a calculées, que les trames impaires sont plus filtrées avec une interpolation bicubique qu'avec une interpolation bilinéaire.

On constate que sur les séquences reconstruites à l'aide de l'interpolation linéaire, les trames impaires sont beaucoup plus filtrées que celles obtenues par l'interpolation de Keys. Cela confirme l'influence du choix de l'interpolateur sur cette partie du processus.

Au vue de ces résultats, on comprend pourquoi le filtrage pour une trame impaire est important. On peut cependant remarquer que, quand la séquence d'image est jouée, c'est à dire que l'on fait défiler les images assez rapidement, on ne remarque pas le flou des trames impaires. Ce mouvement correspond à un déplacement de la position de l'observateur.

Un meilleur interpolateur réduirait ces effets. Il serait également utile pour estimer plus précisément la disparité. Une recherche dans ce sens a été faite. (cf. 7).

5.3 gestion des trames impaires

Le problème avec la solution évoquée ci-dessus est le coût de calcul qu'engendrerait ce nouvel interpolateur. D'autres techniques ont été essayées pour corriger ce problème sur les trames impaires des séquences. Pour ce faire, l'idée qui vient en tête est d'éviter d'avoir à réaliser une interpolation quand dx est proche de 0,5.

Plusieurs essais ont été réalisés en arrondissant les parties de vecteurs de disparités $\alpha\vec{d}$ et $(1 - \alpha)\vec{d}$, ou en arrondissant la position de l'extrémité du vecteur avant interpolation.

Dans le premier cas l'image reste floue et on ne constate pas de différence entre le flou actuel et le flou obtenu précédemment. Dans le second cas, des déformations très importantes sont introduites, rendant le résultat inacceptable.

Aucunes de ces méthodes simples ne permet donc d'éviter ce problème, et les recherches dans cette directions n'ont pas été continuées.

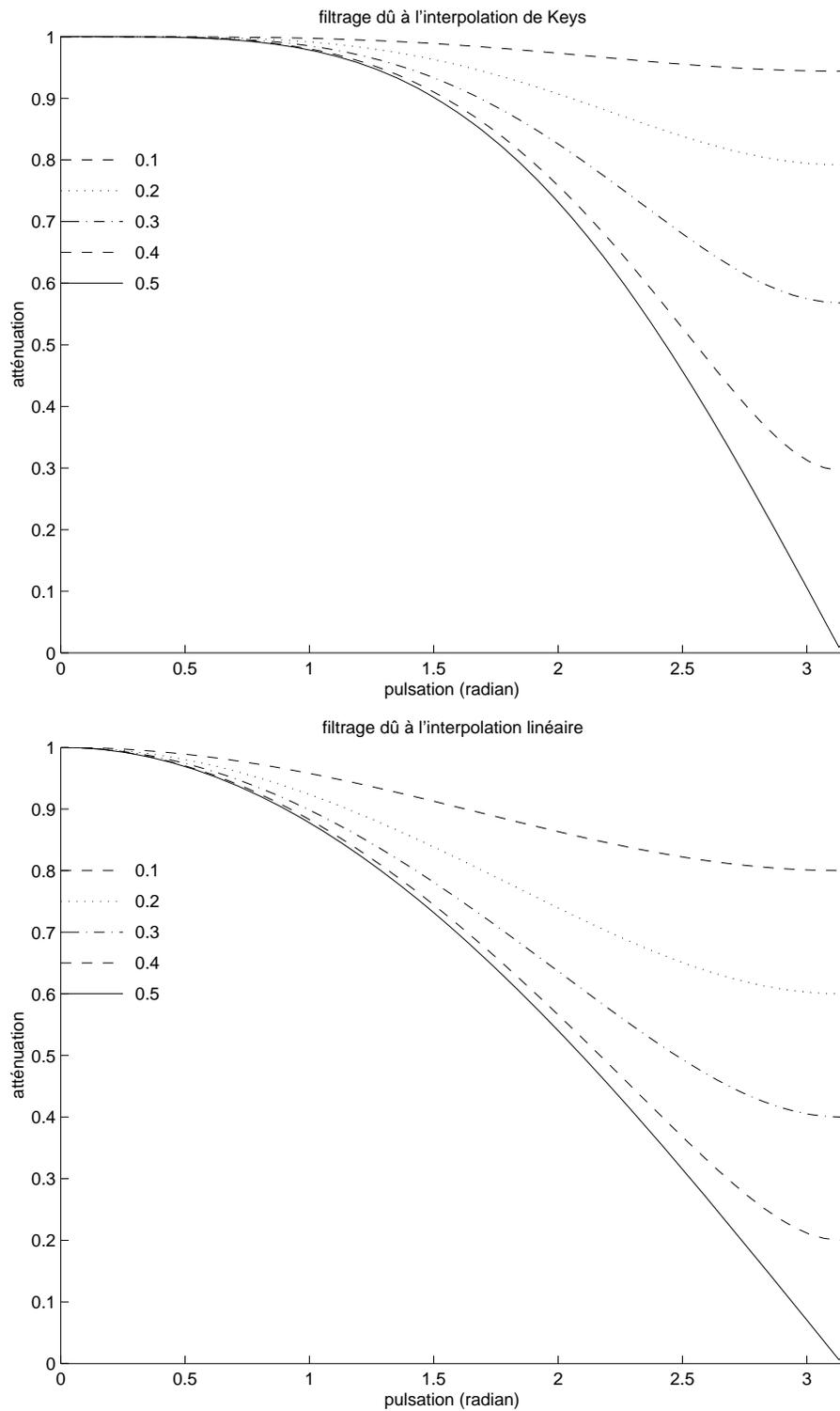


FIG. 8 – réponse en fréquence des filtres correspondant aux deux méthodes d'interpolations pour différents dx . atténuation linéaire.

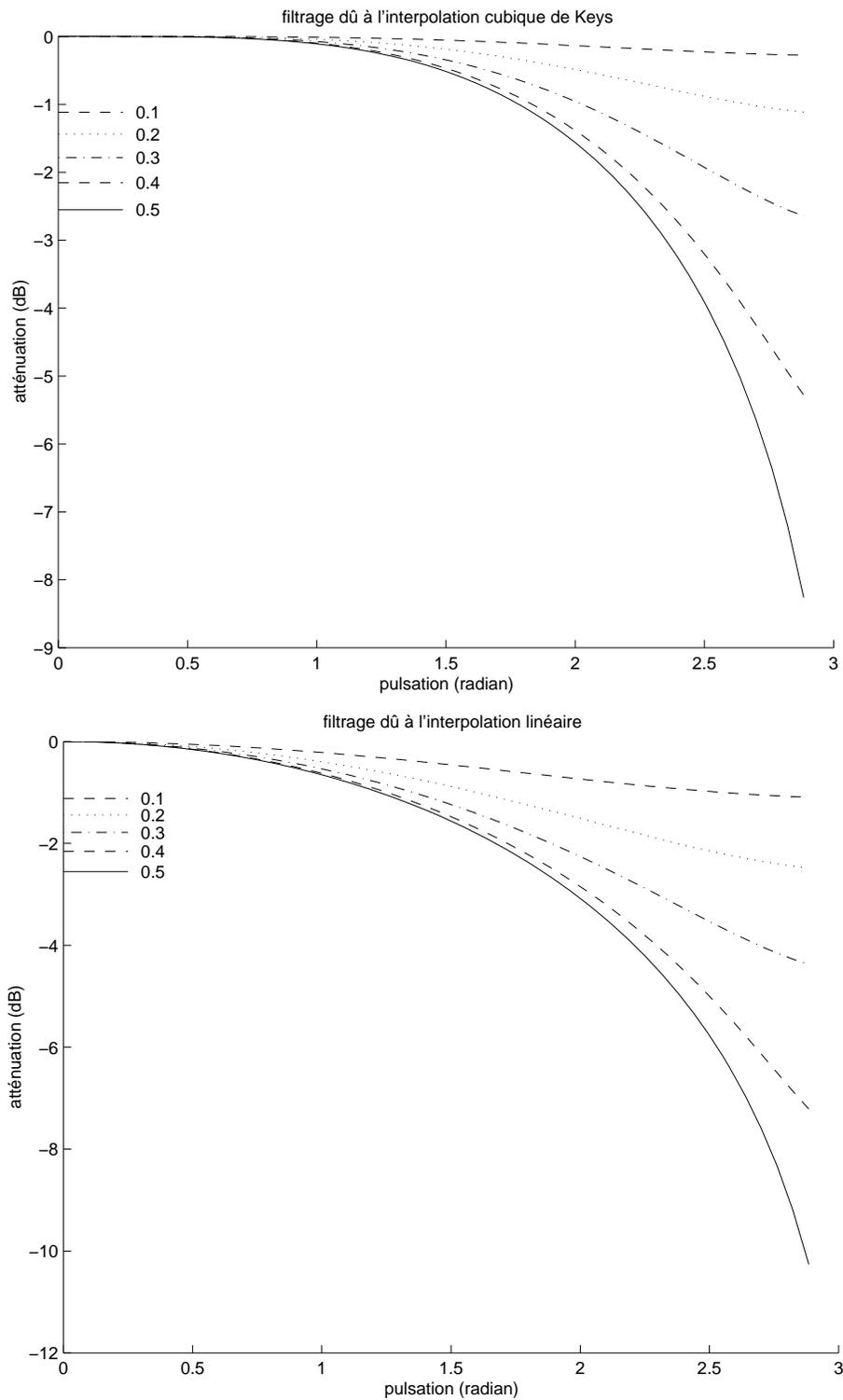


FIG. 9 – réponse en fréquence des filtres correspondant aux deux méthodes d'interpolations pour différents dx . atténuation logarithmique.

6 Influence de la précision sur la disparité

Jusqu'à présent, les champs de disparité étaient calculés à partir d'un programme C qui calculait la disparité avec une précision très importante (`float` en C). Cette précision sera toujours la même pendant le calcul de la disparité, mais le résultat final sera arrondi à diverses précisions.

La première série de tests se fera avec une précision entière, puis selon le résultat, on poursuivra l'expérience avec des précision demi-entière, quart-entière,...

La précision est définie dans le programme par une constante qui prend sa valeur dans `{ABSOLUE, INT, DEMI}` (`ABSOLUE` représente la précision "`float`"). Pour la précision entière, on arrondit à l'aide de la fonction `rint` (cf. la *man page* de `rint`). Pour une précision demi-entière (ou supérieure), on réitère le procédé en passant le double de la valeur à la fonction `arrondi`, et on divise ensuite par 2.

Les résultats sont vraiment très bon. En effet, il faut recourir à la fonction `diff` pour vérifier qu'il y a une différence entre les images ainsi obtenues, et les images obtenues par la méthode de départ. A l'œil, on ne réussit pas à voir une différence entre les deux types d'images.

Cet arrondi à l'avantage de permettre une réduction importante de la taille de stockage des disparités. Cette taille peut, en effet, devenir très importante quand on génère une séquence de disparités. Un autre avantage serait une réduction de la taille du bus de données dans le cas d'une implantation hardware.

Il est inutile de continuer plus loin avec les précision demi-entière, ... Par acquis de conscience, une série de tests a été réalisée avec une précision demi entière, et les résultats obtenus sont identiques à ceux obtenus en précision entière.

7 Interpolation B-spline

Comme abordé dans la section 5.2, la recherche d'un nouvel interpolateur a été effectué. Après consultation des articles publiés pendant la conférence ICIP 1998, une méthode d'approximation nous a semblé intéressante. Il s'agit de famille d'approximateurs qui donnent des résultats optimaux pour un support donné. (se référer à l'article citeunser pour plus de détail). Cette famille est basé sur les fonctions de Bessel.

Comme nous tenons à garder le même support que l'interpolation bicubique de Keys (c'est à dire 4×4 pixels), nous nous limiterons par la suite au fonctions cubiques.

7.1 introduction aux fonctions de Bessel

Soit la fonction $B_1(x) = 1_{[-\frac{1}{2}; \frac{1}{2}]}$, c'est à dire l'indicatrice de l'intervalle $[-\frac{1}{2}; \frac{1}{2}]$. On définit les fonctions de Bessel de degré supérieur par convolution :

$$\forall n > 1 \quad B_n = B_{n-1} * B_1 \quad (1)$$

Les fonctions de Bessel ont des propriétés intéressantes pour notre problème, en voici quelques unes :

- support fini $supp(B_n) = [-\frac{n}{2}; \frac{n}{2}]$.
- facilité du calcul de la dérivée : $B'_n(x) = B_{n-1}(x + \frac{1}{2}) - B_{n-1}(x - \frac{1}{2})$
- $\sum_j B_n(x - j) = 1$

Nous verrons par la suite que les deux fonctions de Bessel qui sont utiles pour la réalisation de l'estimateur sont B_4 et B_2 dont voici les expressions :

$$B_2(x) = \begin{cases} 1+x & \text{si } x \in [-1; 0] \\ 1-x & \text{si } x \in [0; 1] \\ 0 & \text{sinon} \end{cases}$$

et

$$B_4(x) = \begin{cases} \frac{1}{6}x^3 + x^2 + 2x + \frac{4}{3} & \text{si } x \in [-2; -1] \\ -\frac{1}{2}x^3 - x^2 + \frac{2}{3} & \text{si } x \in [-1; -0] \\ \frac{1}{2}x^3 - x^2 + \frac{2}{3} & \text{si } x \in [0; 1] \\ -\frac{1}{6}x^3 + x^2 - 2x + \frac{4}{3} & \text{si } x \in [1; 2] \\ 0 & \text{sinon} \end{cases}$$

7.2 estimateur

Appliquons maintenant les résultats précédents et les principes décrits dans [3]. L'article montre que l'approximation optimal d'une fonction f est réalisée par

$$f_{approx}(x) = \sum_k c(k) \phi_L\left(\frac{x}{T} - k\right) \quad (2)$$

où T est la période d'échantillonnage (1 dans notre cas) et ϕ est une composition linéaire de fonctions de Bessel et de ses dérivées.

$$\phi_L(x) = \sum_{k=0}^{L-1} p_k \frac{d^k}{dx^k} B_L(x) \quad (3)$$

Avec $L = 4$ dans notre cas, on obtient

$$\phi_4(x) = B_4 + \frac{1}{42} B_4'' \quad (4)$$

or la propriété de la dérivations des fonctions de Bessel permet d'obtenir :

$$B_4''(x) = B_2(x+1) + B_2(x-1) - 2B_2(x) \quad (5)$$

Le calcul des coefficient $c(k)$ est décrit dans [4] dans le cas d'une interpolation à l'aide de B-spline cubique. Ce calcul a été étendu à notre fonction optimale. On obtient :

$$f_{approx}(x) = \sum_{k=-\infty}^{+\infty} f(x) \eta(x-k) \quad (6)$$

avec

$$\eta(x) = \frac{5,25\alpha}{1-\alpha^2} \sum_{k=-\infty}^{+\infty} \alpha^{|k|} \phi_4(x-k) \quad (7)$$

où α est le pôle du filtre inverse et vaut $\frac{\sqrt{105}-13}{8}$. La somme décrite dans l'équation 7 est en réalité une somme finie due au fait que le support de la fonction ϕ est lui même fini. De plus, comme la valeur de ce pôle est très petite devant 1, la convergence de la somme infinie de l'équation 6 est extrêmement rapide. On peut donc se limiter à un k maximum relativement faible. Pour les tests suivants, ce maximum a été fixé arbitrairement à 4.

Comme il a été démontré (cf. [5]) que dans un espace de dimension supérieure les fonctions de Bessel et toutes les fonctions utilisés dans les différentes équations précédentes sont à variables séparables, on pourra facilement combiner cet estimateur mono-dimensionnel sur les deux axes des images.

7.3 résultats

L'implémentation de cet approximateur donne des résultats conformes aux attentes. Le temps de calcul est beaucoup plus long que pour l'interpolateur bicubique de Keys, mais la qualité de l'estimation est un peu meilleure. La figure 10 démontre que le filtrage est moins important que celui dû à l'interpolateur de Keys.

En se plaçant dans le cas du filtrage maximale (i.e. $dx = 0,5$), on peut comparer les différentes atténuations pour le filtrage linéaire, bicubique de Keys, spline cubique et spline cubique optimal, comme sur la figure 11.

Malgré ces bons résultats, cet estimateur a deux gros problèmes :

- Le temps de calcul est beaucoup plus important que pour l'interpolation bicubique de Keys (de l'ordre de 10 fois plus long). Même si l'implémentation réalisée est loin d'être optimale, ce temps de calcul pourra être difficilement réduit.
- La fonction n'est pas un interpolateur, mais un estimateur. c'est à dire que les valeurs des pixels connus peuvent être modifiées.

la figure 12 montre le résultat d'une partie de l'image flower agrandie 4 fois par les interpolateur bilinéaire, bicubique de Keys et bidimensionnel B-spline optimal.

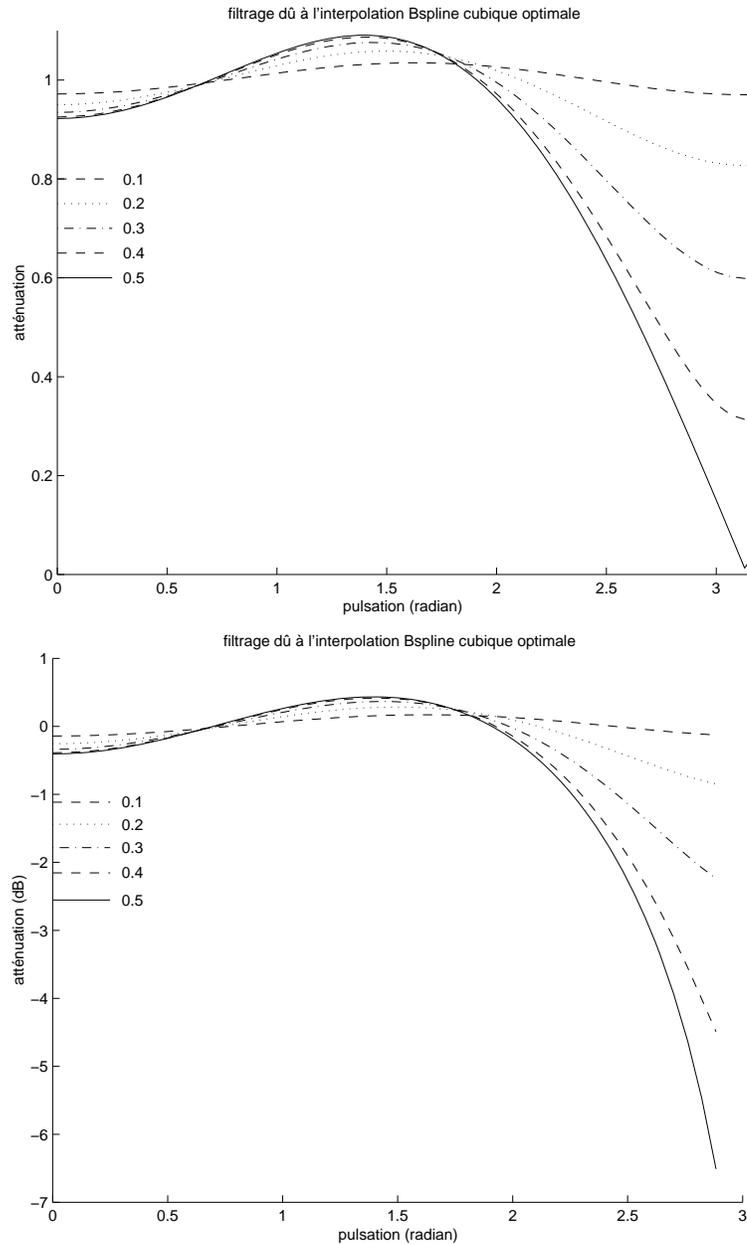


FIG. 10 – réponse en fréquence du filtre correspondant à l'interpolation B-spline optimal pour différents dx . atténuation linéaire et log.

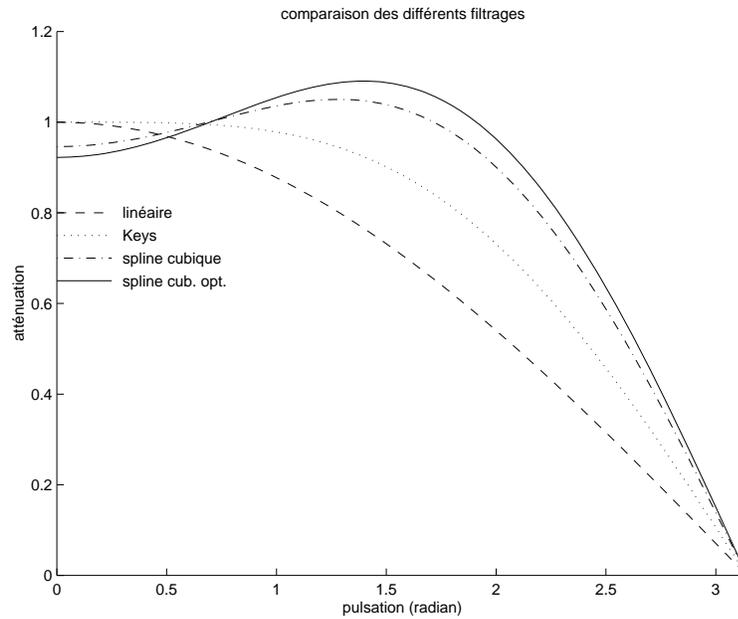


FIG. 11 – atténuations comparées des différentes méthodes d'interpolation.



bilinéaire

bicubique

B-spline

FIG. 12 – Agrandissement par 4 d'un détail de flower par les 3 interpolateurs.

8 Conclusion

Il a donc été montré que l'utilisation d'une image d'origine et d'une seule image reconstruite dans une paire stéréoscopique était tout à fait bénéfique. Cela améliore la qualité de l'image et peut permettre une réduction du nombre d'images à calculer.

Le fait d'utiliser le même champ de disparité (calculé en $\alpha = 0,5$ par exemple) apporte lui aussi un gain de temps potentiel. La qualité de l'image étant tout à fait correcte, cela permet de ne faire qu'une fois le calcul de la disparité, qui est de loin l'étape la plus longue du processus. Il a été prouvé que la propagation de la disparité donne de très bons résultats pour des images reconstruites entre les images gauche et droite. Pour une reconstruction à l'extérieur des images d'origines (i.e. $\alpha < 0$ et $\alpha > 1$), on peut même obtenir une meilleure qualité en propageant la disparité qu'en la recalculant à chaque position.

L'arrondi de la disparité une fois le calcul effectué est lui aussi sans retombées sur l'image obtenue. Cela permettrait de coder les valeurs de la disparité par des entiers, d'où un gain de place non négligeable pour le stockage et/ou la transmission des champs de disparité. On peut aussi penser à l'intérêt de réduire la précision à une valeur entière lors d'une implantation hardware (taille des bus de données).

En ce qui concerne le mode d'interpolation, l'étape de reconstruction peut tout à fait se faire avec une interpolation bilinéaire, car on ne constate aucune perte de qualité. Par contre il semble que l'interpolation bilinéaire ne soit malheureusement pas assez précise pour l'estimation de la disparité. Il faut donc rester avec une interpolation bicubique, coûteuse en calcul. On peut cependant remarquer que si on souhaite réaliser un système de qualité moyenne à prix peu élevé, l'utilisation d'une interpolation bilinéaire pour le calcul de la disparité est acceptable. Par contre il n'y a pas de solution pratique au problème du flou des trames impaires dans la création d'une séquence d'image.

L'estimateur B-spline donne de bons résultats mais le coût en terme de calcul est vraiment trop important. Pour l'instant son utilisation dans des algorithmes comme l'estimation de disparité ou de mouvement est trop coûteuse. Mais dans d'autres petites applications comme l'agrandissement d'images, l'emploi de cet estimateur peut s'avérer efficace.

A Annexe – Utilitaires

Quelques programmes permettant la manipulation des images ViDS ont également été réalisés. Ces programmes sont disponibles dans le répertoire d'exécutables du groupe *VisCom*, et les *pages man* sont toutes disponibles. Ces programmes sont :

Prmd2Dseq : Crée une pyramide de séquences d'images filtrés à partir d'une séquence donnée et d'un fichier décrivant les filtres à utiliser sur chaque composante.

Extinsimg : Extrait à partir d'une séquence une fenêtre tri-dimensionnelle (x, y, t). Le résultat est soit placé dans un nouveau fichier, soit inséré dans une séquence existante.

Comblinimg : Exécute la combinaison linéaire de deux séquences. Ce programme est notamment utile pour effectuer une différence entre deux séquences.

Ces programmes ne sont utilisables que sur les stations SUN.

Des fonctions **C** ont également été ajoutées à la bibliothèque *viscom*. (compilation avec l'option `-lVisCom`). il s'agit de :

Flt2Dupdn : Cette routine filtre une image et effectue un sur- ou sous-échantillonnage.

Prmd2Dimg : Fonction de base de *Prmd2Dseq*, cette routine retourne une pyramide pour une image.

Les *pages man* sont elles aussi disponibles.

Notons cependant que le programme *Prmd2Dseq* donne des résultats différents du script *CASCADE* utilisé sur les stations DEC. Même si l'écart entre les images obtenues par les deux programmes n'est jamais supérieur à une valeur de 1 sur une dynamique de 256 niveaux, les résultats obtenus après estimations de la disparité et reconstruction d'images intermédiaires peuvent présenter des différences très visibles.

Références

- [1] Anthony Mancini, *Disparity estimation and intermediate view reconstruction for novel applications in stereoscopic video*, Master's thesis, McGill University, Dept. Electr. Eng., Feb. 1998
- [2] R.G. Keys, *Cubic convolution interpolation for digital image processing*, IEEE-T-ASSP, pages 1153–1160, volume ASSP-29, number 6, Dec. 1981
- [3] Thierry Blu, Philippe Thévenaz and Michael Unser, *Minimum support interpolators with optimum approximation properties*, pp 242–245, vol. III, ICIP '98

- [4] Michael Unser, Akram Aldroubi and Murray Eden, *Fast B-Spline transforms for continuous image representation and interpolation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 3, pp. 277-285, March 1991.
- [5] Michael Unser, Akram Aldroubi and Murray Eden, *B-Spline signal processing: part II – efficient design an applications* IEEE Transactions on signal processing, vol. 41, no. 2, pp. 834-848, February 1993.