

Université du Québec
INRS-Télécommunications

**RECONSTRUCTION D'IMAGES INTERMÉDIAIRES À PARTIR
D'UNE PAIRE D'IMAGES STÉRÉOSCOPIQUES**

Par
Vincent Finnerty

Mémoire présenté pour l'obtention du grade de Maître ès sciences (M.Sc.) en
Télécommunications

Jury d'évaluation

Résumé

Le but de cette étude est de développer une méthode robuste d'estimation de disparité et d'appliquer cette méthode pour reconstruire des images intermédiaires entre deux images stéréoscopiques.

La méthode développée consiste en ceci : Une méthode de correction locale de luminosité est implantée pour corriger les déséquilibres de luminance d'une paire d'image stéréoscopique afin de réduire les distorsions. Un estimateur de disparité de type relaxation non linéaire robuste, qui permet des discontinuités dans le champ de disparité, est ensuite utilisé pour évaluer le déplacement des points entre les deux images stéréoscopiques. Pour déterminer la valeur d'une image et de ses dérivés spatiales entre les pixels l'interpolation bicubique de Keys est implantés. Finalement les méthodes de reconstruction d'images intermédiaires sont utilisés pour la reconstruction à l'aide du champ de disparité.

Afin de comparer nos résultats, deux estimateurs de disparité classiques, soit March, Horns & Schunck, sont adaptés. L'efficacité des trois estimateurs de disparité est évaluée qualitativement et quantitativement à l'aide la séquence d'images de mouvement synthétique *Birds*, la paire d'image stéréo *Flower* et la séquence d'images *FlowerGarden*.

Table des matières

Résumé	i
Table des matières	iii
Liste des tableaux	v
Table des figures	vii
1 Introduction	1
2 Vidéo Stéréoscopique : De la scène à l'oeil	5
2.1 Lumières, Caméras, Action!	5
2.1.1 Caméras Parallèles	5
2.1.2 Caméras Convergentes	6
2.2 Affichage Stéréoscopiques	8
3 Estimation de Disparité	11
3.1 Définitions	11
3.2 La Base des Estimateurs de Disparité	11
3.2.1 Appariement de Blocs	12
3.2.2 Flot Optique	15
3.2.3 Approche Hiérarchique	17
3.3 Améliorations ou nouvelles approches	20
3.3.1 Flot optique lisse par morceau	20
3.3.2 Évaluation de mouvement avec la détection des zones d'occlusion	20
3.3.3 Estimation cohérente de disparité et de mouvement	21
3.3.4 Amélioration de l'appariement de blocs	22
3.3.5 Appariement de blocs avec dimension de fenêtre variable	23
4 Algorithmes utilisés	25
4.1 Interpolation sur une grille d'échantillonnage	25
4.1.1 Interpolation linéaire	26
4.1.2 Interpolation cubique de Keys	27

4.1.3	Interpolation sur une grille	30
4.2	Correction de luminosité entre deux images	32
4.2.1	Correction globale de luminosité	32
4.2.2	Correction locale de luminosité (par section)	32
4.2.3	Correction bilinéaire de luminosité	33
4.3	Estimation de Disparité	35
4.3.1	Flot optique de March	35
4.3.2	Flot optique de Horn et Schunck	39
4.3.3	Flot optique robuste	42
4.4	Reconstruction d'images intermédiaires	48
4.4.1	Reconstruction exacte	48
4.4.2	Reconstruction par propagation	49
4.4.3	Reconstruction non-uniforme	49
5	Présentation des résultats	53
5.1	La séquence d'images <i>Birds</i>	53
5.1.1	Comparaison du champ de vecteurs	56
5.1.2	Reconstruction d'une image existante	57
5.2	La séquence d'images <i>Flower</i>	62
5.2.1	Reconstruction d'une image existante	62
5.2.2	Reconstruction d'images intermédiaires	67
5.3	La séquence d'images <i>Flower Corrigée</i>	70
5.3.1	Reconstruction d'une image existante	70
5.3.2	Reconstruction d'images intermédiaires	75
5.4	La séquence d'images <i>FlowerGarden</i>	78
5.4.1	Reconstruction d'une image existante	78
5.4.2	Reconstruction d'images intermédiaires	83
6	Conclusion	87
6.1	Apport de la relaxation convexe	87
6.2	Apports de la correction bilinéaire de luminosité	92
6.3	Apports de la relaxation robuste	95
	Bibliographie	98

Liste des tableaux

5.1	Comparaison quantitative des champs de déplacement de <i>Birds</i>	56
5.2	Comparaison quantitative des reconstructions de <i>Birds</i> droite. . . .	58
5.3	Comparaison quantitative des reconstructions de <i>Flower</i> droite. . . .	64
5.4	Comparaison quantitative des reconstructions de <i>Flower</i> droite et <i>Flower</i> droite corrigée.	72
5.5	Comparaison quantitative des reconstructions de <i>FlowerGarden</i> droite. . . .	82
5.6	Comparaison quantitative des reconstructions de <i>FlowerGarden</i> centre. . . .	85

Table des figures

1.1	Interprétation d'un parallaxe positif.	1
1.2	Interprétation d'un parallaxe nul.	2
1.3	Interprétation d'un parallaxe négatif.	2
1.4	Exemple d'un parallaxe excessif.	3
2.1	Configuration de Caméras Parallèles.	6
2.2	Configuration de Caméras Convergentes.	7
2.3	Le stéréoscope.	9
3.1	Exemples de Champs d'Occlusion.	12
3.2	Exemple d'une Procédure d'Appariement de Blocs.	13
3.3	Exemple d'une Recherche Logarithmique Bidimensionnelle	15
3.4	Exemple d'une approche hiérarchique.	18
4.1	Noyau de convolution linéaire et sa dérivé	27
4.2	Noyau de convolution de Keys et sa dérivé	30
4.3	Séparation de l'image I_1 en quadrant.	33
4.4	Quadratique Tronquée en $x=\pm 2$	43
4.5	Comparaison entre la fonction Geman-Mclure et la fonction quadra- tique tronquée	46
4.6	Comparaison des différentes puissances de fonctions Geman-Mclure gé- néralisées	47
4.7	Reconstruction exacte.	48
4.8	Reconstruction par propagation.	50
4.9	Reconstruction non-uniforme.	51
4.10	triangulation de Denaulay.	52
5.1	Séquence d'images <i>Birds</i>	54
5.2	Séquence d'images <i>Birds</i> avec le rectangle de translation.	55
5.3	Composante horizontale du champ de vecteur idéal.	56
5.4	Composante horizontale du champ de vecteur calculé avec la méthode de March.	57
5.5	Composante horizontale du champ de vecteur calculé avec la méthode de Horn & Schunck.	58

5.6	Composante horizontale du champ de vecteur calculé avec la relaxation convexe.	59
5.7	Composante horizontale du champ de vecteur calculé avec la relaxation robuste.	59
5.8	Erreurs dans la reconstruction de <i>Birds</i> pour l'approche de a)March b)Horn & Schunck a)la relaxation convexe b)la relaxation robuste. . .	60
5.9	Reconstruction de <i>Birds</i> droite à l'aide de la relaxation robuste.	61
5.10	Image stéréoscopique <i>Flower</i> gauche.	62
5.11	Image stéréoscopique <i>Flower</i> droite.	63
5.12	Zones de références de <i>Flower</i>	63
5.13	Erreurs dans la reconstruction de <i>Flower</i> droite pour l'approche de a)March b)Horn & Schunck c)la relaxation convexe d)la relaxation robuste. (Blanc=-128 Noir=128)	65
5.14	Reconstruction de <i>Flower</i> droite à l'aide de la relaxation robuste.	66
5.15	Reconstruction d'une vue intermédiaire de <i>Flower</i> à l'aide de l'approche de March.	67
5.16	Reconstruction d'une vue intermédiaire de <i>Flower</i> à l'aide de l'approche de Horn.	68
5.17	Reconstruction d'une vue intermédiaire de <i>Flower</i> à l'aide de l'approche de la relaxation convexe.	69
5.18	Reconstruction d'une vue intermédiaire de <i>Flower</i> à l'aide de l'approche de la relaxation robuste.	69
5.19	Correction de luminosité de l'image <i>Flower</i> droite à l'aide de la correction par section.	70
5.20	Correction de luminosité de l'image <i>Flower</i> droite à l'aide de la correction bilinéaire.	71
5.21	Erreurs dans la reconstruction de <i>Flower</i> droite corrigée pour l'approche de a)March b)Horn & Schunck c)la relaxation convexe d)la relaxation robuste. (Blanc=-221 Noir=221)	73
5.22	Reconstruction à l'aide de l'approche de la relaxation robuste de <i>Flower</i> droite corrigée.	74
5.23	Reconstruction d'une vue intermédiaire de <i>Flower</i> à l'aide de l'approche de March et de la correction de luminosité.	75
5.24	Reconstruction d'une vue intermédiaire de <i>Flower</i> à l'aide de l'approche de Horn & Schunck et de la correction de luminosité.	76
5.25	Reconstruction d'une vue intermédiaire de <i>Flower</i> à l'aide de l'approche de la relaxation convexe et de la correction de luminosité.	77
5.26	Reconstruction d'une vue intermédiaire de <i>Flower</i> à l'aide de l'approche de la relaxation robuste et de la correction de luminosité.	77
5.27	Image <i>FlowerGarden</i> gauche.	78
5.28	Image <i>FlowerGarden</i> centre.	79
5.29	Image <i>FlowerGarden</i> droite.	79

5.30	Zones nouvellement exposées <i>FlowerGarden</i> droite.	80
5.31	Erreurs dans la reconstruction de <i>FlowerGarden</i> droite corrigée pour l'approche de a)March b)Horn & Schunck c)la relaxation convexe d)la relaxation robuste. (Blanc=-128 Noir=128)	81
5.32	Reconstruction de <i>FlowerGarden</i> droite à l'aide de la relaxation robuste.	82
5.33	Zones d'occlusions de <i>FlowerGarden</i> centre.	83
5.34	Erreurs dans la reconstruction de <i>FlowerGarden</i> droite corrigée pour l'approche de a)March b)Horn & Schunck c)la relaxation convexe d)la relaxation robuste. (Blanc=-128 Noir=128)	84
5.35	Reconstruction de <i>FlowerGarden</i> centre à l'aide de la relaxation convexe.	85
6.1	Image Test gauche.	89
6.2	Image Test droite.	90
6.3	Reconstruction de l'image test droite à l'aide de l'approche de March. .	90
6.4	Reconstruction de l'image test droite à l'aide de la relaxation convexe.	91
6.5	Reconstruction de l'image test intermédiaire à l'aide de l'approche de March.	91
6.6	Image <i>FlowerCoin</i> gauche.	92
6.7	Image <i>FlowerCoin</i> droite.	92
6.8	Reconstruction de <i>FlowerCoin</i> droite à l'aide de la relaxation convexe.	92
6.9	Image <i>FlowerCoin</i> gauche corrigée	93
6.10	Image <i>FlowerCoin</i> droite	93
6.11	Reconstruction de la <i>FlowerCoin</i> droite à l'aide de la relaxation convexe et de la correction de luminosité	93
6.12	Composante horizontale du champ de disparité de <i>Birds</i> calculé avec la relaxation robuste à $\alpha = 0.5$	96

Chapitre 1

Introduction

Dans le domaine du divertissement vidéo, on tente sans cesse d'augmenter le réalisme. Qu'il s'agisse de la vidéo couleur ou de la récente télévision haute définition, tout est développé dans le but d'améliorer l'expérience des usagers. La vidéo stéréoscopique promet encore de plus grand réalisme. Une vidéo stéréoscopique est réalisée par de deux caméras filmant la scène à partir de deux points de vues légèrement distants. Ces deux vues sont alors présentées à l'utilisateur de manière à ce que l'image de la caméra gauche soit vue uniquement par l'oeil gauche, et l'image de la caméra droite par l'oeil droit pour donner un effet de relief.

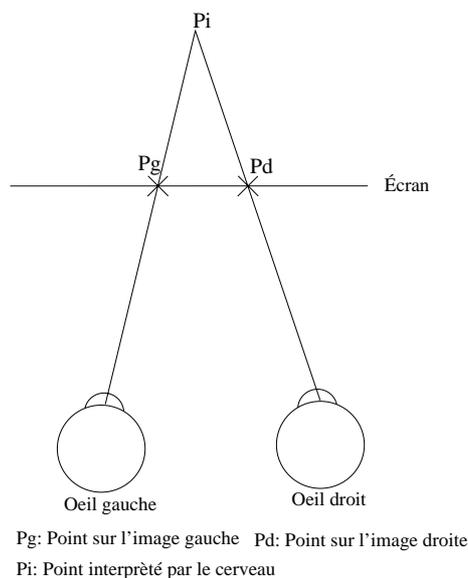


FIG. 1.1 – Interprétation d'un parallaxe positif.

Cet effet de relief est produit par la parallaxe. La parallaxe est le déplacement entre les deux projections d'un même point physique. Les figures 1.1, 1.2 et 1.3 illustrent, respectivement, une parallaxe positive, nulle et négative. Habituellement, les caméras en vidéo stéréoscopique sont ajustées pour que la parallaxe correspondant à un point à

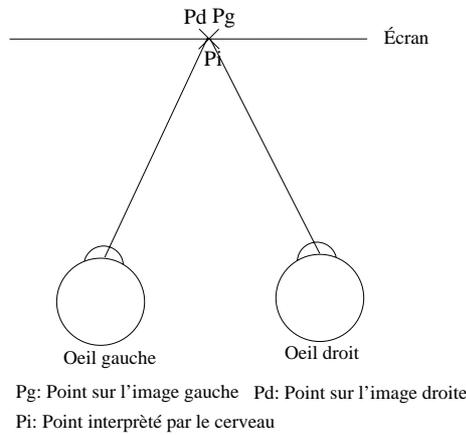


FIG. 1.2 – Interprétation d'un parallaxe nul.

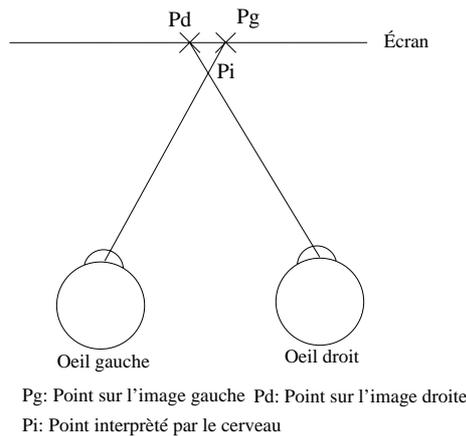


FIG. 1.3 – Interprétation d'un parallaxe négatif.

l'horizon soit de 65mm, c'est-à-dire la distance moyenne entre les pupilles d'un adulte. Lorsqu'un jeune enfant ayant une distance entre les pupilles inférieure à 65mm regarde cette horizon, ses yeux devront prendre une position divergente qui n'est pas naturelle et très inconfortable (voir figure 1.4).

Pour pouvoir ajuster la parallaxe, nous devons avoir des vues intermédiaires à celles prises par les caméras stéréoscopiques. Toutefois, la dimension des caméras et la distance qui les sépare ne permettent pas de caméras intermédiaires. Ainsi, la création automatique d'images intermédiaires entre deux images stéréoscopique s'impose.

Le but de cette étude est de développer une méthode robuste d'estimation de disparité et d'appliquer cette méthode pour reconstruire des images intermédiaires entre deux images stéréoscopiques.

La méthode développée consiste en ceci : Une méthode de correction locale de luminosité est implanté pour corriger les déséquilibres de luminance d'une paire d'image stéréoscopique afin de réduire les distorsions. Un estimateur de disparité de type relaxation non linéaire robuste, qui permet des discontinuités dans le champ de dis-

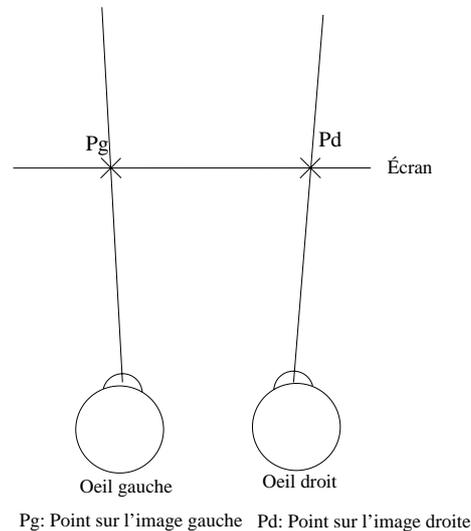


FIG. 1.4 – Exemple d'un parallaxe excessif.

parité, est ensuite utilisé pour évaluer le déplacement des points entre les deux images stéréoscopiques. Pour déterminer la valeur d'une image et de ses dérivés spatiales entre les pixels l'interpolation bicubique de Keys est implantés. Finalement les méthodes de reconstruction d'images intermédiaires sont utilisés pour la reconstruction à l'aide du champ de disparité.

Afin de comparer nos résultats, deux estimateurs de disparité classiques, soit March, Horns & Schunck, sont adaptés. L'efficacité des trois estimateurs de disparité est évaluée qualitativement et quantitativement à l'aide la séquence d'images de mouvement synthétique *Birds*, la paire d'image stéréo *Flower* et la séquence d'images *FlowerGarden*.

Ce mémoire est organisé de la façon suivante : le Chapitre 2 est consacré à la description des différentes méthodes de capture et d'affichage. Le Chapitre 3 traite de différentes méthodes d'estimation de disparité. Le Chapitre 4 comporte une description des différentes méthodes mathématiques que nous avons utilisées pour reconstruire des vues intermédiaires. Les Chapitres 5 et ?? présentent et interprètent les résultats obtenus. Le Chapitre 6 contient la conclusion.

Chapitre 2

Vidéo Stéréoscopique : De la scène à l'oeil

2.1 Lumières, Caméras, Action !

Deux caméras sont utilisés pour la saisie stéréoscopique d'une scène. La distance entre les caméras correspond généralement à la distance entre nos yeux (environ 65 millimètres). Même si la modification de la distance entre les caméras est à la base de certains effets visuels spéciaux, elle n'influencera pas l'estimation de disparité nécessaire à la reconstruction d'images intermédiaires.

2.1.1 Caméras Parallèles

Cette configuration exige deux caméras identiques avec des axes optiques parallèles (la ligne qui passe par le point focal et le centre des capteurs dans la caméra). Soit B la distance entre le centre de la caméra gauche et celui de la caméra droite et soit f la distance focale (nous la supposons égale pour les deux caméras). Définissons un repère dont l'origine est à mi-distance entre les centres du capteur gauche $(-B/2, 0, 0)$ et du capteur droit $(B/2, 0, 0)$ (voir la figure 2.1). Si X représente la position latérale d'un point physique, Y sa hauteur et Z sa profondeur, un point situé à (X, Y, Z) est projeté sur le capteur gauche à la position (x_l, y_l) . Puisque la ligne passant par (X, Y, Z) et $(x_l, y_l, 0)$ traversera également $(-B/2, 0, f)$ (la position du point focal de gauche), nous avons :

$$\begin{pmatrix} x_l \\ y_l \end{pmatrix} = \begin{pmatrix} f \frac{X+B/2}{f-Z} \\ f \frac{Y}{f-Z} \end{pmatrix} \quad (2.1.1)$$

De la même façon, nous avons :

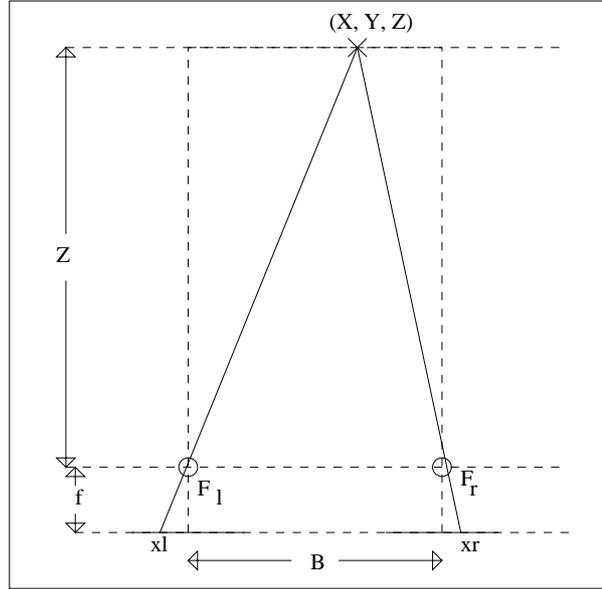


FIG. 2.1 – Configuration de Caméras Parallèles.

$$\begin{pmatrix} x_r \\ y_r \end{pmatrix} = \begin{pmatrix} f \frac{X - B/2}{f - Z} \\ f \frac{Y}{f - Z} \end{pmatrix} \quad (2.1.2)$$

Le déplacement horizontal entre les deux projections du point (X, Y, Z) est :

$$d_x = x_r - x_l = f \frac{X - B/2}{f - Z} - f \frac{X + B/2}{f - Z} = f \frac{-B}{f - Z} \quad (2.1.3)$$

Dans la configuration de caméras parallèles, le déplacement horizontal dépend seulement de la profondeur du point physique. Le déplacement vertical est nul :

$$d_y = y_r - y_l = 0 \quad (2.1.4)$$

Que d_y soit nul confère un avantage pour l'estimation de disparité, il réduit le nombre de variable à estimer. Le déplacement horizontal est positif et il est nul si $Z = \infty$. De cette façon, quand la scène est affichée, l'horizon ($Z = \infty$) semblera être sur l'écran et les objets, apparaîtront plus près que l'écran.

2.1.2 Caméras Convergentes

Cette configuration exige deux caméras identiques. Au lieu d'être parallèles, les axes optiques convergent au point $(0, 0, Z_{conv})$ (voir la figure 2.2). Si β est l'angle entre les axes optiques, on a :

$$Z_{conv} = \frac{B}{2 \tan(\beta)} \quad (2.1.5)$$

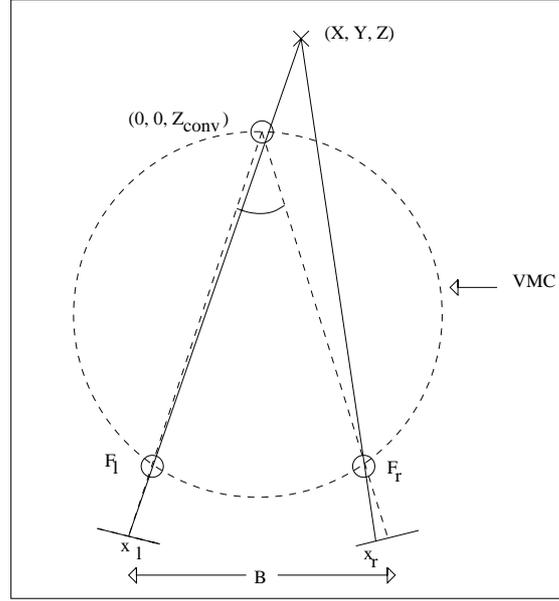


FIG. 2.2 – Configuration de Caméras Convergentes.

Ceci donne [7] :

$$x_l = f \frac{(X + B/2) \cos(\beta) - Z \sin(\beta)}{f - Z \cos(\beta) - (X + B/2) \sin(\beta)} \quad (2.1.6)$$

$$y_l = f \frac{Y}{f - Z \cos(\beta) - (X + B/2) \sin(\beta)} \quad (2.1.7)$$

$$x_r = f \frac{(X - B/2) \cos(\beta) + Z \sin(\beta)}{f - Z \cos(\beta) + (X - B/2) \sin(\beta)} \quad (2.1.8)$$

$$y_r = f \frac{Y}{f - Z \cos(\beta) + (X - B/2) \sin(\beta)} \quad (2.1.9)$$

Dans ces équations, f ne représente plus la composante Z du point focal puisque les axes optiques sont maintenant convergents. Pour $\beta = 0$ nous avons le cas des caméras parallèles. Le déplacement horizontal est maintenant :

$$d_x = -2 \frac{N_x(X, Z)}{D_x(X, Z)} \quad (2.1.10)$$

où :

$$\begin{aligned} N_x(X, Z) = & 4Z^2 \sin(\beta) \cos(\beta) + Z[2B - 4f \sin(\beta) - 4B \cos^2(\beta)] \\ & + B \cos(\beta)[2f - B \sin(\beta)] - 4X^2 \sin^2(\beta) \end{aligned} \quad (2.1.11)$$

et :

$$D_x(X, Z) = 4Z^2 \cos^2(\beta) + 4Z \cos(\beta)[2B - af \sin(\beta) - 4B \cos^2(\beta)] \\ + [4f^2 - 4fB \sin(\beta) + b^2 \sin^2(\beta)] - 4X^2 \sin^2(\beta) \quad (2.1.12)$$

Les points pour lesquels $d_x=0$ sont sur un cercle de centre :

$$\begin{pmatrix} X_c \\ Z_c \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{B \cos(2\beta) + 2f \sin(\beta)}{2 \sin(2\beta)} \end{pmatrix} \quad (2.1.13)$$

et de rayon :

$$r_c = \frac{2 - 2f \sin(\beta)}{2 \sin(2\beta)} \quad (2.1.14)$$

Ce cercle s'appelle le cercle de Vieth-Müller. C'est un cercle qui passe par l'intersection des axes optiques et le point focal de chaque caméra. Lorsque ($Z \rightarrow \infty$), on a :

$$\lim_{Z \rightarrow \infty} d_x = -2f \tan(\beta) \quad (2.1.15)$$

Cette équation et les caractéristiques de l'écran de visualisation, permettent d'ajuster l'angle des caméras pour créer un déplacement horizontal à l'infini, égal à la distance entre les yeux.

Le déplacement vertical est :

$$d_y = y_r - y_l = -8f \frac{XY \sin(\beta)}{D_y(X, Y, Z)} \quad (2.1.16)$$

Où :

$$D_y(X, Y, Z) = 4f\{f - B \sin(\beta) - 2Z \cos(\beta)\} + \sin^2(\beta)\{\beta^2 - 4X^2\} \\ + 2BZ \sin(2\beta) + 4Z^2 \cos^2(\beta) \quad (2.1.17)$$

2.2 Affichage Stéréoscopiques

On décrit ici le fonctionnement général des affichages stéréoscopiques les plus utilisés actuellement ([14]) : stéréoscope, anaglyphes, autostéréoscope, polarisation et multiplexage.

Le premier affichage stéréoscopique à être inventé fut le stéréoscope. Charles Wheatstone a proposé en 1832 ce qui était probablement le premier stéréoscope fonctionnel. Le stéréoscope consiste en une boîte rectangulaire dont la partie supérieure est ouverte. Deux ouvertures à une extrémité, permettent de voir les deux images à l'autre extrémité. Au milieu du stéréoscope, un séparateur est placé entre les images et entre nos yeux (voir figure 2.3).

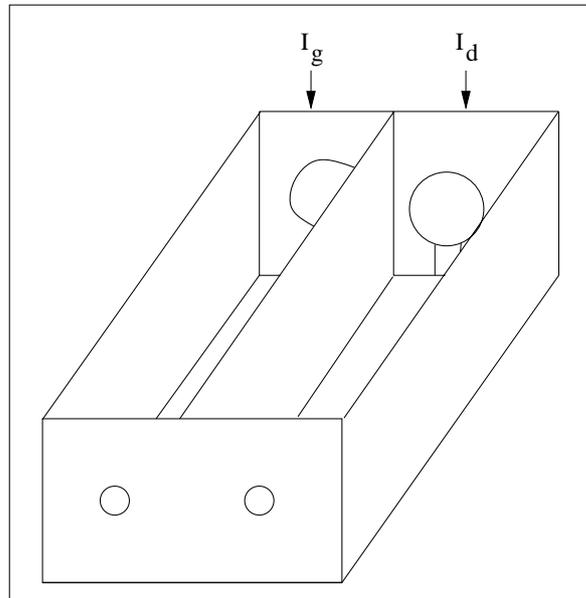


FIG. 2.3 – Le stéréoscope.

Un anaglyphes est un couple d'images stéréoscopiques en deux couleurs complémentaires (habituellement rouge et bleu). Si on enregistre ces deux images monochromes sur la même bande, on peut les visualiser séparément avec des lunettes qui filtrent une couleur ou l'autre pour chaque oeil donnant ainsi une impression de profondeur. Cette technique ne permet pas la diffusion de vidéo stéréoscopique couleur.

L'autostéréoscope où les images gauche et droite sont présentées mutiplexées spatialement sur un même écran : les colonnes impaires contiennent l'image gauche, et les colonnes paires l'image droite. Devant l'écran se trouve un autre écran qui filtre pour créer un impression de profondeur. À moins d'avoir un système de localisation permettant le déplacement de l'utilisateur, celui-ci doit rester au centre pour conserver cette impression de profondeur.

Le principe de polarisation permet des systèmes d'affichage plus souples. Ici, l'utilisateur porte des lunettes polarisées, légères et peu coûteuses. L'écran est équipé d'un dispositif de polarisation qui encode les images gauche et droite de polarisation perpendiculaire. Les verres polarisés captent alors seulement l'image appropriée. Cette méthode permet l'utilisation de couleurs et peut accoutumer plusieurs usagers. C'est une méthode pratique pour des cinémas (Imax), mais n'est pas adaptée pour le divertissement à domicile.

Finalement, il y a l'affichage à multiplexage temporel adapté pour le visionnement sur un écran de télévision. Ici, les images gauche et droite sont affichées une après l'autre. Des lunettes d'obstruction synchronisées à l'affichage permettent de créer un effet de profondeur.

Chapitre 3

Estimation de Disparité

3.1 Définitions

L'estimation de disparité consiste à déterminer la correspondance entre la projection d'un point physique sur les deux images stéréoscopiques. Selon [12] si la luminance I du point a dans l'image A et du point b dans l'image B ont été définies par le même point de la scène, nous disons que le a et b correspondent.

Il est possible que le point a n'ait aucune correspondance dans l'image B . Nous appelons ce point un point d'occlusion d'image A et fait donc partie de la zone d'occlusion de l'image A . Dans ce cas-ci il est impossible de trouver une correspondance, seulement une pseudo-correspondance peut être trouvée. L'évaluation de pseudo-correspondance n'est pas une tâche bien définie puisque toute l'information concernant ce point est perdue dans l'image B . Réciproquement, on appelle point nouvellement exposé de l'image B un point b n'ait aucune correspondance dans l'image A .

La figure 3.1 contient un exemple d'évaluation de zones d'occlusion pour une séquence de trois images. Un pixel donné dans une image peut être : un point d'occlusion, un point nouvellement exposé ou un point visible. Pour l'image 1, il n'y a aucun point nouvellement exposé, puisque c'est la première image de la séquence. Pour les mêmes raisons, l'image 3 n'a aucun point d'occlusion.

3.2 La Base des Estimateurs de Disparité

L'estimation de disparité est un problème mal posé, c'est pourquoi il est nécessaire d'ajouter des conditions pour solutionner le problème. Par exemple, des propriétés des objets solides et de la lumière pourraient être intégrées dans la résolution du problème. Habituellement, ces contraintes sont exprimées sous forme de fonctions de coût.

L'hypothèse suivante est la plus commune : si un point dans une image correspond

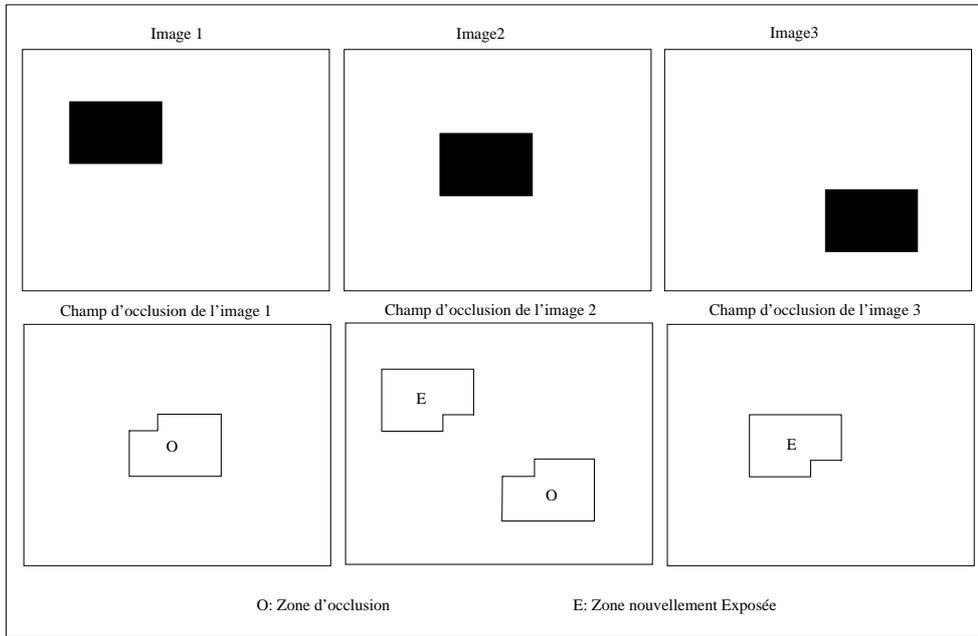


FIG. 3.1 – Exemples de Champs d'Occlusion.

physiquement à un autre point dans l'autre image, il doit avoir la même luminance.

Il est possible d'ajouter l'hypothèse suivante : Si a (dans l'image A) correspond à b (dans l'image B), une zone contenant a devrait apparier une zone contenant b . Les pixels voisins font probablement partie d'un seul objet. Une approche d'appariement de régions consiste à présélectionner les régions de l'image A qui correspondent aux objets et à les apparier à des régions de l'image B . Une voie plus simple est de séparer l'image A en blocs, et de trouver un bloc correspondant dans l'image B pour chaque bloc ainsi découpé (approche d'appariement de blocs).

L'approche du flot optique consiste à ajouter une contrainte de lissage à la supposition de conservation de luminance. La disparité de deux pixels voisins devrait être semblable. Par conséquent, le champ de disparité devrait être lisse.

3.2.1 Approche d'Appariement de Blocs (Block Matching)

L'appariement de blocs consiste à diviser la première image en blocs et de trouver la meilleure position pour chaque bloc dans la deuxième image. Voici le procédé général de l'approche (voir figure 3.2 pour un exemple) :

1. Séparation de l'image A en blocs de $N \times M$
2. Sélection d'un bloc de l'image A , déplacement à toutes les positions d'une région de recherche choisie de l'image B et évaluation de la fonction de coût.
3. Choix de la position de l'image B où la fonction de coût est minimale.
4. Le vecteur correspondant est associé au bloc.

5. Répéter jusqu'à ce que tous les blocs dans l'image A soient appariés.

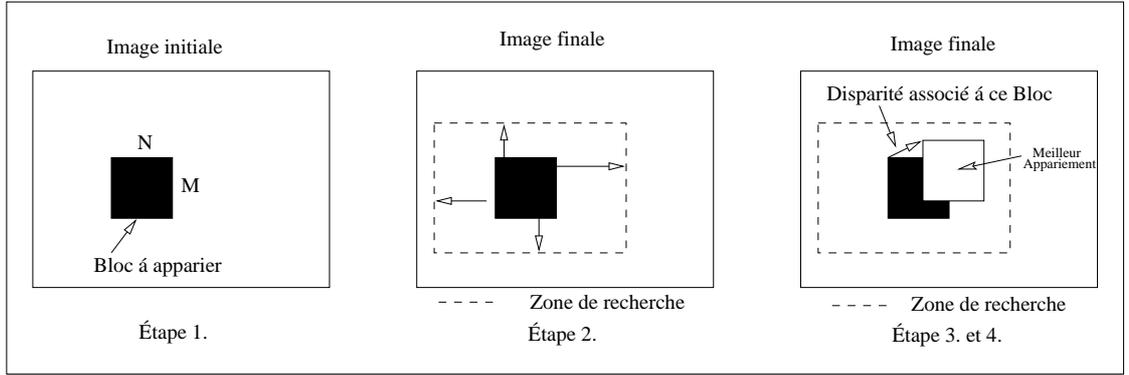


FIG. 3.2 – Exemple d'une Procédure d'Appariement de Blocs.

Types de Fonction de Coût

Le type de fonction de coût utilisée influencera l'efficacité de l'algorithme. Étant donné $D = B_2 - B_1$ la matrice de différence de luminance des blocs et $d_{i,j}$ la composante (i, j) de D , nous pouvons définir plusieurs types d'erreur :

L'erreur quadratique :

$$Q = \sum_{i,j} d_{i,j}^2 \quad (3.2.1)$$

qui est souvent utilisée puisqu'elle exprime l'énergie contenu dans l'erreur.

L'erreur absolue :

$$A = \sum_{i,j} |d_{i,j}| \quad (3.2.2)$$

qui est moins sensible aux grandes différences que l'erreur quadratique.

L'erreur quadratique tronquée :

$$E_{QT} = \sum_{i,j} QT_{\sigma}(d_{i,j}) \quad (3.2.3)$$

$$QT_{\sigma}(x) = \begin{cases} x^2 & x < \sigma \\ \sigma^2 & x \geq \sigma \end{cases} \quad (3.2.4)$$

L'erreur de Geman-McLure :

$$GM(\sigma) = \sum_{i,j} \rho_{GM}(d_{i,j}, \sigma) \quad (3.2.5)$$

où :

$$\rho_{GM}(x, \sigma) = \frac{x^2}{1 + (\frac{x}{\sigma})^2} \quad (3.2.6)$$

Puisque GM et E_{QT} sont bornés, les grandes différences ont toutes un poids similaire dans l'erreur.

L'Erreur de Hamming :

$$H = \sum_{i,j} h(d_{i,j}) \quad (3.2.7)$$

où :

$$h(x) = \begin{cases} 0 & \text{si } x = 0 \\ 1 & \text{sinon} \end{cases} \quad (3.2.8)$$

Ici l'amplitude des différences $d_{i,j}$ n'importe pas, car la fonction H représente le nombre de pixels erronés.

Une autre façon de modéliser des similitudes entre les blocs serait d'utiliser la corrélation.

$$\rho = \frac{\frac{1}{N} \sum_{i,j} B1_{ij} B2_{ij} - \frac{1}{N} \sum_{i,j} B1_{ij} \frac{1}{N} \sum_{i,j} B2_{ij}}{\sigma_1 \sigma_2} \quad (3.2.9)$$

où :

$$\sigma_k^2 = \frac{1}{N} \sum_{i,j} Bk_{ij}^2 - \left(\frac{1}{N} \sum_{i,j} Bk_{ij} \right)^2 \quad k = 1, 2 \quad (3.2.10)$$

ρ est le coefficient de corrélation, il est maximum ($\rho = 1$) si et seulement si les deux blocs sont entièrement corrélés, N est le nombre de pixels dans chaque bloc et σ_k^2 est la variance dans d'un bloc k . Si un la luminance d'un bloc est constante (i.e $b_{ij} = Cst \forall i, j$), sa variance sera nulle et (3.2.9) non définie. Dans ce cas, nous utiliserons une autre définition :

$$C = \frac{\frac{1}{N} \sum_{i,j} B1_{ij} B2_{ij}}{c_1 c_2} \quad (3.2.11)$$

où :

$$c_k^2 = \frac{1}{N} \sum_{i,j} Bk_{ij}^2 \quad k = 1, 2 \quad (3.2.12)$$

La corrélation (dans (3.2.9) et (3.2.11)) est égale à 1 si $B1 = \alpha B2$. Puisque la luminance des blocs n'a pas besoin d'être égale (mais proportionnelle) pour qu'ils soient considérés correspondants, la corrélation ne reflète pas l'hypothèse de luminance constante. Par contre, elle n'est pas sensible aux changements d'illumination et aux différences de luminance entre les caméras.

Types de Recherches

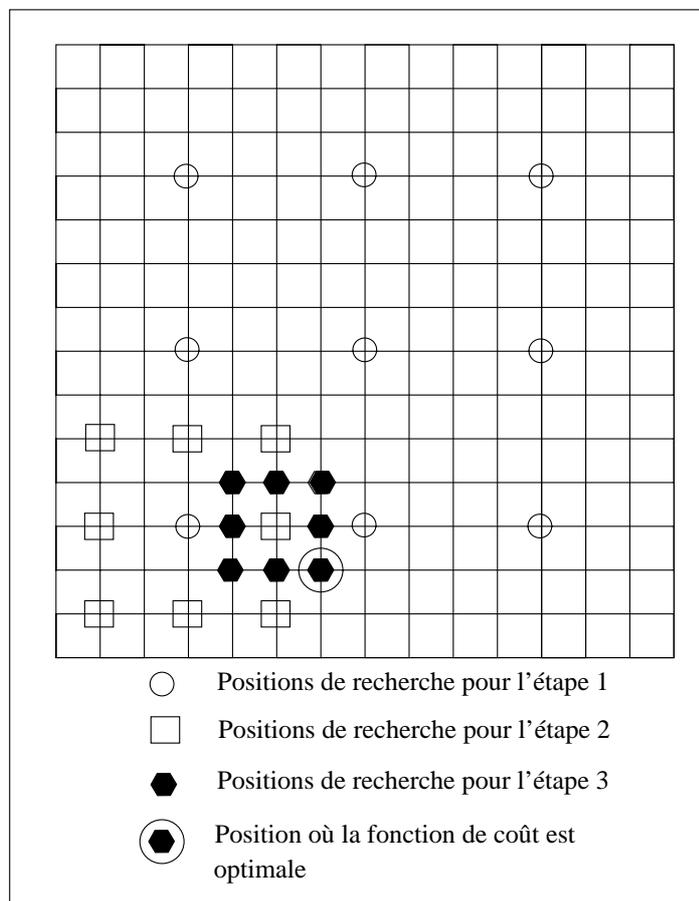


FIG. 3.3 – Exemple d’une Recherche Logarithmique Bidimensionnelle à trois étapes.

La recherche exhaustive consiste à regarder chaque position de la zone de recherche pour trouver la position qui minimise la fonction de coût. Cette méthode assure de trouver la meilleure correspondance, mais cela implique beaucoup de calculs.

La recherche logarithmique bidimensionnelle consiste à séparer la zone de recherche en $N \times M$ régions (habituellement 3×3) d’aire égale. La fonction de coût est évaluée au centre de chacune des régions et celle où la fonction de coût est optimale est choisie pour être la nouvelle zone de recherche (voir la figure 3.3). Ce processus continue jusqu’à ce que la précision obtenue soit suffisante dans chacune des deux directions. Beaucoup de calculs peuvent être évités grâce à cette méthode.

3.2.2 Approche du Flot Optique

Il est également plus facile (même si mathématiquement équivalent) de résoudre le problème pour une séquence temporelle d’image plutôt qu’une paire d’image stéréoscopique.

La première hypothèse est la conservation de la luminance pour un point physique

donné. Soit $I(x, y, t)$ l'intensité au point de coordonnées (x, y) et au temps t de l'image, cette hypothèse s'écrit :

$$I(x + u\delta t, y + v\delta t, t + \delta t) = I(x, y, t) \quad (3.2.13)$$

Où la projection (x, y) au temps t d'un point physique s'est déplacé à une vitesse de (u, v) pendant δt secondes. Si nous dérivons la série de Taylor du côté gauche de l'équation (3.2.13) :

$$\frac{\partial x}{\partial t} \frac{\partial I}{\partial x} + \frac{\partial y}{\partial t} \frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} = 0 \quad (3.2.14)$$

$$\Rightarrow [u, v, 1] \bullet \nabla I = 0 \quad (3.2.15)$$

Tout $[u, v, 1]$ perpendiculaires au gradient de I sont solutions de (3.2.15). Ainsi les solutions possibles de (3.2.15) définissent une ligne dans le plan de vitesse.

Le coût associé à l'hypothèse de constance de l'intensité est :

$$e_c = \int \int (uI_x + vI_y + I_t)^2 dx dy \quad (3.2.16)$$

La deuxième hypothèse est stipule que le flot optique est lisse : des points voisins ont des vitesses voisines. Dans [8] le coût qui modélise la continuité du champ de vecteur est :

$$e_s = \int \int ((u_x^2 + u_y^2) + (v_x^2 + v_y^2)) dx dy \quad (3.2.17)$$

Le cout est minimum pour un champ de vecteur constant. Plus les variation du champ sont importante, plus le coût est grand.

La fonction de coût totale à minimiser est donc :

$$E = e_c + \lambda e_s = \int \int (uI_x + vI_y + I_t)^2 + \lambda((u_x^2 + u_y^2) + (v_x^2 + v_y^2)) dx dy \quad (3.2.18)$$

λ est un paramètre de pondération de lissage du champ. Un grand λ donne un champ très lisse.

Pour minimiser un problème de la forme :

$$\int \int F(u, u_x, u_y, v, v_x, v_y) dx dy \quad (3.2.19)$$

Les équations d'Euler-Lagrange correspondantes sont :

$$\frac{\partial F}{\partial u} - \frac{\partial}{\partial x} \left(\frac{\partial F}{\partial u_x} \right) - \frac{\partial}{\partial y} \left(\frac{\partial F}{\partial u_y} \right) = 0 \quad (3.2.20)$$

$$\frac{\partial F}{\partial v} - \frac{\partial}{\partial x} \left(\frac{\partial F}{\partial v_x} \right) - \frac{\partial}{\partial y} \left(\frac{\partial F}{\partial v_y} \right) = 0 \quad (3.2.21)$$

Dans notre cas :

$$F = (I_x u + I_y v + I_t)^2 + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2) \quad (3.2.22)$$

En résolvant (3.2.20) et (3.2.21) nous trouvons :

$$\nabla^2 u = \frac{1}{\lambda}(I_x u + I_y v + I_t)I_x \quad (3.2.23)$$

$$\nabla^2 v = \frac{1}{\lambda}(I_x u + I_y v + I_t)I_y \quad (3.2.24)$$

On isole et discrétise u et v afin de résoudre le problème. Pour cette tâche, [8] propose l'algorithme itératif suivant :

$$u_{kl}^{n+1} = \overline{u_{kl}^{(n)}} - \frac{\overline{I_x u_{kl}^{(n)}} + \overline{I_y v_{kl}^{(n)}} + I_t}{\lambda + (I_x^2 + I_y^2)} I_x \quad (3.2.25)$$

$$v_{kl}^{n+1} = \overline{v_{kl}^{(n)}} - \frac{\overline{I_x u_{kl}^{(n)}} + \overline{I_y v_{kl}^{(n)}} + I_t}{\lambda + (I_x^2 + I_y^2)} I_y \quad (3.2.26)$$

Où la fonction $\overline{u_{ij}}$ est une moyenne locale qui ne tient pas compte du point lui-même :

$$\overline{u_{ij}} = \frac{u_{i-1j} + u_{i+1j} + u_{ij-1} + u_{ij+1}}{4} \quad (3.2.27)$$

Cette méthode donne de bons résultats seulement pour de petits déplacements. Pour de grands déplacements, cette méthode doit être modifiée et combinée avec une approche hiérarchique.

3.2.3 Approche Hiérarchique (ou progressive)

L'approche hiérarchique peut améliorer les méthodes d'évaluation de disparité en abaissant le coût de calcul et en agrandissant la zone de recherche. Elle consiste à prendre les images initiales (A et B) et sous-échantillonner par un facteur f , n fois (typiquement $f=2$). Deux pyramides de $n + 1$ images sont ainsi définies : l'image originale A s'appelle A_0 , la plus petite version de A est nommée A_n , l'image originale B s'appelle B_0 et la plus petite version de B est nommée B_n .

Le procédé commence par une estimation de disparité grossière du champ D_n à partir de l'image A_n vers l'image B_n . Cette première approximation est utilisée comme point de départ pour l'évaluation au niveau $n - 1$ (en utilisant A_{n-1} et B_{n-1}) en faisant un sur-échantillonnage par un facteur f (en agrandissant les vecteurs d'un facteur f). Alors que nous descendons les niveaux, la précision de l'approximation s'améliore. La figure 3.4 contient un exemple d'approche hiérarchique.

Combinée avec une recherche exhaustive d'appariement de blocs, on obtient l'équivalent d'une recherche logarithmique mais avec un filtrage qui évite de trouver un minimum local.

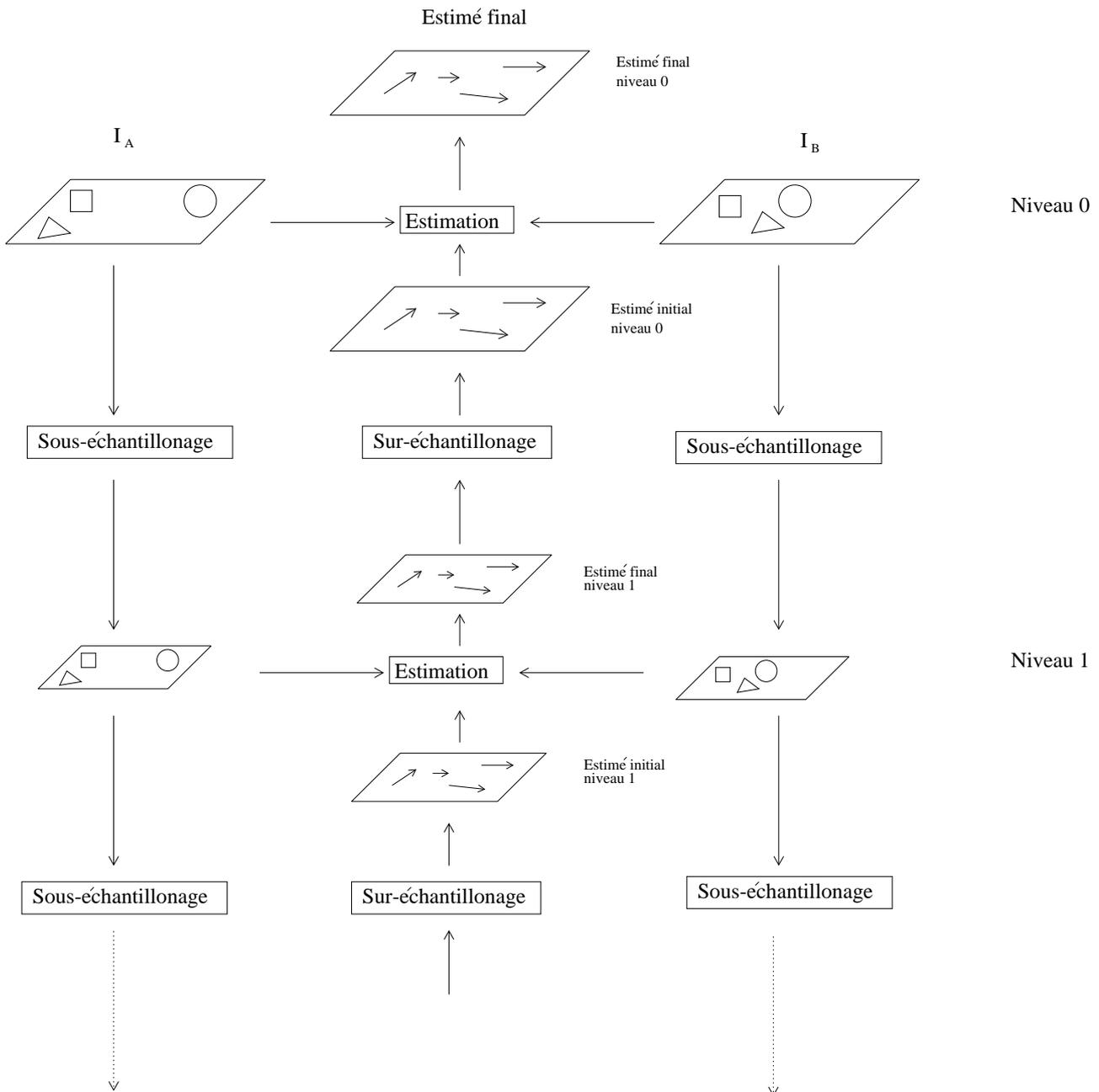


FIG. 3.4 – Exemple d'une approche hiérarchique.

Utilisée avec le flot optique, cette méthode permet de plus grands déplacements et abaisse le nombre de calculs. Pour de petit déplacement, la méthode de flot optique converge rapidement. Puisque la première étape utilise les plus petites images, le nombre de pixels est très faible, donc l'algorithme nécessite très peu de calculs. Les prochains niveaux comportent de plus grands déplacements mais l'estimation de disparité approximative du niveau précédent peut-être utilisée pour débiter l'algorithme. Ainsi, seulement quelques itérations devraient être nécessaires pour actualiser le champs.

3.3 Améliorations ou nouvelles approches

Cette section contient un bref survol d'approches spécifiques développées par différents auteurs. Certaines sont de légers modifications des méthodes existantes, d'autres sont des approches complètement nouvelles.

3.3.1 Flot optique lisse par morceau

Selon [11], bien que la contrainte de lissage semble raisonnable, elle ne devrait pas être appliquée partout. La présence de différents objets dans la scène cause des discontinuités dans le champ de disparité. Ainsi un estimateur de disparité devrait permettre des discontinuités dans le champ de vecteur.

Étant donné un champ de discontinuité $w_{i,j}$, dans (3.2.25) et (3.2.26) remplaçons la fonction $\overline{u_{i,j}}$ par la fonction $u_{i,j}^*$ qui considère les discontinuités :

$$u_{i,j}^* = \frac{w_{i-1,j}u_{i-1,j} + w_{i+1,j}u_{i+1,j} + w_{i,j-1}u_{i,j-1} + w_{i,j+1}u_{i,j+1}}{w_{i-1,j} + w_{i+1,j} + w_{i,j-1} + w_{i,j+1}} \quad (3.3.1)$$

Grâce à la fonction de poids w , la fonction de moyenne modifiée ne moyenne pas où il y a des discontinuités. Le problème se situe dans la connaissance de la fonction w . Le champ de discontinuité doit être connu pour que la méthode fonctionne. Ceci signifie que nous devons savoir où les objets sont placés dans l'image. Segmenter une image réelle en objets distincts est un problème très complexe qui n'est pas abordé dans ce texte.

3.3.2 Évaluation de mouvement avec la détection des zones d'occlusion

Dans [4] une méthode pour évaluer le mouvement en présence d'occlusion, des zones nouvellement exposées et des discontinuités est proposée. Contrairement à [11], il n'est pas nécessaire de connaître les zones d'occlusion ou les discontinuités. La méthode consiste à trouver le champ de mouvement, les zones d'occlusions et les zones de discontinuité conjointement.

Une fonction de coûts de quatre termes est définie par :

$$C = C_i + \lambda_d C_d + \lambda_l C_l + \lambda_o C_o \quad (3.3.2)$$

C_i est une fonction de coût liée à la vraisemblance des prévisions de mouvement.

$$C_i = \sum_x E_u(x, d_t(x), O_t(x))^2 \quad (3.3.3)$$

où $E_u(x, d_t(x), O_t(x))$ est l'erreur d'appariement de la valeur d'intensité $u(x)$, considérant le déplacement $d_t(x)$ à la position x et $O_t(x)$ le champ d'occlusion à trois valeurs (point d'occlusion, point nouvellement exposé, point en mouvement).

$$E_u(x, d_t(x), O_t(x)) = \tilde{u}_t(x) - \alpha(O_t(x))\tilde{u}_{t-}(x - \Delta_t d_t(x)) - (1 - \alpha(O_t(x)))\tilde{u}_{t+}(x + (1 - \Delta_t)d_t(x)) \quad (3.3.4)$$

où \tilde{u} est une interpolation de u , Δ_t est la position intermédiaire entre les images et $\alpha(O_t(x))$ est défini comme :

$$\alpha(O_t(x)) = \begin{cases} 1 - \Delta_t & \text{si } x \text{ est un point visible} \\ 0 & \text{si } x \text{ est dans une zone nouvellement exposée} \\ 1 & \text{si } x \text{ est dans une zone d'occlusion} \end{cases} \quad (3.3.5)$$

C_d est une fonction de coût liée au champ de disparité. Elle favorise le lissage spatial et permet des discontinuités (en utilisant le champ de discontinuité) :

$$C_d = \sum_x \sum_{x' \in \eta(x)} |d_t(x) - d_t(x')| \cdot (1 - l_t(x, x')) \quad (3.3.6)$$

Où $\eta(x)$ est un voisinage de x et la fonction $l_t(x, x')$ est définie entre deux pixels voisins comme suit :

$$l_t(x, x') = \begin{cases} 1 & \text{s'il existe une discontinuité dans le champ} \\ 0 & \text{si le champ est continu} \end{cases} \quad (3.3.7)$$

Les fonctions de coûts C_l et C_o sont associées aux champs de discontinuité et d'occlusion. Si certaines discontinuités ne sont pas plausibles C_l deviendra très important. La même approche est utilisée pour C_o (voir [4] pour les détails).

Une fois la fonction de coût complètement définie, il ne reste que sa minimisation. Ce n'est pas une tâche simple parce que le nombre de variables est énorme : 5 variables par point d'échantillonnage.

3.3.3 Estimation cohérente de disparité et de mouvement

Le but de cette méthode d'appariement proposée dans [5] résout le problème en combinant différentes stratégies, exploitant la redondance spatiale et temporelle de la scène tridimensionnelle. La méthode est la suivante :

1. Un appariement hiérarchique de blocs est appliqué à 2 paires d'images. Au total, 4 champs de disparité (de gauche à droite \vec{d}_g et de droite à gauche \vec{d}_d au temps t et $t+1$) et 2 champs de mouvement (pour l'image gauche \vec{m}_g et l'image droite \vec{m}_d) sont estimés.
2. Puisque les zones homogènes ne sont pas bien appariées dans des approches d'appariement de blocs, les vecteurs correspondant à ces zones sont remplacés par une interpolation de vecteurs plus fiables.

3. Un contrôle de cohérence bidirectionnel est exécuté pour éliminer des vecteurs non cohérents (c'est-à-dire les vecteurs qui ne respectent pas la condition suivante pour un ϵ donné) :

$$\|\vec{d}_g(\vec{z}, t) + \vec{d}_d(\vec{z} + \vec{d}_g, t)\| < \epsilon \quad (3.3.8)$$

4. Pour remplacer les vecteurs non cohérents, il faut choisir leur remplacement dans un ensemble de 11 vecteurs : six du premier niveau d'appariement hiérarchique, quatre provenant des points environnants où le champs est déjà évalué et un provenant d'une prédiction temporelle compensée par le mouvement (au temps $t - 1$).
5. Une fonction de coût dépendant de la corrélation, de la continuité et d'une cohérence des quatre champs est définie pour nous aider à choisir le meilleur vecteur parmi les onze candidats de l'ensemble de vecteur défini précédemment. La cohérence des quatre champs est définie comme suit :

$$\delta = \|\vec{d}_g(\vec{z}, t) + \vec{m}_d[\vec{z} + \vec{d}_g(\vec{z}, t), t] - \vec{d}_g[\vec{z} + \vec{m}_g(z, t), t + 1] - \vec{m}_g(z, t)\| \quad (3.3.9)$$

La méthode est complexe à implanter (à noter qu'ici une version simplifiée est présentée : la nature exacte de la fonction de coût n'est pas définie ; pour plus de détails se reporter à [5]). Il est nécessaire de calculer deux champs de disparité et les deux champs de mouvement à chaque temps t . La méthode contient plusieurs paramètres et seuils. Trouver de bonnes valeurs pour une scène donnée pourrait-être une tâche compliquée.

3.3.4 Amélioration de l'appariement de blocs

Avec quelques modifications, l'appariement de blocs peut devenir une estimation de disparité très efficace. Dans [10] une méthode améliorée d'appariement de blocs est présentée.

En utilisant la géométrie des caméras, on peut déterminer le déplacement vertical maximum pour limiter la zone de recherche.

Il est possible qu'il y ait un déséquilibre de luminosité ou de chrominance entre les deux caméras. Dans [10] une transformation linéaire est employée sur les composantes pour égaliser la moyenne et la variance des deux images. De cette façon, la luminosité d'un point physique devrait être égale dans l'image gauche et l'image droite, facilitant l'identification de correspondance. Il est possible d'appliquer cette technique aux ajustements de luminance et de chrominance.

Au lieu d'utiliser seulement le niveau de luminosité pour estimer la position optimale, [10] propose de modifier la fonction de coût pour inclure les composantes de chrominance. Dans les régions de faible détails, les composantes de chrominance peuvent offrir de l'information supplémentaire pour établir les correspondances.

Pour lever l'ambiguïté dans des régions homogènes, une contrainte de lissage, qui transforme le problème en un algorithme itératif, est ajoutée.

Selon [13], la robustesse est le pourcentage de mauvais points tolérés avant que l'estimateur obtienne une évaluation arbitrairement mauvaise. [10] utilise l'opérateur robuste de Geman McLure comme fonction de coût. Ceci permet un meilleur appariement de blocs et une détection automatique de blocs problématiques. Cette détection est exécutée en appliquant un seuil différentiel : si le meilleur appariement d'un bloc donné donne une erreur plus grande qu'un certain seuil, le bloc est considéré problématique. Les blocs problématiques sont divisés en quatre blocs de taille égale. Le processus est ensuite appliqué aux plus petits blocs. Cette étape peut être appliquée plusieurs fois mais une grosseur de bloc minimale doit être définie.

Dans [10], on a réglé les principaux défauts de l'appariement de blocs et ajouté une approche de correction de luminosité.

3.3.5 Appariement de blocs avec dimension de fenêtre variable

Cette technique proposée dans [6] a été développée pour résoudre le même problème que la structure de quadtree : éviter que les blocs recouvrent les discontinuités du champ de disparité. Initialement, un appariement hiérarchique de bloc est exécuté dans les deux directions (de gauche à droite et de droite à gauche). Une carte de fiabilité est alors définie en utilisant le contrôle de cohérence bidirectionnel et la courbure de la surface de corrélation.

Par la suite, un champ de potentiel est défini en utilisant la carte de fiabilité. La fenêtre (ou le bloc) est petite au départ et est soumise à des forces internes augmentant la grandeur de la fenêtre (ces forces sont produites par une certaine quantité d'énergie cinétique). Les forces externes qui sont produites par le champ de potentiel, arrêtent l'expansion de la fenêtre. Les forces extérieures deviennent plus fortes dans les zones de basse fiabilité. Ainsi, un bloc commence petit et subit une expansion jusqu'à ce qu'il atteigne un équilibre. Une fois à l'équilibre, un bloc similaire à ce bloc est cherché dans la seconde image (voir [6] pour plus de détails).

Chapitre 4

Algorithmes utilisés

4.1 Interpolation sur une grille d'échantillonnage

Dans la section 4.3 il sera nécessaire de déterminer des valeurs de la luminosité de l'image et de ses différentes dérivées à des endroits où l'image n'est pas définie, soit entre les pixels. Il sera donc nécessaire de faire une interpolation. Considérons tout d'abord le cas d'une fonction f définie sur un échantillonnage uniforme unidimensionnel Λ .

Pour interpoler à l'intérieur d'un échantillonnage uniforme, il est pratique d'utiliser une convolution avec une fonction prédéterminée (appelée noyau de convolution), parce que c'est simple et rapide. Posons g , notre nouvelle fonction interpolée :

$$g(x) = \sum_k f(x_k)N\left(\frac{x - x_k}{h}\right) \quad (4.1.1)$$

où $x_k \in \Lambda$, $k \in \mathbb{N}$, $x \in \mathbb{R}$, h est la distance entre deux échantillons consécutifs et N le noyau de convolution. La fonction N doit être symétrique. En effet, si la fonction $g(x)$ est l'interpolation de $f(x)$, la fonction $g_i(x) := g(-x)$ doit être l'interpolation de $f(-x)$. Selon 4.1.1

$$\begin{aligned} g(x) &= \sum_k f(x_k)N\left(\frac{x - x_k}{h}\right) \\ \Rightarrow g(-x) &= \sum_k f(-x_k)N\left(\frac{x_k - x}{h}\right) \end{aligned}$$

Or :

$$g_i(x) = \sum_k f(-x_k)N\left(\frac{x - x_k}{h}\right)$$

Puisque $g_i(x) = g(-x)$

$$\begin{aligned} N\left(\frac{x_k - x}{h}\right) &= N\left(\frac{x - x_k}{h}\right) \\ \Rightarrow N(s) &= N(-s) \quad \forall s \end{aligned}$$

Puisqu'il sera aussi question de la dérivé de la fonction interpolée, le noyau N doit être différentiable presque partout car :

$$g'(x) = \frac{1}{h} \sum_k f(x_k) N' \left(\frac{x - x_k}{h} \right) \quad (4.1.3)$$

4.1.1 Interpolation linéaire

Un noyau d'interpolation linéaire est une composition de deux fonctions linéaires respectivement définies sur les intervalles $(-1,0)$ et $[0,1)$. Puisque la fonction N doit être symétrique :

$$N(s) = \begin{cases} A|s| + B & \text{si } |s| < 1 \\ 0 & \text{si } |s| \geq 1 \end{cases} \quad (4.1.4)$$

Par continuité :

$$\begin{aligned} A + B &= N(1^-) = N(1^+) = 0 \\ \Rightarrow A &= -B \end{aligned} \quad (4.1.5)$$

Pour que $g(x_k) = f(x_k) \forall x_k$ on doit avoir $N(n) = \delta_n^0 \forall n \in \mathbb{Z}$:

$$\Rightarrow N(0) = B = 1 \quad (4.1.6)$$

$$\Rightarrow A = -1 \quad (4.1.7)$$

Donc, notre noyau de convolution, qui est tracé à la figure 4.1, peut s'écrire :

$$N(s) = \begin{cases} -|s| + 1 & \text{si } |s| < 1 \\ 0 & \text{si } |s| \geq 1 \end{cases} \quad (4.1.8)$$

Définissons Δx comme la distance entre x et l'échantillon précédent (étant x_i) alors :

$$g(x) = \left(1 - \frac{\Delta x}{h}\right) f(x_i) + \frac{\Delta x}{h} f(x_{i+1}) \quad (4.1.9)$$

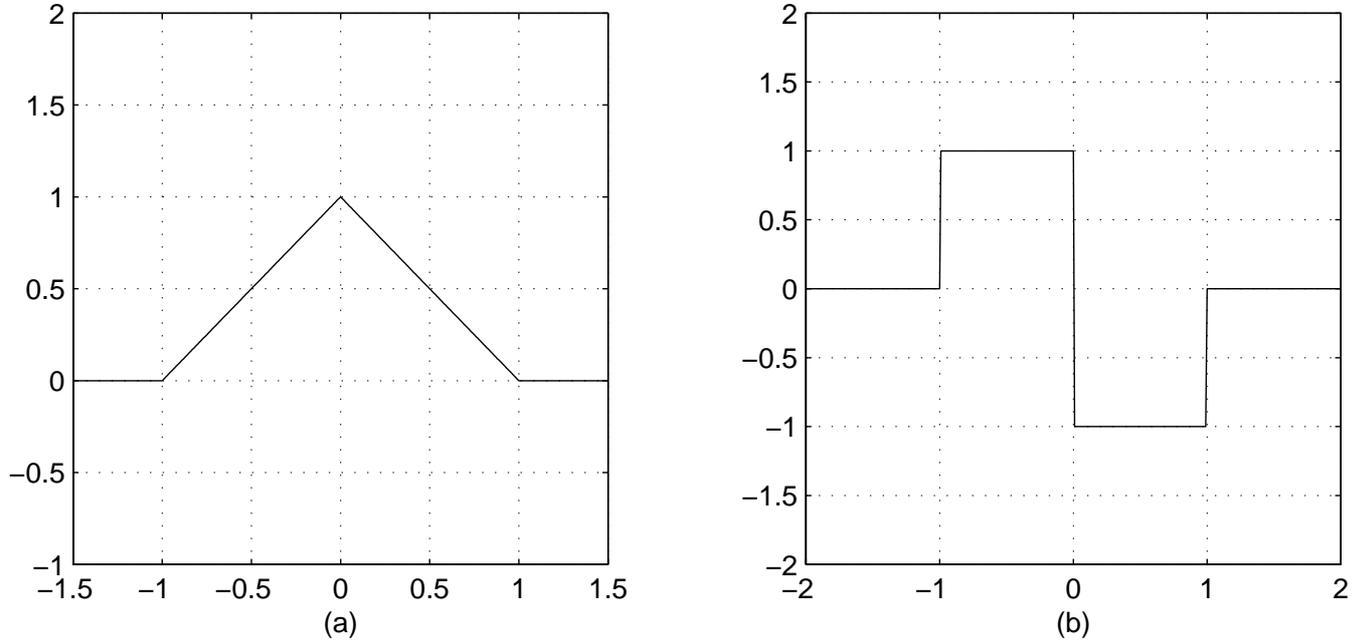


FIG. 4.1 – Noyau de convolution linéaire (a) et sa dérivé (b).

Si nous ne pouvons utiliser que deux échantillons, l'interpolation linéaire est la meilleure approche d'interpolation possible.

4.1.2 Interpolation cubique de Keys

Le noyau de convolution cubique est une composition de quatre fonctions polynomiales de degré 3 définies respectivement sur les intervalles $(-2,-1]$, $(-1,0)$, $[0,1)$ et $[1,2)$. Il est nul partout ailleurs. Il a la forme suivante :

$$N(s) = \begin{cases} A_1|s|^3 + B_1|s|^2 + C_1|s| + D_1 & \text{si } |s| < 1 \\ A_2|s|^3 + B_2|s|^2 + C_2|s| + D_2 & \text{si } 1 \geq |s| < 2 \\ 0 & \text{si } |s| \geq 2 \end{cases} \quad (4.1.10)$$

Les premières conditions à vérifier doivent être la continuité et la cohérence (c'est-à-dire $g(x_k) = f(x_k)$) :

$$1 = N(0) = D_1 \quad (4.1.11)$$

$$0 = N(1^-) = A_1 + B_1 + C_1 + D_1 \quad (4.1.12)$$

$$0 = N(1^+) = A_2 + B_2 + C_2 + D_2 \quad (4.1.13)$$

$$0 = N(2^-) = 8A_2 + 4B_2 + 2C_2 + D_2 \quad (4.1.14)$$

Puisque nous avons huit inconnues et seulement quatre équations, nous pouvons aussi supposer la continuité de sa dérivé :

$$-C_1 = N'(0^-) = N'(0^+) = C_1 \quad (4.1.15)$$

$$3A_1 + 2B_1 + C_1 = N'(1^-) = N'(1^+) = 3A_2 + 2B_2 + C_2 \quad (4.1.16)$$

$$12A_2 + 4B_2 + C_2 = N'(2^+) = N'(2^-) = 0 \quad (4.1.17)$$

Il manque une équation pour compléter le système. Posons $A_2 = a$ et trouvons la solution en fonction de a :

$$N(s) = \begin{cases} (a+2)|s|^3 - (a+3)|s|^2 + 1 & si \quad 0 < |s| < 1 \\ a|s|^3 - 5a|s|^2 + 8a|s| - 4a & si \quad 1 < |s| < 2 \\ 0 & si \quad |s| \geq 2 \end{cases} \quad (4.1.18)$$

L'idée de Keys [9] est de supposer que la fonction f , avant l'échantillonnage, admet un développement en série de Taylor de degré 2. Ainsi, il a comparé le développement de f avec celui de g afin d'en égaliser le plus de termes possibles. Soit x_j l'échantillon précédent x et soit $s = \frac{x-x_j}{h}$, alors :

$$\frac{x-x_k}{h} = \frac{x-x_j+x_j-x_k}{h} = s+j-k. \quad (4.1.19)$$

L'équation (4.1.1) devient :

$$g(x) = \sum_k f(x_k)N(s+j-k) \quad (4.1.20)$$

Puisque N est nul partout, sauf sur l'intervalle $(-2,2)$, et que $s \in (0,1)$ il est possible d'écrire explicitement la somme de (4.1.20) :

$$g(x) = f(x_{j-1})N(s+1) + f(x_j)N(s) + f(x_{j+1})N(s-1) + f(x_{j+2})N(s-2) \quad (4.1.21)$$

Or, par (4.1.18) nous avons :

$$N(s+1) = as^3 - 2as^2 + as \quad (4.1.22)$$

$$N(s) = (a+2)s^3 - (a+3)s^2 + 1 \quad (4.1.23)$$

$$N(s-1) = -(a+2)s^3 + (2a+3)s^2 - as \quad (4.1.24)$$

$$N(s-2) = -as^3 + as^2 \quad (4.1.25)$$

En substituant les équations (4.1.22) à (4.1.25) dans (4.1.21), puis en regroupant les termes selon les différentes puissances de s , nous obtenons :

$$\begin{aligned} g(x) &= [a(f(x_{j+2}) - f(x_{j-1})) + (a+2)(f(x_{j+1}) - f(x_j))]s^3 \\ &+ [2a(f(x_{j+1}) - f(x_{j-1})) + 3(f(x_{j+1}) - f(x_j)) + a(f(x_{j+2}) - f(x_j))]s^2 \\ &- a(f(x_{j+1}) - f(x_{j-1}))s \\ &+ f(x_j) \end{aligned} \quad (4.1.26)$$

Si f est deux fois continûment dérivable dans l'intervalle $[x_{j-1}, x_{j+2}]$, alors selon le théorème de Taylor :

$$f(x_{j+1}) = f(x_j + h) = f(x_j) + f'(x_j)h + \frac{f''(x_j)h^2}{2} + O(h^3) \quad (4.1.27)$$

De la même façon :

$$f(x_{j-1}) = f(x_j - h) = f(x_j) - f'(x_j)h + \frac{f''(x_j)h^2}{2} + O(h^3) \quad (4.1.28)$$

$$f(x_{j+2}) = f(x_j + 2h) = f(x_j) + f'(x_j)2h + f''(x_j)2h^2 + O(h^3) \quad (4.1.29)$$

En substituant (4.1.27) à (4.1.29) dans (4.1.26), la fonction interpolée peut s'écrire :

$$\begin{aligned} g(x) &= -(2a + 1)[2hf'(x_j) + h^2f(x_j)]s^3 \\ &\quad + \left[(6a + 3)hf'(x_j) + (4a + 3)\frac{f''(x_j)h^2}{2} \right] s^2 \\ &\quad - 2ahf'(x_j)s + f(x_j) + O(h^3) \end{aligned} \quad (4.1.30)$$

Il est aussi possible d'estimer $f(x)$ par expansion de Taylor :

$$f(x) = f(x_j + sh) = f(x_j) + f'(x_j)sh + \frac{f''(x_j)(sh)^2}{2} + O(h^3) \quad (4.1.31)$$

On peut donc écrire :

$$\begin{aligned} f(x) - g(x) &= (2a + 1)[2hf'(x_j) + h^2f(x_j)]s^3 \\ &\quad - (2a + 1)[3hf'(x_j) + h^2f(x_j)]s^2 \\ &\quad + (2a + 1)shf'(x_j) + O(h^3) \end{aligned} \quad (4.1.32)$$

Si nous posons $a = -1/2$, les trois premiers termes du développement de Taylor seront identiques pour f et g . Substituons $a = -1/2$ dans l'équation (4.1.18) pour obtenir le noyau de convolution cubique de Keys (voir figure 4.2) :

$$N(s) = \begin{cases} \frac{3}{2}|s|^3 - \frac{5}{2}|s|^2 + 1 & si \quad 0 < |s| < 1 \\ -\frac{1}{2}|s|^3 + \frac{5}{2}|s|^2 - 4|s| + 2 & si \quad 1 < |s| < 2 \\ 0 & si \quad |s| \geq 2 \end{cases} \quad (4.1.33)$$

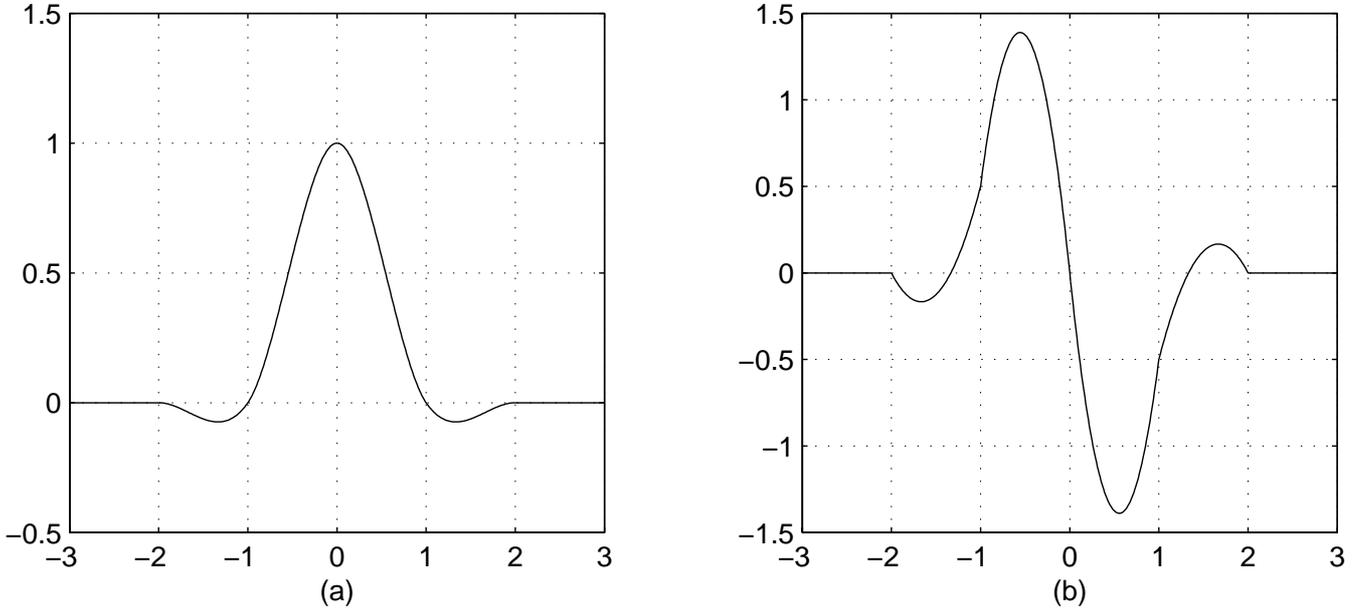


FIG. 4.2 – Noyau de convolution de Keys (a) et sa dérivé (b).

4.1.3 Interpolation sur une grille

Pour interpoler une fonction sur une grille, une approche similaire est utilisée, avec un noyau de convolution bidimensionnel :

$$g(x, y) = \sum_k \sum_l f(x_k, y_l) N_{2d} \left(\frac{x - x_k}{h_h}, \frac{y - y_l}{h_v} \right) \quad (4.1.34)$$

L'interpolation bidimensionnelle peut aussi s'effectuer comme une suite d'interpolations unidimensionnelles :

$$g(x, y) = \sum_k N_x \left(\frac{x - x_k}{h_h} \right) \sum_l f(x_k, y_l) N_y \left(\frac{y - y_l}{h_v} \right) \quad (4.1.35)$$

L'équation (4.1.35) est équivalente à l'équation (4.1.34) à condition de supposer que N_{2d} est séparable, c'est-à-dire :

$$N_{2d} \left(\frac{x - x_k}{h_h}, \frac{y - y_l}{h_v} \right) = N_x \left(\frac{x - x_k}{h_h} \right) N_y \left(\frac{y - y_l}{h_v} \right) \quad (4.1.36)$$

De cette façon, n'importe quel noyau de convolution unidimensionnel peut être utilisé, deux fois, pour faire une interpolation bidimensionnelle. Puisqu'on suppose que l'image est isotrope, il est inutile d'utiliser deux noyaux différents pour N_x et N_y . De plus, puisque notre unité de mesure est le pixel, nous pouvons supposer, sans perdre de généralité, que $h_h = h_v = 1$. Réécrivons l'équation (4.1.35) :

$$g(x, y) = \sum_k \sum_l f(k, l) N(x - k) N(y - l) \quad (4.1.37)$$

En supposant que le noyau N est une fois différentiables les dérivés spatiales de g sont :

$$\frac{\partial g(x, y)}{\partial x} = \sum_k \sum_l f(k, l) N'(x - k) N(y - l) \quad (4.1.38)$$

$$\frac{\partial g(x, y)}{\partial y} = \sum_k \sum_l f(k, l) N(x - k) N'(y - l) \quad (4.1.39)$$

En supposant que le noyau N est deux fois différentiables les dérivés spatiales d'ordre 2 de g sont :

$$\frac{\partial^2 g(x, y)}{\partial x^2} = \sum_k \sum_l f(k, l) N''(x - k) N(y - l) \quad (4.1.40)$$

$$\frac{\partial^2 g(x, y)}{\partial x \partial y} = \sum_k \sum_l f(k, l) N'(x - k) N'(y - l) \quad (4.1.41)$$

$$\frac{\partial^2 g(x, y)}{\partial y^2} = \sum_k \sum_l f(k, l) N(x - k) N''(y - l) \quad (4.1.42)$$

4.2 Correction de luminosité entre deux images

Lorsque deux images de la même scène sont prises par deux caméras différentes, il peut y avoir un changement de luminosité d'une caméra à l'autre. Puisque toutes les versions de flot optique décrites dans la prochaine section reposent sur le fait que la luminosité est la même pour chacune des projections bidimensionnelles d'un point tridimensionnel, un changement de luminosité peut être problématique.

4.2.1 Correction globale de luminosité

Souvent, une simple transformation linéaire appliquée à la luminosité de l'une des deux images s'avère suffisante pour balancer leur luminosité. Soient I_1 l'image de référence et I_2 l'image à ajuster. Appelons J_2 la version corrigée de I_2 .

$$J_2(i, j) = aI_2(i, j) + b \quad (4.2.1)$$

où a et b sont des paramètres à déterminer. Puisque l'image J_2 devrait ressembler statiquement à l'image I_1 , les paramètres a et b doivent être déterminés pour que les deux premiers moments (la moyenne et la l'écart-type) de I_1 et J_2 soient égaux. En supposant que μ_1 et σ_1 sont respectivement la moyenne et l'écart-type de I_1 et que μ_2 et σ_2 sont ceux de I_2 :

$$\kappa = \frac{I_2(i, j) - \mu_2}{\sigma_2} \quad (4.2.2)$$

est un processus centré de variance unitaire.

$$J_2(i, j) = \sigma_1 \kappa + \mu_1 \quad (4.2.3)$$

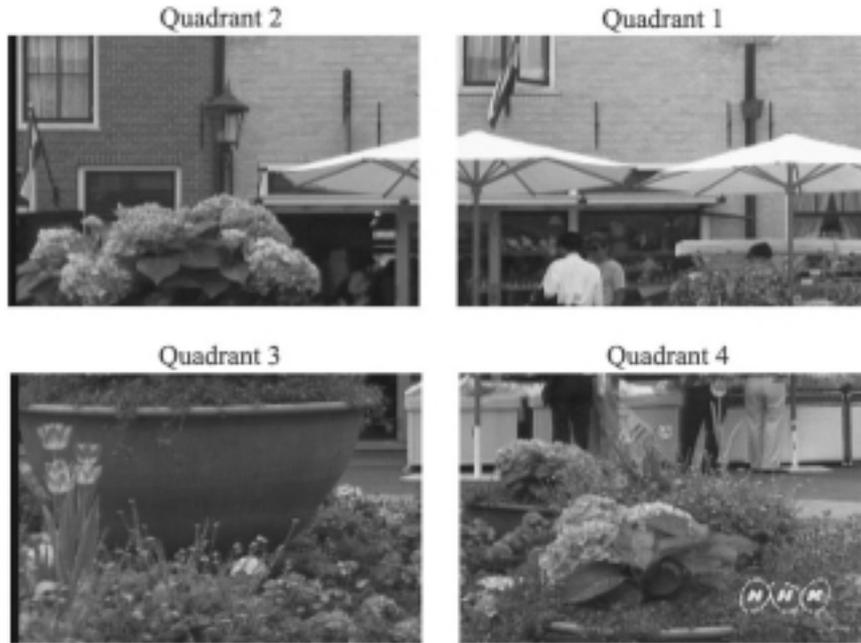
est alors un processus de moyenne μ_1 et d'écart type σ_1 . La transformation linéaire à appliquer à toute l'image I_2 est donc :

$$J_2(i, j) = \frac{\sigma_1}{\sigma_2} I_2(i, j) + \left(\mu_1 - \mu_2 \frac{\sigma_1}{\sigma_2} \right) \quad (4.2.4)$$

Cette transformation linéaire est dite correction globale car les paramètres de correction sont les mêmes pour l'image en entier.

4.2.2 Correction locale de luminosité (par section)

Parfois, seule une correction locale de luminosité peut résoudre le problème d'ajustement de luminosité. La première méthode consiste à séparer les images en sections et effectuer des transformations linéaires différentes à chacune des sections. Par exemple, séparons les images en quadrants (voir figure 4.3).

FIG. 4.3 – Séparation de l'image I_1 en quadrant.

Soient I_{kl} le l^e quadrant ($l \in \{1, 2, 3, 4\}$) de l'image I_k ($k \in \{1, 2\}$), si μ_{kl} et σ_{kl} sont respectivement la moyenne et l'écart type du quadrant I_{kl} , la transformation appliquée sur chacun des quadrants est donc :

$$J_{2l}(i, j) = \frac{\sigma_{1l}}{\sigma_{2l}} I_{2l}(i, j) + (\mu_{1l} - \mu_{2l} \frac{\sigma_{1l}}{\sigma_{2l}}) \quad (4.2.5)$$

L'image J_2 sera reconstruite à partir des quatre quadrants J_{2l} . Il est important de limiter le nombre de sections car, plus il y aura de sections, moins la correspondance entre chacune des paires de sections I_{1l} et I_{2l} sera justifié.

4.2.3 Correction bilinéaire de luminosité

Il est aussi possible de déterminer une fonction de correction continue afin de corriger les différences locales de luminosité. L'avantage sur la méthode précédente est d'éviter les discontinuités de luminosité entre les quadrants.

$$J_2(i, j) = a(i, j)I_2(i, j) + b(i, j) \quad (4.2.6)$$

Il suffit de déterminer la correction nécessaire à certains endroits de l'image, d'interpoler (ou extrapoler) pour trouver la correction à effectuer ailleurs. Les images sont séparées en quadrants comme dans la figure 4.3, puis la moyenne et la variance sont déterminées pour chacun des quadrants. Nous pouvons alors supposer que nous

connaissons a et b au centre de chacun des quadrants. En utilisant la même notation qu'à la section 4.2.2 :

$$a_1 = a\left(\frac{3h}{4}, \frac{v}{4}\right) = \frac{\sigma_{11}}{\sigma_{21}} \quad b_1 = b\left(\frac{3h}{4}, \frac{v}{4}\right) = m_{11} - m_{21} \frac{\sigma_{11}}{\sigma_{21}} \quad (4.2.7)$$

$$a_2 = a\left(\frac{h}{4}, \frac{v}{4}\right) = \frac{\sigma_{12}}{\sigma_{22}} \quad b_2 = b\left(\frac{h}{4}, \frac{v}{4}\right) = m_{12} - m_{22} \frac{\sigma_{12}}{\sigma_{22}} \quad (4.2.8)$$

$$a_3 = a\left(\frac{h}{4}, \frac{3v}{4}\right) = \frac{\sigma_{13}}{\sigma_{23}} \quad b_3 = b\left(\frac{h}{4}, \frac{3v}{4}\right) = m_{13} - m_{23} \frac{\sigma_{13}}{\sigma_{23}} \quad (4.2.9)$$

$$a_4 = a\left(\frac{3h}{4}, \frac{3v}{4}\right) = \frac{\sigma_{14}}{\sigma_{24}} \quad b_4 = b\left(\frac{3h}{4}, \frac{3v}{4}\right) = m_{14} - m_{24} \frac{\sigma_{14}}{\sigma_{24}} \quad (4.2.10)$$

où h et v sont les dimensions horizontale et verticale de l'image. Puisque nous connaissons chacun des paramètres de correction sur quatre points de l'image, répartis de manière uniforme, nous pouvons utiliser un interpolation/extrapolation bilinéaire. Ainsi nous trouvons :

$$a(i, j) = (\Delta x)(1 - \Delta y)a_1 + (1 - \Delta x)(1 - \Delta y)a_2 \\ + (1 - \Delta x)(\Delta y)a_3 + (\Delta x)(\Delta y)a_4 \quad (4.2.11)$$

$$b(i, j) = (\Delta x)(1 - \Delta y)b_1 + (1 - \Delta x)(1 - \Delta y)b_2 \\ + (1 - \Delta x)(\Delta y)b_3 + (\Delta x)(\Delta y)b_4 \quad (4.2.12)$$

où Δx et Δy sont les distances normalisées horizontale et verticale par rapport au centre du quadrant 2. Soit :

$$\Delta x = \frac{2\left(i - \frac{h}{4}\right)}{h} \quad (4.2.13)$$

$$\Delta y = \frac{2\left(j - \frac{v}{4}\right)}{v} \quad (4.2.14)$$

Ensuite, les ajustements a et b sont appliqués à l'image I_2 entière comme dans l'équation (4.2.6). Ainsi, la fonction d'ajustement est continue, et permet une correction locale de luminosité. On pourrait généraliser et séparer les images en 16 sections pour faire une interpolation bicubique sur les paramètres de correction, plus il y aura de section, moins chaque paire de sections I_{1l} et I_{2l} vont correspondre.

4.3 Estimation de Disparité

4.3.1 Flot optique de March

Dans cette section l'approche hiérarchique est utilisée pour prendre en compte les grands déplacements. L'approche de flot optique doit donc être adaptée pour des déplacements supérieur à 1 pixel. L'approche de Horn et Schunck n'est pas adaptée à cause de l'hypothèse que u et v doivent être faibles (équation (3.2.14)). Par contre, l'approche utilisée par March dans [11] ne dépend pas de la grandeur de u et v .

Le problème est le suivant : soit une image $I(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}$. Dans le cas d'une séquence d'image le temps intervient :

$$I(x, y, t) : \mathbb{R}^2 \times \mathbb{N} \rightarrow \mathbb{R} \quad (4.3.1)$$

Soient deux réalisations (deux images) prises à des instants différents. Nommons-les I_1 et I_2 :

$$I_1(x, y) = I(x, y, t_0) \quad (4.3.2)$$

$$I_2(x, y) = I(x, y, t_0 + 1) \quad (4.3.3)$$

$$x \in \mathbb{R}, \quad y \in \mathbb{R}$$

On suppose qu'il existe deux fonctions $u(x, y)$ et $v(x, y)$ représentant le mouvement, tel que $u : \mathbb{R}^2 \rightarrow \mathbb{R}$, $v : \mathbb{R}^2 \rightarrow \mathbb{R}$, que u et v sont de classe C^2 . Alors :

$$I_1(x - \alpha u, y - \alpha v) = I_2(x + (1 - \alpha)u, y + (1 - \alpha)v) \quad (4.3.4)$$

où $\alpha \in (0, 1)$ est la position intermédiaire entre I_1 et I_2 où l'on calcule le champ de disparité. La présente discussion propose une manière de calculer ces deux fonctions. Puisque les arguments de I_1 et I_2 admettent des valeurs réelles et que ces fonctions sont définies, en pratique, pour des arguments entiers, il faut utiliser une interpolation. On utilisera une interpolation bicubique de Keys pour évaluer les valeurs de $I(x, y)$ pour des x et y quelconques (voir section 4.1.2).

La fonction de coût associée à l'équation (4.3.4) est la suivante :

$$e_c = \int \int \left((I_1(x - \alpha u, y - \alpha v) - I_2(x + (1 - \alpha)u, y + (1 - \alpha)v)) \right)^2 dx dy \quad (4.3.5)$$

La minimisation de (4.3.5) n'est pas suffisante pour que le problème admette une solution unique. Alors, nous supposons également que le mouvement (u, v) pour un pixel est semblable à celui de ses voisins. De cette supposition nous obtenons une deuxième condition à minimiser :

$$e_s = \int \int (\lambda_u(u_x^2 + u_y^2) + \lambda_v(v_x^2 + v_y^2)) dx dy \quad (4.3.6)$$

où $u_x = \frac{\partial u}{\partial x}$, λ_u et λ_v sont des paramètres libres permettant de donner un poids à la minimisation de (4.3.6) en plus de la possibilité de pouvoir accorder une importance relative entre les déplacements en x et en y . Dans le cas présent, c'est-à-dire des images stéréoscopiques, $\lambda_v \gg \lambda_u$ puisque l'on considère un déplacement latéral entre les deux images .

En minimisant la somme de (4.3.5) et (4.3.6) nous obtenons le problème de minimisation classique d'une fonction de la forme :

$$\int \int F(u, v, u_x, u_y, v_x, v_y) dx dy \quad (4.3.7)$$

où

$$F = \left(I_1(x - \alpha u, y - \alpha v) - I_2(x + (1 - \alpha)u, y + (1 - \alpha)v) \right)^2 + \lambda_u(u_x^2 + u_y^2) + \lambda_v(v_x^2 + v_y^2) \quad (4.3.8)$$

Les équations de d'Euler-Lagrange s'écrivent :

$$\frac{\partial F}{\partial u} - \frac{\partial}{\partial x} \left(\frac{\partial F}{\partial u_x} \right) - \frac{\partial}{\partial y} \left(\frac{\partial F}{\partial u_y} \right) = 0 \quad (4.3.9)$$

$$\frac{\partial F}{\partial v} - \frac{\partial}{\partial x} \left(\frac{\partial F}{\partial v_x} \right) - \frac{\partial}{\partial y} \left(\frac{\partial F}{\partial v_y} \right) = 0 \quad (4.3.10)$$

Soient (x_0, y_0) les coordonnées initiales du pixel et (x_1, y_1) les coordonnées finales. On a :

$$(x_0, y_0) = (x - \alpha u(x, y), y - \alpha v(x, y)) \quad (4.3.11)$$

$$(x_1, y_1) = (x + (1 - \alpha)u(x, y), y + (1 - \alpha)v(x, y)) \quad (4.3.12)$$

Les différents termes de l'équation (4.3.9) :

$$F_u = 2[I(x_1, y_1, t + \Delta t) - I(x_0, y_0, t)] \cdot [(1 - \alpha)I_x(x_1, y_1, t + \Delta t) + \alpha I_x(x_0, y_0, t)] \quad (4.3.13)$$

$$\frac{\partial F_{u_x}}{\partial x} = \frac{2\partial \lambda_u u_x}{\partial x} = 2\lambda_u u_{xx} \quad (4.3.14)$$

$$\frac{\partial F_{u_y}}{\partial y} = \frac{2\partial \lambda_u u_y}{\partial y} = 2\lambda_u u_{yy} \quad (4.3.15)$$

En remplaçant ces termes dans l'équation :

$$F_u - \frac{\partial F_{u_x}}{\partial x} - \frac{\partial F_{u_y}}{\partial y} = F_u - 2\lambda_u \nabla^2 u = 0$$

$$\Rightarrow F_u = 2\lambda_u \nabla^2 u \quad (4.3.16)$$

De la même façon :

$$F_v = 2\lambda_v \nabla^2 v \quad (4.3.17)$$

Renommons maintenant les termes composant F_u et F_v

$$\tilde{I}_x(x, y) = (1 - \alpha)I_x(x_1, y_1, t + \Delta t) + \alpha I_x(x_0, y_0, t) \quad (4.3.18)$$

$$\tilde{I}_y(x, y) = (1 - \alpha)I_y(x_1, y_1, t + \Delta t) + \alpha I_y(x_0, y_0, t) \quad (4.3.19)$$

$$\tilde{I}_t(x, y) = I(x_1, y_1, t + \Delta t) - I(x_0, y_0, t) \quad (4.3.20)$$

Par l'interpolation bicubique de Keys, on peut évaluer la fonction et de la dérivé en dehors grille d'échantillonnage (voir section 4.1.3).

En comparant (4.3.13) est égale à l'équation (4.3.16) réécrivons :

$$\tilde{I}_x \cdot \tilde{I}_t = \lambda_u \nabla^2 u \quad (4.3.21)$$

de la même façon :

$$\tilde{I}_y \cdot \tilde{I}_t = \lambda_v \nabla^2 v \quad (4.3.22)$$

Il est inutile de déterminer $u(x, y)$ et $v(x, y)$ partout, ils seront évaluer seulement où $I(x, y)$ est correctement défini (sur la grille d'échantillonnage). Trouvons les expressions représentant $\nabla^2 u$ et $\nabla^2 v$ dans le cas discret. Pour un pixel dans l'image ayant comme coordonnées (i, j) :

$$\frac{\partial u_{i,j}}{\partial x} = u_{i+1/2,j} - u_{i-1/2,j}$$

$$\frac{\partial^2 u_{i,j}}{\partial x^2} = u_{i+1,j} - u_{i,j} + u_{i-1,j} - u_{i,j} = -2u_{i,j} + u_{i+1,j} + u_{i-1,j}$$

$$\frac{\partial^2 u_{i,j}}{\partial y^2} = -2u_{i,j} + u_{i,j+1} + u_{i,j-1}$$

$$\Rightarrow \nabla^2 u = 4(\overline{u_{i,j}} - u_{i,j}) \quad (4.3.23)$$

Où la fonction $\overline{u_{i,j}}$ est une moyenne locale :

$$\overline{u_{i,j}} = \frac{u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}}{4} \quad (4.3.24)$$

Nous avons aussi :

$$\nabla^2 v = 4(\overline{v_{i,j}} - v_{i,j}) \quad (4.3.25)$$

En remplaçant (4.3.23) et (4.3.25) dans (4.3.21) et (4.3.22) on obtient :

$$\begin{aligned} \tilde{I}_x \cdot \tilde{I}_t &= 4(\overline{u_{i,j}} - u_{i,j}) \\ \tilde{I}_y \cdot \tilde{I}_t &= 4(\overline{v_{i,j}} - v_{i,j}) \end{aligned}$$

$$\Rightarrow u_{i,j} = \overline{u_{i,j}} - \frac{\tilde{I}_x \tilde{I}_t}{4\lambda_u} \quad (4.3.26)$$

$$v_{i,j} = \overline{v_{i,j}} - \frac{\tilde{I}_y \tilde{I}_t}{4\lambda_v} \quad (4.3.27)$$

On détermine $u_{i,j}$ et $v_{i,j} \forall (i,j) \in (0, M-1) \times (0, N-1)$ par la méthode itérative de Jacobi :

$$u_{i,j}^{(n+1)} = \overline{u_{i,j}^{(n)}} - \frac{\tilde{I}_x^{(n)} \tilde{I}_t^{(n)}}{4\lambda_u} \quad (4.3.28)$$

$$v_{i,j}^{(n+1)} = \overline{v_{i,j}^{(n)}} - \frac{\tilde{I}_y^{(n)} \tilde{I}_t^{(n)}}{4\lambda_v} \quad (4.3.29)$$

Le calcul débute avec un mouvement initial $(u, v) = 0 \forall (x, y)$ pour le dernier niveau hiérarchique (la paire d'images la plus petite). Une fois que le processus itératif 4.3.28 et 4.3.29 converge, ce champ est interpolé et amplifié pour le propager au niveau inférieur (à une image plus grande). L'approximation du niveau précédent a alors été utilisée pour débiter les itérations de l'estimation en cours.

4.3.2 Flot optique de Horn et Schunck

Pour obtenir une version hiérarchique de l'algorithme de Horn et Schunck, il est nécessaire de modifier l'approche afin de permettre de grands déplacements. De plus, il faut ajouter le paramètre α qui permet le calcul du champ de disparité à une position arbitraire.

La première contrainte est la conservation de la luminance d'un point physique donné. Étant donné $I(x, y, t)$ la fonction d'intensité de l'image, nous pouvons facilement écrire cette contrainte :

$$I(x + (1 - \alpha)u, y + (1 - \alpha)v, t + 1) = I(x - \alpha u, y - \alpha v, t) \quad (4.3.30)$$

Soient $(u_0(x, y), v_0(x, y))$ une approximation initiale de $(u(x, y), v(x, y))$, et

$$(\Delta u, \Delta v) = (u, v) - (u_0, v_0), \quad (4.3.31)$$

alors l'équation (4.3.30) devient :

$$\begin{aligned} I(x + (1 - \alpha)u_0 + (1 - \alpha)\Delta u, y + (1 - \alpha)v_0 + (1 - \alpha)\Delta v, t + 1) \\ = I(x - \alpha u_0 - \alpha \Delta u, y - \alpha v_0 - \alpha \Delta v, t) \end{aligned} \quad (4.3.32)$$

où le point physique qui était aux coordonnées $(x - \alpha u_0, y - \alpha v_0)$ au temps t a voyagé à la vitesse (u, v) pendant une unité de temps après t . Calculons la série de Taylor du côté gauche de l'équation (4.3.32) autour du point $(x + (1 - \alpha)u_0, y + (1 - \alpha)v_0, t + 1)$ pour le terme de gauche et autour du point $(x - \alpha u_0, y - \alpha v_0, t)$ pour le terme de droite. En regroupant tous les termes du même côté de l'égalité :

$$\tilde{I}_t(x, y) + \Delta u \tilde{I}_x(x, y) + \Delta v \tilde{I}_y(x, y) = 0 \quad (4.3.33)$$

où :

$$\begin{aligned} \tilde{I}_t(x, y) &= I(x + (1 - \alpha)u_0, y + (1 - \alpha)v_0, t + 1) \\ &\quad - I(x - \alpha u_0, y - \alpha v_0, t) \end{aligned} \quad (4.3.34)$$

$$\begin{aligned} \tilde{I}_x(x, y) &= (1 - \alpha) \frac{\partial I}{\partial x}(x + (1 - \alpha)u_0, y + (1 - \alpha)v_0, t + 1) \\ &\quad + \alpha \frac{\partial I}{\partial x}(x - \alpha u_0, y - \alpha v_0, t) \end{aligned} \quad (4.3.35)$$

$$\begin{aligned} \tilde{I}_y(x, y) &= (1 - \alpha) \frac{\partial I}{\partial y}(x + (1 - \alpha)u_0, y + (1 - \alpha)v_0, t + 1) \\ &\quad + \alpha \frac{\partial I}{\partial y}(x - \alpha u_0, y - \alpha v_0, t) \end{aligned} \quad (4.3.36)$$

La deuxième contrainte est la continuité du flot optique. Elle favorise la ressemblance entre les vecteurs voisins. Dans [8] la fonction de coût qui modélise la continuité du champ de vecteur est ce qui suit :

$$e_s = \sum_{i,j} \left[\frac{\lambda}{2} \sum_{(k,l) \in \nu(i,j)} ((u_{ij} - u_{kl})^2 + (v_{ij} - v_{kl})^2) \right] \quad (4.3.37)$$

La fonction de coût e_s est minimale pour un champ de vecteur constant. λ est un paramètre qui permet de donner plus ou moins d'importance au lissage des composantes du champ et $\nu(i,j)$ un voisinage de (i,j) contenant les pixels $(i+1,j)$, $(i,j+1)$, $(i-1,j)$ et $(i,j-1)$.

Nous voulons également minimiser e_c , qui sera la fonction de coût pour l'équation (4.3.33).

$$e_c = \sum_{i,j} \left(\Delta u(i,j) \tilde{I}_x(i,j) + \Delta v(i,j) \tilde{I}_y(i,j) + \tilde{I}_t(i,j) \right)^2 \quad (4.3.38)$$

Nous devons donc minimiser la somme :

$$E = \sum_{i,j} \left[\left(\Delta u(i,j) \tilde{I}_x(i,j) + \Delta v(i,j) \tilde{I}_y(i,j) + \tilde{I}_t(i,j) \right)^2 + \frac{\lambda}{2} \sum_{(k,l) \in \nu(i,j)} \left[(u_{ij} - u_{kl})^2 + (v_{ij} - v_{kl})^2 \right] \right] \quad (4.3.39)$$

Pour minimiser E par rapport à $u(i,j)$ et $v(i,j)$, il suffit de résoudre $\frac{\partial E}{\partial u(i,j)} = 0$ et $\frac{\partial E}{\partial v(i,j)} = 0$.

$$\begin{aligned} \frac{\partial E}{\partial u(i,j)} &= 2\tilde{I}_x(i,j) \left(\tilde{I}_x(i,j) \Delta u(i,j) \right. \\ &\quad \left. + \tilde{I}_y(i,j) \Delta v(i,j) + \tilde{I}_t(i,j) \right) \\ &\quad + 2\lambda(u(i,j) - \overline{u(i,j)}) \end{aligned} \quad (4.3.40)$$

$$\begin{aligned} \frac{\partial E}{\partial v(i,j)} &= 2\tilde{I}_y(i,j) \left(\tilde{I}_x(i,j) \Delta u(i,j) \right. \\ &\quad \left. + \tilde{I}_y(i,j) \Delta v(i,j) + \tilde{I}_t(i,j) \right) \\ &\quad + 2\lambda(v(i,j) - \overline{v(i,j)}) \end{aligned} \quad (4.3.41)$$

où la fonction $\overline{u(i,j)}$ est une moyenne locale sur $\nu(i,j)$, soit :

$$\overline{\Delta u_{ij}} = \frac{\Delta u_{i-1j} + \Delta u_{i+1j} + \Delta u_{ij-1} + \Delta u_{ij+1}}{4} \quad (4.3.42)$$

dans ce qui suit, l'index (i,j) est laissé implicite pour alléger la notation. Puisque les équations (4.3.40) et (4.3.41) doivent être nulles pour minimiser E , nous avons donc :

$$\Delta u(\lambda + \tilde{I}_x^2) + \Delta v \tilde{I}_x \tilde{I}_y = \lambda(\overline{\Delta u} + \overline{u_0} - u_0) - \tilde{I}_x \tilde{I}_t \quad (4.3.43)$$

$$\Delta u \tilde{I}_x \tilde{I}_y + \Delta v(\lambda + \tilde{I}_y^2) = \lambda(\overline{\Delta v} + \overline{v_0} - v_0) - \tilde{I}_y \tilde{I}_t \quad (4.3.44)$$

$$\begin{pmatrix} \lambda + \tilde{I}_x^2 & \tilde{I}_x \tilde{I}_y \\ \tilde{I}_x \tilde{I}_y & \lambda + \tilde{I}_y^2 \end{pmatrix} \begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix} = \begin{pmatrix} \lambda(\overline{\Delta u} + \overline{u_0} - u_0 + \tilde{I}_x \tilde{I}_t) \\ \lambda(\overline{\Delta v} + \overline{v_0} - v_0 + \tilde{I}_y \tilde{I}_t) \end{pmatrix} \quad (4.3.45)$$

Si nous inversons la matrice de l'équation (4.3.45) nous obtenons :

$$\Delta u = \overline{\Delta u} + \overline{u_0} - u_0 - \frac{\left((\overline{\Delta u} + \overline{u_0} - u_0) \tilde{I}_x + (\overline{\Delta v} + \overline{v_0} - v_0) \tilde{I}_y + \tilde{I}_t \right) \tilde{I}_x}{\lambda + \tilde{I}_x^2 + \tilde{I}_y^2} \quad (4.3.46)$$

$$\Delta v = \overline{\Delta v} + \overline{v_0} - v_0 - \frac{\left((\overline{\Delta u} + \overline{u_0} - u_0) \tilde{I}_x + (\overline{\Delta v} + \overline{v_0} - v_0) \tilde{I}_y + \tilde{I}_t \right) \tilde{I}_y}{\lambda + \tilde{I}_x^2 + \tilde{I}_y^2} \quad (4.3.47)$$

Qui nous mène au processus itératif de Jacobi suivant :

$$u^{n+1} = \overline{u^{(n)}} + - \frac{\left((\overline{u^{(n)}} - u_0) \tilde{I}_x + (\overline{v^{(n)}} - v_0) \tilde{I}_y + \tilde{I}_t \right) \tilde{I}_x}{\lambda + \tilde{I}_x^2 + \tilde{I}_y^2} \quad (4.3.48)$$

$$v^{n+1} = \overline{v^{(n)}} + - \frac{\left((\overline{u^{(n)}} - u_0) \tilde{I}_x + (\overline{v^{(n)}} - v_0) \tilde{I}_y + \tilde{I}_t \right) \tilde{I}_y}{\lambda + \tilde{I}_x^2 + \tilde{I}_y^2} \quad (4.3.49)$$

Au dernier niveau hiérarchique, u_0 et v_0 sont inconnues, ils seront donc considérés nuls. Pour que 4.3.33 soit valide, Δu et Δv doivent toujours être entre -1 et 1. Suffisamment de niveaux hiérarchiques devront être utilisé afin que u et v au dernier niveau soient entre -1 et 1 pour toute l'image.

4.3.3 Flot optique robuste

Le but de cette étude est de développer une méthode robuste d'estimation de disparité. Le flot optique tel que vu par Horn et Schunck [8] ou March [11] est intéressant mais a ses limites en ce qui concerne les discontinuités dans le champ de mouvement. En effet, nous observons certaines déformations du mouvement qui ne correspondent à aucun mouvement réel aux endroits où devraient se trouver les frontières qui séparent des zones de déplacements différents. C'est une version filtrée du champ de déplacement qui remplace le déplacement idéal.

Jusqu'à présent, dans cet ouvrage, seules des fonctions de coûts quadratiques ont été considérées pour résoudre le problème du flot optique. Cette fonction de coût augmente son influence plus les deux valeurs de mouvements de pixels voisins sont différents. L'influence d'une fonction est définie par la dérivée de la fonction de coût (voir [13]).

Ainsi, lorsqu'on utilise la fonction quadratique comme fonction de coût pour le lissage du champs, les discontinuités (un grand écart entre les déplacements voisins) sont interdites puisque la fonction d'influence est monotone croissante.

Par contre, une fonction robuste a un champs d'influence limité car sa dérivée converge vers 0. Il est intéressant d'utiliser une fonction de coût robuste puisqu'elle devrait permettre les discontinuités dans le champ de déplacement. La fonction quadratique tronquée en est un exemple. Elle est définie par :

$$QT_\sigma(x) = \begin{cases} x^2 & x < \sigma \\ \sigma^2 & x \geq \sigma \end{cases} \quad (4.3.50)$$

La fonction quadratique tronquée (figure 4.4) influence seulement pour les valeurs qui sont inférieures à σ en valeur absolue ($\sigma = 2$ dans l'exemple). Afin d'utiliser une fonction de coût robuste, solutionnons d'abord le problème du flot optique pour une fonction quelconque [2].

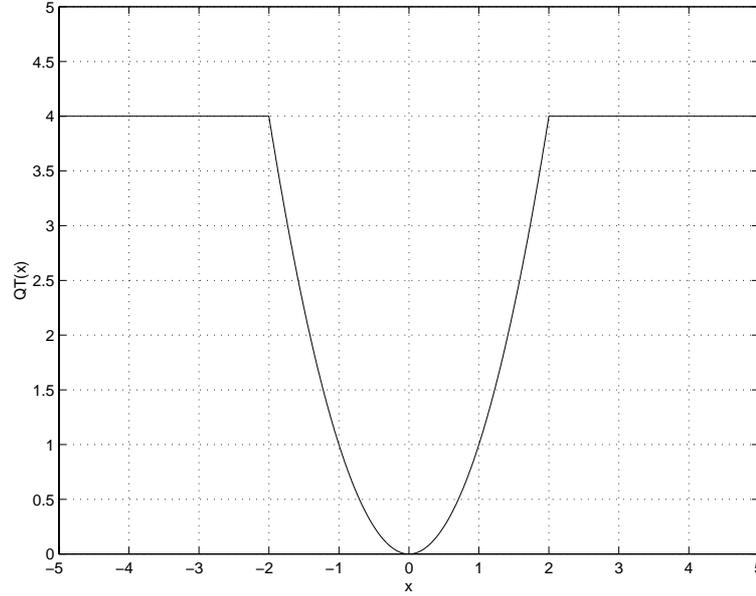
Soient $I(i, j, 0) : \mathbb{N}^2 \rightarrow \mathbb{R}$, $I(i, j, 1) : \mathbb{N}^2 \rightarrow \mathbb{R}$ les images numériques, $u_{ij} : \mathbb{N}^2 \rightarrow \mathbb{R}$, $v_{ij} : \mathbb{N}^2 \rightarrow \mathbb{R}$ les mouvements estimés pour le pixel (i, j) et $\alpha \in [0, 1]$ une position arbitraire entre l'image au temps 0 et l'image au temps 1. Nous définissons la fonction de coût total comme :

$$E = \sum_{i,j} \left[(\tilde{I}_t(i, j))^2 + \sum_{(k,l) \in \nu_{ij}} [\lambda_u \rho(u_{ij} - u_{kl}, \sigma) + \lambda_v \rho(v_{ij} - v_{kl}, \sigma)] \right] \quad (4.3.51)$$

où

$$\begin{aligned} \tilde{I}_t(i, j) := & I(i + (1 - \alpha)u_{ij}, j + (1 - \alpha)v_{ij}, 1) \\ & - I(i + -\alpha u_{ij}, j - \alpha v_{ij}, 0) \end{aligned} \quad (4.3.52)$$

$\rho(x, \sigma)$ est une fonction robuste de paramètre σ . λ_u et λ_v sont les paramètres de lissage des mouvements horizontaux et verticaux.

FIG. 4.4 – Quadratique Tronquée en $x=\pm 2$.

En utilisant une descente de gradient de type relaxation non linéaire (SOR [3]) pour minimiser l'équation (4.3.51) on obtient :

$$u_{ij}^{(n+1)} = u_{ij}^{(n)} - \frac{\omega}{T(u_{ij})} \frac{\partial E}{\partial u_{ij}} \quad (4.3.53)$$

$$v_{ij}^{(n+1)} = v_{ij}^{(n)} - \frac{\omega}{T(v_{ij})} \frac{\partial E}{\partial v_{ij}} \quad (4.3.54)$$

où $\omega \in (0, 2]$. ω est un paramètre qui peut être calculé (voir [3]) pour optimiser la vitesse de convergence et, par définition :

$$T(u_{ij}) \geq \sup \left(\frac{\partial^2 E}{\partial u_{ij}^2} \right), \quad \forall (i, j) \quad (4.3.55)$$

$$T(v_{ij}) \geq \sup \left(\frac{\partial^2 E}{\partial v_{ij}^2} \right), \quad \forall (i, j) \quad (4.3.56)$$

$T(u)$ et $T(v)$ ont pour effet de diminuer la longueur du pas dans les endroits où la dérivé varie rapidement ; la prudence est de mise lorsque la dérivé est instable.

Explicitement, les différentes composantes des équations (4.3.53) et (4.3.54) sont :

$$\frac{\partial E}{\partial u_{mn}} = 2\tilde{I}_t(m, n)\frac{\partial \tilde{I}_t(m, n)}{\partial u_{mn}} + 2 \sum_{(k,l) \in \nu_{mn}} \lambda_u \phi(u_{mn} - u_{kl}, \sigma) \quad (4.3.57)$$

$$\begin{aligned} \frac{\partial \tilde{I}_t(m, n)}{\partial u_{mn}} &= (1 - \alpha) \frac{\partial}{\partial x} I(i + (1 - \alpha)u_{mn}, j + (1 - \alpha)v_{mn}, 1) \\ &\quad + \alpha \frac{\partial}{\partial x} I(i - \alpha u_{mn}, j - \alpha v_{mn}, 0) \\ &:= \tilde{I}_x(m, n) \end{aligned} \quad (4.3.58)$$

où $\phi(x, \sigma) = \frac{d\rho}{dx}$. L'équation (4.3.58) est renommé $\tilde{I}_x(m, n)$. L'équation (4.3.58) est une combinaison linéaire des dérivés horizontales évaluées aux positions initiales et finales. Nous avons donc :

$$\frac{\partial E}{\partial u_{mn}} = 2\tilde{I}_t(m, n)\tilde{I}_x(m, n) + 2 \sum_{(k,l) \in \nu_{mn}} \lambda_u \phi(u_{mn} - u_{kl}, \sigma) \quad (4.3.59)$$

Puis, de la même façon :

$$\frac{\partial E}{\partial v_{mn}} = 2\tilde{I}_t(m, n)\tilde{I}_y(m, n) + 2 \sum_{(k,l) \in \nu_{mn}} \lambda_v \phi(v_{mn} - v_{kl}, \sigma) \quad (4.3.60)$$

où :

$$\tilde{I}_y(m, n) := \frac{\partial \tilde{I}_t(m, n)}{\partial v_{mn}} \quad (4.3.61)$$

Pour calculer $T(u)$ et $T(v)$ nous devons calculer la dérivé seconde de E par rapport à u et par rapport à v .

$$\frac{\partial^2 E}{\partial u_{mn}^2} = 2\tilde{I}_x^2(m, n) + 2\tilde{I}_t(m, n)\frac{\partial \tilde{I}_x(m, n)}{\partial u_{mn}} + 2 \sum_{(k,l) \in \nu_{mn}} \lambda_u \phi'(u_{mn} - u_{kl}, \sigma) \quad (4.3.62)$$

$$\frac{\partial^2 E}{\partial v_{mn}^2} = 2\tilde{I}_y^2(m, n) + 2\tilde{I}_t(m, n)\frac{\partial \tilde{I}_y(m, n)}{\partial v_{mn}} + 2 \sum_{(k,l) \in \nu_{mn}} \lambda_v \phi'(v_{mn} - v_{kl}, \sigma) \quad (4.3.63)$$

Nous pouvons donc poser $T(u)$ et $T(v)$:

$$T(u) = 2 \left(\tilde{I}_x^2(m, n) + \tilde{I}_t(m, n)\tilde{I}_{xx}(m, n) + 4\lambda_u \sup(\phi'(x, \sigma)) \right) \quad (4.3.64)$$

$$T(v) = 2 \left(\tilde{I}_y^2(m, n) + \tilde{I}_t(m, n)\tilde{I}_{yy}(m, n) + 4\lambda_v \sup(\phi'(x, \sigma)) \right) \quad (4.3.65)$$

où :

$$\begin{aligned} \tilde{I}_{xx}(m, n) = (1 - \alpha)^2 I_{xx}(m + (1 - \alpha)u_{mn}, n + (1 - \alpha)v_{mn}, 1) \\ - \alpha^2 I_{xx}(m - \alpha u_{mn}, n - \alpha v_{mn}, 0) \end{aligned} \quad (4.3.66)$$

$$\begin{aligned} \tilde{I}_{yy}(m, n) = (1 - \alpha)^2 I_{yy}(m + (1 - \alpha)u_{mn}, n + (1 - \alpha)v_{mn}, 1) \\ - \alpha^2 I_{yy}(m - \alpha u_{mn}, n - \alpha v_{mn}, 0) \end{aligned} \quad (4.3.67)$$

Le processus itératif de relaxation complet, pour un ρ quelconque, est le suivant :

$$u_{ij}^{n+1} = u_{ij}^n - \frac{\tilde{I}_t(i, j)\tilde{I}_x(i, j) + \lambda_u \sum_{(k,l) \in \nu_{ij}} \phi(u_{ij} - u_{kl}, \sigma)}{\tilde{I}_x^2(i, j) + \tilde{I}_t(i, j)\tilde{I}_{xx}(i, j) + 4\lambda_u \sup(\phi'(x, \sigma))} \quad (4.3.68)$$

$$v_{ij}^{n+1} = v_{ij}^n - \frac{\tilde{I}_t(i, j)\tilde{I}_y(i, j) + \lambda_v \sum_{(k,l) \in \nu_{ij}} \phi(v_{ij} - v_{kl}, \sigma)}{\tilde{I}_y^2(i, j) + \tilde{I}_t(i, j)\tilde{I}_{yy}(i, j) + 4\lambda_v \sup(\phi'(x, \sigma))} \quad (4.3.69)$$

Les équations (4.3.64) et (4.3.65) respectent les conditions (4.3.55) et (4.3.56) dans la mesure où ϕ' est une fonction bornée. Ceci n'est pas le cas avec la fonction quadratique tronquée car sa dérivée n'est pas continue. Considérons donc la fonction robuste de Geman-McLure :

$$\rho(x, \sigma) = \frac{x^2}{\left(\frac{x}{\sigma}\right)^2 + 1} \quad (4.3.70)$$

Nous nous intéresserons surtout à sa dérivée (fonction d'influence), car seul ϕ apparait dans les équations (4.3.68) et (4.3.69) :

$$\phi(x, \sigma) = \frac{\partial \rho(x, \sigma)}{\partial x} = \frac{2x}{\left(\left(\frac{x}{\sigma}\right)^2 + 1\right)^2} \quad (4.3.71)$$

Dans les cas limites, ($|x| \ll \sigma$ ou $|x| \gg \sigma$) les fonctions de Geman-McLure se comportent comme la fonction quadratique tronquée (voir figure 4.5). Bien sur, les fonctions de Geman-McLure ont une dérivée seconde bornée car la fonction ϕ est continue et bornée.

Il est aussi possible de généraliser les fonctions de Geman-McLure afin d'augmenter la ressemblance. Il suffit d'augmenter le degré du dénominateur afin qu'il converge plus rapidement vers 1 si $|x| \ll \sigma$ ou vers $\left(\frac{x}{\sigma}\right)^2$ si $|x| \gg \sigma$. Soit la famille de fonctions :

$$\rho_n(x, \sigma) = \frac{x^2}{\left(\left(\frac{x}{\sigma}\right)^{2n} + 1\right)^{1/n}}, \quad n \in \mathbb{N} \quad (4.3.72)$$

$$\phi_n(x, \sigma) = \frac{2x}{\left(\left(\frac{x}{\sigma}\right)^{2n} + 1\right)^{(1+1/n)}}, \quad n \in \mathbb{N} \quad (4.3.73)$$

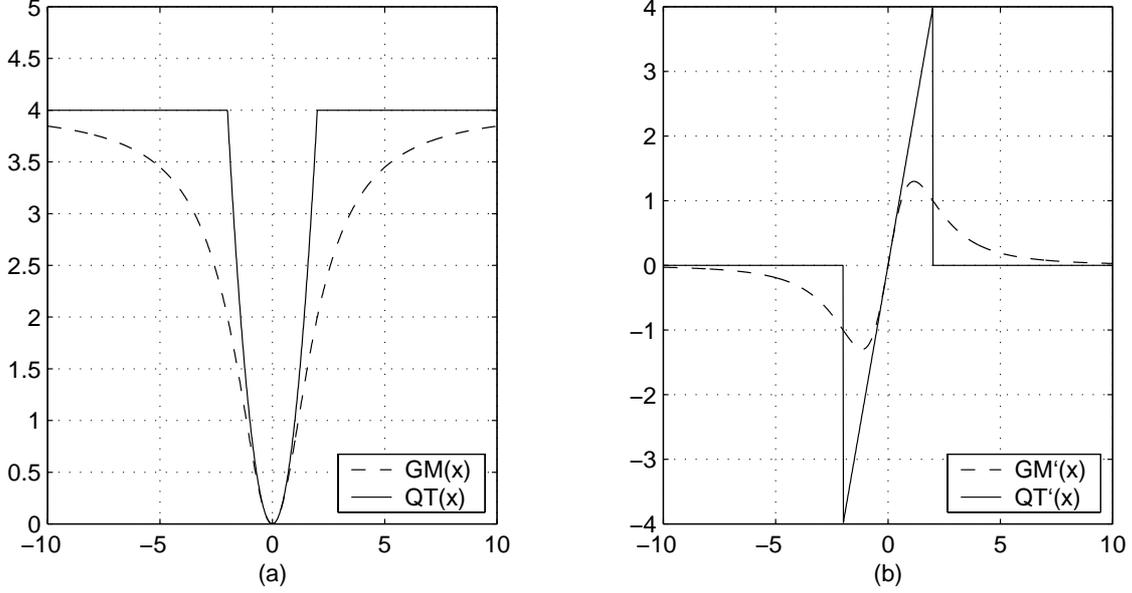


FIG. 4.5 – (a) Comparaison entre la fonction Geman-McLure et la fonction Quadratique Tronquée (b) et leurs dérivés.

Plus n est grand, plus la fonction de coût ρ_n converge vers la fonction quadratique tronquée, et plus sa fonction d'influence ϕ_n converge vers la dérivée de celle-ci (figure 4.6).

En supposant que l'on utilise la fonction de Geman-McLure généralisée ρ_n , les équations (4.3.68) et (4.3.69) peuvent se réécrire :

$$u_{ij}^{n+1} = u_{ij}^n - \frac{\tilde{I}_t(i, j)\tilde{I}_x(i, j) + \lambda_u \sum_{(k,l) \in \nu_{ij}} \phi_n(u_{ij} - u_{kl}, \sigma)}{\tilde{I}_x^2(i, j) + \tilde{I}_t(i, j)\tilde{I}_{xx}(i, j) + 8\lambda_u} \quad (4.3.74)$$

$$v_{ij}^{n+1} = v_{ij}^n - \frac{\tilde{I}_t(i, j)\tilde{I}_y(i, j) + \lambda_v \sum_{(k,l) \in \nu_{ij}} \phi_n(v_{ij} - v_{kl}, \sigma)}{\tilde{I}_y^2(i, j) + \tilde{I}_t(i, j)\tilde{I}_{yy}(i, j) + 8\lambda_v} \quad (4.3.75)$$

Puisque ρ n'est pas une fonction convexe, et que \tilde{I}_t n'est pas linéaire, la fonction de coût total E a plusieurs minima locaux. Afin de trouver la meilleure solution, une approche hiérarchique est utilisée avec un σ suffisamment grand pour que ρ soit toujours localement convexe. Si σ est suffisamment supérieur à toutes les différences de déplacement qu'il peut y avoir entre deux pixels voisins la fonction ρ sera toujours localement convexe.

On utilise ensuite cette approximation dite convexe pour recommencer l'estimation sur les images originales avec un σ plus petit. En diminuant progressivement le σ , on obtient des discontinuités de plus en plus fines.

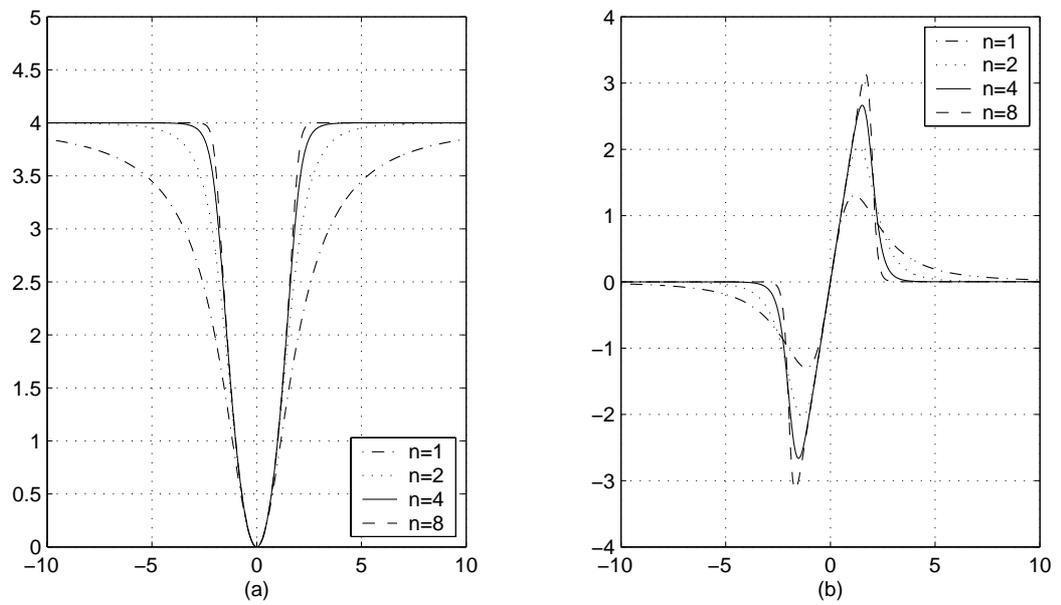


FIG. 4.6 – (a) Comparaison des différentes puissances de fonctions Geman-McLure généralisées (b) et de leurs dérivés.

4.4 Reconstruction d'images intermédiaires

Un champ de disparité décrit le déplacement de luminosité d'une image à l'autre. En étudiant ces déplacements, il est possible de reconstruire des images intermédiaires à l'aide d'une interpolation compensée par la disparité. Tout dépendant de la position du pivot du champ de déplacement (c'est-à-dire α), différentes stratégies de reconstruction seront utilisées.

4.4.1 Reconstruction exacte

On appelle reconstruction exacte la reconstruction d'une image située exactement à l'endroit où le champ de vecteur a été calculé (c'est-à-dire son point de pivot). Dans ce cas, tous les vecteurs du champs de déplacement passent par une grille d'échantillonnage uniforme (voir figure 4.7).

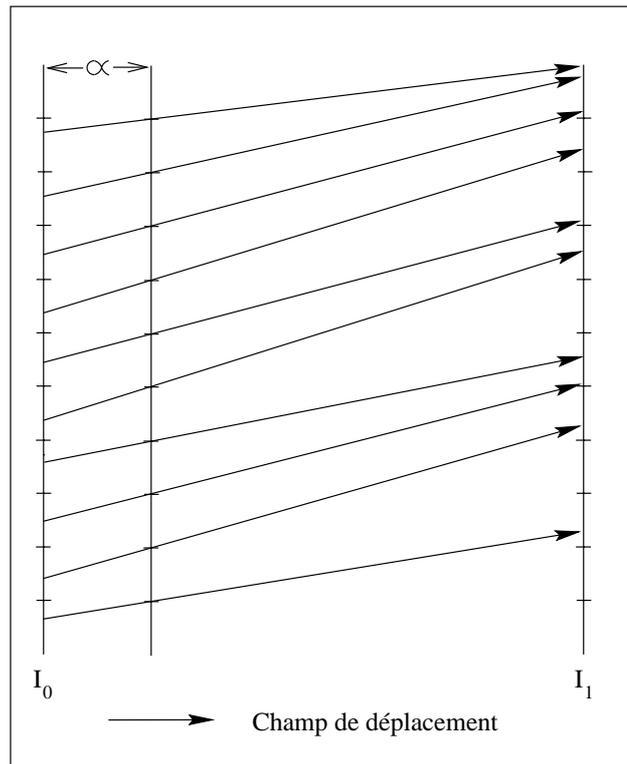


FIG. 4.7 – Reconstruction exacte.

Alors la reconstruction $R_\alpha(i, j)$ peut s'écrire :

$$R_\alpha(i, j) = (1 - \alpha)I_0(i - \alpha u_\alpha(i, j), j - \alpha v_\alpha(i, j)) + \alpha I_1(i + (1 - \alpha)u_\alpha(i, j), j + (1 - \alpha)v_\alpha(i, j)) \quad (4.4.1)$$

$\alpha \in [0, 1]$ étant la position entre les deux images originales I_0 et I_1 où nous voulons calculer la reconstruction R_α . Le champ de vecteur (u_α, v_α) doit avoir été évalué à la position α . $R_\alpha(i, j)$ est une combinaison linéaire du point où le vecteur $(u_\alpha(i, j), v_\alpha(i, j))$ débute sur l'image I_0 et le point où il se termine sur l'image I_1 . Plus de poids sera accordé à l'image originale la plus proche.

Pour reconstruire des images avec la reconstruction exacte, le champ de vecteur doit avoir été préalablement calculé pour chacune des positions. Pour générer 20 images intermédiaires, l'algorithme d'estimation de disparité devra être utilisé 20 fois, et ce en prenant une valeur de α appropriée pour chacune des reconstructions.

4.4.2 Reconstruction par propagation

Dans le cas où le champ de vecteur ne peut être évalué à chacune des positions, nous devons utiliser une autre approche. La reconstruction par propagation utilise un seul champ de vecteur évalué en α_0 (habituellement $\alpha_0=0.5$) puis utilisé pour toutes les images reconstruites :

$$R_\alpha(i, j) = (1 - \alpha)I_0(i - \alpha u_{\alpha_0}(i, j), j - \alpha v_{\alpha_0}(i, j)) + \alpha I_1(i + (1 - \alpha)u_{\alpha_0}(i, j), j + (1 - \alpha)v_{\alpha_0}(i, j)) \quad (4.4.2)$$

Ainsi, il est supposé que $u_\alpha(i, j) = u_{\alpha_0}(i, j)$ pour chacun des pixels (i, j) . La figure 4.8 démontre l'effet de cette supposition. Puisque le temps de calcul de la reconstruction est négligeable par rapport à celui de l'estimation de disparité et qu'ici le champ de disparité n'est calculé qu'une seule fois, le temps de calcul total est réduit d'un facteur égal au nombre d'images à reconstruire.

Si nous estimons le champ en $\alpha_0 = 0.5$, l'approximation sera évidemment valide pour $\alpha \approx 0.5$. Pour le cas où $\alpha \approx 0$ (ou $\alpha \approx 1$), beaucoup plus d'importance sera accordée au petit segment du vecteur et ainsi R_α ressemblera beaucoup à I_0 (ou à I_1). Pour le cas où $\alpha \approx 0.25$ (ou $\alpha \approx 0.75$) l'approximation ne devrait pas causer de problème majeur dans la mesure où le champ de disparité est suffisamment lisse.

4.4.3 Reconstruction non-uniforme

Une autre option existe dans le cas où le champ de vecteur a été déterminé à une seule position. Plutôt que de considérer le champ à la position α comme étant égal à celui en α_0 nous pouvons prendre en considération les véritables intersections entre le plan de reconstruction et le champ de vecteur calculé. La figure 4.9 nous en donne un exemple.

En termes mathématiques, l'intersection du vecteur située à la position (i, j) du champ centré en α_0 et du plan de reconstruction situé en α est : $(i + (\alpha - \alpha_0)u_{\alpha_0}(i, j), j + (\alpha - \alpha_0)v_{\alpha_0}(i, j))$. Un tel point a très peu de chances de se retrouver sur la grille

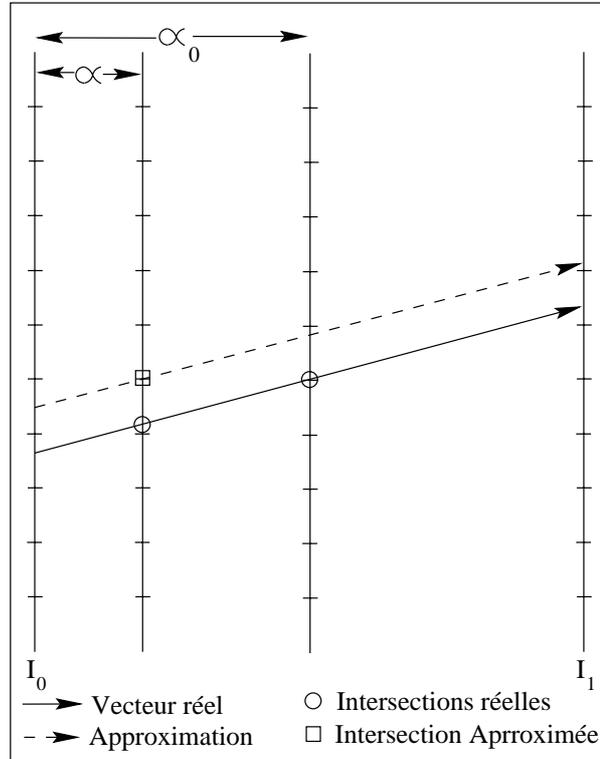


FIG. 4.8 – Reconstruction par propagation.

d'échantillonnage et formera donc, avec toutes les autres intersections, un ensemble non-uniforme d'échantillons :

$$R_\alpha(i + (\alpha - \alpha_0)u_{\alpha_0}, j + (\alpha - \alpha_0)v_{\alpha_0}) = (1 - \alpha)I_0(i - \alpha_0u_{\alpha_0}, j - \alpha_0v_{\alpha_0}) + \alpha I_1(i + (1 - \alpha_0)u_{\alpha_0}, j + (1 - \alpha_0)v_{\alpha_0}) \quad (4.4.3)$$

Ainsi la reconstruction R_α est connue sur un ensemble de points semblables à la partie droite de la figure 4.9. Il est nécessaire de reconstruire une grille uniforme afin que l'image soit d'un format lisible. Pour ce faire, une triangulation de Denaulay est effectuée. Pour bien comprendre la triangulation de Denaulay, nous passerons par le concept du diagramme de Voronoï.

Soit S un ensemble de N points d'un espace euclidien E^n et soit le point $p \in S$. La cellule de Voronoï du point p , notée $V(p)$, est l'ensemble des points éléments de E^n qui sont plus près de p que de n'importe lequel autre point de S . Le diagramme de Voronoï est la décomposition de E^n en N cellules de Voronoï.

La triangulation de Denaulay est la décomposition de E^2 en triangles dont les arêtes sont des segments de droites reliant les points de l'ensemble S qui sont éléments de deux cellules de Voronoï voisines.

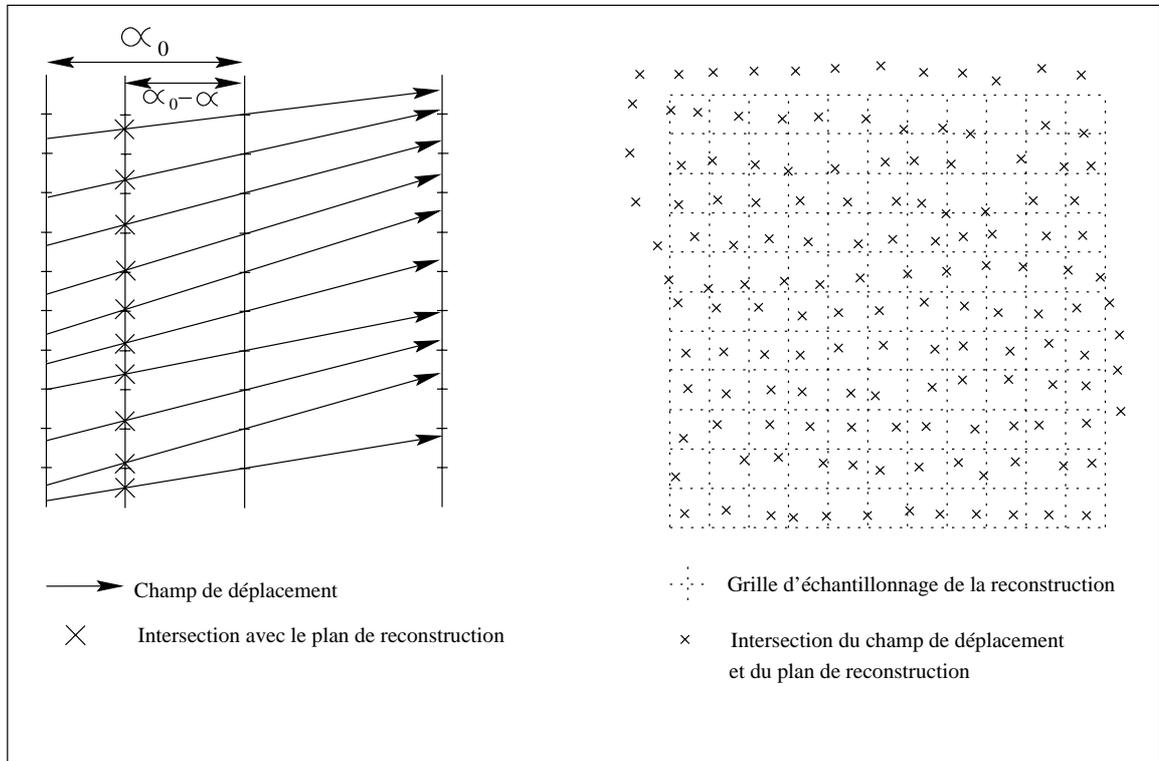


FIG. 4.9 – Reconstruction non-uniforme.

La figure 4.10 présente un exemple de triangulation de Denaulay. Étant donné l'ensemble de points S en (a), tous les segments de droites sont tracés par relier tous les points de l'ensemble en (b). Seul ceux qui correspondent à une frontière de Voronoï sont conservés en (c) pour former la triangulation de Denaulay en (d).

Une fois l'image séparée en triangles, la plus grande partie des pixels se trouve à l'intérieur d'un triangle. La valeur de ces pixels est déterminée à l'aide du plan qui intersecte les trois sommets du triangle. Si un pixel se trouve sur une arête d'un triangle, une simple interpolation linéaire donnera la valeur de la reconstruction à ce pixel. Et pour les pixels qui se trouvent sur un sommet, la valeur du pixel sera égale à ce dernier.

Ainsi, il est possible de reconstruire une image définie sur une grille d'échantillonnage uniforme à partir d'un échantillonnage non-uniforme.

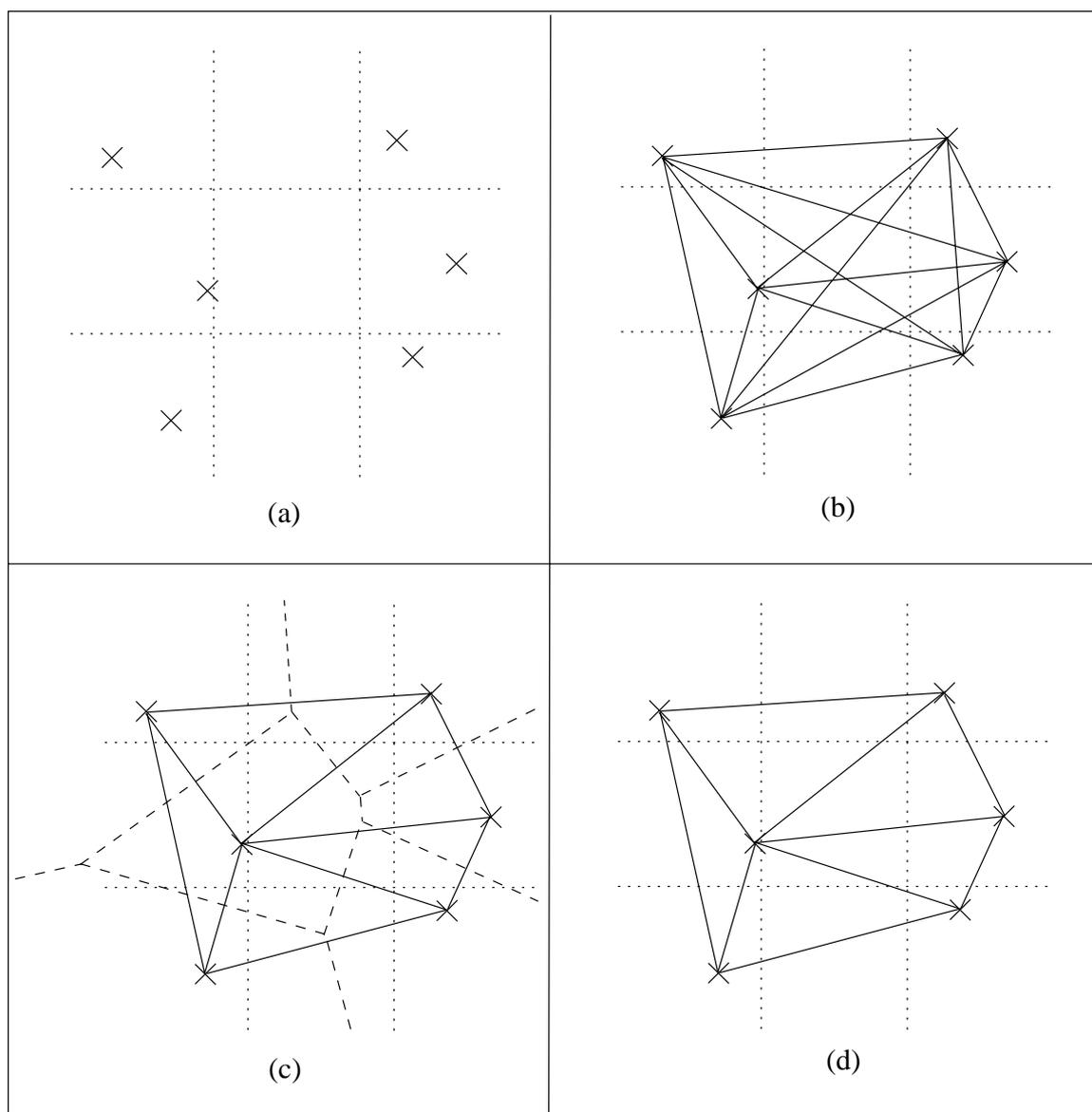


FIG. 4.10 – triangulation de Delaunay.

Chapitre 5

Présentation des résultats

Trois séquences d'images seront utilisées et quatre méthodes seront mises à l'épreuve sur chacune des images. Tout dépendant de la nature de la séquence d'images, différents tests seront utilisés pour comparer les approches d'estimation de disparité.

Les quatre méthodes comparées sont les suivantes : March, Horn & Schunck, relaxation convexe et relaxation robuste. L'algorithme dit de March est utilisé tel qu'il est exposé dans la section 4.3.1. L'algorithme de Horn & Schunck est décrit à la section 4.3.2. Les algorithmes de relaxation sont décrits à la section 4.3.3. La relaxation convexe est l'approximation initiale ($\sigma = \sigma_M$) utilisée pour la relaxation robuste.

Puisque que le format de ce document nécessitait une modification des dimensions des images présentées, que l'impression est une forme de filtrage et que certains résultats sont présenté seulement sous forme de différences, tous les résultats se retrouvent en version originale sur la page web suivante :

www.inrs-telecom.quebec.ca/users/viscom/francais/publications/Finn02thesis

5.1 La séquence d'images *Birds*

La séquence d'images *Birds* présentée à la figure 5.1 est composée de l'image originale (*Birds* gauche) de dimension 480×256 et d'une seconde image créée à partir de la première, avec une translation de deux pixels vers la gauche d'un rectangle englobant l'oiseau de gauche (figure 5.2).

Birds gauche*Birds droite*FIG. 5.1 – Séquence d'images *Birds*.

Birds gauche



Birds droite



FIG. 5.2 – Séquence d'images *Birds* avec le rectangle de translation.

5.1.1 Comparaison du champ de vecteurs

Pour comparer la qualité du champ de vecteur, la valeur exacte du champ réel est requise, ce qui est impossible pour une séquence d'images réelles. Ici, nous utilisons une séquence de mouvement synthétique. Ainsi, les valeurs du champ de disparité exact de la séquence *Birds* sont connues et il est facile de donner une mesure sur la qualité du champ de disparité : l'erreur quadratique moyenne (EQM).

$$EQM(\vec{D}, \vec{d}) = \sum_{i=1}^M \sum_{j=1}^N \frac{\|\vec{D}(i, j) - \vec{d}(i, j)\|^2}{MN} \quad (5.1.1)$$

où M et N sont les dimensions de l'image.

La figure 5.3 représente la composante horizontale du champ de vecteur idéal. Les figures 5.4 à 5.7 représentent la composante horizontale des champs de vecteurs obtenus avec chacune des méthodes. Dans ces figures, un point blanc signifie $u = 0$ et un point noir $u = -2$.



FIG. 5.3 – Composante horizontale du champ de vecteur idéal.

	EQM
March	0.0349
Horn & Schunck	0.0235
Relaxation convexe	0.0371
Relaxation robuste	0.0079

TAB. 5.1 – Comparaison quantitative des champs de déplacement de *Birds*.

L'algorithme de relaxation convexe est l'approche qui donne un champ de disparité



FIG. 5.4 – Composante horizontale du champ de vecteur calculé avec la méthode de March.

le moins conforme. Par contre, il est un excellent point de départ pour l'algorithme de relaxation robuste qui a dépassé d'un facteur 3 toutes les autres approches.

5.1.2 Reconstruction d'une image existante

Une fois le champ de vecteur calculé, il est possible de reconstruire des images existantes à partir de l'image originale. Pour ce faire, le champ de disparité est calculé avec un pivot sur l'image droite ($\alpha = 1$), puis, à l'aide de ce champ et de l'image gauche, l'image droite est reconstruite. Soient u_{ij} et v_{ij} les composantes horizontale et verticale du champ de vecteur au pixel (i, j) et I_0 l'image gauche, alors la reconstruction R_1 est :

$$R_1(i, j) = I_0(i - u_{ij}, j - v_{ij}) \quad (5.1.2)$$

Ainsi, il est possible de comparer quantitativement l'image reconstruite R_1 avec la véritable image droite I_1 . Comme mesure, prenons le rapport signal à bruit de pointe (RSBP) :

$$RSBP(R_1, I_1) = 20 \cdot \log_{10} \left(\frac{255}{\sqrt{EQM(R_1, I_1)}} \right) \quad (5.1.3)$$

En comparant plusieurs images reconstruites, celle qui aura le RSBP le plus élevé sera de meilleure qualité (donc celle qui ressemblera le plus à l'originale). La figure 5.8 montre l'erreur des différentes approches. Chacune des approches est comparée à l'image originale (*Birds* droite) puis la différence est présentée. Un code de couleur est utilisé pour représenter cette différence. Un ton de gris moyen, signifie que l'erreur

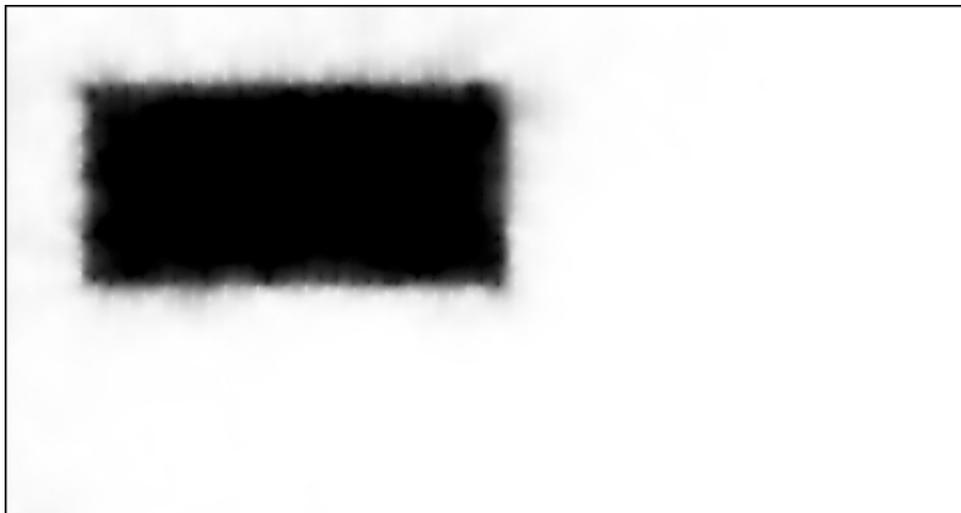


FIG. 5.5 – Composante horizontale du champ de vecteur calculé avec la méthode de Horn & Schunck.

	RSBP
March	35.5065 dB
Horn & Schunck	59.0535 dB
Relaxation convexe	56.8907 dB
Relaxation robuste	70.4162 dB

TAB. 5.2 – Comparaison quantitative des reconstructions de *Birds* droite.

est près de 0. Plus la différence est inférieure à 0 pour un pixel donné, plus il sera blanc. Plus la différence est supérieure à 0 pour un pixel donné, plus il sera noir. Il y a saturation de noir ou de blanc à ± 9 .

Les RSBP obtenus démontrent qu'il y a un problème avec la l'approche de March. L'image de différence de la figure 5.8 présente des distortions aux becs, aux pattes et à l'oeil de l'oiseau de gauche puis à tout l'oiseau de droite.

La relaxation convexe et l'approche de Horn & Schunck donne des résultats similaires. Donc, le surlissage du champ de disparité obtenu à l'aide de la relaxation convexe ne s'est pas fait au dépend de l'intensité lumineuse. Les données numériques démontrent qu'il y a eu une amélioration de la qualité de la reconstruction pour l'approche de la relaxation robuste. Cette qualité supérieure n'est pas visible à l'oeil nu mais est visible sur les images de différence (figure 5.8b) et 5.8c)). La figure 5.9 présente la reconstruction offrant le meilleur RSBP soit celle obtenue avec la relaxation robuste.

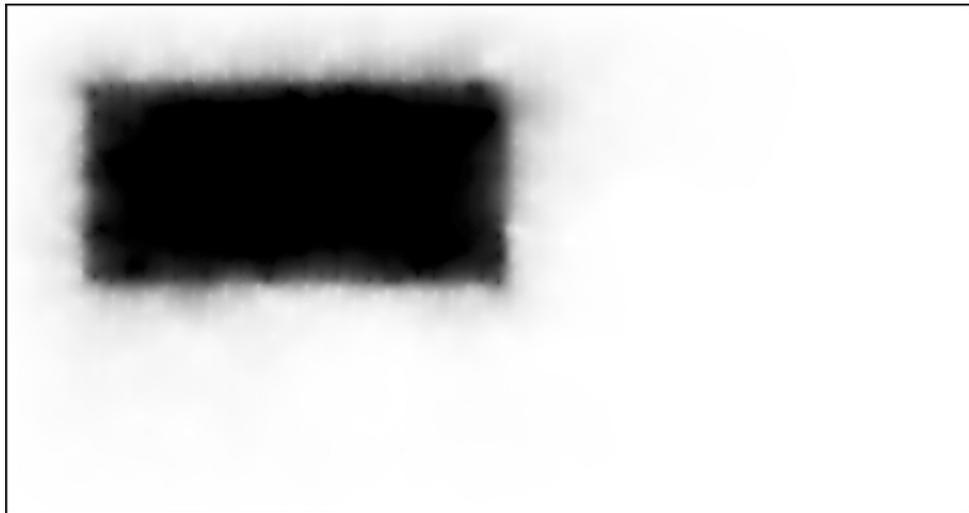


FIG. 5.6 – Composante horizontale du champ de vecteur calculé avec la relaxation convexe.

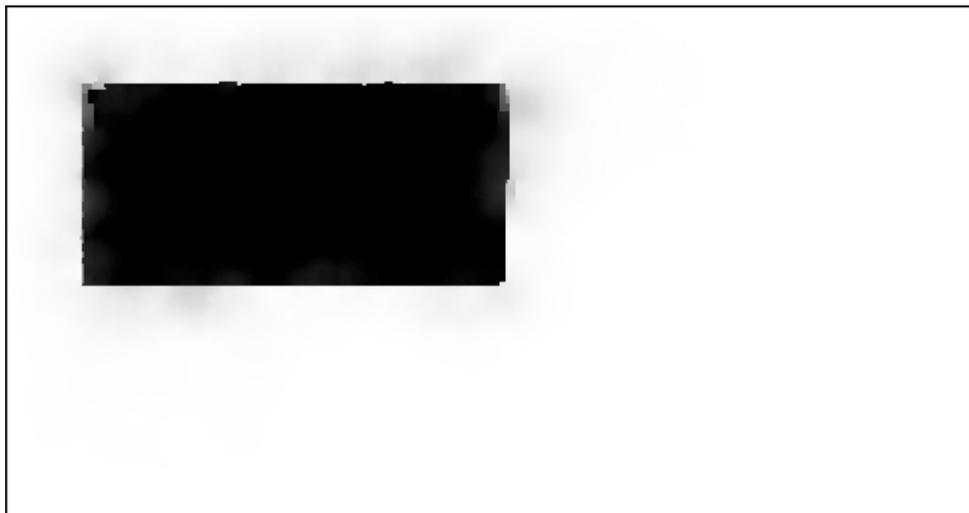


FIG. 5.7 – Composante horizontale du champ de vecteur calculé avec la relaxation robuste.

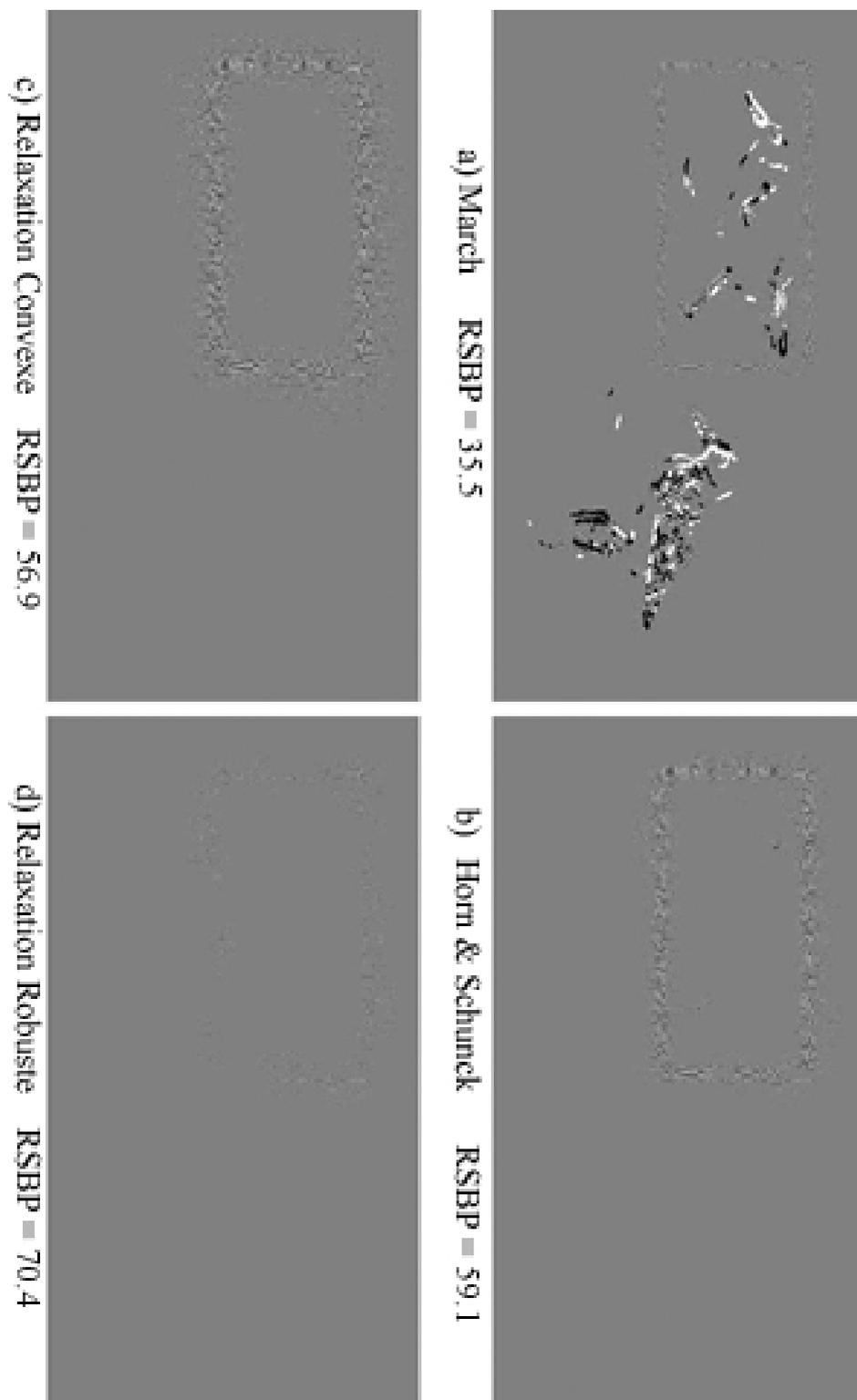


FIG. 5.8 – Erreurs dans la reconstruction de *Birds* pour l'approche de a) March b) Horn & Schunck a) la relaxation convexe b) la relaxation robuste.



FIG. 5.9 – Reconstruction de *Birds* droite à l'aide de la relaxation robuste.

5.2 La séquence d'images *Flower*



FIG. 5.10 – Image stéréoscopique *Flower* gauche.

La séquence d'images *Flower* (figure 5.10 et 5.11), est une paire d'images stéréoscopiques qui ont des dimensions de 720×480 .

5.2.1 Reconstruction d'une image existante

Pour des images stéréoscopiques réelles telle que *Flower*, il est impossible de comparer les champs de vecteurs puisque le champ idéal est inconnu. La reconstruction d'images existantes devient donc une étape cruciale. Avant l'interprétation quantitative du RSBP, une interprétation qualitative est nécessaire afin de déterminer les faiblesses des différentes approches. Pour ce faire, l'image reconstruite est comparée à l'image originale en affichant alternativement l'une et l'autre sur un même écran. Ceci permet de trouver les différences entre les deux images. Pour faire de même, le lecteur peut télécharger les différents résultats à la page web suivante :

www.inrs-telecom.quebec.ca/users/viscom/francais/publications/Finn02thesis

La figure 5.12 indique les différentes zones problématiques. Lorsqu'une zone sera décrite pour une approche en particulier, c'est à cette figure qu'on se réfère.



FIG. 5.11 – Image stéréoscopique *Flower* droite.

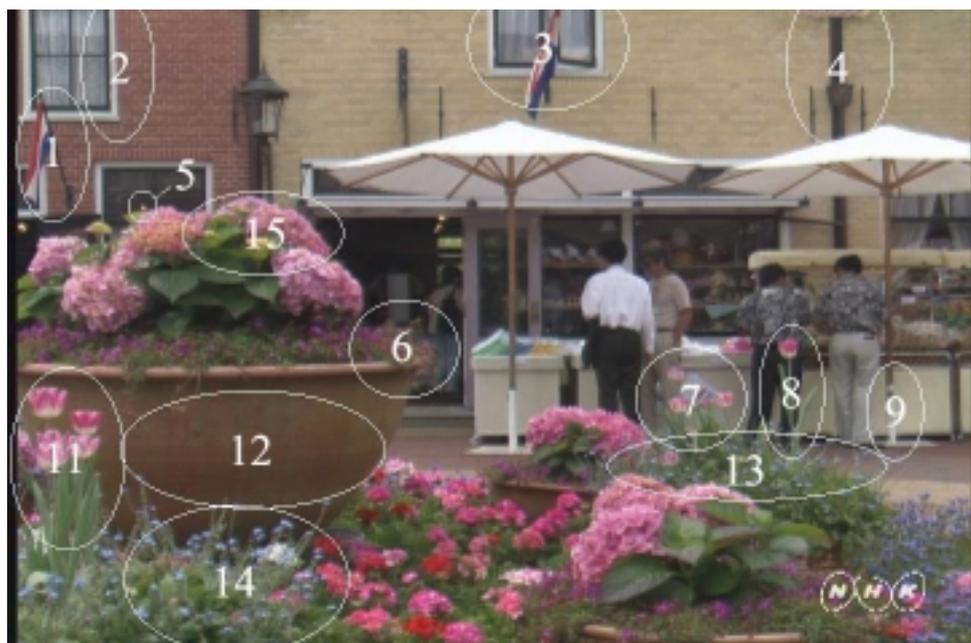


FIG. 5.12 – Zones de références de *Flower*.

Dans la reconstruction de *Flower* droite à l'aide de l'approche de March (figure 5.13a) le principal défaut est évidemment le bruit visible un peu partout dans l'image. Ce bruit se retrouve surtout où il y a des discontinuités dans la luminosité de l'image. Cette distorsion est particulièrement observable sur les drapeaux et sur les cadres de fenêtres. Le point jaune sous la zone 1 a disparu, ce qui est compréhensible puisqu'il n'est pas dans l'image originale de gauche. De légères déformations du pot (zone 12), des fleurs sur le bord de ce pot (zone 6) et des fleurs de la zone 7 sont notées. Il y a aussi une déformation majeure des fleurs et des tiges dans les zones 8, 11 et 14. De plus, il y a un effet de bord surtout notable sur la bordure gauche.

Dans la reconstruction de *Flower* droite à l'aide de l'approche de Horn & Schunck (figure 5.13b), il n'y a pas d'erreur globale comme dans l'approche de March. On remarque de légères déformations des drapeaux dans les zones 1 et 3 et du poteau de la zone 4. Des déformations sont présentes où il devrait y avoir des discontinuités dans le champ de disparité, soit le point jaune de la zone 5 et les fleurs des zones 6, 7, 8, 13 et 15. Les fleurs qui se trouvent à l'avant du pot (zones 11 et 14) semblent floues. De façon générale, cette reconstruction reste beaucoup plus agréable à l'oeil que celle de l'approche de March.

Dans la reconstruction de *Flower* droite à l'aide de la relaxation convexe (figure 5.13c), des déformations flagrantes du cadre de la fenêtre de la zone 2, du drapeau de la zone 3 et du poteau de la zone 4 se font tout d'abord remarquer. Le pied du parasol se trouvant dans la zone 9 semble aussi affecté par ce type de déformations. Encore une fois, il y a des problèmes aux endroits où il y a des discontinuités dans le champ de disparité : l'effet est léger sur les fleurs des zones 6, 7 et 15 mais est important sur la fleur et sa tige dans la zone 8. Les fleurs dans les zones 11 et 14 semblent encore une fois déformées et floues. Il y a un effet de bord assez prononcé sur la bordure du bas et faible sur la bordure du haut.

Dans la reconstruction de *Flower* droite à l'aide de la relaxation robuste (figures 5.13d et 5.14), les déformations du cadre de la fenêtre de la zone 2, du drapeau de la zone 3 et du poteau de la zone 4 sont toujours présentes mais moins importantes que pour la relaxation convexe. Les problèmes aux discontinuités du champ de disparité (zones 6, 7, 8 et 15) sont beaucoup moins marqués que dans le cas précédent. Toutefois, les bordures de la tige et des feuilles de la fleur de la zone 11 sont encore déformées.

	RSBP
March	27.4294 dB
Horn & Schunck	26.1551 dB
Relaxation convexe	29.1016 dB
Relaxation robuste	33.4741 dB

TAB. 5.3 – Comparaison quantitative des reconstructions de *Flower* droite.

La figure 5.14 présente la reconstruction offrant le meilleur RSBP, soit celle obtenue avec la relaxation robuste.

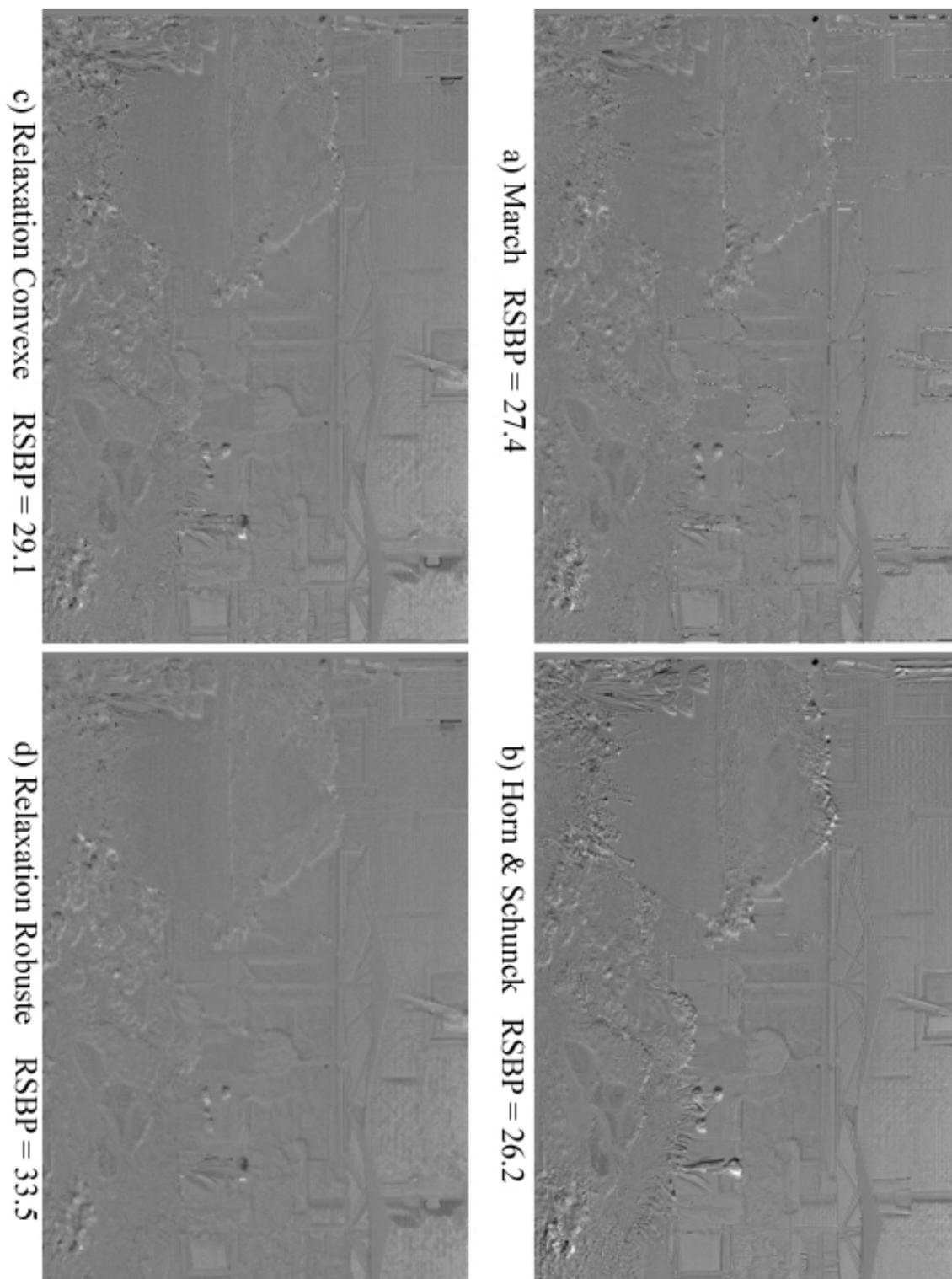


FIG. 5.13 – Erreurs dans la reconstruction de *Flower* droite pour l'approche de a)March b)Horn & Schunck c)la relaxation convexe d)la relaxation robuste. (Blanc=128 Noir=128)



FIG. 5.14 – Reconstruction de *Flower* droite à l'aide de la relaxation robuste.

5.2.2 Reconstruction d'images intermédiaires

Comparons maintenant les reconstructions de vues intermédiaires, soient des images reconstruites à l'aide des deux images originales et du champ de déplacement calculé à l'aide des différentes méthodes. La méthode de reconstruction utilisée est telle que décrite à la section 4.4.3. Il est impossible de comparer ces reconstructions avec les originaux pour une image séréoscopique puisque ces images n'ont jamais existées. Par contre, une analyse qualitative peut être faite sur la séquence reconstruite. Dans ce texte, seule la vue du centre est présentée pour chacune des approches. Pour voir la totalité des reconstructions (10 images couleurs par approche) le lecteur peut consulter la page web :

www.inrs-telecom.quebec.ca/users/viscom/francais/publications/Finn02thesis



FIG. 5.15 – Reconstruction d'une vue intermédiaire de *Flower* à l'aide de l'approche de March.

Dans les images intermédiaires reconstruites à l'aide de l'approche de March (figure 5.15), le bruit se trouvant près des discontinuités dans la luminosité a disparu. Par contre, le manque de discontinuité dans le champs de disparité se fait sentir : la tige et les feuilles de la fleur dans la zone 8 semblent brisées, les fleurs de la zone 6 et le point jaune de la zone 5 sont dédoublés, les fleurs des zones 12 et 13 sont floues. La bordure supérieure gauche contient une mince ligne mauve qui correspond probablement à un manque d'information lors de la reconstruction.

Pour les images intermédiaires reconstruites à l'aide de l'approche de Horn & Schunck (figure 5.16), les effets dus au manque de discontinuité dans le champ de



FIG. 5.16 – Reconstruction d’une vue intermédiaire de *Flower* à l’aide de l’approche de Horn.

disparité sont notables sur la tige dans la zone 8, sur les fleurs des zones 6, 7 et 13 et le point jaune en 5. Un léger effet de bord sur la bordure gauche est présent.

En ce qui concerne les images intermédiaires reconstruites à l’aide de l’approche de la relaxation convexe (figure 5.17), où il devrait y avoir des discontinuités dans le champ de mouvement, il y a encore des problèmes : dans la zone 8, la tige semble cassée et les feuilles sont dédoublées, le point jaune de la zone 5 est dédoublé et les fleurs bleues dans le côté droit de la zone 13 sont déformées et floues. Un effet de bord important se trouve sur le haut de la bordure gauche.

Enfin, dans les images intermédiaires reconstruites à l’aide de l’approche de la relaxation robuste (figure 5.18), les mêmes défauts que dans le cas précédent sont notés. En plus, un effet de bord s’est ajouté sur la bordure supérieure.



FIG. 5.17 – Reconstruction d'une vue intermédiaire de *Flower* à l'aide de l'approche de la relaxation convexe.



FIG. 5.18 – Reconstruction d'une vue intermédiaire de *Flower* à l'aide de l'approche de la relaxation robuste.

5.3 La séquence d'images *Flower* Corrigée

Les images originales de *Flower* comportent une différence locale de luminosité entre l'image gauche et l'image droite qui influence la qualité des résultats des estimateurs de disparité. Pour corriger ces différences, une correction locale de luminosité est nécessaire. Puisque la correction locale par section (section 4.2.2) crée des discontinuités dans la luminosité (visibles entre les quadrants 1 et 2 de la figure 5.19), une correction bilinéaire sera plutôt utilisée pour corriger l'image droite (voir figure 5.20). Les images corrigées auront donc les mêmes dimensions que les images originales soit 720×480 .



FIG. 5.19 – Correction de luminosité de l'image *Flower* droite à l'aide de la correction par section.

5.3.1 Reconstruction d'une image existante

Toutes les estimations de disparité et les reconstructions faites dans la section 5.2 sont reprises en remplaçant *Flower* droite originale (figure 5.11) par la correction de luminosité bilinéaire de *Flower* droite (figure 5.20).

Dans la reconstruction à l'aide de l'approche de March de *Flower* droite corrigée (figure 5.21a), le principal défaut est toujours le bruit aux discontinuités dans la luminosité de l'image. Il y a toujours une déformation des objets se trouvant là où il devrait y avoir des discontinuités dans le champ de mouvement. Aucune amélioration majeure n'est notable par rapport au résultat obtenu avec la même approche sur les



FIG. 5.20 – Correction de luminosité de l'image *Flower* droite à l'aide de la correction bilinéaire.

images originales. De plus, une disparition des fleurs devant le pot entre les zones 12 et 14 est notée. Finalement, dans la partie supérieure de la bordure gauche, une partie de l'image a été reconstruite dans la bordure noire.

En comparant la reconstruction à l'aide de l'approche de Horn & Schunck de *Flower* droite (figure 5.21b) corrigée à la reconstruction de l'originale (figure 5.13b), plusieurs améliorations sont notées. D'abord, le drapeau de la zone 1, le drapeau et le cadre de la fenêtre de la zone 3, le poteau de la zone 4, le pied du parasol de la zone 9 et les fleurs de la zone 11 sont moins déformés. La tige de la fleur de la zone 8 ressemble plus à la véritable tige même si elle est toujours problématique. Les défauts dus au manque de discontinuité dans le champ de mouvement sont toujours présents.

L'approche de la relaxation convexe semble aussi donner de meilleures reconstructions de *Flower* droite corrigée (figure 5.21c) que de l'originale (figure 5.13c). Le cadre de fenêtre de la zone 2, le drapeau de la zone 3, le poteau de la zone 4, le pied du parasol de la zone 9 et les fleurs de la zone 11 et 14 ne comportent plus d'erreurs marquées. Toutefois, les endroits où il devrait y avoir des discontinuités dans le champ de vecteurs sont toujours affectés : soient les zones 5, 6, 7, 8, 9 et 13. Par contre, la fleur de la zone 8 est mieux reconstruite même si la tige est toujours affectée. Les effets de bord sont toujours présents.

La reconstruction de *Flower* droite corrigée à l'aide de la relaxation robuste comporte seulement quelques défauts puisque la correction de luminosité semble avoir réglé les problèmes du cadre de la zone 2, du drapeau de la zone 3, du poteau de

	RSBP	
	Originale	Corrigée
March	27.4294 dB	26.2478 dB
Horn & Schunck	26.1551 dB	27.6577 dB
Relaxation convexe	29.1016 dB	31.4550 dB
Relaxation robuste	33.4741 dB	36.3578 dB

TAB. 5.4 – Comparaison quantitative des reconstructions de *Flower* droite et *Flower* droite corrigée.

la zone 4 et de la fleur de la zone 8. La tige et la feuille de la zone 8 sont toujours problématiques et les effets de bord sont toujours présents sur la bordure gauche et la bordure inférieure.

La figure 5.22 présente la reconstruction offrant le meilleur RSBP, soit celle obtenu avec la relaxation robuste.

À l'exception de l'approche de March, tous les algorithmes ont bénéficiés de la correction de luminosité, spécialement la relaxation robuste qui surpasse maintenant de plusieurs dB chacune des autres approches.

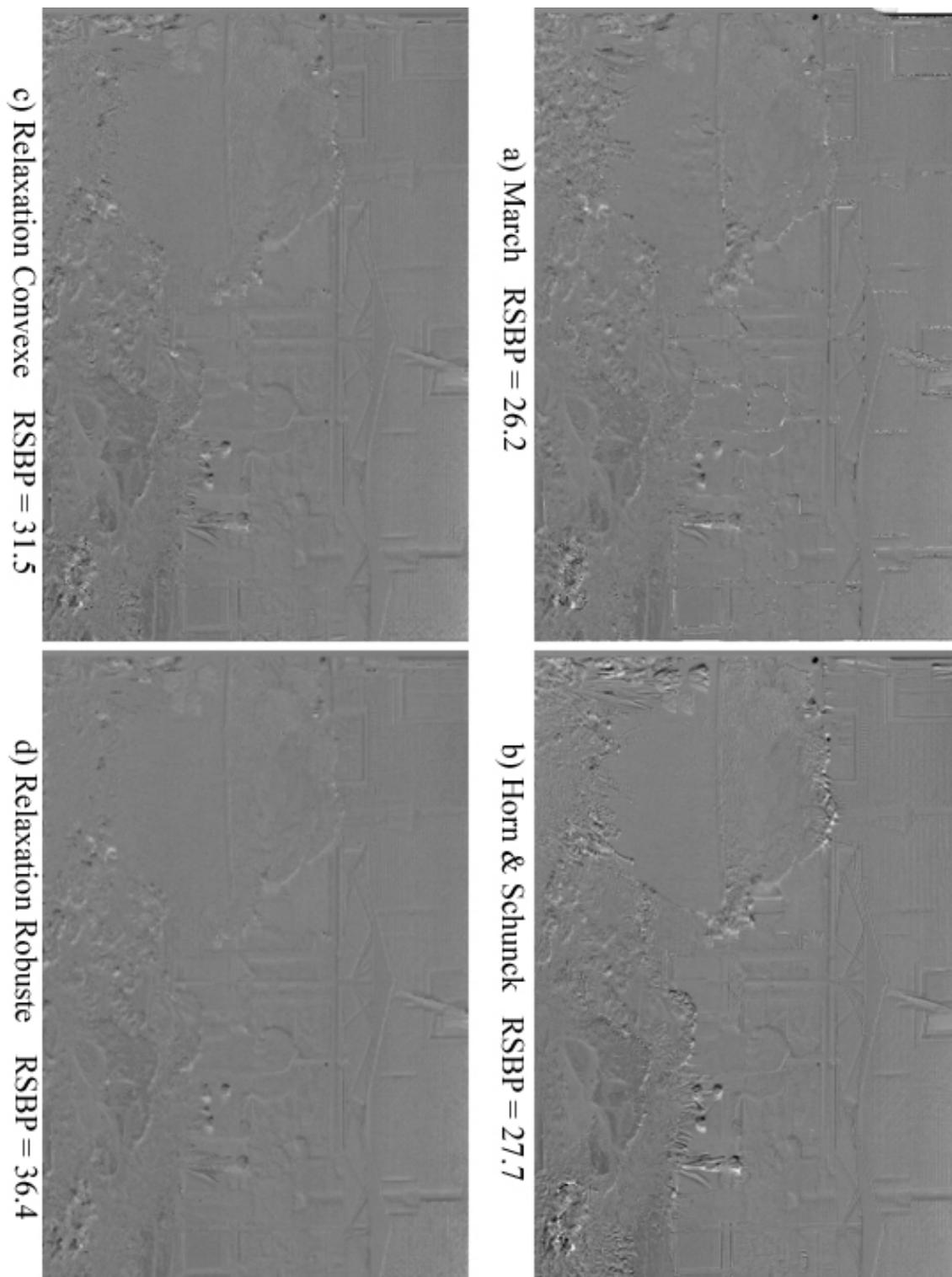


FIG. 5.21 – Erreurs dans la reconstruction de *Flower* droite corrigée pour l'approche de a)March b)Horn & Schunck c)la relaxation convexe d)la relaxation robuste. (Blanc=221 Noir=221)



FIG. 5.22 – Reconstruction à l'aide de l'approche de la relaxation robuste de *Flower* droite corrigée.

5.3.2 Reconstruction d'images intermédiaires

La comparaison des différentes approches pour la reconstruction d'images intermédiaires a été reprise suite à la correction de luminosité. Seule la vue du centre est présentée pour chacune des approches. Pour voir la totalité des reconstructions en couleurs, le lecteur peut se reporter à la page web :

www.inrs-telecom.quebec.ca/users/viscom/francais/publications/Finn02thesis



FIG. 5.23 – Reconstruction d'une vue intermédiaire de *Flower* à l'aide de l'approche de March et de la correction de luminosité.

Pour la reconstruction de vues intermédiaires de *Flower* à l'aide de l'approche de March et de la correction de luminosité (figure 5.23), le même phénomène que dans la section 5.2.2 est remarqué : le bruit dans les discontinuités de luminosité n'est pas présent. Aucune différence majeure avec la reconstruction de vues intermédiaires des images originales n'est notée. Il y a une légère amélioration de la tige de la zone 8 et 11 et des fleurs de la zone 14. L'effet de bord est aussi légèrement moins important.

Pour l'approche de Horn & Schunck, la correction de luminosité a contribué à améliorer légèrement la reconstruction de vues intermédiaires (figure 5.24) pour quelques endroits dans l'image. Entre autres, le côté gauche du cadre de la fenêtre gauche et la tige de la zone 8 sont mieux reconstruits. Le reste ne semble pas être affecté par la correction de luminosité.

La reconstruction de vues intermédiaires de *Flower* à l'aide de l'approche de la relaxation convexe et de la correction de luminosité (figure 5.25) est très similaire à celle obtenue avant la correction de luminosité à quatre améliorations près : le côté



FIG. 5.24 – Reconstruction d’une vue intermédiaire de *Flower* à l’aide de l’approche de Horn & Schunck et de la correction de luminosité.

gauche du cadre de la fenêtre de gauche, les tiges des zones 8 et 11 et les fleurs de la zone 14.

Pour ce qui est de la reconstruction de vues intermédiaires de *Flower* à l’aide de l’approche de la relaxation robuste et de la correction de luminosité (figure 5.26), on note quelques améliorations par rapport à la reconstruction de l’image originale (figure 5.18) : le côté gauche du cadre de de la fenêtres gauche, les tiges des zones 8 et 11 et les fleurs de la zone 14. Par contre, les résultats sont très similaires à la relaxation convexe.



FIG. 5.25 – Reconstruction d'une vue intermédiaire de *Flower* à l'aide de l'approche de la relaxation convexe et de la correction de luminosité.



FIG. 5.26 – Reconstruction d'une vue intermédiaire de *Flower* à l'aide de l'approche de la relaxation robuste et de la correction de luminosité.

5.4 La séquence d'images *FlowerGarden*

La séquence temporelle d'images *FlowerGarden* sont des images de dimensions 720×480 (figures 5.27 à 5.29) prises par une seule caméra qui fait un travelling vers la droite. Seules les trames 0, 2 et 4 ont été utilisées pour éviter les considérations d'entrelacement.



FIG. 5.27 – Image *FlowerGarden* gauche.

5.4.1 Reconstruction d'une image existante

Tout comme pour les images *Flower* et *Birds*, nous avons reconstruit l'image droite à l'aide des champs de vecteurs calculés avec les différentes méthodes. Chacune des méthodes tentera d'estimer le champ de déplacement de l'image *FlowerGarden* gauche vers *FlowerGarden* droite, avec comme pivot l'image droite. Avec ce champ de déplacement et l'image gauche, une reconstruction de l'image droite sera créée et comparée avec l'image droite originale.

Par contre, aucune évaluation qualitative ne sera donnée ici, nous jugeons que les erreurs sont toutes aux mêmes endroits, soient les zones nouvellement exposés (voir figure 5.30) et que l'amplitude de ces erreurs est bien représentée par le RSBP.



FIG. 5.28 – Image *FlowerGarden* centre.



FIG. 5.29 – Image *FlowerGarden* droite.



FIG. 5.30 – Zones nouvellement exposées *FlowerGarden* droite.

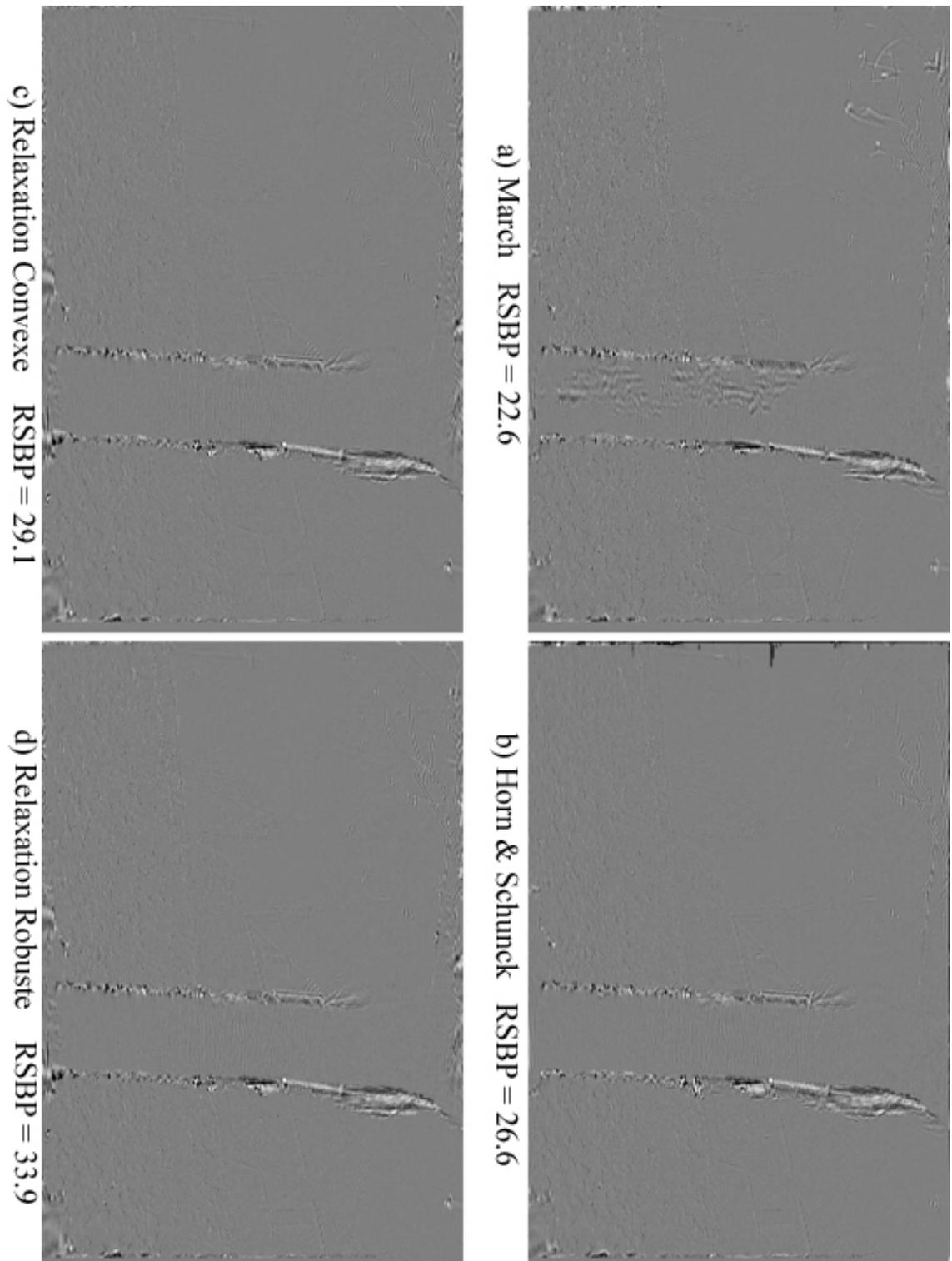


FIG. 5.31 – Erreurs dans la reconstruction de *FlowerGarden* droite corrigée pour l'approche de a)March b)Horn & Schunck c)la relaxation convexe d)la relaxation robuste. (Blanc=-128 Noir=128)

	RSBP
March	22.6252 dB
Horn & Schunck	26.6434 dB
Relaxation convexe	29.0526 dB
Relaxation robuste	33.9082 dB

TAB. 5.5 – Comparaison quantitative des reconstructions de *FlowerGarden* droite.

La figure 5.32 présente la reconstruction offrant le meilleur RSBP, soit celle obtenue avec la relaxation robuste.



FIG. 5.32 – Reconstruction de *FlowerGarden* droite à l'aide de la relaxation robuste.

5.4.2 Reconstruction d'images intermédiaires

Pour *FlowerGarden*, une image intermédiaire réelle existe puisque c'est une séquence temporelle d'images. Il est ainsi possible d'utiliser le RSBP pour comparer la reconstruction de l'image *FlowerGarden* centre. Le champ de déplacement a été estimé de l'image *FlowerGarden* gauche vers *FlowerGarden* droite avec chacune des méthodes. Le centre des deux images ($\alpha = 0.5$) a été choisi comme pivot. À l'aide de ce champ de déplacement, de l'image gauche et de l'image droite, une reconstruction d'images intermédiaires sera effectuée et l'image du centre sera comparée avec l'image *FlowerGarden* centre. Dans ce texte, seules les images du centre seront utilisées, le lecteur peut se référer au site web suivant pour voir la totalité des images reconstruites en couleurs :

www.inrs-telecom.quebec.ca/users/viscom/francais/publications/Finn02thesis

Encore une fois, les erreurs de chacune des méthodes sont situées aux mêmes endroits, les zones d'occlusions et les zones nouvellement exposées (voir figure 5.33). Puisque l'image du centre de la reconstruction peut être comparée à l'originale et que le RSBP est représentatif de l'importance des erreurs contenues dans ces zones, aucune évaluation qualitative sera ne présentée dans cette section.

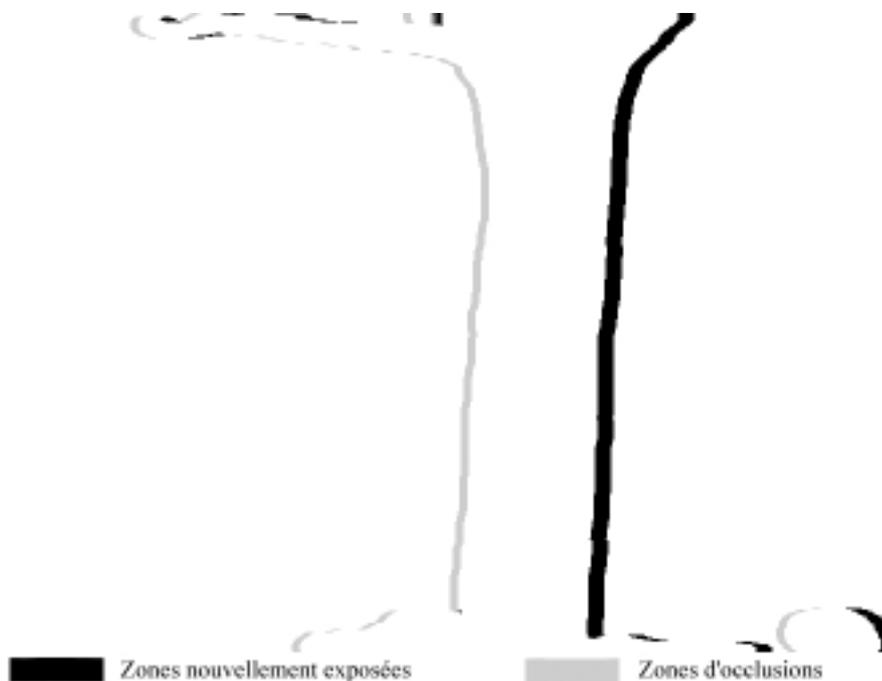


FIG. 5.33 – Zones d'occlusions de *FlowerGarden* centre.

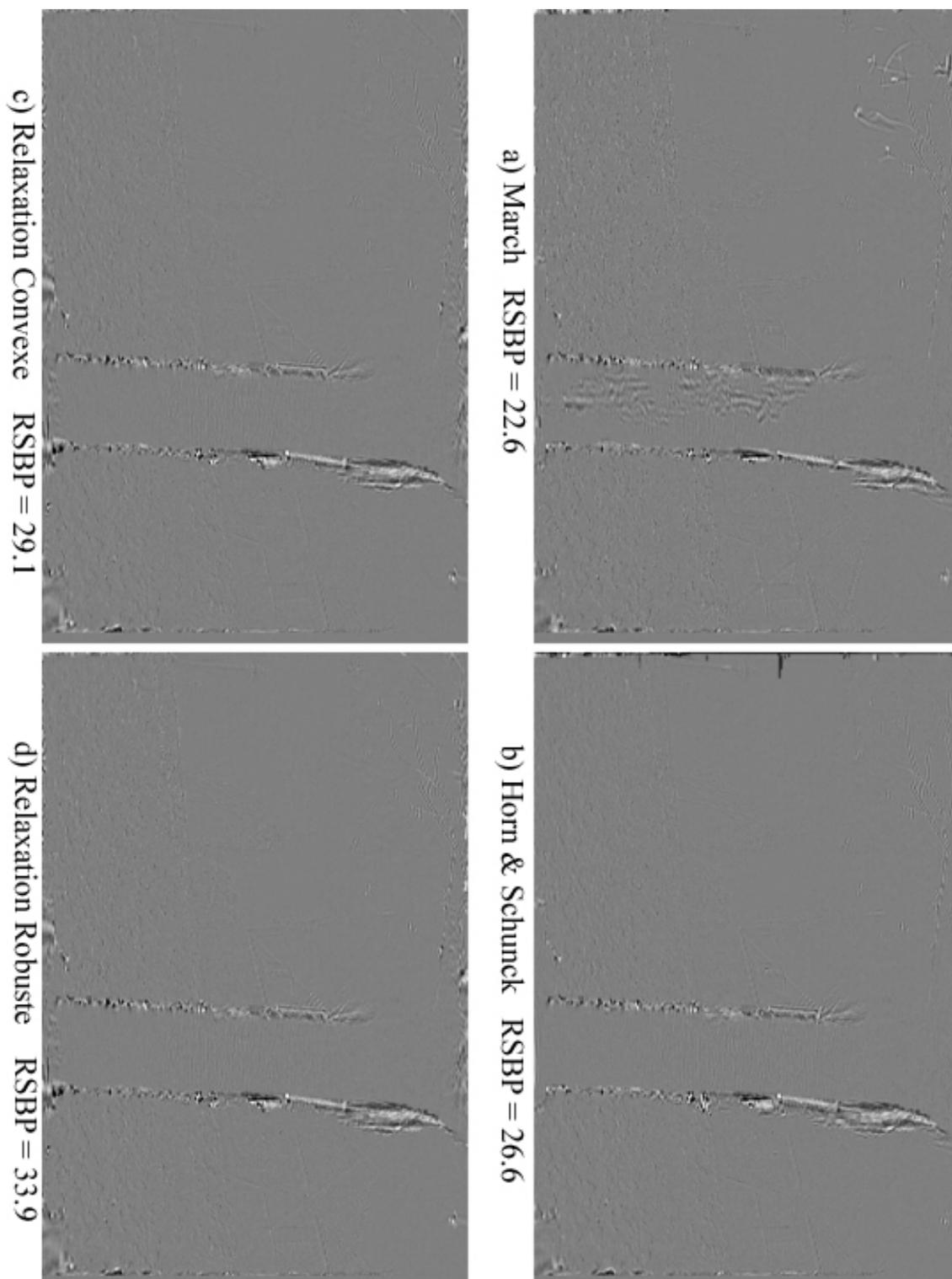


FIG. 5.34 – Erreurs dans la reconstruction de *FlowerGardendroite* corrigée pour l'approche de a) March b) Horn & Schunck c) la relaxation convexe d) la relaxation robuste. (Blanc=-128 Noir=128)

	RSBP
March	26.8597 dB
Horn & Schunck	25.6815 dB
Relaxation convexe	27.3909 dB
Relaxation robuste	27.2575 dB

TAB. 5.6 – Comparaison quantitative des reconstructions de *FlowerGarden* centre.

La figure 5.35 présente la reconstruction offrant le meilleur RSBP, soit celle obtenue avec la relaxation convexe.

FIG. 5.35 – Reconstruction de *FlowerGarden* centre à l'aide de la relaxation convexe.

Chapitre 6

Conclusion

Cette étude avait pour but de développer une méthode robuste d'estimation de disparité et de l'appliquer pour reconstruire des images intermédiaires entre deux images stéréoscopiques. Le présent chapitre a pour but d'interpréter et d'expliquer les différents résultats obtenus selon les aspects suivants : l'apport de la relaxation convexe par rapport à l'approche de March, l'apport de la correction bilinéaire de luminosité et l'apport de la relaxation robuste.

6.1 Apport de la relaxation convexe

L'approche de March, lorsque l'estimation de disparité est effectuée sur l'une des images ($\alpha = 0$ ou 1), comporte des anomalies où la luminosité est discontinue. Bien que dans le cas discret la relaxation convexe ait une fonction de coût équivalente à celle de l'approche de March elle n'engendre pas ce type d'anomalies. Essayons de comprendre pourquoi.

$$E_r = \sum_{i,j} \left[(\tilde{I}_t(i, j))^2 + \sum_{(k,l) \in \nu_{ij}} [\lambda_u(u_{ij} - u_{kl})^2 + \lambda_v(v_{ij} - v_{kl})^2] \right] \quad (6.1.1)$$

L'approche de March utilise une fonction de coût continue :

$$E = \int \int \left((\tilde{I}_t(x, y))^2 + \lambda_u(u_x^2 + u_y^2) + \lambda_v(v_x^2 + v_y^2) \right) dx dy$$

Ensuite, les équations d'Euler-Lagrange sont utilisées pour trouver une équation dans le cas continu. Après la discrétisation de cette équation, une approche de relaxation linéaire est utilisée sur cette équation afin d'en trouver la solution itérative suivante :

$$u_{i,j}^{(n+1)} = \overline{u_{i,j}}^{(n)} - \frac{\tilde{I}_x^{(n)} \tilde{I}_t^{(n)}}{4\lambda_u} \quad (6.1.2)$$

$$v_{i,j}^{(n+1)} = \overline{v_{i,j}}^{(n)} - \frac{\tilde{I}_y^{(n)} \tilde{I}_t^{(n)}}{4\lambda_v} \quad (6.1.3)$$

qui peut se réécrire ainsi :

$$u_{i,j}^{(n+1)} = u_{i,j}^{(n)} - \frac{\tilde{I}_x^{(n)} \tilde{I}_t^{(n)} + 4\lambda_u(u_{i,j}^{(n)} - \overline{u_{kl}}^{(n)})}{4\lambda_u} \quad (6.1.4)$$

$$v_{i,j}^{(n+1)} = v_{i,j}^{(n)} - \frac{\tilde{I}_y^{(n)} \tilde{I}_t^{(n)} + 4\lambda_v(v_{i,j}^{(n)} - \overline{v_{kl}}^{(n)})}{4\lambda_v} \quad (6.1.5)$$

La relaxation convexe utilise une approche de relaxation non-linéaire pour trouver une solution à la fonction de coût discrétisée. Une approche non-linéaire est plus appropriée qu'une approche linéaire, car les images sont des objets non-linéaires. On obtient ainsi :

$$u_{ij}^{(n+1)} = u_{ij}^{(n)} - \frac{\tilde{I}_t^{(n)} \tilde{I}_x^{(n)} + 2\lambda_u(u_{ij}^{(n)} - \overline{u_{kl}}^{(n)})}{\tilde{I}_x^{2(n)} + \tilde{I}_t^{(n)} \tilde{I}_{xx}^{(n)} + 8\lambda_u} \quad (6.1.6)$$

$$v_{ij}^{n+1} = v_{ij}^n - \frac{\tilde{I}_t^{(n)} \tilde{I}_y^{(n)} + 2\lambda_v(v_{ij}^{(n)} - \overline{v_{kl}}^{(n)})}{\tilde{I}_y^{2(n)} + \tilde{I}_t^{(n)} \tilde{I}_{yy}^{(n)} + 8\lambda_v} \quad (6.1.7)$$

Lorsque nous comparons les équations (6.1.4) et (6.1.6) et les équations (6.1.5) et (6.1.7), la différence majeure se retrouve au dénominateur. Le dénominateur de la relaxation convexe comporte un terme en \tilde{I}_x^2 ou \tilde{I}_y^2 qui ralentit la relaxation dans les endroits où l'image varie rapidement. C'est exactement à ces endroits que l'approche de March donne des anomalies.

Une séquence test qui permettra de vérifier ce phénomène est présentée aux figures 6.1 et 6.2.

Les deux images sont identiques à l'exception d'un décalage latéral de 5 pixels vers la droite de l'image de droite (le cadre noir ne fait pas parti des images originales, il a été ajouté par la suite pour ajouter de la clareté à la présentation).

En effectant la reconstruction de l'image test droite avec l'approche de March, l'image présentée à la figure 6.3 est obtenue.

On peut observer au quart de la hauteur, à partir du haut, le même type de phénomène que les images plus complexes. Le seul facteur qui peut entraîner cette anomalie est la dérivé spatiale de l'image. Si on regarde maintenant la reconstruction de l'image droite à l'aide de la relaxation convexe présentée à la figure 6.4, on n'observe aucune erreur similaire.

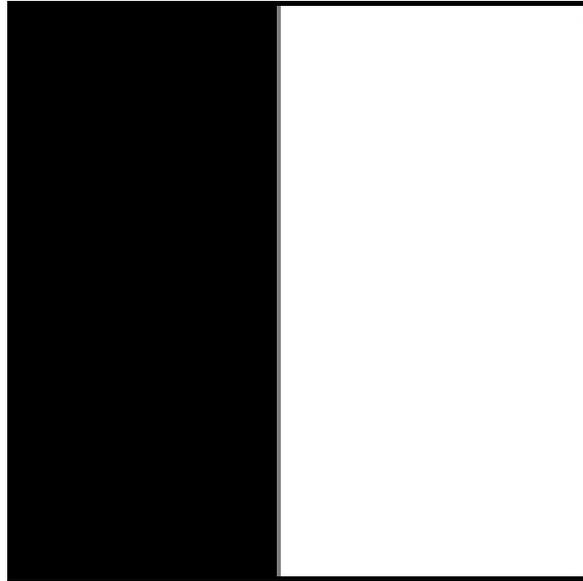


FIG. 6.1 – Image Test gauche.

On avait aussi noté dans le chapitre précédent que ce phénomène ne semblait pas se produire pour des reconstructions de vues intermédiaires. Nous avons donc reconstruit une image test intermédiaire pour $\alpha = 0.6$ avec la méthode de March. Effectivement, le résultat (présenté à la figure 6.5) ne comporte pas ce genre d'anomalies. Cette plus grande stabilité vient probablement du terme \tilde{I}_x^2 qui est maintenant une combinaison linéaire de la dérivé horizontale sur l'image gauche et l'image droite. Par contre, quand la reconstruction se fait en $\alpha=1$, seule l'image gauche est considérée dans \tilde{I}_x^2 . Évidemment, dans cet exemple, \tilde{I}_y^2 est nul partout.

L'approche de relaxation convexe offre des résultats plus stables que l'approche de March lorsque l'estimation de disparité est effectuée sur l'une des images. Autrement les images produites par l'approche de March sont acceptables.

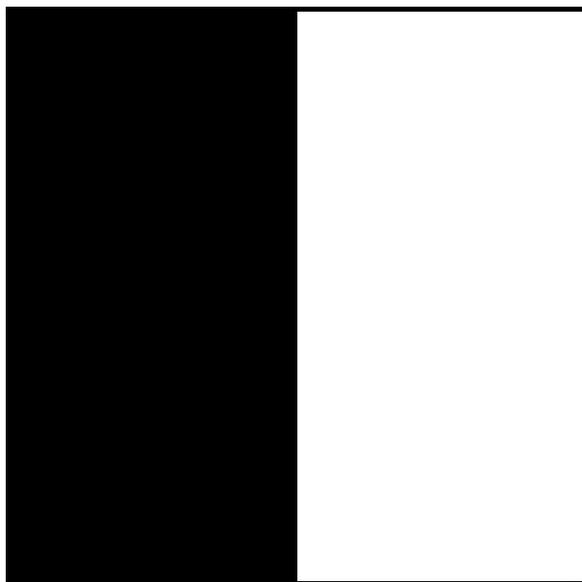


FIG. 6.2 – Image Test droite.



FIG. 6.3 – Reconstruction de l'image test droite à l'aide de l'approche de March.

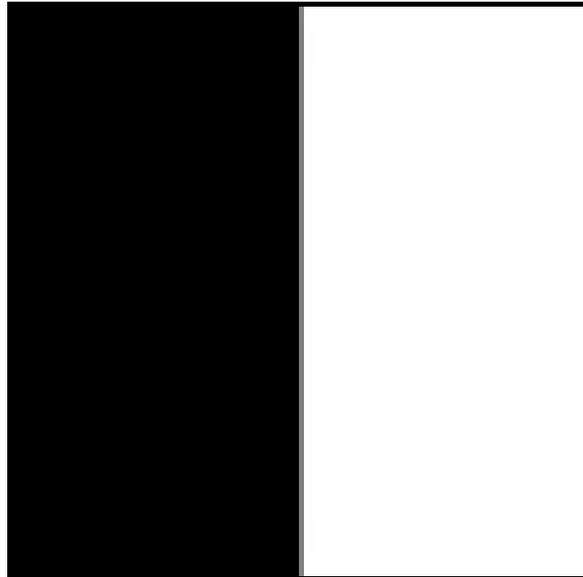


FIG. 6.4 – Reconstruction de l'image test droite à l'aide de la relaxation convexe.

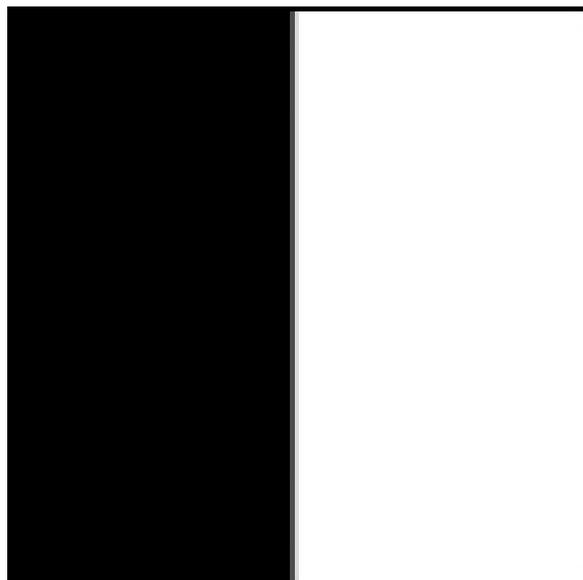


FIG. 6.5 – Reconstruction de l'image test intermédiaire à l'aide de l'approche de March.

6.2 Apports de la correction bilinéaire de luminosité

Les besoins de développer une méthode de correction locale de luminosité se sont fait ressentir lorsque nous avons observé l'anomalie qui se trouve sur le poteau supérieur gauche pour la relaxation convexe. A priori, il n'y a aucune raison pour que l'estimation soit mauvaise puisqu'il n'y a pas de discontinuité dans le champ de déplacement et aucune zone d'occlusion dans cette région. Par contre, localement, il y a une différence dans la luminosité. Pour vérifier qu'il n'y a pas d'autre facteur contributoire, on a fait une estimation de disparité et une reconstruction sur cette partie des images présentées aux figures 6.6 et 6.7 ont été faites.



FIG. 6.6 – Image *FlowerCoin* gauche.



FIG. 6.7 – Image *FlowerCoin* droite.



FIG. 6.8 – Reconstruction de *FlowerCoin* droite à l'aide de la relaxation convexe.

L'anomalie qui était présente dans l'image complète l'est toujours dans figure 6.8. L'image *FlowerCoin* est alors soumise à une correction globale de luminosité, ce qui

correspondrait à une correction de luminosité par sections sur *Flower*. Cette image est présentée à la figure 6.9, l'image droite originale à la figure 6.7 et la reconstruction de l'image droite à la figure 6.11.



FIG. 6.9 – Image *FlowerCoin* gauche corrigée



FIG. 6.10 – Image *FlowerCoin* droite



FIG. 6.11 – Reconstruction de la *FlowerCoin* droite à l'aide de la relaxation convexe et de la correction de luminosité

L'anomalie a disparue. Une correction locale de luminosité sur l'image entière est nécessaire afin d'éliminer cette anomalie. En effet une fois la méthode implantée et utilisée, certaines améliorations sont remarquées. Dans le cas de la reconstruction de *Flower* droite, les améliorations qualitatives et quantitatives étaient visibles pour toutes les méthodes, à l'exception de celle de March (voir tableau 5.4). Pour ce qui est des reconstructions de vues intermédiaires, toutes les méthodes ont bénéficié légèrement de la correction de luminosité.

La correction bilinéaire de luminosité, qui a été développée pour corriger les déséquilibres de luminance de la paire d'image stéréoscopique *Flower*, semble donc avoir rempli son mandat. Par contre, elle n'a été testée que sur cette unique paire. Il pourrait être intéressant de vérifier si cette méthode s'avère aussi efficace sur d'autres images stéréoscopiques offrant des conditions similaires à *Flower*.

6.3 Apports de la relaxation robuste

L'introduction de l'opérateur robuste à l'approche du flot optique avait pour but d'admettre les discontinuités dans le champ de vecteurs de disparité afin d'obtenir un champ de meilleure qualité. Les approches traditionnelles utilisaient un opérateur quadratique et interdisaient catégoriquement les discontinuités. Puisque la fonction d'influence d'un opérateur robuste est limitée, cette interdiction devrait maintenant être levée.

Ceci peut être observé à l'aide des champs de déplacement engendrés avec la séquence d'images *Birds* (voir figures 5.4 à 5.6). Alors que les autres méthodes ont un champ continu, la relaxation robuste laisse des discontinuités sur presque tout le périmètre du rectangle en déplacement (voir figure 5.7). Ceci a pour effet de donner une meilleure estimation de disparité (tableau 5.1) qui se transpose en une meilleure reconstruction (tableau 5.2).

On note aussi cette amélioration de manière qualitative et quantitative pour la reconstruction de *Flower* droite et de *Flower* droite corrigée (tableau 5.4). Par contre, pour la reconstruction de vues intermédiaires, toutes les améliorations observées de manière qualitative avec les reconstructions de *Flower* droite et *Flower* droite corrigée, ne sont pas présentes. Les différences entre les approches semblent moindre lorsque le pivot du champ de déplacement n'est pas localisé sur une des images originales.

On observe encore une fois ce phénomène, mais de manière qualitative, pour la séquence d'images *FlowerGarden*. Pour la reconstruction de *FlowerGarden* droite (tableau 5.5), les méthodes donnent des résultats de qualités différentes, l'approche de March étant de faible qualité et la relaxation robuste offrant, de loin, la meilleure reconstruction. Dans la reconstruction de *FlowerGarden*centre (tableau 5.6), les différences de qualité sont réparties autrement. L'approche de Horn & Schunck est en dernière position, précédée de celle de March, qui réussit beaucoup mieux pour une reconstruction de vues intermédiaires. L'approche de la relaxation convexe et de la relaxation robuste donnent essentiellement les mêmes résultats. Dans le cas d'une reconstruction de vues intermédiaires (seulement 1.7 dB sépare les quatre méthodes, alors que dans le cas de la reconstruction de l'image droite, 5.3 dB les séparaient).

Pour mieux comprendre ce qui se passe pour une reconstruction de vues intermédiaires voici les champs de disparité de *Birds*, calculé à $\alpha = 0.5$.

L'opérateur robuste a créé des discontinuités aux bons endroits même si le pivot était situé en $\alpha = 0.5$. Ce qui nous porte à croire que la complexité de l'image y est aussi pour quelque chose. Pour des images réelles, il est plus difficile de trouver une correspondance exacte ($\tilde{I}_t = 0$). Si la solution parfaite n'est pas disponible même en créant une discontinuité, l'algorithme trouvera une solution satisfaisante, en déplaçant les deux extrémités du vecteurs, souvent sans créer de discontinuité.

De façon générale, les approches de relaxations fonctionnent assez bien, de manière quantitative et qualitative. Pour ce qui est de la relaxation robuste, elle ne se démarque que si elle sert à reconstruire une image existante ou un cas simple. Puisque l'objectif

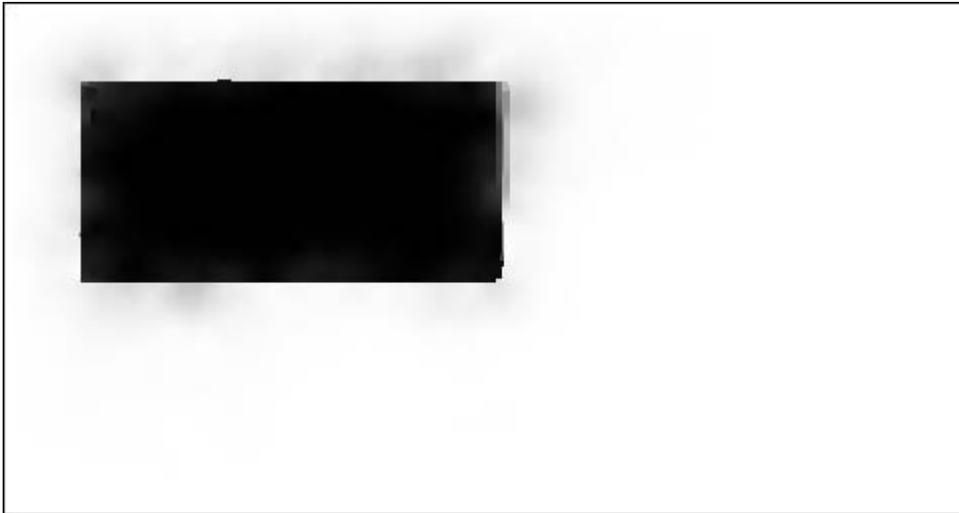


FIG. 6.12 – Composante horizontale du champ de disparité de *Birds* calculé avec la relaxation robuste à $\alpha = 0.5$.

était de créer de nouvelles images, on voit mal l'utilité d'un opérateur robuste dans ce cadre.

Il existe cependant d'autres possibilités pour cette méthodes. Il est possible d'utiliser un champ de disparité centré sur une image existante afin d'estimer la topologie de la scène. C'est à dire transformer la fonction de déplacement en une fonction de profondeur. Cette profondeur pourrait alors être utilisée telle quelle ou pour segmenter l'images en objets distincts. Puisque la topologie peut être déterminée à partir d'une image existante, la relaxation robuste devrait offrir des résultats intéressant à ce problème.

Bibliographie

- [1] V. Bhaskaran and Konstantinides. *Image and Video Compression Standards : Algorithms and Architectures*. Kluwer Academic Publishers, 1999.
- [2] M.J. Black. *Robust Incremental Optical Flow*. Doctorat thesis in computer science, Yale University, USA, 1992.
- [3] A. Blake and A. Zisserman. *Visual Reconstruction*. The MIT Press, Cambridge, Massachusets, 1987.
- [4] R. Depommier and E. Dubois. Motion estimation with detection of occlusion area. INRS-Télécommunication, 1992.
- [5] M. Ebroul Izquiero. Stereo matching for enhanced telepresence in 3d video-communications. *IEEE transactions on circuits and systems for video technology*, 7(4) :629–643, august 1997.
- [6] M. Ebroul Izquiero. Disparity/segmentation analysis : Matching with an adaptive window and depth-driven segmentation. *IEEE on circuits and systems for video technology*, 9(4) :589–607, june 1999.
- [7] Ruggero Elia Hendrik Franich. *Disparity Estimation in Stereoscopic Images*. Doctorat thesis in electrical engineering, College Dekanen, Italie, 1996.
- [8] B.K.P. Horn. *Robot Vision*. MIT press, Cambridge, 1986.
- [9] Robert G. Keys. Cubic convolution interpolation for digital image processing. *IEEE Trans. Acoustics, Speech, and Signal Processing*, ASSP-29(6) :1153–1160, december 1981.
- [10] A. Mancini. Disparity estimation and intermediate view reconstruction for novel application in stereoscopic video. Masters thesis in electrical engineering, McGill University, Montreal, 1998.
- [11] R. March. Computation of stereo disparity using regularisation. *Pattern Recognition Letters*, 8 :181–187, october 1988.
- [12] A. Redert, E. Hendriks, and J. Biemond. Correspondence estimation in image pairs. *IEEE Signal Processing Magazine*, 16(3), may 1999.
- [13] P. J. Rousseau and A. M. Leroy. *Robust Regression and Outlier Detection*. John Wiley & Son, Inc., New-York, 1987.

- [14] I. Sexton and P. Surman. Stereoscopic and autostereoscopic display systems. *IEEE Signal Processing Magazine*, 16(3), may 1999.
- [15] M. G. Srintzis and Malassiotis. Object-based coding of stereoscopic and 3d image sequences. *IEEE Signal Processing Magazine*, 16(3), may 1999.