

Filtrage EPSF pour l'amélioration d'image

Bruno Migeon, Véronique Serfaty

▶ **To cite this version:**

Bruno Migeon, Véronique Serfaty. Filtrage EPSF pour l'amélioration d'image. [Rapport de recherche] RR-2175, INRIA. 1994. inria-00074498

HAL Id: inria-00074498
https://inria.hal.science/inria-00074498

Submitted on 24 May 2006

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

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

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Filtrage EPSF pour l'amélioration d'image

Bruno MIGEON, Véronique SERFATY

N° 2175

Janvier 1994

PROGRAMME 4

Robotique,
image
et vision



ISSN 0249-6399

INRIA

Filtrage EPSF pour l'amélioration d'image

Bruno MIGEON*, Véronique SERFATY**

Programme 4 — Robotique, image et vision
Projet Syntim

Rapport de recherche n° 2175 — Janvier 1994 — 44 pages

Résumé :

Ce rapport se veut être une revue du filtrage EPSF (*Edge Preserving Smoothing Filtering*) et un résumé d'une étude que nous avons menée.

Nous avons implanté un certain nombre de filtres se réclamant de cette famille, et étudié leur comportement sur différents types d'images (réelles et de synthèse) pour tenter de dégager sommairement leurs avantages et inconvénients, dans le but de pouvoir mieux choisir tel ou tel en fonction d'un besoin particulier de notre application. On présente le principe de chacun de ces filtres et citons brièvement leurs qualités et défauts. Enfin, nous illustrons l'intérêt du filtrage EPSF en montrant l'amélioration qu'il peut apporter au niveau d'étapes fondamentales dans le cadre d'un processus de reconstruction 3D.

Mots-clé : traitement d'image, amélioration d'image, lissage.

(Abstract: pto)

*. migeon@bora.inria.fr
**. serfaty@bora.inria.fr

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
Téléphone : (33 1) 39 63 55 11 – Télécopie : (33 1) 39 63 53 30

Edge Preserving Smoothing Filtering for Image Enhancement

Abstract:

This report is a review on Edge Preserving Smoothing Filtering. We have implemented several filters of this category and studied their behaviour on different images (synthetic and real). We tried to consider the advantages and disadvantages of each method to be able to choose the right filter for the right application. We describe briefly each filter and show the improvement obtained by the use of EPSF in some fundamental steps of a 3D reconstruction process.

Key-words: image processing, image enhancement, smoothing.

Table des matières

1 Introduction **3**

2 Recherche bibliographique et tentative de classement **5**
2.1 Méthodes optimales 5
2.2 Méthodes structurelles: filtrage DPSF 6
2.3 Méthodes adaptatives: filtrage EPSF 6

3 Objectif des tests réalisés **8**

4 Filtres implantés et résultats **10**
4.1 Préambule 10
4.2 ITEN 10
4.3 GIF 12
4.4 EGPR 13
4.5 KNN 15
4.6 SNN 16
4.7 ASMT 17
4.8 AFUGM 18
4.9 NOPEL 20
4.10 ASAD 21
4.11 AFUMEP 22

5 Tableau de synthèse **26**

6 Application aux paires d'images stéréo **28**
6.1 Introduction 28
6.2 Intérêt pour la calibration 28
6.2.1 Calibration avec une mire à carrés 28
6.2.2 Calibration avec une mire à disques 29
6.3 Intérêt pour la segmentation 34
6.4 Intérêt pour l'analyse photométrique 36

7 Conclusion **40**

Table des figures

1 Appellation des luminances des pixels de la fenêtre W 11
2 Barre tournant autour de (x, y) 13
3 La réalisation des masques comprend 4 pentagones, 4 hexagones et un carré 13
4 Pour chaque paire symétrique, la valeur la plus proche de celle du pixel central est sélectionnée (représentée par un point noir) 16
5 La courbe des statistiques d'ordre est décomposée en différentes zones (*pentes* et *segments*) par une analyse morphologhique. 19
6 Tableau de synthèse 27
7 Mire à carrés. (a): image originale (b): image filtrée avec EGPR 30
8 Extraction des contours. (a): sur l'image originale (b): sur l'image filtrée avec EGPR 30
9 Mire à disques. (a): image originale (b): image filtrée avec ASAD 32
10 Différence entre image originale et image filtrée . (a): image des pixels dont l'intensité a changé de plus de 3 unités (b): valeur absolue de la différence normalisée sur [0..255] (en inverse vidéo) 32
11 Partie de la coupe de l'image de différence de la figure 10(b) à la ligne 210 33
12 Segmentation des images de la figure 9. (a): image originale (b): image filtrée 35
13 Partie de la coupe de l'image originale présentée à la figure 9(a) à la ligne 210. 35
14 Partie de la coupe de l'image filtrée présentée à la figure 9(b) à la ligne 210. 36
15 Scène de bureau. (a): image originale (b): image filtrée avec AFUMEP 37
16 Extraction des contours des images de la figure 15 (a): image originale (b): image filtrée 38
17 Visualisation en fausses couleurs des images de la figure 15. (a): image originale (b): image filtrée 38

1 Introduction

Une des tâches les plus importantes en traitement d'image est le filtrage, visant à éliminer le bruit et à améliorer l'interprétation des images.

Dans de nombreux domaines d'applications (vision robotique et industrielle, stéréo-vision, image de profondeur, imagerie satellitaire ou médicale...), le prétraitement constitue une étape essentielle (et parfois même primordiale) pour le bon déroulement des différentes phases du processus de traitement spécifique à l'application.

Le lissage par filtrage linéaire, très largement utilisé [And72, And74, And77, Hua71], permet une très bonne atténuation du bruit et offre ainsi un lissage correct. En revanche, il estompe les bords francs et atténue les détails en introduisant un flou ambiant sur l'image. Ceci rend délicate la détermination des limites entre les différents objets dans l'image, ce qui est très dommageable, car la plupart des applications (pour ne pas dire la totalité) reposent sur la possibilité de connaître de façon aussi précise que possible la localisation des différents contours présents dans l'image (qu'il s'agisse d'une approche par régions ou d'une approche par contours). Par ailleurs, le filtrage linéaire n'est pas du tout approprié pour l'élimination des forts bruits impulsionnels.

Le filtrage médian [Tuk71, Tuk74], très couramment utilisé en traitement du signal 1D [Rab75, Jay76], permet d'éliminer le bruit impulsionnel, de lisser les données lorsqu'il est utilisé de façon iterative (mais il ne peut pas lisser autant que le lissage par filtrage linéaire), et de préserver les bords. En 2D, il est également largement employé en amélioration d'image [Pra78, Hua81, Ros82], mais tend à arrondir les angles (lorsque la fenêtre de filtrage est carrée), ce qui est regrettable.

Depuis une quinzaine d'années, de nombreux filtres ont vu le jour, dans le but de lisser le signal pour atténuer le bruit dû au système d'imagerie, tout en préservant les bords. On les regroupe souvent sous le label de "Edge Preserving Smoothing Filtering" [Chi83, Har87, Nag79]. On limitera notre propos à un certain nombre de ces filtres de type EPSF auxquels on s'est intéressé.

L'objet du travail présenté était de:

1. Faire une recherche bibliographique (avec une tentative de classement par catégories) sur le filtrage EPSF, pour ensuite sélectionner un certain nombre de filtres "reconnus" comme étant assez intéressants au vu

des articles de leurs auteurs et des différentes évaluations comparatives publiées.

2. Implanter l'ensemble des filtres choisis en les intégrant à Xanalyse (logiciel en perpétuelle évolution, développé au sein du projet Syntim, et regroupant toutes les différentes étapes successives du processus de traitement des paires d'images stéréoscopiques pour la reconstruction et la reconnaissance de scènes 3D).

3. Tester chacun de ces filtres sur différents types d'images afin d'évaluer leurs avantages et inconvénients respectifs, pour ensuite pouvoir choisir tel ou tel filtre dans le but d'améliorer les résultats de telle ou telle étape essentielle du processus de reconstruction (parmi lesquelles on peut citer la calibration, la segmentation et l'étude photométrique), résultats dont la qualité est déterminante pour l'obtention d'une reconstruction 3D la meilleure possible.

4. Appliquer ces filtres aux images de scènes à reconstruire et voir dans quelle mesure ils peuvent améliorer les différentes étapes du processus de reconstruction 3D.

2 Recherche bibliographique et tentative de classement

Depuis une quinzaine d'années, une littérature abondante a vu le jour sur le lissage visant à atténuer le bruit tout en préservant les bords, voire les détails. Il existe de multiples approches du problème, ce qui explique le nombre important de filtres développés.

Ainsi, il est très difficile et délicat de regrouper ces diverses approches en classes ou catégories, de façon strictement hiérarchique ou taxonomique, car les filtres qui en résultent peuvent parfois se ressembler beaucoup, tant par leurs résultats que par leurs caractéristiques, voire même leurs implantations. (Par exemple, les filtres classiques *médian* et *moyenneur* peuvent être considérés comme des cas particuliers de nombreux filtres basés sur une approche statistique [Lee85, Gan89]). De ce fait, il est impossible d'établir un classement exhaustif de tous ces filtres, dans la mesure où un grand nombre d'entre eux seraient susceptibles d'appartenir à plusieurs catégories.

Toutefois, au vu des différents articles étudiés, on peut se risquer à faire un classement assez large et sans prétention, mais qui a au moins le mérite d'exister, en trois grandes familles, que l'on peut ensuite "redécouper" en différentes classes ou catégories:

2.1 Méthodes optimales

Le but des méthodes de filtrage de cette famille est de trouver le filtre optimal en terme de rapport signal sur bruit. On trouve notamment dans cette famille les filtres d'ordre qui, en prenant en compte la distribution statistique des perturbations, permettent d'améliorer les performances en réduction de bruit, par rapport au filtrage linéaire, et la mise au point d'opérateurs optimaux peut se faire selon différentes approches (approximation, maximum de vraissemblance, estimation bayesienne) [Bol92]. Certains filtres sont basés sur une solide approche statistique, reposant notamment sur la théorie des estimateurs robustes [ANd72, Hub81], et forment entre autres les catégories des L-filtres [Bov83, Lee85, Mal79, Mal80], des M-filtres [Stu79, Hub81, Lee85], et des R-filtres [Gan89].

Ils sont surtout utilisés en traitement du signal 1D [Lee85, Gan89], et leur emploi n'est pas toujours aisé, notamment à cause du fait qu'ils nécessitent souvent la connaissance de la distribution statistique des perturbations.

2.2 Méthodes structurelles: filtrage DPSF

On pourrait attribuer à cette famille de filtres le label de filtrage DPSF pour signifier "Detail Preserving Smoothing Filtering", ce qu'un certain nombre d'auteurs emploie souvent [Nie87, Arc89].

On peut distinguer deux catégories dans cette famille, qui sont d'une part les variantes structurelles du filtre médian (multistage median filter) à savoir multiétage (UMMF, BMMF) [Nie87, Nie88, Arc89, Wan90] et hybrides (UFIRMHF, BFIRMHF) [Hei85, Hei87], et d'autre part les filtres de morphologie mathématique (2DCO) [Ste86, Ste87, Mar87]. Ces filtres utilisent des sous-structures unidirectionnelles et visent à préserver les détails fins du type ligne plutôt que les bords.

Le filtrage DPSF opère à un niveau très fin de l'image, et son utilisation n'est intéressante que pour des applications nécessitant un tel niveau de détails, ce qui n'est à priori pas notre cas dans une analyse globale de l'image (mais on peut très bien envisager l' utiliser ensuite, au niveau local, dans certaines zones).

2.3 Méthodes adaptatives: filtrage EPSF

L'idée générale des méthodes de lissage adaptatif est d'appliquer un opérateur "changeant" (*versatile* en anglais) qui s'adapte lui même à la topographie locale du signal à filtrer. Tous les filtres de cette famille peuvent s'utiliser de façon itérative, et la qualité du lissage est sensée augmenter avec le nombre d'itérations.
Notre intérêt s'est porté sur cette famille de filtres.

De nombreux filtres ont été développés, et on peut encore les classer dans différentes catégories (qui se chevauchent parfois plus ou moins) au sein de cette même grande famille, à savoir:

- Les filtres de lissage par convolution, dont la singularité réside dans le choix des coefficients du masque de convolution. Ceux-ci ne sont pas fixés à l'avance mais déterminés de façon adaptative. Ces filtres sont très liés à une détection de contours "sous-entendue" (ITEN, GIF).

- Les filtres de lissage sélectif adaptatif utilisant un ensemble organisé de sous-masques, et opérant une *moyenne* ou un *médian* sur un seul sous-masque, sélectionné à l'aide d'un critère d'homogénéité (EGPR).

- Les filtres de lissage sélectif adaptatif utilisant un nombre fixé de pixels. Contrairement à ceux de la catégorie précédente, ces filtres n'utilisent

pas de sous-masques mais un nombre défini de pixels, sélectionnés de façon adaptative dans la fenêtre de travail, selon différents critères d'homogénéité (KNN), de symétrie (SNN)...

- Les filtres de lissage sélectif adaptatif utilisant un nombre de pixels variable. Ces filtres n'utilisent ni sous-masques ni nombre de pixels fixé à l'avance comme dans les deux catégories précédentes. Ils utilisent un nombre adaptatif de pixels, ceux-ci étant sélectionnés car reconnus comme étant significatifs, à l'aide d'un critère d'homogénéité (ASMT, AFUGM).

- Les filtres "débruiteurs" basés sur l'idée maîtresse qu'un bruit se retrouve sous forme d'une valeur extrémale dans la fenêtre d'étude (NOPEL).

- Les filtres inspirés par des principes physiques tels que la diffusion anisotropique de la chaleur (ASAD) ou le principe du maximum d'entropie (AFUMEP).

3 Objectif des tests réalisés

L'étude des caractéristiques des nombreux filtres développés a été faite, pour la plupart, par leurs auteurs. Par ailleurs, certaines évaluations comparatives entre quelques filtres ont déjà été effectuées et publiées [Chi83, Mas85, Har87, Imm91, Wu92].

Les images-test utilisées sont, soit des images réelles, illustrant ainsi l'intérêt du filtre dans le cadre d'une application [Nag79, Har87, Imm91, Sai91, Sho92], soit des images de synthèse perturbées par un bruit additif parfaitement connu, ce qui permet de comparer les images idéale, bruitée et filtrée [Lev77, Nag79, Har87, Imm91].

L'évaluation d'un filtre EPSF doit se faire en analysant l'image filtrée éventuellement de façon globale (pour voir la qualité du lissage par exemple) mais surtout de facon locale, en certains endroits "intéressants" de l'image (bords, angles, flou, bruit), pour juger de la qualité du filtrage.

De ce fait, l'évaluation d'un filtre EPSF ne peut guère se faire correctement que de façon visuelle, car il est quasiment impossible de quantifier le degré de qualité du filtre à l'aide d'un outil "calculatoire" local et non pas global.

L'instrument de mesure le plus utilisé est alors le système visuel humain [Lev77, Sho92, Imm91, Won93], et on peut également avoir recours à un système de notation (du type "mauvais","moyen", "bon" [Imm91] ou du type "note sur 20" [Mas85]) pour juger de la qualité du filtrage.

Aucun filtre n'est parfait et ne convient à tout type d'image. Ainsi, dans le cadre d'une application précise, le choix du filtre à utiliser ne peut se faire qu'en connaissant d'une part les caratéristiques générales de l'image à traiter, et d'autre part les caractéristiques des différents filtres dont on dispose (c'est-à-dire leurs avantages et inconvénients, et leurs aptitudes générales), afin de pouvoir répondre le mieux possible au besoin.

Dès lors, les tests que nous avons effectués, reposent sur une analyse essentiellement visuelle des résultats obtenus sur de nombreuses images-test (à l'aide d'un certain nombre d'outils), le but étant de juger de la qualité des différents filtres implantés, au regard d'un certain nombre de critères importants tels que:

– l'élimination du bruit impulsionnel,

- la préservation des lignes pour les filtres DPSF (petits détails),
- la préservation des bords (il ne faut pas qu'il y ait introduction de flou au niveau d'un bord franc, ni même d'aggravation du flou au niveau d'un bord déjà flou),
- la préservation des angles,
- l'amplitude du lissage,
- la continuité du lissage (c'est à dire faire attention à ce que le filtre ne crée pas de discontinuités du type grandes "marches d'escalier" ou bien du type "mosaïques". Par ailleurs, si on est en présence d'un dégradé parfait, un bon lissage continu doit le préserver).

Pour ce faire, nous avons utilisé de nombreuses images-test, réelles et de synthèse, dans le but d'étudier tous les points importants précités, et pour détecter d'éventuels cas particuliers.

Nous avons également utilisé quelques outils tels qu'un utilitaire (CoupeXGraph) permettant de zoomer et de voir très en détail le profil d'une ligne (ou d'une colonne) de l'image, et un autre utilitaire (Xsurf) permettant une représentation 3D zoomée de toute partie sélectionnée de l'image. Ces deux outils sont particulièrement utiles, notamment lorsque les variations d'intensité sont trop faibles (l'oeil ne pouvant pas les discerner) et pour faire une étude fine de ce qui se passe aux bords, aux angles...

4 Filtres implantés et résultats

4.1 Préambule

Pour pouvoir distinguer les différents filtres présentés avec un minimum de clarté (et pour éviter toute confusion fâcheuse), il est nécessaire de pouvoir les identifier, et donc de leur attribuer un nom (tous n'en ont pas dans la littérature).

Ainsi, chaque filtre présenté a un nom (ou étiquette) constitué de majuscules, qui représente un sigle d'un libellé pouvant être:

- le nom couramment utilisé pour ce filtre (lorsqu'il en a un),
- le titre de l'article de référence relatif au filtre en question,
- un libellé créé spécialement pour l'occasion, et assez significatif de l'approche utilisée pour la conception du filtre désigné.

Dans ce chapitre, au début de chaque paragraphe relatif à un nouveau filtre, le nom de celui-ci est utilisé pour le désigner, le label dont il est le sigle est explicité, et l'article de référence du filtre est donné.

4.2 ITEN

ITEN : ITerative ENhancement of noisy images
référence : [Lev77]

Ce filtre consiste à faire une moyenne pondérée sur un masque (3x3), où la détermination des coefficients de pondération est basée sur l'hypothèse de quatre directions possibles pour un bord passant par le pixel central de la fenêtre, à savoir 0^o, 45^o, 90^o et 135^o.

Soient f l'image originale, g l'image filtrée, et $W(i,j)$ la fenêtre (3×3) centrée sur le point (i,j); le filtre est défini de la façon suivante:

$$g(i,j) = \sum_{k=-1}^{+1} \sum_{l=-1}^{+1} D^*(k,l) f(i+k, j+l)$$

où D^* est la matrice D (définie ci-après) après normalisation, de telle sorte que le poids central soit égal à $1/9$ au lieu de 1, et que la somme de tous les coefficients soit égale à 1:

$$D = \begin{pmatrix} \alpha\gamma\epsilon & \alpha\gamma\eta & \alpha\zeta\eta \\ \gamma\epsilon\theta & 1 & \delta\zeta\eta \\ \beta\epsilon\theta & \beta\delta\theta & \beta\delta\zeta \end{pmatrix}$$

avec

$$\begin{aligned}
\alpha &= exp(-|\,(a + b + c) - (d + e + f)\,|/\sigma)\\
\beta &= exp(-|\,(g + h + i) - (d + e + f)\,|/\sigma)\\
\gamma &= exp(-|\,(a + b + d) - (c + e + g)\,|/\sigma)\\
\epsilon &= exp(-|\,(a + d + g) - (b + e + h)\,|/\sigma)\\
\zeta &= exp(-|\,(c + f + i) - (b + e + h)\,|/\sigma)\\
\eta &= exp(-|\,(b + c + f) - (a + e + i)\,|/\sigma)\\
\theta &= exp(-|\,(d + g + h) - (a + e + i)\,|/\sigma)
\end{aligned}$$

σ étant un paramètre empirique et *a, b, c, d, e, f, g, h* et *i* étant les luminances des pixels de la fenêtre W, comme l'illustre la figure 1.

a	b	c
d	e	f
g	h	i

FIG. 1 - *Appellation des luminances des pixels de la fenêtre W*

ITEN élimine assez mal le bruit impulsionnel, et n'y parvient qu'à condition d'être itéré un grand nombre de fois (comme un moyenneur classique). Il préserve les bords francs nets, mais arrondit les angles. S'il y a très peu de flou, ITEN peut réhausser les bords francs. Si le flou est important, son application aggrave la situation en déplaçant des frontières, en déformant des bords courbes (par création de plusieurs bords droits noyés dans du flou) et en déformant sévèrement les angles. L'amplitude du lissage est très bonne à condition d'itérer longtemps (ce qui améliore le résultat).

Par ailleurs, le paramètre empirique σ détermine l'amplitude du lissage. Plus σ est grand plus le lissage est important, mais au détriment des bords, des angles et de la netteté, puisque l'image peut finir noyée dans un flou ambiant intense si σ est trop grand. Il vaut mieux prendre une valeur de σ raisonnable et itérer davantage afin d'obtenir un bon lissage tout en préservant le mieux possible les bords. Lorsqu'on itère, le lissage obtenu est bien continu et respecte les dégradés, même si quelques petites discontinuités résiduelles subsistent très curieusement (généralement là où se trouvait une densité plus importante d'impacts de bruit impulsionnel, que le masque (3×3) ne suffit pas à résorber).

4.3 GIF

GIF : Gradient Inverse Filter
référence : [Wan81]

Ce filtre consiste à faire la moyenne pondérée des luminances de tous les pixels de la fenêtre (3×3) centrée sur le pixel (i, j) à traiter, où le coefficient attribué à chaque pixel est, à un coefficient de normalisation près, l'inverse du gradient. L'idée de base étant d'accorder un poids moindre aux pixels qui ont une différence importante avec le pixel central, de façon à bien lisser dans une région et ne pas trop lisser au niveau des bords.

L'algorithme est le suivant:

1. on calcule l'inverse des gradients des pixels:

$$h_{k,l} = \begin{cases} \dfrac{1}{|f(i+k, j+l) - f(i,j)|} & \text{si} \quad f(i+k, j+l) \neq f(i,j) \\ \dfrac{1}{2} & \text{si} \quad f(i+k, j+l) = f(i,j) \end{cases}$$

où $k, l = \{-1, 0, +1\}$

2. on calcule les coefficients de pondération

$$w_{k,l} = \frac{1}{2} \cdot \frac{h_{k,l}}{\sum_{k \neq 0} \sum_{l \neq 0} h_{k,l}} \qquad \text{pour } (k,l) \neq (0,0) \qquad \text{et } w_{0,0} = \tfrac{1}{2}$$

3. on calcule la réponse du filtre par convolution:

$$g(i,j) = \sum_k \sum_l w_{k,l} \cdot f(i+k, j+l)$$

GIF élimine le bruit impulsionnel par lissages successifs en itérant un grand nombre de fois, tout comme ITEN. Il ne préserve ni les bords ni les angles, mais les dégrade en introduisant un léger flou, qui ne s'étend pas pour autant dans le cadre d'un processus itératif. En effet, sur un bord franc parfait, la première itération crée systématiquement une couronne de flou de deux pixels d'épaisseur au niveau du contour, mais celle-ci ne s'étend pas au cours des itérations suivantes. Il en va de même pour les angles: ils sont arrondis à la première itération, mais ce phénomène ne s'étend pas ou presque pas par la suite.

GIF ne réhausse pas les bords flous et parfois même les aggrave, et on a souvent à faire à une image noyée dans du flou après un usage itératif de

GIF. L'amplitude du lissage effectué dépend du type d'images traitées, mais reste en général faible, voire insuffisant. La continuité du lissage quant à elle, est moyenne.

4.4 EGPR

EGPR : EdGe PReserving smoothing technique
référence : [Nag79]

Le principe de cette technique consiste à faire tourner une "barre allongée" (sous-masque) autour du point (x, y) à traiter (voir fig. 2), et à sélectionner la position pour laquelle le voisinage de (x, y) formée par la barre soit le plus homogène possible, et de faire ensuite la moyenne des pixels de ce voisinage. L'idée étant que le voisinage sur lequel on effectue un moyennage ne doit pas contenir de bord, car sinon on bruite ce bord.

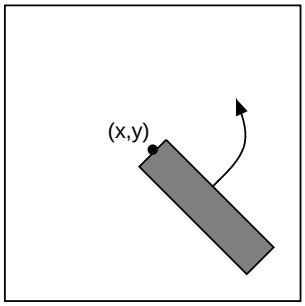


FIG. 2 - *Barre tournant autour de* (x, y)

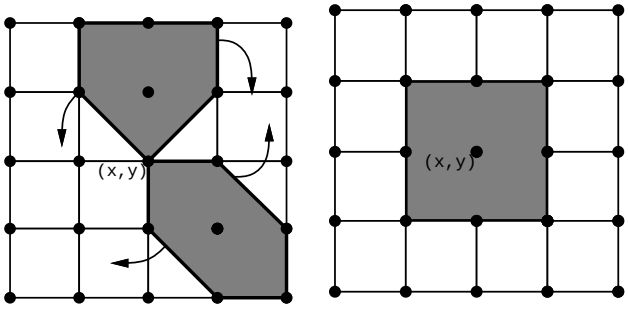


FIG. 3 - *La réalisation des masques comprend 4 pentagones, 4 hexagones et un carré*

En pratique, on utilise un masque (5×5) dans lequel on définit neuf sous-masques dont huit en forme de barre (quatre pentagones, quatre hexagones et un carré), qui sont construits de façon discrète de la manière indiquée par la figure 3; l'algorithme est alors le suivant:

1. tourner le sous-masque barre autour du point (x, y)
2. détecter la position du sous-masque dont la variance en luminance est minimale
3. attribuer au point (x, y) la moyenne des luminances des pixels du masque correspondant à la position sélectionnée

Plus formellement, soient f l'image originale et g l'image filtrée, et soient $W_1(i,j), \cdots, W_9(i,j)$ les neuf sous-masques de la fenêtre (5×5) centrée en (i,j), on a:

$$g(i,j) = \frac{1}{N_p} \sum_{(k,l) \in W_p(i,j)} f(k,l)$$

où $W_p(i,j)$ est le p-ème sous-masque de la fenêtre $W(i,j)$ centrée en (i,j)
$N_p = \text{Card } W_p(i,j)$

et où p est choisi tel que $\sigma^2(W_p) = \min_r(\sigma^2(W_r))$, $\sigma^2(W_r)$ étant la variance des luminances des pixels du sous-masque W_r.

EGPR atténue très bien le bruit impulsionnel et ce d'autant mieux qu'il est utilisé de façon itérative. Il ne l'élimine pas brutalement en une fois comme le fait un *médian* par exemple, mais par le biais du moyennage sur le sous-masque le plus homogène. Ainsi, une valeur aberrante le devient de moins en moins grâce au lissage. L'utilisation des neuf sous-masques fait qu'EGPR préserve bien les bords francs, droits et même courbes. Les angles sont bien conservés: s'ils sont droits, ils le restent parfaitement, mais s'ils sont très aigüs, leur pointe (le pixel sommet) est supprimée. Toutefois, lorsque cela se produit, l'angle n'est pas trop dénaturé car même dans le cadre d'un processus itératif, il ne continue pas à être rongé comme c'est le cas avec de nombreux autres filtres (*moyenneur classique* ou *médian standard* notamment).

Utilisé de façon itérative, EGPR arrive également à réhausser les bords flous (et même très flous) de façon impressionnante. En effet, un pixel flou entre deux régions tend à se rapprocher de plus en plus de la valeur moyenne du sous-masque homogène dont il fait partie, et petit à petit, la zone de flou diminue jusqu'à disparaître. Mais, bien qu'il réhausse les bords, il ne les restitue pas toujours très bien et peut détériorer ainsi les frontières. EGPR a un pouvoir de lissage très important et efficace, qui s'améliore avec le nombre

d'itérations, comme on peut s'y attendre s'agissant d'une moyenne non pondérée. Toutefois, l'utilisation judicieuse des sous-masques, et plus exactement la sélection du meilleur sous-masque sur lequel on effectue le lissage, permet de ne pas introduire un flou ambiant sur l'image, qui est le fléau et le défaut majeur des méthodes classiques de moyennage.

En revanche, les résultats concernant la continuité du lissage ne sont pas bons. Dans une région à lisser, EGPR tend à lisser de façon homogène sur le sous-masque choisi sous l'hypothèse de constance par morceaux, et le résultat du lissage sur un sous-masque (en un endroit de l'image), tend à devenir constant. Si une région est ou doit être constante alors qu'elle est bruitée (même fortement), EGPR est parfait car il restitue la constance de la région. Par contre, si une même région n'est pas constante mais constituée d'un dégradé continu, même non bruité, EGPR tend à créer des marches importantes, des mosaïques constantes éparpillées (dues au *piecewise constant* sur les sous-masques) et même à modifier et dénaturer les lignes des gradients.

4.5 KNN

KNN : K-Nearest Neighbor filter
référence : [Dav78]

Ce filtre remplace la valeur du pixel central d'une fenêtre carrée (masque) par la moyenne des luminances des K pixels les plus proches du pixel central (en luminance). K est un paramètre du filtre et correspond au nombre maximum de pixels de la fenêtre carrée (pixel central exclu) qui peuvent être du même côté d'un segment de droite passant par le pixel central. (5 pour une fenêtre (3×3), 14 pour une fenêtre (5×5),...,$(2N+3)N$ pour une fenêtre $(2N+1) \times (2N+1)$).

Remarque: le filtre décrit est également appelé KNN-mean, car un *moyennage* est effectué. Si cette opération de moyennage sur les luminances des pixels sélectionnés est remplacée par un *médian*, on définit alors le filtre KNN-median.

KNN élimine très bien le bruit impulsionnel. Il préserve les bords francs droits, mais peut parfois détériorer les bords courbes en les rendant linéaires par morceaux lorsque la taille du masque est trop petite (assez rare). Par exemple, il tend à polygonaliser un petit disque, et ce de façon d'autant plus marquée que la taille du filtre est petite. KNN arrondit tous les angles, de plus en plus à chaque itération, jusqu'à obtenir une courbure qu'il ne puisse plus modifier. Plus la taille du filtre est grande, plus les angles sont arrondis. Il redresse très bien les bords flous en les réhaussant petit à petit à chaque

itération. KNN effectue également un bon lissage, dont la continuité est assez correcte (ni excellente ni mauvaise). L'amplitude et la continuité du lissage s'améliorent lorsque la taille du filtre augmente.

4.6 SNN

SNN : Symmetric Nearest Neighbor filter
référence : [Har87]

Le filtre SNN utilise à la fois l'information spectrale (au travers de la contrainte de proximité en luminance: "les plus proches voisins") et l'information spatiale (au travers d'un critère de symétrie) (voir fig. 4) .

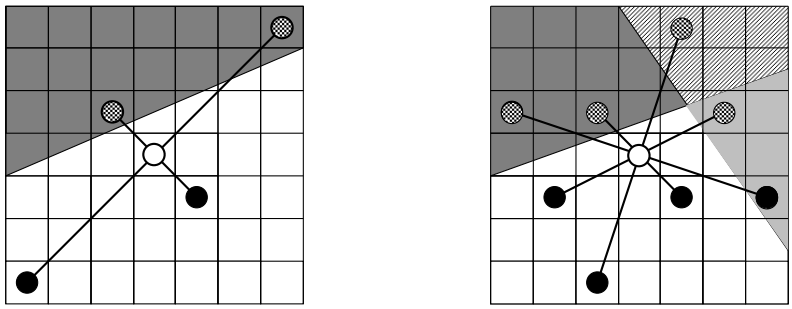


FIG. 4 - *Pour chaque paire symétrique, la valeur la plus proche de celle du pixel central est sélectionnée (représentée par un point noir)*

Pour calculer la nouvelle valeur du pixel central d'une fenêtre carrée, on sélectionne la moitié des luminances des pixels de la fenêtre en gardant, pour chaque paire de pixels symétrique autour du pixel central, la valeur la plus proche de celle du pixel central. Dans le cas d'égalité au niveau d'une paire, on sélectionne la moyenne de cette paire. Ensuite, on attribue soit la *moyenne* des luminances sélectionnés (SNN-mean) soit la *médiane* (SNN-median).

Plus formellement, pour une fenêtre carrée $(2N+1) \times (2N+1)$ centrée sur le pixel (x, y), on sélectionne les luminances de la façon suivante:

Pour chaque paire de pixels symétriques $\{(x+i, y+j), (x-i, y-j)\}$, $-N \leq i, j \leq N$

$$\begin{array}{rl} \text{sélectionner} & f(x+i,y+j) \\ \text{si} & |f(x,y)-f(x+i,y+j)| < |f(x,y)-f(x-i,y-j)| \end{array}$$

$$\begin{array}{rl} \text{sélectionner} & f(x-i,y-j) \\ \text{si} & |f(x,y)-f(x+i,y+j)| > |f(x,y)-f(x-i,y-j)| \end{array}$$

$$\begin{array}{rl} \text{sélectionner} & \frac{1}{2}[f(x+i,y+j)+f(x-i,y-j)] \\ \text{si} & |f(x,y)-f(x+i,y+j)| = |f(x,y)-f(x-i,y-j)| \end{array}$$

SNN élimine assez bien le bruit impulsionnel, préserve bien les bords francs nets et parfaits (pas de flou), mais ronge légèrement les pointes des angles en les atténuant, et ce d'autant plus que la taille du masque est grande. Il parvient bien à réhausser les bords flous, mais lorsqu'ils sont entachés de bruit de quantification, SNN les restaure mal et introduit de nombreuses discontinuités qui détériorent les frontières. Bien que ce phénomène s'atténue quand la taille du masque augmente, il ne disparaît jamais totalement et cela peut être gênant. L'amplitude du lissage est moyenne, voire bonne, et s'améliore avec la taille du masque. La continuité varie également avec la taille du masque, et peut aller du médiocre au correct, mais il subsiste toujours des marches et de petites mosaïques constantes éparpillées, surtout si le bruit de quantification est intense.

4.7 ASMT

ASMT : the Average Selected sMoothing Technique
référence : [Asa81]

Ce filtrage consiste à attribuer au pixel central (x,y) de la fenêtre (3×3) de travail centrée en (x,y), la moyenne des luminances des pixels de la fenêtre, dont la différence (en luminance) avec le pixel central est inférieure à la moyenne des différences entre le pixel central et ses huit voisins.

Autrement dit, si f est l'image originale et g l'image filtrée, on a:

$$g(i,j) = \frac{1}{\sum\limits_{(k,l)\in N(i,j)} \omega(k,l;i,j)} \sum_{(k,l)\in N(i,j)} \omega(k,l;i,j) f(k,l)$$

$$\text{avec} \quad \omega(k,l;i,j) = \begin{cases} 1 & \text{si} \quad |f(i,j)-f(k,l)| \le \beta(i,j) \\ 0 & \text{sinon} \end{cases}$$

$$\text{et} \quad \beta(i,j) = \frac{1}{8} \sum_{(k,l)\in N_8(i,j)} |f(i,j)-f(k,l)|$$

$N_8(i,j)$ étant le voisinage 8-connexe du point central (i,j) défini par: $N_8(i,j) = \{(k,l)/\max(|i-k|,|j-l|) = 1\}$, et $N(i,j)$ étant la fenêtre (3×3) entière, c'est à dire $N(i,j) = N_8(i,j) \cup \{(i,j)\}$.

ASMT atténue le bruit impulsionnel petit à petit lorsqu'il est utilisé de façon itérative. Il préserve bien les bords francs et les angles. Lorsqu'ils sont nets, les angles ne sont absoluement pas rongés ni arrondis, même lorsqu'ils sont aigüs. ASMT peut réhausser les bords flous à condition que la zone de flou soit faible. Si cette zone est importante, ASMT est capable de réhausser une structure arrondie sur un fond constant, mais pour une structure anguleuse, il crée parfois des artefacts assez sévères au niveau des angles, des incurvations de bords droits, et aussi des modifications de frontières assez bizarres et inattendues, ce qui peut être grave de conséquence. L'amplitude du lissage d'ASMT est moyenne à correcte mais n'atteint pas celle d'EGPR par exemple. En effet, le lissage reste très local (3×3) et ne permet pas un bon lissage d'une trame de texture ou de tissu par exemple. Le lissage respecte assez bien les dégradés continus parfaits (non bruités-image de synthèse), mais lorsqu'ils sont bruités, de très petites mosaïques quasi-constantes apparaissent, de façon nettement moins flagrante qu'avec EGPR néanmoins.

4.8 AFUGM

AFUGM : Adaptative Filtering Using Gradient Morphology
référence : [Sho92]

Cette méthode est basée sur l'analyse morphologique de la courbe créée par la liste des luminances triées (statistiques d'ordre) de la fenêtre (3×3) centrée sur le pixel à traiter (le fait de travailler sur une fenêtre de taille (3×3) assurant l'adjacence de tous les pixels de la fenêtre au pixel central). En analysant la pente du gradient de cette courbe autour du pixel à traiter, on divise l'ensemble des statistiques d'ordres de la fenêtre en différentes zones (pentes ou segments), en utilisant un critère morphologique (voir fig. 5). La réponse du filtre est alors adaptée en fonction de l'appartenance du pixel central à telle ou telle zone.

L'algorithme peut s'écrire de la façon suivante:

1. trier dans l'ordre croissant les luminances des éléments de la fenêtre $W(x,y)$ donnant la liste des statistiques d'ordre $(L(W_1), L(W_2), \cdots, L(W_9))$ relative à la liste de pixels $(W_1, W_2, \cdots, W_9)$ ($L(W_i)$ étant la luminance du pixel W_i)

2. considérer la courbe formée par cette liste de valeurs (avec $(W_1, W_2, \cdots, W_9)$ en abscisses et $(L(W_1), L(W_2), \cdots, L(W_9))$ en ordonnées)

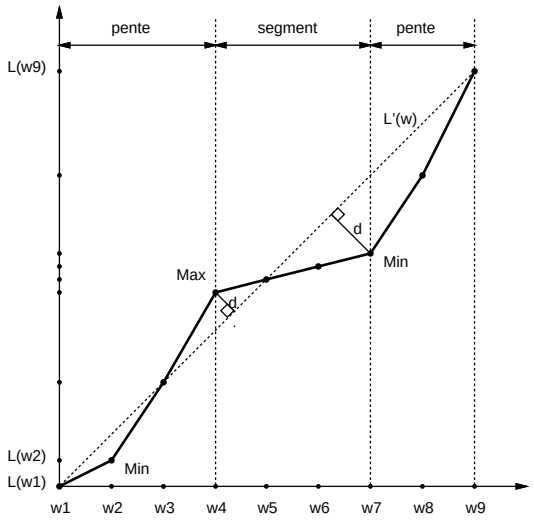


FIG. 5 - *La courbe des statistiques d'ordre est décomposée en différentes zones (pentes et segments) par une analyse morphologhique.*

3. considérer la fonction linéaire $L'(W)$ reliant $(W_1, L(W_1))$ à $(W_9, L(W_9))$

4. déterminer les distances euclidiennes d de chaque point $(W_i, L(W_i))$ à la droite $L'(W)$

5. déterminer les minima et maxima locaux discrets des distances à $L(W)$.
   Un maximum local est défini comme étant un point situé au dessus de la droite $L'(W)$, dont la distance euclidienne le séparant de la droite $L'(W)$ est un maximum.
   Un minimum local est défini comme étant un point situé en dessous de la droite $L'(W)$, dont la distance euclidienne le séparant de la droite $L'(W)$ est un maximum.

6. définir la morphologie de la courbe en différentes parties ou zones, à savoir pentes ou segments.
   Une pente est définie comme étant un ensemble de points compris entre
   -soit un minimum local et un maximum local
   -soit un minimum local et W_9
   -soit W_1 et un maximum local
   Un segment est défini comme étant un ensemble de points compris entre
   -soit un maximum local et un minimum local
   -soit W_1 et un minimum local
   -soit un maximum local et W_9

7. déterminer la réponse du filtre en fonction de la position du pixel central W_c à traiter.
   Supposons le pixel W_c situé sur la zone $[W_i, \cdots, W_{i+n}]$. Si $[W_i, \cdots, W_{i+n}]$ est un segment, on fait la moyenne sur ce segment; sinon, on choisit la valeur la plus proche de $L(W_c)$ qui appartient à un segment, et qui est donc soit W_i soit W_{i+n}.

AFUGM ne supprime le bruit impulsionnel que s'il y en a très peu (un impact au plus dans la fenêtre de travail (3×3)), car sinon, il ne peut rien faire, même en itérant. Il préserve parfaitement les bords (droits et courbes), ainsi que les angles (même aigüs) lorsqu'ils sont parfaitement nets, mais il n'est pas capable du tout de réhausser les bords flous. La taille du masque étant figée à (3×3), on ne peut pas adapter AFUGM à un type d'image donnée, et comme le nombre de pixels sélectionnés est souvent faible, AFUGM travaille à un niveau de détail trop petit, et le lissage effectué est très faible, voire quasi-imperceptible.

4.9 NOPEL

NOPEL : NOise Peak ELimination filter
référence : [Imm91]

Ce filtre repose sur l'idée de base qu'un pixel peut être défini comme un bruit si sa valeur est extrémale parmi l'ensemble des valeurs des pixels de l'image vus à travers une fenêtre carrée (3×3) centrée sur ce pixel.

Soient f l'image originale et g l'image filtrée et W la fenêtre (3×3) centrée sur le pixel (x, y), NOPEL s'exprime alors comme suit:

$$g(x,y) = \begin{cases} NLVM & \text{si} \quad [(f(x,y) = MAX) \quad \text{et} \quad (NLVM \neq MIN)] \\ NHVM & \text{si} \quad [(f(x,y) = MIN) \quad \text{et} \quad (NHVM \neq MAX)] \\ f(x,y) & \text{sinon} \end{cases}$$

$$\text{avec} \qquad NLVM = \max_{(x,y)} \{W(x,y) < MAX\}$$

$$NHVM = \min_{(x,y)} \{W(x,y) > MIN\}$$

$$\text{et} \qquad MAX = \max_{(x,y)} W(x,y)$$

$$MIN = \min_{(x,y)} W(x,y)$$

NOPEL élimine très bien le bruit impulsionnel dans le cas général d'une image réelle, mais il n'est pas parfait. En général, NOPEL préserve parfaitement les bords et les angles lorsque ceux-ci sont très nets, mais lorsqu'il y a du flou, non seulement il ne réhausse pas les bords, mais en plus, il arrive qu'il les détériore davantage. Le lissage que fait NOPEL est d'une ampleur et d'une continuité limitées puisqu'il se contente de remplacer un point bruit par une valeur existante.

4.10 ASAD

ASAD : Adaptative Smoothing using Anisotropic Diffusion
référence : [Sai91]

Ce filtre utilise un paramètre k servant de *scale dimension parameter* qui détermine l'amplitude des bords à préserver durant le processus de lissage. Il consiste à convoluer l'image de façon itérative par un petit masque (3×3) dont les coefficients de pondération correspondent à une mesure du degré de continuité du signal image en chaque point. Ce processus itératif engendre une implantation de l'équation de diffusion anisotropique de la chaleur (similaire notamment, à celle proposée par Perona et Malik [Per90]), en résolvant l'équation:

$$\frac{\partial I}{\partial t} = \nabla(c \nabla I)$$

(I étant ici l'image, c le coefficient de diffusion, et t le temps).

Soit $I^t(x,y)$ le signal image au point (x,y) à l'itération t. On définit $d^t(x,y)$ comme étant l'amplitude du gradient en (x,y) calculé sur une fenêtre (3×3). La fonction $w^t(x,y)$ mesurant la valeur ou le degré de continuité en (x,y) est définie par:

$$w^t(x,y) = f(d^t(x,y)) = e^{-\frac{(d^t(x,y))^2}{2k^2}}$$

$$\text{avec} \qquad d^t(x,y) = \sqrt{G_x^2 + G_y^2}$$

$$\text{et} \qquad G_x = \frac{\partial I^t(x,y)}{\partial x} \qquad\qquad G_y = \frac{\partial I^t(x,y)}{\partial y}$$

k étant le paramètre d'échelle.

Le signal lissé à l'itération *(t+1)* est alors calculé en chaque point par:

$$I^{t+1}(x,y) = \frac{1}{N^t(x,y)} \sum_{i=-1}^{+1} \sum_{j=-1}^{+1} I^t(x+i, y+j) w^t(x+i, y+j)$$

$$\text{avec} \qquad N^t(x,y) = \sum_{i=-1}^{+1} \sum_{j=-1}^{+1} w^t(x+i, y+j)$$

La particularité du filtre ASAD est son paramètre d'echelle k, dont le choix est très important, car il détermine complètement le résultat du filtre. Celui-ci permet d'adapter ASAD à un type d'image, à une application donnée, ou à un besoin précis. Ce paramètre k permet en fait de conditionner la "hauteur de la marche" au delà de laquelle il n'y aura pas de lissage, et même au contraire, où il y aura du réhaussement de contour (flous). Plus k est grand plus le lissage est important et généralisé, et l'image peut finir noyée dans le flou car très peu de marches (donc de gradients) sont suffisamment importants pour être respectés. A l'inverse, plus k est petit, plus le lissage est faible et plus les discontinuités sont mises en valeur car reconnues significatives. Ainsi, avec k bien choisi (en fonction du besoin), on peut répondre à tous les critères auxquels on s'intéresse ici (ou presque).

ASAD élimine très bien le bruit impulsionnel, préserve parfaitement les bords francs (droits et courbes), il conserve un peu moins bien les angles puisqu'il leur atténue la pointe, et réhausse très bien les bords flous, de façon rapide (peu d'itérations) et impressionnante. ASAD effectue un lissage d'une bonne amplitude, mais son défaut, comme celui de nombreux autres filtres, est que son lissage n'est pas toujours très continu, et tend à produire des surfaces constantes par morceaux.

4.11 AFUMEP

AFUMEP : Adaptative Filter Using the Maximum Entropy Principle
références : [Won93]

Le filtre présenté ici est un filtre que nous avons développé en nous inspirant d'une récente méthode de *clustering* développée par Wong [Won93], basée sur la théorie de l'information et la mécanique statistique.

Soient x une donnée et y un centre de classe, on choisit de mesurer la compacité de la classe à l'aide de la fonction coût $e(x) = (x-y)^2$. Considérons la contribution $P(x)$ de la donnée x envers le *cluster* y. En maximisant l'entropie $-\sum_x P(x) log P(x)$ sous la contrainte $\sum_x P(x)e(x) = C$, on obtient: $P(x) = e^{-\beta(x-y)^2}/Z$, où $Z = \sum_x e^{-\beta(x-y)^2}$. Pour faire le lien avec la thermodynamique, on définit "l'énergie libre" $F = -\frac{1}{\beta} log Z$. Or, on sait qu' à l'équilibre, tout système thermodynamique tend vers la configuration qui minimise son "énergie libre". Autrement dit, on veut que $\partial F/\partial y = 0$, ce qui entraîne:

$$y = \sum_x \frac{x e^{-\beta(x-y)^2}}{\sum_x e^{-\beta(x-y)^2}} \tag{1}$$

Le raisonnement a été ici restreint à un ensemble de données unidimensionnelles, mais il peut se généraliser aux données multidimensionnelles en considérant une autre fonction coût $e(x)$ prenant en compte l'information spatiale. Ainsi, pour un signal 1D où y est l'intensité désirée (après filtrage) du signal à l'instant x, on a:

$$y = \sum_i \frac{y_i e^{-\alpha(x_i-x)^2} e^{-\beta(y_i-y)^2}}{\sum_i e^{-\alpha(x_i-x)^2} e^{-\beta(y_i-y)^2}} \tag{2}$$

Dans le cas d'une image 2D, soit z_i l'intensité du i-ème pixel de coordonnés (x_i, y_i), on a:

$$z = \sum_i \frac{z_i \theta_i \delta_i}{\sum_i \theta_i \delta_i} \tag{3}$$

avec $\theta_i = e^{-\alpha d_p^2}$ $\delta_i = e^{-\beta d_l^2}$

et $d_p = \sqrt{(x_i - x)^2 + (y_i - y)^2}$ $d_l = (z_i - z)$

α est un paramètre de mesure d'échelle lié au domaine d'entrée (notion de distance spatiale), et β est un paramètre de mesure d'échelle lié au domaine de sortie (notion de distance spectrale).

L'équation (1), de même que la (2) et la (3), sont des équations à résoudre en y (en z pour la (3)). Wong a proposé une méthode de résolution de l'équation (1) dans le cadre de sa méthode de *clustering*. Cette méthode consiste à résoudre l'équation numériquement (la solution ne peut pas se calculer analytiquement) en itérant un schéma du type $y = F(y)$ pour converger vers la solution.

Dans le cadre de la méthode de *clustering* développée par Wong, le schéma itératif que l'on vient de décrire, est lui même englobé dans une boucle itérative. Celle-ci consiste à partir d'une valeur de β très élevée (pour laquelle tous les pixels sont des régions) et à la faire décroître petit à petit à chaque

itération ($\beta_{i+1} = \beta_i/1.05$), jusqu'à ce qu'il n'y ait plus de *merging*.

Ce processus permet de générer un arbre structuré représentant l'évolution de l'agrégation des *clusters* dans l'espace des échelles. Ensuite, l'étude des différentes branches (bifurcations) de l'arbre permet à Wong de déterminer les "bons" *clusters*, parmi tous les clusters dans l'espace des échelles, à l'aide d'un critère lié à "l'énergie libre fractionnelle" (FFE). Toutefois, le choix de ce critère est délicat, comme le reconnaît Wong: "*It is very difficult to define a universaly accepted criterion because clusters really need to be interpreted in the context of the specific application*". Ainsi, la méthode (processus itératif) proposée par Wong et décrite plus haut, est très intéressante et adaptée à sa technique de *clustering*, mais elle ne semble pas forcément convenir, ou être la mieux appropriée pour le filtrage.

Dans l'implémantation que nous avons développée, les deux paramètres d'échelle α et β sont choisis par l'utilisateur. Ils sont liés aux notions de distances spatiale et spectrale qui conditionnent entièrement le résultat du filtre, et il ne faut donc pas les lier à la notion de *merging* si l'on veut pouvoir adapter le filtre à un type d'image, en fonction d'une application donnée (et donc orienter le résultat vers ce qu'on en attend). En effet, les valeurs des paramètres α et β, ainsi que la valeur initiale de l'intensité, déterminent l'optimum local vers lequel va converger le processus itératif.

Par ailleurs, sur une image, il est inutile de considérer tous le pixels, car au delà d'une certaine distance, l'information apportée est très négligeable voire numériquement nulle ("zéro machine"), et le filtrage prend énormément de temps inutilement. Ainsi, on choisit de travailler sur une fenêtre carrée de taille $(2n+1) \times (2n+1)$ centrée sur le pixel courant à traiter, sachant d'une part, que plus n est grand, plus le lissage sera important, et d'autre part, que pour un α donné, il ne sert strictement à rien de prendre n supérieur à $\sqrt{-log\epsilon_m/\alpha}$, ($\epsilon_m$ étant la plus petite valeur positive non nulle disponible numériquement sur le calculateur utilisé).

Soient α et β déterminés par l'utilisateur et $(2n+1) \times (2n+1)$ la taille de la fenêtre de travail, le filtre que nous avons développé s'utilise de façon itérative et s'exprime comme suit:

$$I^{t+1}(x,y) = \frac{1}{N^t(x,y)} \sum_{i=-n}^{+n} \sum_{j=-n}^{+n} I_{ij}^t(x,y)\theta_{ij}^t(x,y)\delta_{ij}^t(x,y)$$

avec $$N^t(x,y) = \sum_{i=-n}^{+n} \sum_{j=-n}^{+n} \theta_{ij}^t(x,y)\delta_{ij}^t(x,y)$$

et $$\theta_{ij}^t(x,y) = e^{-\alpha[p_{ij}^t(x,y)]^2} \qquad \delta_{ij}^t(x,y) = e^{-\beta[l_{ij}^t(x,y)]^2}$$

$$p_{ij}^t(x,y) = \frac{\sqrt{(x_i-x)^2+(y_j-y)^2}}{\sqrt{L_x^2+L_y^2}} \qquad l_{ij}^t(x,y) = \frac{I_{ij}^t(x,y)-I^t(x,y)}{I_{max}-I_{min}}$$

avec les notations $I_{ij}^t(x,y) = I^t(x_i,y_i)$, $x_i = x+i$ et $y_j = x+j$

La particularité et l'intérêt du filtre AFUMEP réside dans ses deux paramètres d'échelle α et β. Le premier est lié à la notion de distance spatiale (entre pixels) et permet de pondérer l'information apportée par un pixel en fonction de sa distance au pixel traité ("confiance spatiale"), tandis que le second, lié à la notion de distance spectrale (entre intensités) permet de pondérer l'information apportée par un pixel en fonction de sa propre valeur, au regard de l'intensité du pixel traité ("confiance spectrale"). Plus β est grand, moins le lissage est important, car la "hauteur de la marche" au delà de laquelle on interdit de lisser, est faible, et donc toutes les dicontinuités (même faibles) subsistent, et à la limite, l'image ne change pas. A l'inverse, plus β est faible, plus le lissage est important. En effet, si on regarde l'équation *(3)*, on constate que pour $\beta = \infty$, toute donnée z_i, avec θ_i non nul, satisfait l'équation, et lorsque β décroît, le nombre de solutions decroît. De même, quand $\alpha = \infty$, θ_i vaut 1 si $x_i = x$ et vaut zéro sinon, et tous les pixels sont ainsi préservés. En revanche, une faible valeur de α permet à plus de voisins de contribuer au lissage. Par ailleurs, pour pouvoir profiter pleinement de l'effet de la mesure de "confiance spatiale" dosée par α, il ne faut pas utiliser une taille de masque trop petite (une taille de (3×3) est ridicule et sans aucun sens). Ainsi, avec des paramètres α et β bien choisis, en fonction du besoin et du résultat souhaité, on peut répondre à tous les critères auxquels on s'intéresse ici.

AFUMEP permet d'éliminer le bruit impulsionnel, de préserver parfaitement les bords (droits ou courbes) et les angles, de réhausser les bords flous (mais pas aussi bien que le fait ASAD). Quant au lissage, son amplitude et sa continuité sont impressionnantes et sans égales parmi tous les filtres implantés et testés.

5 Tableau de synthèse

Le tableau présenté dans cette section se veut être un récapitulatif succinct des aptitudes des différents filtres implantés, au regard des critères auxquels on s'est essentiellement intéressés.

Comme on l'a vu, la qualité d'un filtre peut varier de manière sensible en fonction du type d'image, de différents cas particuliers, ou encore des paramètres utilisés. Il est donc difficile de les noter puisque selon les cas, pour un critère donné, un même filtre peut être jugé très différemment.

Aussi, pour noter l'aptitude à répondre aux différents critères considérés des différents filtres implantés, on a choisi d'utiliser un système de notation volontairement assez vague, comprenant cinq notes possibles (A:"très bon", B:"bon", C:"moyen", D:"mauvais", E:"très mauvais").

Le tableau présenté à la figure 6, en reflétant les aptitudes générales des filtres, permet d'avoir une bonne idée d'ensemble et de sélectionner un certain nombre de filtres pour un besoin donné. Néanmoins, pour plus de précisions, il convient de se reporter aux commentaires sur chacun des filtres.

	ITEN	GIF	EGPR	KNN	SNN	ASMT	AFUGM	NOPEL	ASAD	AFUMEP
élimination du bruit impulsionnel	CD	C	AB	A	B	B	CD	A	A	A
préservation des bords nets	BC	DE	BCD	ABC	CD	BC	AB	BCD	A	A
préservation des angles	D	CD	BCD	E	CD	A	AB	AB	B	AB
redressement des bords flous	CDE	E	AB	B	C	BC	E	E	A	AB
amplitude du lissage	AB	CDE	B	AB	ABC	B	E	D	A	A
continuité du lissage	C	D	E	C	CD	CD		D	BC	A

FIG. 6 - *Tableau de synthèse*

6 Application aux paires d'images stéréo

6.1 Introduction

La qualité des résultats obtenus dans le cadre d'un processus de reconstruction de scènes 3D dépend pour beaucoup de la précision des résultats des étapes fondamentales telles que la calibration, la segmentation, l'analyse photométrique, dont la réussite est conditionnée par la qualité des informations manipulées. Ainsi, par exemple, la calibration (qui se doit d'être la plus précise possible) souffre des distortions photométriques et géométriques introduites par le système d'acquisition, la segmentation est souvent incertaine pour départager nettement et précisément des régions adjacentes (flou et manque de netteté des frontières), quant à l'analyse des gradients d'irradiance, elle est fortement perturbée par l'importance du bruit de quantification.

L'objet de cette section est de s'intéresser à l'application du filtrage EPSF pour le prétraitement de paires d'images stéréo, dans le cadre du processus de reconstruction 3D, et de voir dans quelle mesure il est possible d'améliorer les différentes phases en amont des reconstructions 3D géométrique et photométrique.

6.2 Intérêt pour la calibration

L'analyse des paires stéréoscopiques nécessite, notamment pour la reconstruction tridimensionnelle, la connaissance du déplacement entre les deux vues, ainsi que les déformations produites par la caméra.

Le processus de calibration consiste à extraire de l'image de la mire, des caractéristiques qui permettront d'estimer les paramètres du modèle caméra, son orientation et son positionnement (qu'on appelle respectivement paramètres intrinsèques et extrinsèques). La qualité des résultats dépend directement de la précision obtenue sur les attributs extraits.

6.2.1 Calibration avec une mire à carrés

La calibration de caméras à l'aide d'une mire à carrés est la technique développée par l'équipe du projet Robotvis (INRIA - Sophia-Antipolis). Elle consiste à placer dans le champ des caméras une mire biplanaire à la surface de laquelle sont dessinés des carrés pleins. Les paramètres sont estimés en minimisant au sens des moindres carrés la différence de localisation entre

l'ensemble des images des points d'intérêt et la projection du modèle tridimensionnel de la mire. Le problème, non linéaire, est résolu itérativement en peu d'étapes, car le point initial obtenu par la méthode de Toscani est proche de la solution recherchée [Tos87]. Dans cette méthode, les points d'intérêt en question, qui sont les attributs à extraire sur les images de la mire, sont les coins des carrés. Ceux-ci sont déterminés par le calcul des intersections des droites passant par les bords des carrés de la mire, projetés sur les images. Pour cela, on considère les droites qui approchent au mieux les bords (optimisation), présents sur l'image des contours extraits.

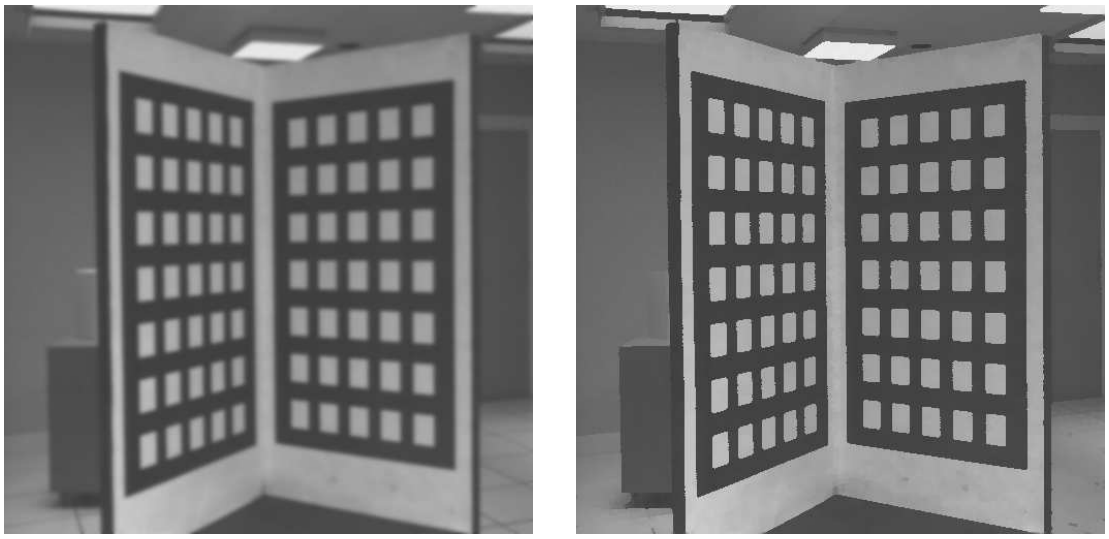
La précision de cette technique, qui utilise les images de contours, est d'autant plus mauvaise que l'image originale est floue, que les angles des carrés sont arrondis, ou que les bords ne sont pas très droits. Or, ces désagréments se rencontrent fréquemment sur des images réelles, et des erreurs importantes peuvent apparaître sur l'estimation des paramètres [Tar91].

Un filtre permettant de réhausser les bords flous en les restituant bien nets et droits, de ne pas arrondir les angles plus qu'ils ne le sont et même de les restaurer, et d'éliminer le bruit impulsionnel présent sur les bords (qui modifie les contours) convient alors parfaitement. En effet, les contours extraits sont alors de meilleure qualité (voir figures 7 et 8), ce qui permet d'extraire les attributs intéressants (coins) de façon plus précise, et donc d'améliorer la qualité de la calibration. Ainsi par exemple, on peut utiliser les filtres AFUMEP et ASAD sans problème, EGPR et SNN si les angles demeurent assez proches des angles droits malgré la projection (ce qui est souvent le cas), ASMT si le flou et le bruit de quantification ne sont pas trop importants.

6.2.2 Calibration avec une mire à disques

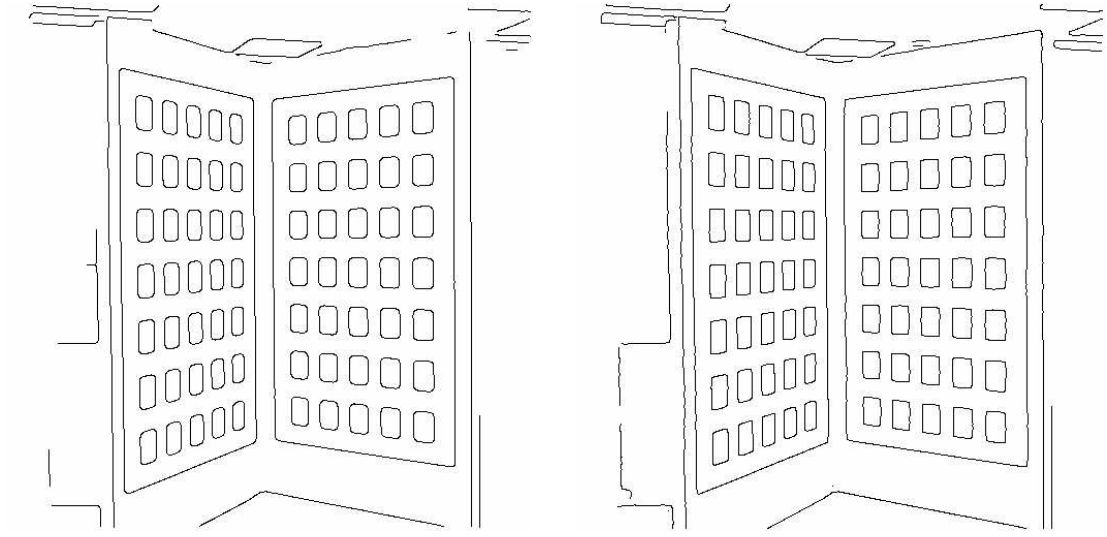
Même dans les meilleures conditions d'utilisation, la technique de calibration précédente, basée sur l'utilisation d'une mire à carrés, est d'une précision limitée sur la localisation des points caractéristiques, et l'estimation des paramètres peut être entachée d'erreurs importantes, même à partir d'une paire d'images synthétiques de mire [Tar91].

Dans le but d'obtenir des résultats de calibration plus précis, une autre approche a été explorée. On sait que la qualité des résultats dépend directement de la précision obtenue sur les attributs extraits. Les moments géométriques de l'image d'une forme plane sur un fond uniforme peuvent être calculés avec une grande précision (subpixel), notamment le barycentre. L'idée est alors de les utiliser comme attributs dans la calibration. Les formes elliptiques paraissant plus particulièrement intéressantes, du fait des propriétés qu'ont les



(a) (b)

FIG. 7 - *Mire à carrés. (a): image originale (b): image filtrée avec EGPR*



(a) (b)

FIG. 8 - *Extraction des contours. (a): sur l'image originale (b): sur l'image filtrée avec EGPR*

coniques quand elles subissent une transformation perspective, le choix s'est orienté vers l'utilisation d'une mire biplanaire à la surface de laquelle sont dessinés des disques pleins et non plus des carrés.

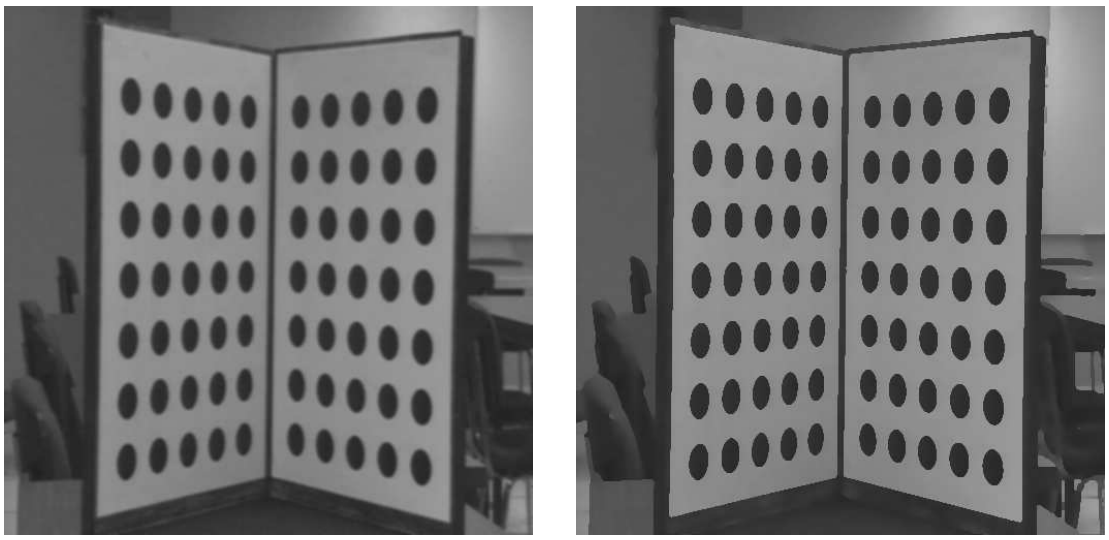
Cette technique est due à J.-P. Tarel qui l'a développée au sein du projet Syntim [Tar94]. Tout comme dans la technique précédente, les paramètres sont estimés par une méthode d'optimisation visant à minimiser la différence de localisation entre l'ensemble des images des points d'intérêt (attributs extraits) et la projection du modèle tridimensionnel de la mire. Mais ici, les attributs sont les barycentres des disques de la mire. Le problème qui se pose est d'extraire les attributs avec la plus grande précision possible.

Une première méthode consiste à calculer les barycentres à partir des contours. Elle offre des résultats assez similaires à ceux de la première technique de calibration, mais a l'avantage d'être plus simple. Lorsque le bruit de quantification est très présent sur l'image, un filtrage EPSF permet une légère amélioration en offrant une meilleure image des contours, mais en général, cela revient au même.

Une seconde méthode donnant de très bons résultats a été développée. Dans le voisinage de chacune des ellipses (projections des disques de la mire sur la paire d'images stéréo), il est possible de distinguer trois zones, à savoir: une zone dont on est sûr qu'il s'agisse du disque, une zone dont on est sûr qu'il s'agisse du fond, et une autre, transitoire, qui est le flou entre les deux premières zones. Cette méthode consiste, pour chaque ellipse, à corriger les gradients d'intensité en normalisant l'image entre 0 et 1 pour chaque pixel du voisinage décrit, et de calculer ainsi le barycentre par une moyenne pondérée, faisant intervenir aussi les pixels de la zone de floue [Tar94].

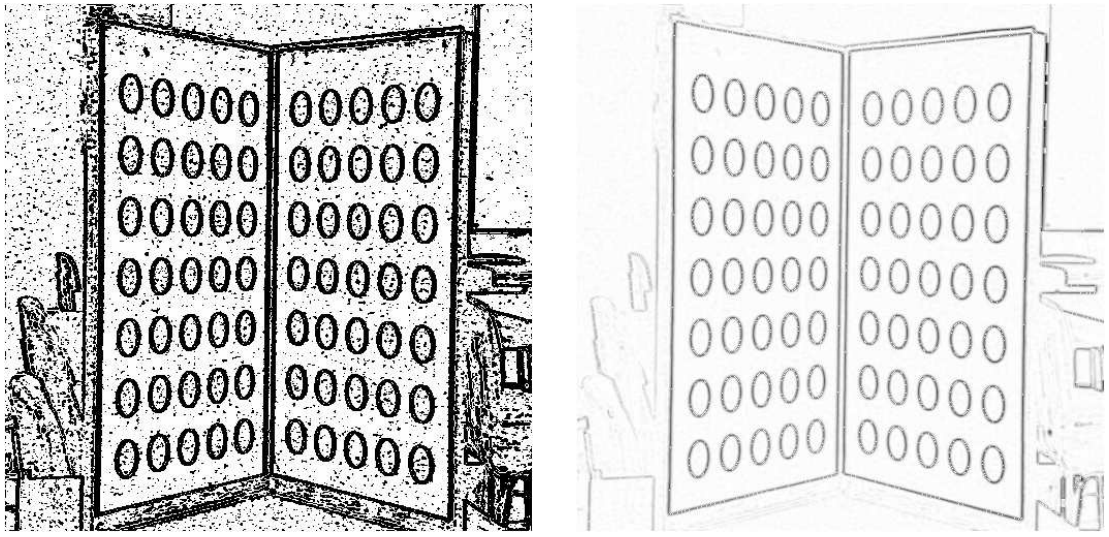
Cette dernière technique, qui donne de bons résultats en utilisant l'information floue, perdrait sa particularité et son efficacité, si un filtrage réhaussant les bords flous était effectué au préalable. Toutefois, l'utilisation du filtrage EPSF en tant qu'outil donnant des informations supplémentaires semble être une perspective intéressante à plusieurs points de vue.

Tout d'abord, la délimitation des trois zones prédéfinies pour chacune des ellipses, est actuellement faite de façon empirique alors qu'on pourrait la rendre automatique et adaptée à chaque cas. En effet, en utilisant un filtre tel que ASAD, qui réhausse très bien les bords flous tout en respectant fidèlement la continuité des frontières courbes, si on regarde les pixels qui ont changés de valeur entre l'image originale et l'image filtrée, on dispose directement des zones de flou autour des ellipses (voir figures 9 et 10(a)).



(a) (b)

FIG. 9 - *Mire à disques. (a): image originale (b): image filtrée avec ASAD*



(a) (b)

FIG. 10 - *Différence entre image originale et image filtrée . (a): image des pixels dont l'intensité a changé de plus de 3 unités (b): valeur absolue de la différence normalisée sur [0..255] (en inverse vidéo)*

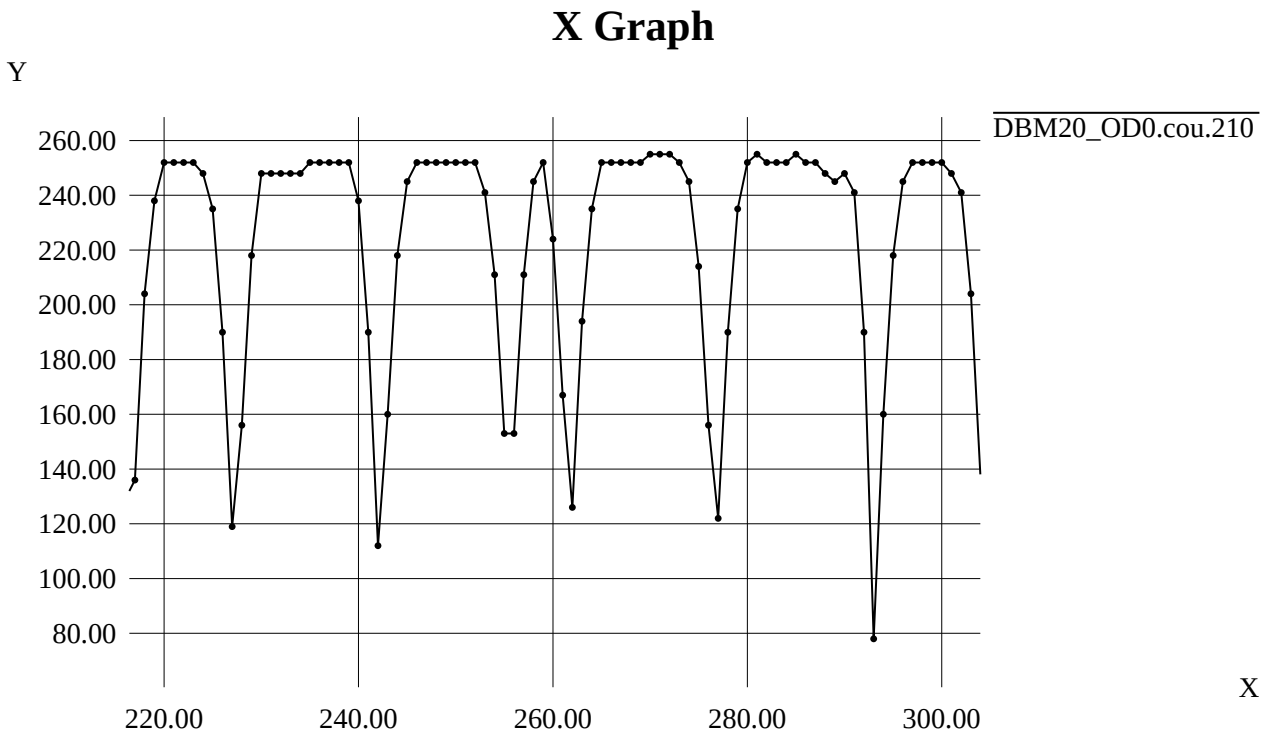


FIG. 11 - *Partie de la coupe de l'image de différence de la figure 10(b) à la ligne 210*

Par ailleurs, en faisant la différence (en valeur absolue) entre l'image originale et l'image filtrée avec ASAD, on dispose pour chaque pixel du voisinage considéré d'une ellipse donnée, de la distance en intensité, qui le sépare du bord presque idéalement numérisé (la véritable frontière étant vraisemblablement contenue dans un pixel) (voir figures 10(b) et 11). Ceci laisse entrevoir la possibilité de pouvoir améliorer encore cette technique de calibration, notamment en la généralisant au cas du flou quelconque, alors qu'elle fonctionne actuellement sous l'hypothèse d'un flou parfaitement symétrique.

6.3 Intérêt pour la segmentation

L'opération de segmentation a pour but de décomposer et partitionner l'image en différentes régions. Cette tâche se retrouve dans tous les domaines d'application du traitement d'image, et de très nombreuses méthodes existent, souvent très liées aux applications en question. Notamment, au sein du projet Syntim, plusieurs méthodes ont été utilisées, parmi lesquelles certaines y ont même vu le jour (*Split and Merge*, segmentation descentante coopérative de paires d'images stéréoscopiques, segmentation par minimisation d'une énergie, segmentation floue) [Vin91, Ran92, Ack92, Bou94].

Dans le cadre du processus de reconstruction/reconnaissance de scènes 3D, l'étape de segmentation précède généralement celle de la mise en correspondance (*matching*) qui consiste à apparier les régions segmentées sur chacune des deux images stéréo [Gag92], ou sont menées de façon coopérarive [Ran92]. Ensuite, la reconstruction 3D s'appuie sur les régions (facettes dans le cas d'un monde supposé polyédrique) segmentées et *matchées* pour déterminer (reconstruire) la région 3D dont les deux régions sont les projections sur les plans caméras [Véz91]. De petites erreurs de discrétisation sur les régions peuvent être assez graves de conséquence au niveau de la reconstruction 3D, et ne peuvent donc pas être négligées, même sur des images de synthèse [Fel92, FEl92]. De ce fait, la qualité de la segmentation est d'une importance capitale, et il est nécessaire qu'elle soit la plus précise possible.

Ainsi, le bruit présent dans l'image, et notamment les zones de flou qui créent souvent des ambiguïtés d'appartenance, sont des sources d'ennuis et d'erreurs, que le filtrage EPSF peut atténuer, voire éliminer.

La figure 12 montre le résultat de la segmentation des images de mire à disques de la figure 9 (originale et filtrée). On voit notamment très bien sur la figure 12(a) que les zones de flou autour des disques sont considérés comme des régions. Sur la figure 12(b) en revanche, les bords flous ayant été réhaussés par ASAD, les régions segmentées sont correctes. Les figures 13 et

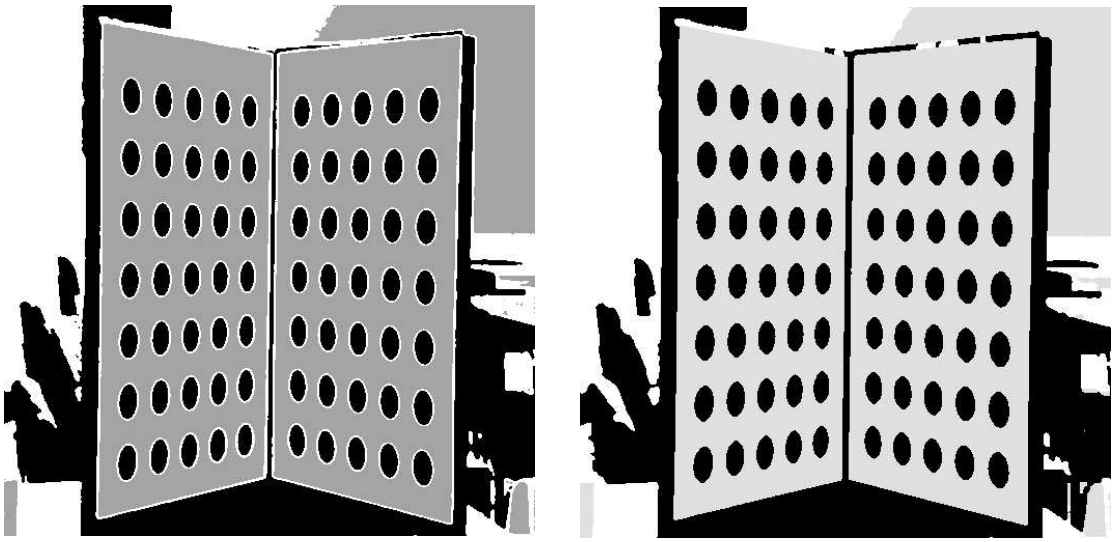


FIG. 12 - *Segmentation des images de la figure 9. (a): image originale (b): image filtrée*

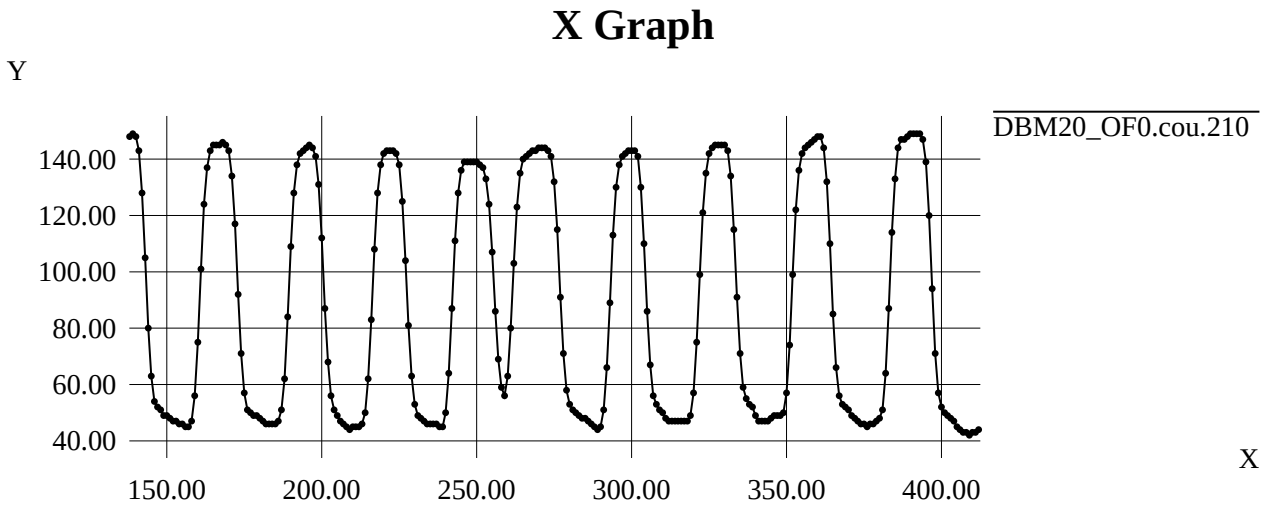


FIG. 13 - *Partie de la coupe de l'image originale présentée à la figure 9(a) à la ligne 210.*

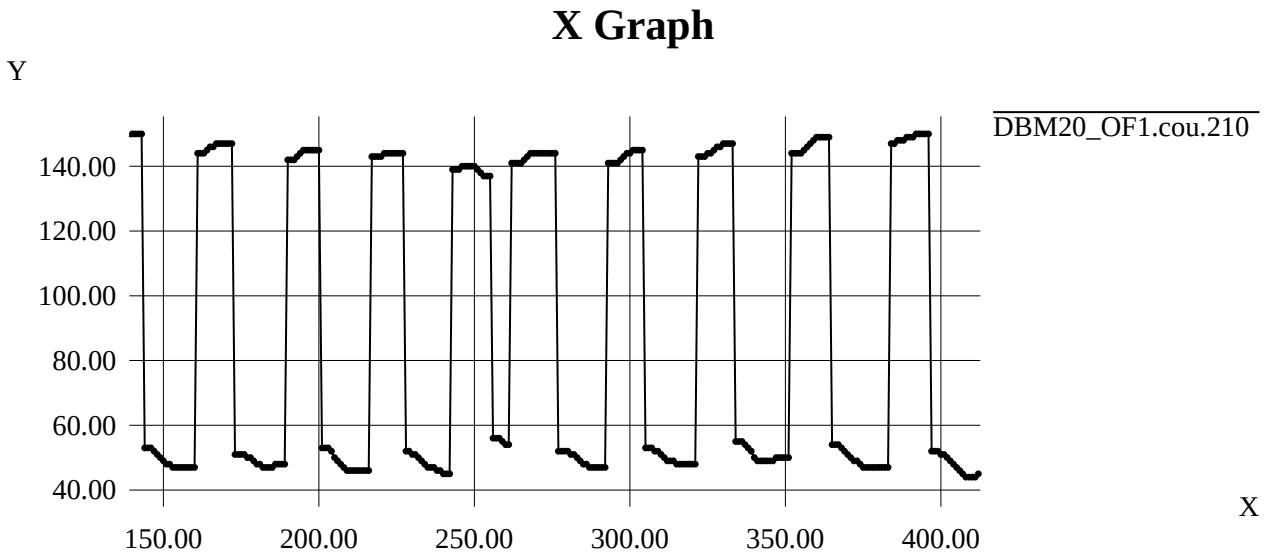


FIG. 14 - *Partie de la coupe de l'image filtrée présentée à la figure 9(b) à la ligne 210.*

14 illustrent le réhaussement du flou sur un extrait de la coupe à la ligne 210 des images de la figure 9.

6.4 Intérêt pour l'analyse photométrique

L'objet de l'analyse photométrique est d'avoir une meilleure compréhension de la scène. En complément des aspects géométriques 3D, l'analyse doit estimer les paramètres photométriques des objets présents dans l'environnement (réflectance, couleur...). Deux phénomènes physiques contribuent de façon conjointe à la formation de la luminance des images: d'une part, les propriétés de réflectance des objets, et d'autre part, les caractéristiques énergétiques et géométriques des sources qui les illuminent. Ainsi, la connaissance des sources lumineuses doit être considérée comme partie intégrante de la segmentation d'images et, plus généralement, de l'analyse de scènes.

La fonction d'intensité mesurée sur chacune des régions dépend des propriétés photométriques de la facette 3D dont est issue chaque région et de celles des sources lumineuses participant (directement ou pas) à l'éclairement de cette facette. L'analyse photométrique consiste alors à modéliser ces phénomènes radiométriques et à estimer les paramètres des modèles d'éclairement et de réflexion choisis, c'est-à-dire déterminer la position des sources

lumineuses de la scène et leur valeur d'émittance, et déterminer les valeurs de réflectance pour chaque face des polyèdres (objet) vus sur la paire d'images stéréo.

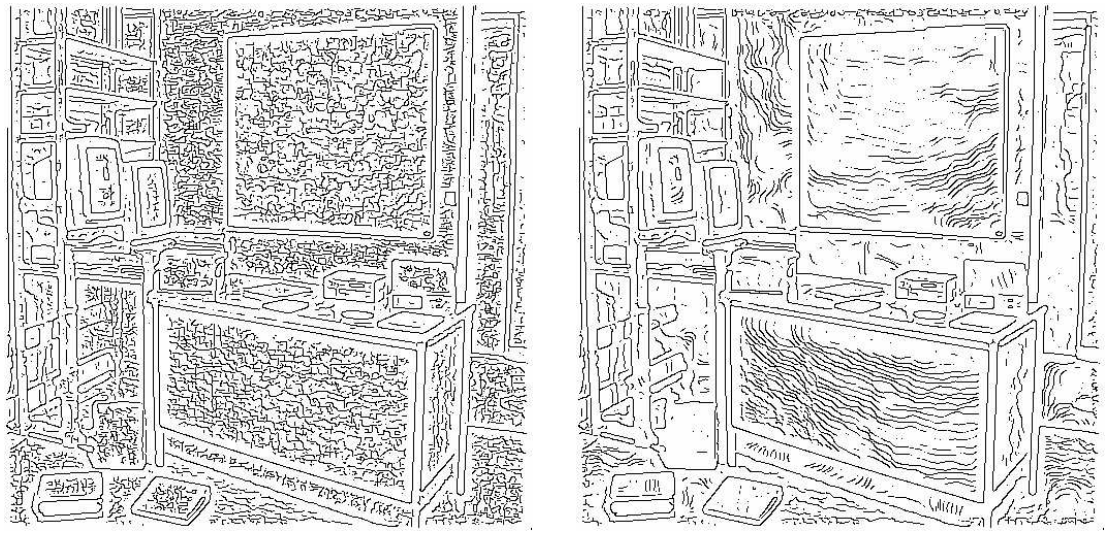
L'étude photométrique passe par l'analyse des gradients d'irradiance [Ser92, SEr92], qui est fortement perturbée par l'importance du bruit de quantification. Pour résoudre ce problème, il faut utiliser un filtre qui puisse éliminer le bruit de quantification, qui n'introduise pas de flou, et surtout qui lisse de manière très continue pour respecter les gradients et les surfaces d'isoluminances (éviter l'écueil des mosaïques et/ou des grandes marches). Parmi tous les filtres implantés et testés, seul AFUMEP répond à ces conditions, et il offre des résultats excellents et même inespérés, comme l'illustrent les figures qui suivent.



(a) (b)

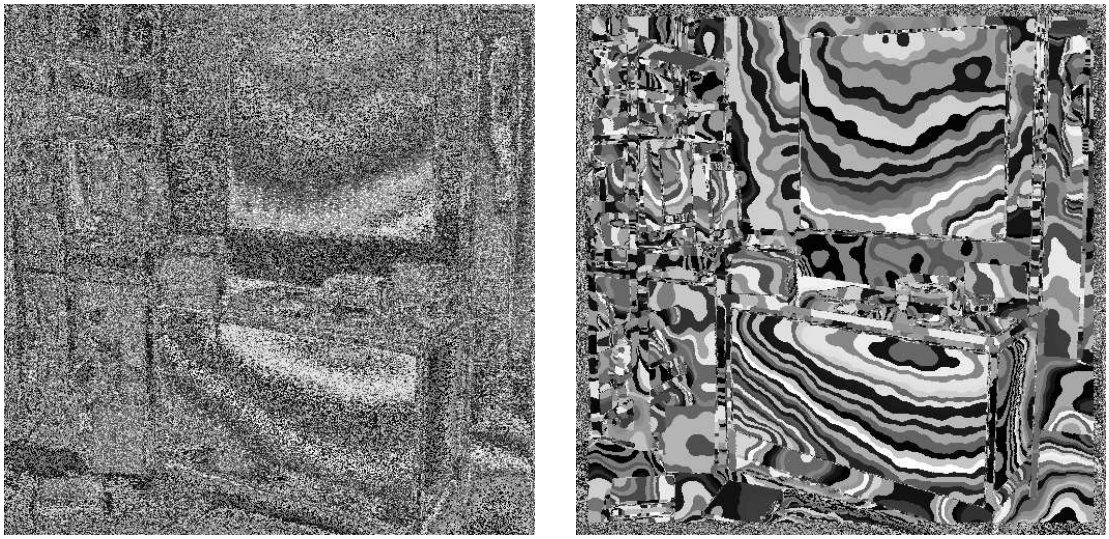
FIG. 15 - *Scène de bureau. (a): image originale (b): image filtrée avec AFUMEP*

La figure 15 représente l'image originale et l'image filtrée d'une scène de bureau. La figure 16, qui représente les images de contours des images de la figure 15, permet de voir à quel point les gradients (avant seuillage) de l'image originale sont inexploitables car très bruités, alors que ceux de l'image filtrée sont bien restaurés et exploitables (très visible sur les grandes facettes). Enfin, la figure 17 correspond aux deux images présentées à la figure 15, visualisées à l'aide de fausses couleurs, ce qui permet d'illustrer parfaitement ce propos. Sur l'image filtrée, notamment au niveau du tableau et de la face avant du



(a) (b)

FIG. 16 - *Extraction des contours des images de la figure 15 (a): image originale (b): image filtrée*



(a) (b)

FIG. 17 - *Visualisation en fausses couleurs des images de la figure 15. (a): image originale (b): image filtrée*

bureau, on voit très nettement les différentes régions d'isoluminance, et leur forme caractéristique.

7 Conclusion

Dans ce rapport, un certain nombre de filtres se réclamant de la famille EPSF (*Edge Preserving Smoothing Filtering*) sont présentés. Ils ont été implantés et testés sur différents types d'images, afin d'observer leur comportement, de dégager leurs points forts et points faibles, pour ensuite être en mesure d'adapter plus aisément le filtrage à une application précise.

Cette étude a permis de mettre en évidence les avantages et inconvénients de chacun d'eux, au regard de différents critères. Une telle évaluation rend plus facile le choix de tel(s) ou tel(s) filtre(s) pour un besoin bien défini.

Plus particulièrement, dans le cadre de notre application, à savoir la reconstruction/reconnaissance de scènes 3D, nous avons montré l'intérêt du filtrage EPSF pour l'amélioration d'étapes essentielles telles que la calibration, la segmentation, et surtout l'analyse photométrique.

Références

[Ack92] **A. Ackah-Miezan, A. Gagalowicz,** *Energy Minimizing Segmentation of an Image.* Proc. of International Symposium on Computer and Information Sciences ISCIS VII, pp 631-634, Antalya, Turkey, 2-4 Nov., 1992

[ANd72] **D. F. Andrew and al.,** *Robust Estimation of Location.* Princeton, NJ: Princeton University Press, 1972

[And72] **H. C. Andrew,** *N topics in search of an editorial: Heuristics, superresolution, and bibliography.* Proc. IEEE 06, pp 891-894, 1972

[And74] **H. C. Andrew,** *Digital Image Restoration: A survey.* Computer, 8, pp 36-45, 1974

[And77] **H. C. Andrew, B. R. Hunt,** *Digital Image Restoration.* Prentice-Hall, Englewood Cliffs, N. J., 1977

[Asa81] **T. Asano, N. Yokoya,** *Image segmentation scheme for low level computer vision.* Pattern Recognition, 1981

[Bol92] **P. Bolon,** *Filtrage d'ordre, vraissemblance et optimalité des prétraitements d'image.* Traitement du Signal, vol. 9, No 3, pp 225-250, 1992

[Bou94] **N. Boujemaa, G. Stamon, A. Gagalowicz,** *Modélisation floue pour la segmentation d'images.* Congrès AFCET-RFIA, Paris, France, 11-14 jan., 1994

[Bov83] **A. C. Bovik, T. S. Huang, D. C. Munson,** *A generalization of median filtering using linear combinations of order statistics.* IEEE Trans. Acoust.,Speech, Signal Processing, vol. ASSP-31, pp 1342-1350, 1983

[Chi83] **R. T. Chin, C. L. Yeh,** *Quantitative Evaluation of Some Edge-Preserving Noise-Smoothing Techniques.* CVGIP 23, pp 67-91, 1983

[Dav78] **L. S. Davis, A. Rosenfeld,** *Noise cleaning by iterated local averaging.* IEEE Trans. Syst. Man. Cybern., vol. SMC-8, pp 705-710, 1978

[Fel92] **J. Feldmar, G. Jourdan,** *Vision stéréoscopique: reconstruction géométrique et reconnaissance de modèles.* Rapport de DEA, INRIA, 1992

[FEl92] **J. Feldmar, G. Jourdan,** *Reconstruction géométrique et reconnaissance de modèles.* Rapport d'activité INRIA - programme 4 - projet Syntim, 1992

[Gag92] **A. Gagalowicz,** *Collaboration Between Computer Graphics and Computer Vision.* Progress in Image Analysis and Processing II, edited by World Publishing, pp 229-258, 1992

[Gan89] **P. P. Gandhi, I. S. Song, S. A. Kassam,** *Nonlinear Smoothing Filters Based on Rank Estimates of Location.* IEEE Trans. Acoust.,Speech, Signal Processing, vol. ASSP-37, No 9, pp 1359-1379, 1989

[Har87] **D. Harwood, M. Subbarao, H. Hakalahti, L. Davis,** *A new class of edge-preserving smoothing filters.* Pattern Recognition Letters, 6, pp 155-162, 1987

[Hei85] **P. Heinonen, Y. Neuvo,** *Smoothed median filters with FIR substructures.* Proc. IEEE ICASSP, Tampa, FL, pp 49-52, 1985

[Hei87] **P. Heinenen, Y. Neuvo,** *FIR median hybrid filters.* IEEE Trans. Acoust.,Speech, Signal Processing, vol. ASSP-35, pp 832-838, 1987

[Hua71] **T. S. Haung, W. F. Schreiber, D. J. Tretiak,** *Image Processing.* Proc. IEEE 59, pp 1586-1608, 1971

[Hua81] **T. S. Huang,** *Two-Dimensionnal Digital Image Processing II: Transforms and Median Filters.* New York: Springer-Verlag, pp 3-6, 1981

[Hub81] **P. Huber,** *Robust Statistics.* New York: Wiley, 1981

[Imm91] **M. Imme,** *A Noise Peak Elimination Filter.* CVGIP: Graphic, Models and Image Processing, vol. 53, No. 2, pp 204-211, 1991

[Jay76] **N. S. Jayant,** *Average and median-based smoothing techniques for improving speech quality in the presence of transmission errors.* IEEE Trans. Commun., vol. COM-24, pp 1043-1045, 1976

[Lee85] **Y. H. Lee, S. A. Kassam,** *Generalized Median Filtering and Related Nonlinear Filtering Techniques.* IEEE Trans. Acoust.,Speech, Signal Processing, vol. ASSP-33, No 3, 1985

[Lev77] **A. Lev, S. W. Zucker, A. Rosenfeld,** *Iterative Enhancement of Noisy Images.* IEEE Trans. Syst. Man. Cybern., vol. SMC-7, pp 435-442, 1977

[Mar87] **P. A. Maragos, R. W. Shafer,** *Morphological filters - Part II.* IEEE Trans. Acoust.,Speech, Signal Processing, vol. ASSP-35, 1987

[Mas85] **G. A. Mastin,** *Adaptative Filters for Digital Image Noise Smoothing: An Evaluation.* CVGIP, vol. 31, pp 103-121, 1985

[Mal79] **C. L. Mallows,** *Some theoreical results on Tukey's 3R smoother.* Proc. Heidelberg WorkShop Smoothing Tech. Curve Est., pp 77-90, 1979

[Mal80] **C. L. Mallows,** *Some theory of nonlinear smoothers.* Ann. Stat., vol. 8, pp 695-715, 1980

[Nag79] **M. Nagao, T. Matsuyama,** *Edge Preserving Smoothing.* Computer Graphics and Image Processing, 9, pp 394-407, 1979

[Per90] **P. Perona, J. Malik,** *Scale-Space and Edge Detection Using Anisotropic Diffusion.* IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-12, No 7, pp 629-639, 1990

[Pra78] **W. K. Pratt,** *Digital Image Processing.* New York: Wiley, 1978

[Rab75] **L. R. Rabiner, M. R. Sambur,C. E. Schmidt,** *Applications of a nonlinear smoothing algorithm to speech processing.* IEEE Trans. Acoust.,Speech, Signal Processing, vol. ASSP-23, pp 552-557, 1975

[Ran92] **S. Randriamasy,** *Segmentation descendante coopérative en régions de paires d'images stéréoscopiques.* Thèse de Doctorat, Université Paris 9 - Dauphine, Sept., 1992

[Ros82] **A. Rosenfeld, A. C. Kak,** *Digital Picture Processing.* vol. 1, 2nd ed., New York: Academic, 1982

[Sai91] **P. Saint-Marc, J. S. Chen, G. Médioni,** *Adaptative Smoothing: A General Tool for Early Vision.* IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-13, N. 6, pp 514-529, 1991

[Sho92] **R. P. H. M. Shoenmakers, J. H. T. Stakenborg,** *Adaptative Filtering Using Morphological Filtering.* Proc. 11th IAPR (Int. Conf. Patt. Recogn), pp 240-243, 1992

[Ser92] **V. Serfaty, A. Ackah-Miezan, E. Lutton, A. Gagalowicz,** *Towards a "Visual" Model for Robot Vision: From Wire-Frame to Photometric Representation.* Proc. of IMAGE'COM93, 23-25 mars, Bordeaux, France, pp 255-260, 1992

[SEr92] **V. Serfaty, A. Ackah-Miezan, E. Lutton,** *Influence de l'éclairage en analyse photométrique.* Rapport d'activité INRIA - programme 4 - projet Syntim, 1992

[Stu79] **W. Stuetzle, Y. Mittal,** *Some comments on the asymptotic behavior of robust smoothers.* Robustness in Statistics, R. L. Launer and G. Wilkinson Eds., New York: Academic, pp 191-195, 1979

[Tar91] **J.-P. Tarel,** *Calibration de caméras.* Rapport d'activité INRIA - programme 4 - projet Syntim, 1991

[Tar94] **J.-P. Tarel, A. Gagalowicz,** *Calibration de caméra à base d'ellipses.* Congrès AFCET-RFIA, Paris, France, 11-14 jan., 1994

[Tos87] **G. Toscani,** *Systèmes de calibration et perception du mouvement en vision artificielle.* Ph.D Thesis, Université Paris-Sud, 1987

[Tuk71] **J. W. Tukey,** *Exploratory Data Analysis.* Addison-Wesley, Readin. Mass., 1971

[Tuk74] **J. W. Tukey,** *Nonlinear (nonsuperposable) methods for smoothing data.* Conf. EASCON, p. 673, 1974

[Véz91] **J.-M. Vézien, A. Gagalowicz,** *Reconstruction 3D basée sur une analyse en régions d'une paire d'images stéréoscopiques.* Congrès AFCET-RFIA, Lyon, France, 25-29 nov., 1991

[Vin91] **L. Vinet, A. Gagalowicz,** *Mise en correspondance de régions utilisant la cohérence de leurs voisinages.* Congrès AFCET-RFIA, Lyon, France, 25-29 nov., 1991

[Wan81] **D. Wang, A. Vanucci, C. Li,** *Image enhancement by gradient inverse weighted smoothing scheme.* Computer Graphics and Image Processing, 15, pp 167-181, 1981

[Wan90] **X. Wang, D. Wang,** *on the max/median filters.* IEEE Trans. Acoust.,Speech, Signal Processing, vol. ASSP-38, pp 1473-1475, 1990

[Won93] **Y. F. Wong,** *Clustering Data by Melting.* Neural Computation, 5, pp 89-104, 1993

[Wu92] **W. Y. Wu, M. J. Wang, C. M. Liu,** *Performane Evaluation of Some Noise Reduction Methods.* CVGIP: Graphical Models and Image Processing, vol. 54, No. 2, pp 134-146, 1992



Unité de recherche INRIA Lorraine, Technôpole de Nancy-Brabois, Campus scientifique,
615 rue de Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399