



HAL
open science

Mesure de similarité par blocs via les normes triangulaires et l'intégrale de Sugeno - Application à la détection de contours

Hoel Le Capitaine, Thomas Batard, Carl Frelicot, Michel Berthier

► To cite this version:

Hoel Le Capitaine, Thomas Batard, Carl Frelicot, Michel Berthier. Mesure de similarité par blocs via les normes triangulaires et l'intégrale de Sugeno - Application à la détection de contours. Rencontres Francophone sur la Logique Floue et ses Applications, LFA, 2007, Nîmes, France. pp.49-56. hal-00292614

HAL Id: hal-00292614

<https://hal.science/hal-00292614v1>

Submitted on 15 May 2009

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.

Mesure de similarité par blocs via les normes triangulaires et l'intégrale de Sugeno - Application à la détection de contours

Blockwise similarity via triangular norms and Sugeno integrals - Application to edge detection

Hoel Le Capitaine*

Thomas Batard*

Carl Frélicot*

Michel Berthier*

* Laboratoire Mathématiques, Images et Applications (MIA)

Université de La Rochelle Avenue M. Crépeau, 17042 La Rochelle Cedex1, prénom.nom@univ-lr.fr

Résumé :

Dans de nombreux domaines, comme par exemple l'aide à la décision, des valeurs numériques dans $[0, 1]$ sont disponibles et il peut être intéressant de détecter celles qui sont similaires. Dans cet article, nous proposons un opérateur d'agrégation adapté à ce problème, capable de détecter des blocs de valeurs similaires. Il combine les valeurs à agréger et une fonction noyau à l'aide de normes triangulaires et d'intégrales de Sugeno. Sa valeur permet d'estimer la similarité par blocs de valeurs, à différents niveaux. A des fins illustratives, nous appliquons l'opérateur à la détection de contours en traitement d'images. La similarité est alors utilisée comme une mesure d'homogénéité locale d'intensité. Les liens existants avec la morphologie mathématique floue sont soulignés.

Mots-clés :

Similarité, Normes triangulaires, Intégrales floues, Morphologie mathématique floue, Détection de contours.

Abstract:

In many fields, e.g. decision-making, numerical values in $[0, 1]$ are available and one is often interested in detecting which are similar. In this paper, we propose an aggregation operator suitable for this problem, which is able to detect blockwise similar values. It combines the values to be aggregated and a kernel function using triangular norms and Sugeno integrals. Its output value allows to estimate blockwise similarity of values, at different levels. For illustration purpose, we apply the operator to edge detection in the field of image processing. Similarity is then used as a local homogeneity measure of intensity. Links with fuzzy mathematical morphology are underlined.

Keywords:

Similarity, Triangular norms, Fuzzy integrals, Fuzzy mathematical morphology, Edge detection.

1 Introduction

Le cadre de cet article est la mesure de similarité par blocs de valeurs d'un c -uplet à valeurs dans $[0, 1]^c$. Nous proposons un nouvel

opérateur d'agrégation, fondé sur l'utilisation de normes triangulaires [7] et d'intégrales de Sugeno [11], combinant les c valeurs et des fonctions noyau. L'utilisation de noyaux permet, via leur paramètre de résolution, de mesurer la similarité à différents niveaux.

De nombreuses applications sont possibles, citons en deux. La reconnaissance des formes tout d'abord, où les valeurs à agréger représentent le degré d'appartenance d'une forme à une classe donnée (mode supervisé) ou sa contribution à la définition d'un groupe (mode non supervisé). L'opérateur, capable de détecter des similarités, peut permettre de rejeter en ambiguïté une forme à classer ou de mesurer la tendance de plusieurs groupes à n'en constituer qu'un. La détection de contours en traitement d'images ensuite, où les valeurs à agréger sont, après fuzzification, les niveaux d'intensité au voisinage d'un pixel. L'opérateur peut alors permettre de détecter une zone homogène et ainsi infirmer ou non la présence d'un point de contour.

Le reste de l'article est composé de quatre parties. En section 2, nous rappelons brièvement les différents travaux [3, 9] qui nous ont amenés à considérer des fonctions d'agrégation d'un nouveau genre. Ensuite, à la section 3, nous présentons l'opérateur de similarité par blocs. Sa définition, ses propriétés mathématiques y sont développées et des exemples numériques montrant son fonctionnement sont donnés. En-

fin, la dernière partie (section 4) est consacrée à l'application de l'opérateur proposé à la détection de contours en traitement d'images. Les liens entre ce nouvel opérateur et ceux issus de la morphologie mathématique floue [10, 5, 1] y sont soulignés.

2 Travaux antérieurs

Pour les applications que nous visons (voir section 1), nous nous sommes particulièrement intéressés aux fonctions – ou opérateurs – d'agrégation de la forme $\Phi : [0, 1]^c \rightarrow [0, 1]$, $u = \{u_1, u_2, \dots, u_c\} \mapsto \Phi(u)$. Parmi celles-ci, citons les normes triangulaires (ou t-normes) et leur forme duale, les conormes triangulaires (ou t-conormes). Le Tableau 1 présente les quelques normes dites classiques; ce sont celles que nous utiliserons dans les exemples numériques des sections suivantes. Pour une étude très complète, le lecteur intéressé peut se référer à [7]. Par pure convenance d'écriture, nous supposons dorénavant que le c -uplet u est ordonné par valeurs décroissantes, c'est-à-dire que $u_1 \geq u_2 \geq \dots \geq u_c$.

Tableau 1 – Exemples de t-normes et t-conormes basiques

Standard	$a \top_S b = \min(a, b)$
	$a \perp_S b = \max(a, b)$
Algébrique	$a \top_A b = ab$
	$a \perp_A b = a + b - ab$
Lukasiewicz	$a \top_L b = \max(a + b - 1, 0)$
	$a \perp_L b = \min(a + b, 1)$

Dans [9], nous avons défini l'opérateur \perp^k , baptisé *OU flou d'ordre k* , comme suit. Soient \mathcal{P} l'ensemble des parties de $C = \{1, 2, \dots, c\}$ et $\mathcal{P}_k = \{A \in \mathcal{P} : |A| = k\}$, où $|A|$ représente le cardinal du sous-ensemble A , alors

$$\perp_{i=1,c}^k u_i = \bigtop_{A \in \mathcal{P}_{k-1}} \left(\perp_{j \in C \setminus A} u_j \right) \quad (1)$$

où \top est une t-norme, et \perp sa t-conorme duale.

Il généralise la notion de “ $k^{\text{ème}}$ plus grand”, avec k dans C . En particulier, lorsque les normes standard sont utilisées, $\perp = u_k$. Pour la démonstration de cette propriété et d'autres que cet opérateur vérifie, ainsi que pour des applications, le lecteur intéressé peut se référer à [9]. Par combinaison de \perp^k , il est possible de construire un *OU exclusif flou* qui étend le OU exclusif binaire [3] :

$$\perp_{i=1,c} u_i = \left(\perp_{i=1,c}^1 u_i \right) \top \left(\perp_{i=1,c}^2 u_i / \perp_{i=1,c}^1 u_i \right) \quad (2)$$

où $\perp^1 = \perp$ et \top représente un complément flou, par exemple $\bar{a} = 1 - a$. Le terme à droite de \top pénalise le terme à gauche. Il affaiblit donc le \perp^1 , sauf si \perp^2 est suffisamment faible devant \perp^1 , autrement dit sauf si le 1^{er} plus grand a une valeur significativement plus élevée que le 2^{ème} plus grand et par conséquent que les autres valeurs de u .

Cette propriété nous a conduits à imaginer une généralisation à d'autres ordres que 1 et 2 le quotient complémenté dans (2) afin de matérialiser le concept de similarité entre valeurs de u . Cette similarité par paires $(j, k) \in C \times C$, ou plutôt par blocs de valeurs entre u_j et u_k est l'objet des travaux présentés ici.

3 Similarité par blocs

3.1 Définition

Un opérateur de similarité par blocs est, par définition, une famille de fonctions $\Phi_{j,k} : [0, 1]^c \rightarrow [0, 1]$, $(j, k) \in C \times C$, $j < k$, qui satisfait les quatre propriétés suivantes :

(P1) $\Phi_{j,k}(u) = 0$ lorsque $u_j = 1$ et $u_k = 0$

(P2) $\Phi_{j,k} = 1 \Leftrightarrow u_j = u_k$

(P3) $\forall 0 \leq \varepsilon \leq u_{j-1} - u_j$,
 $\Phi_{j,k}(u_1, \dots, u_j + \varepsilon, \dots, u_c) \leq$
 $\Phi_{j,k}(u_1, \dots, u_j, \dots, u_c)$

(P4) $\forall 0 \leq \varepsilon \leq u_{k-1} - u_k$,
 $\Phi_{j,k}(u_1, \dots, u_k + \varepsilon, \dots, u_c) \geq$
 $\Phi_{j,k}(u_1, \dots, u_k, \dots, u_c)$

Ainsi, avec (P1), la similarité est minimale lorsque la différence aux bornes de l'intervalle considérée est maximale. Inversement, la similarité est maximale lorsque la différence est minimale avec (P2). Lorsque l'on augmente la plus grande des valeurs du bloc, la valeur de similarité de ce bloc diminue, (P3), et enfin, la similarité augmente si l'on augmente la plus petite des valeurs du bloc avec (P4). La fonction $\Phi_{j,k}$ mesure la similarité des u_i appartenant au bloc délimité par les indices j et k . En particulier, $\Phi_{1,c}$ mesure la similarité totale de u . On déduit de la propriété 2 que si $u_1 = u_2 = \dots = u_c$, alors $\Phi_{j,k} = 1$.

La solution la plus simple, consistant à considérer les quotients $\Phi_{j,k} = \frac{\perp^k}{\perp^j}$, n'est pas satisfaisante dans la mesure où l'opérateur ainsi défini ne satisfait pas les propriétés requises.

3.2 Le nouvel opérateur

Dans le contexte décrit précédemment, l'intégrale de Sugeno (voir [11] pour des définitions complètes ainsi que des exemples) s'écrit sous la forme suivante :

$$\mathcal{S}_\mu(u) = \max_{i=1,c} [\min(u_i, \mu\{i, \dots, c\})]$$

μ étant une mesure floue sur C . Cette définition peut être généralisée en remplaçant (min, max) par un couple de t-norme et t-conorme (\top, \perp) quelconque. On obtient alors une intégrale floue, relativement à une mesure floue sur les ensembles $\{i, \dots, c\}$, qui permet de mesurer, de quantifier la similarité des u_i par rapport au jeu de t-norme et t-conorme choisi.

Si l'on note $A_i = \{j, u_j \geq u_i\}$ et que l'on considère la famille de mesures floues μ_k avec :

$$\mu_k(A_i) = \begin{cases} 0 & \text{si Card}(A_i) < k \\ 1 & \text{sinon} \end{cases} \quad (3)$$

en posant

$$\perp^k(u) = \perp_{i=1,\dots,c}^k u_i \top \mu_k(A_i) \quad (4)$$

on vérifie facilement que

$$\perp^k(u) = \begin{cases} \perp_{i=k,\dots,c}^k u_i & \text{si } u_{k-1} > u_k \\ \perp_{i=j,\dots,c}^k u_i & \text{sinon, } j \text{ étant défini par} \\ & u_{j-1} > u_j = \dots = u_k \end{cases} \quad (5)$$

en décomposant

$$\perp^k(u) = \left(\perp_{i=1,\dots,k-1}^k u_i \top \mu_k(A_i) \right) \perp \left(\perp_{i=k,\dots,c}^k u_i \top \mu_k(A_i) \right)$$

, avec la convention : $u_0 > u_1$; les opérateurs ci-dessus sont évidemment différents de ceux définis en 2, nous adoptons cependant la même notation pour ne pas alourdir le texte.

Les quotients $\Phi_{j,k} = \frac{\perp^k}{\perp^j}$ correspondant donnent naissance à un opérateur de similarité qui satisfait bien les propriétés P1 à P4 pour les jeux de t-normes et t-conormes standard et algébrique. Compte tenu du choix des μ_k , il apparaît clairement que ce procédé ne tient compte, dans la mesure de similarité entre u_j et u_k , que des seuls u_l pour $l \geq k+1$ (et pas des u_l pour $l < j$). Il est donc nécessaire, pour rendre les valeurs comprises entre u_j et u_k plus significatives, d'avoir recours à des fonctions noyaux symétriques centrées en l , que nous noterons $\mathcal{K}_\lambda(i, l)$ dans la suite.

Nous définissons finalement un opérateur de similarité par blocs à l'aide des fonctions :

$$\Phi_{j,k}(u) = \begin{cases} \frac{\perp_{i=\frac{k+j}{2}}^k u_i \top \mathcal{K}_\lambda(i,k)}{j} & \text{si } k-j \text{ est pair} \\ \frac{\perp_{i=\frac{k+j}{2}}^k u_i \top \mathcal{K}_\lambda(i,j)}{j} & \text{si } k-j \text{ est impair} \end{cases} \quad (6)$$

La symétrie est ici introduite afin de donner de moins en moins d'importance au $i^{\text{ème}}$ ensemble

flou au fur et à mesure que i s'éloigne de j et k lorsque $j \leq i \leq k$. De nombreuses fonctions noyaux peuvent être utilisées. Une des caractéristiques de ces fonctions est de faire intervenir un paramètre qui, dans notre contexte, va jouer le rôle de résolution en contrôlant la zone d'influence que l'on veut prendre en considération. Traitons, à titre d'exemple, le cas particulier du noyau gaussien $\mathcal{K}_\lambda(i, l) = \mathcal{N}_\lambda(i, l)$ défini par (voir Figure 1) :

$$\mathcal{N}_\lambda(i, l) = \exp \frac{-(i-l)^2}{\lambda} \quad (7)$$

Lorsque λ tend vers 0, ce noyau devient un di-

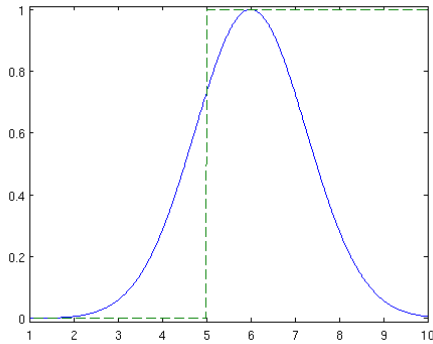


Figure 1 – Noyau $\mathcal{N}_{10}(i, l)$ et mesure cardinale avec $u_5 > u_6$ pour $l = 6$

rac δ_k centré en k , la convergence n'étant pas uniforme par continuité de \mathcal{K} et discontinuité de δ_k . Si les normes triangulaires utilisées sont continues, alors on a :

$$\lim_{\lambda \rightarrow 0} \Phi_{j,k}(u) = \begin{cases} 1 & \text{si } u_j = 0 \\ \frac{u_k}{u_j} & \text{sinon} \end{cases} \quad (8)$$

et

$$\lim_{\lambda \rightarrow +\infty} \Phi_{j,k}(u) = \begin{cases} \frac{\prod_{i=\frac{k+j}{2}}^k u_i}{\prod_{i=\frac{k+j}{2}}^j u_i} & \text{si } k-j \text{ est pair} \\ \frac{\prod_{i=\frac{k+j}{2}}^k u_i}{\prod_{i=\frac{k+j+1}{2}}^j u_i} & \text{si } k-j \text{ est impair} \\ 1 & \text{if } u_j = 0 \end{cases} \quad (9)$$

Ces résultats montrent comment il est possible d'ajuster les poids qui sont donnés aux valeurs intermédiaires entre u_j et u_k : la contribution de u_{j+1}, \dots, u_{k-1} est faible si λ est proche de zéro, et augmente avec λ . On notera cependant que $\Phi_{i-1,i} = u_{i-1}/u_i$ ne dépend pas de λ dans \mathbb{R}^+ . Cela veut dire qu'augmenter λ ne contribuera pas à rendre deux u_i consécutifs plus similaires, mais pourra par contre augmenter la similarité des blocs de taille plus importante. On remarquera également que si les u_i sont constants (quel que soit $i \in \{j, \dots, k\}$), alors $\Phi_{j,k}(u) = 1$ et ce quel que soit (j, k) (en accord avec la propriété 2).

3.3 Exemples numériques

Si l'on se donne un u , le calcul de $\Phi_{j,k}(u)$ pour tout $(i, j) \in C \times C$ donne lieu à un tableau symétrique, comme par exemple les tableaux 2 et 3 pour $u = \{0.9, 0.8, 0.68, 0.51, 0.5, 0.48, 0.32, 0.1\}$ avec des noyaux \mathcal{N}_1 et \mathcal{N}_5 , respectivement. Les valeurs de $\Phi_{j,k}(u)$ diminuent au fur et à mesure que k augmente, pour un j fixé. La détection de similarités consiste simplement à explorer la partie triangulaire supérieure du tableau et à comparer ces valeurs à un seuil s spécifié par l'utilisateur de la manière suivante :

- pour $j = 1, c$ (ligne)
 - pour $k = j + 1, c$ (colonne)
- si $\Phi_{j,k}(u) > s$, alors $\{u_j, \dots, u_k\}$

sont similaires. Afin de comprendre le comportement de l'opérateur en fonction de λ ,

comparons la 4^{ème} ligne de chaque tableau, c'est à dire $\Phi_{4,k}$ pour $k = 4, \dots, 8$, et supposons le seuil $s = 0.9$:

- pour une résolution de $\lambda = 1$, $\Phi_{4,4}$, $\Phi_{4,5}$ et $\Phi_{4,6}$ sont supérieurs à s et les u_k correspondants, $\{0.51, 0.5, 0.48\}$ sont détectés comme similaires. Les faibles valeurs de $\Phi_{4,7}$ et $\Phi_{4,8}$ indiquent que $u_7 = 0.32$ comme $u_8 = 0.1$ ne sont pas similaires aux u_k précédents.
- pour une résolution de $\lambda = 5$, $\Phi_{4,7}$ devient supérieur au seuil s et $u_7 = 0.32$ est maintenant considéré comme similaire à $\{0.51, 0.5, 0.48\}$, alors que $u_8 = 0.1$ ne l'est toujours pas.

4 Application à la détection de contours

4.1 Principe et procédure

La détection de contours est une étape-clé en traitement d'images qui consiste à extraire les zones de fort changement d'intensité dans une image. De nombreuses méthodes ont été proposées dans la littérature, du plus simple opérateur estimant la dérivée, l'opérateur de Sobel [4], jusqu'à l'estimateur de Canny [2]. En présence de bruit, ces méthodes sont plus ou moins efficaces et ce problème continue à faire l'objet de nombreux travaux de recherche.

Soit une image I dont l'intensité $int_I(p)$ de chaque pixel p a été ramenée dans $[0,1]$ (fuzzification). Considérons, pour chaque pixel, un voisinage $V(p)$ constitué des pixels p' situés dans une fenêtre de taille $c = x \times y$ incluant p , par exemple centrée en p . Les intensités $int_I(p')$ des pixels de $V(p)$ forment un c -uplet u de valeurs que nous pouvons passer en argument de l'opérateur de similarité par blocs $\Phi_{j,k}(u)$ défini par (6). Si $V(p)$ est relativement homogène, la valeur de $\Phi_{1,c}(u)$ sera élevée. Au contraire, si $V(p)$ contient un ou plusieurs points de contour matérialisés par des intensités peu similaires, alors $\Phi_{1,c}(u)$ aura une valeur faible. Selon le même principe, mais en considérant un bloc ($j > 1, k < c$), l'opérateur est alors à rapprocher d'un filtre médian pondéré permettant en

outre d'éliminer les valeurs d'intensité extrêmes résultant par exemple d'un bruit impulsionnel.

Nous proposons d'obtenir une image I' de

Tableau 2 – Valeurs de $\Phi_{j,k}$ avec $(\perp, \top)_S$ et $\mathcal{K}_\lambda(i, j) = \mathcal{N}_1(i, j)$

$\Phi_{j,k}$	u	0.9	0.8	0.68	0.51	0.5	0.48	0.32	0.1
u	j, k	1	2	3	4	5	6	7	8
0.9	1	1.00	0.89	0.76	0.57	0.56	0.53	0.36	0.11
0.8	2	0.89	1.00	0.85	0.64	0.62	0.60	0.40	0.12
0.68	3	0.76	0.85	1.00	0.75	0.74	0.71	0.47	0.15
0.51	4	0.57	0.64	0.75	1.00	0.98	0.94	0.63	0.19
0.5	5	0.56	0.62	0.74	0.98	1.00	0.96	0.64	0.20
0.48	6	0.53	0.60	0.71	0.94	0.96	1.00	0.67	0.21
0.32	7	0.36	0.40	0.47	0.63	0.64	0.67	1.00	0.31
0.1	8	0.11	0.12	0.15	0.19	0.20	0.21	0.31	1.00

Tableau 3 – Valeurs de $\Phi_{j,k}$ avec $(\perp, \top)_S$ et $\mathcal{K}_\lambda(i, j) = \mathcal{N}_5(i, j)$

$\Phi_{j,k}$	u	0.9	0.8	0.68	0.51	0.5	0.48	0.32	0.1
u	j, k	1	2	3	4	5	6	7	8
0.9	1	1.00	0.89	0.76	0.59	0.57	0.56	0.53	0.36
0.8	2	0.89	1.00	0.85	0.67	0.64	0.63	0.60	0.40
0.68	3	0.76	0.85	1.00	0.75	0.75	0.74	0.71	0.47
0.51	4	0.59	0.67	0.75	1.00	0.98	0.98	0.94	0.63
0.5	5	0.57	0.64	0.75	0.98	1.00	0.96	0.96	0.64
0.48	6	0.56	0.63	0.74	0.98	0.96	1.00	0.67	0.67
0.32	7	0.53	0.60	0.71	0.94	0.96	0.67	1.00	0.31
0.1	8	0.36	0.40	0.47	0.19	0.64	0.67	0.31	1.00

contours à partir de l'image I selon la procédure suivante :

- fuzzification : pour chaque pixel p de I ,
 - $int_I(p) \leftarrow int_I(p) / \max_{p' \in I} int_I(p')$
- initialisation (choix des paramètres)
 - taille de la fenêtre ($x \times y$)
 - t-norme et t-conorme duale
 - fonction noyau et résolution \mathcal{K}_λ
 - taille de bloc $(j, k) \in (x \times y)^2$
- parcours de I : pour chaque pixel p de I ,
 - $u \leftarrow \{ int_I(p') \text{ de } V(p) \}$
 - $\Phi_{j,k}(u)$
- défuzzification : pour chaque pixel p de I ,
 - $int_{I'}(p) \leftarrow (\Phi_{j,k}(u) \times \max_{p' \in I} int_I(p'))$

On obtient une famille de détecteurs de contours paramétrée à la fois par le couple de normes (\top, \perp) et la fonction noyau \mathcal{K}_λ de



Figure 2 – Image originale - Gradient morphologique - Similarité locale avec $\mathcal{N}_{3,18}$ et $(\top, \perp)_S$

l'opérateur $\Phi_{j,k}(u)$. Les résultats que nous donnons ici reposent sur l'utilisation des normes classiques du Tableau 1. Parmi les différentes possibilités pour la fonction noyau, nous avons choisi le noyau gaussien défini par l'équation (7). Arrêtons-nous sur le choix du paramètre de résolution λ qui s'avère prépondérant puisqu'il influe sur la contribution des valeurs à agréger, selon différentes situations :

- **la normalisation** $\sum_{i=1}^c u_i = 1$: même si une telle contrainte n'a a priori pas lieu d'être dans cette application, les u_i tendent alors tous vers 0 et deviennent ainsi similaires, même pour une taille de fenêtre raisonnable. Nous recommandons dans ce cas de choisir une valeur de λ faible, ($\lambda < 2$).
- **la taille de la fenêtre** : si elle est importante, la valeur du noyau tend vers 0 puisque $(k-j)$ augmente. Nous préconisons dans ce cas de choisir une valeur de λ assez élevée, ($\lambda > 5$).
- **la moyenne** : pour une même étendue de valeurs, la détection de contour peut échouer si les valeurs d'intensité sont faibles car le quotient $\Phi_{j,k}$ peut prendre des valeurs relativement différentes, par exemple $\frac{0.05}{0.10} = 0.20$ et $\frac{0.90}{0.95} = 0.95$. Pour pallier ce problème, nous proposons de calculer $\overline{\Phi_{j,k}(\bar{u})}$ avant la défuzzification et de défuzzifier la valeur de :

$$\nabla_{j,k}(u) = \overline{\Phi_{j,k}(\bar{u})} - \Phi_{j,k}(u) \quad (10)$$

au lieu de $\Phi_{j,k}(u)$ dans la procédure ci-dessus.

4.2 Résultats

Dans cette section, plusieurs expériences sont conduites afin de valider la méthode proposée. Pour l'ensemble des images, nous utilisons une taille de fenêtre de $c = 9 = 3 \times 3$, et des valeurs de $j = 1$ et $k = 9$. En d'autres termes, nous ne nous préoccupons pas d'un éventuel bruit impulsif.

La première expérience propose d'utiliser une image d'échiquier sans bruit (figure 2) et sa version bruitée (bruit additionnel gaussien) (figure 3). Comme on peut le voir sur la figure 3, le paramètre de résolution a plusieurs effets : en l'augmentant, les contours deviennent plus précis, et les zones fortement bruitées ou texturées sont lissées. Ceci s'explique finalement en voyant λ comme une intervalle d'étalement de taille variable où les valeurs seront prises en compte. On voit en figure 2 que l'opérateur de similarité locale détecte bien les contours sur une image nette, au même titre que le gradient morphologique. Pourtant, même avec une valeur faible, ($\lambda = 1$), les contours sont très vite supprimés. En pratique, on choisira donc une valeur de λ élevée lorsque l'on voudra éliminer des zones fortement bruitées, et à l'inverse une valeur faible lorsque l'image n'est pas bruitée.

Pour la deuxième expérience, on utilise une image de la base de segmentation de Berkeley [8] pour laquelle des contours ont été définis par des humains. La figure 4 montre que les discontinuités de bruit sont supprimées tandis

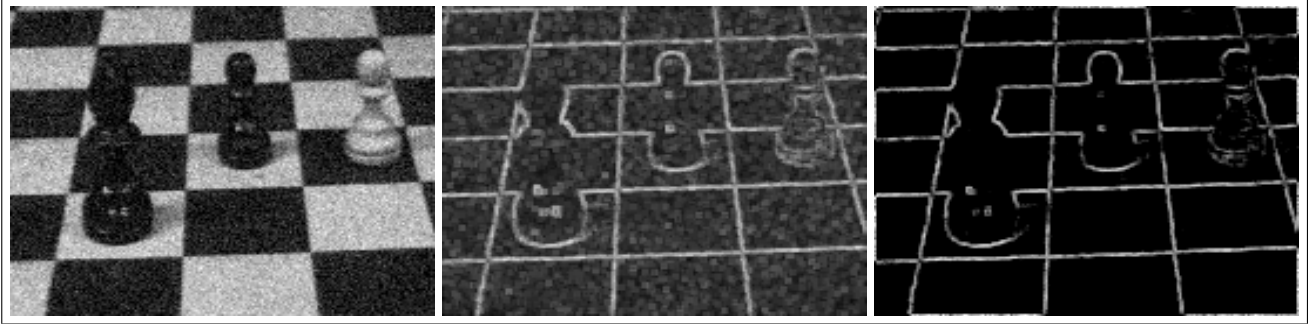


Figure 3 – Image originale bruitée - Gradient morphologique - $\mathcal{N}_{3,18}$ et $(\top, \perp)_S$



Figure 4 – Image originale - Gradient morphologique - Similarité locale avec $\mathcal{N}_{3,18}$ et $(\top, \perp)_L$

que les bords des objets sont conservés, alors que le gradient morphologique ne supprime pas le bruit du fond. Les contours obtenus grâce à cette méthode sont identiques à ceux désirés par les humains. De manière générale, l'utilisation de différentes normes triangulaires n'influe pas beaucoup sur le résultat final : l'effet "filtrage" est moins strict avec $(\top, \perp)_S$ que avec $(\top, \perp)_L$. Il pourrait être intéressant de mettre en oeuvre des normes triangulaires paramétriques (Yager, Dombi, etc ...).

4.3 Liens avec la morphologie mathématique floue

La morphologie mathématique est une théorie initiée dans un premier temps par J. Serra [10] qui consiste à transformer linéairement une image en vue de diverses applications : filtrage, segmentation, reconnaissance de formes. Cette théorie est de nature ensembliste, c'est ce qui justifie son extension au cadre flou, où

les opérateurs binaires sont remplacés par leurs équivalents flous [1]. La définition proposée par I. Bloch et H. Maitre pour la dilatation floue est la suivante :

$$(u \oplus \mu)(x) = \sup_{y \in X} \top [u(x - y), \mu(y)] \quad (11)$$

où μ est l'élément structurant flou et u l'image. Ainsi, le maximum de tous les minima est la valeur résultat à la position x . L'opération duale, l'érosion floue est définie par :

$$(u \ominus \mu)(x) = \inf_{y \in X} \perp [\overline{u(x - y)}, \mu(y)] \quad (12)$$

ou de manière équivalente par

$$u \ominus \mu = \overline{(u \oplus \mu)} \quad (13)$$

Le gradient morphologique prend alors la forme

$$\rho_\mu = (u \oplus \mu) - (u \ominus \mu) \quad (14)$$

On notera le lien entre (10) et (14) : le gradient, et donc la détection des bords des objets, est obtenu par soustraction d'une érosion à

une dilatation. Un certain nombre de propriétés requises ont été listées, en particulier dans [1], pour guider la construction d'opérateurs de morphologie floue. On pourra noter parmi celles-ci l'invariance par translation, la semi-continuité, la croissance, l'idempotence et enfin et surtout, la commutation de la dilatation et de l'érosion ainsi définies avec le suprémum et l'infimum, respectivement. L'intégrale de Sugeno a déjà fait l'objet d'applications à la morphologie floue (cf [6]), et peut être utilisée telle que définie en section 3.2 en construisant une μ -mesure floue. Cette opération, qui n'est pas croissante, fixe sa mesure floue de manière spatiale, comme le noyau dans Φ . Dans le cas de l'utilisation des normes triangulaires, les quatre propriétés indispensables sont respectées (invariance par translation, compatible avec les homothéties, connaissance locale, et semi-continuité, voir [1]). Notre opérateur Φ , selon les normes triangulaires, peut donc être vu comme le quotient de la dilatation des éléments minimaux sur les éléments maximaux, mais avec un élément structurant flou symétrique, comme on l'a vu en section 4.1. Ces dilatations peuvent être qualifiées de standard, algébrique ou de Lukasiewicz, selon la famille de normes triangulaires utilisée et respectant ou non les propriétés requises.

5 Conclusion

Dans cet article, nous avons proposé un nouvel opérateur qui estime, pour un c -uplet à valeurs dans $[0, 1]$ donné, la similarité de certaines de ses composantes. Fondé sur des normes triangulaires et l'intégrale de Sugeno, il combine les valeurs et une fonction noyau à définir. Nous avons démontré comment la définition de cet opérateur lui permet de détecter des similarités par blocs à différents niveaux de résolution grâce au noyau introduit.

Parmi les applications qui peuvent être envisagées, nous avons choisi d'étudier un problème de grande importance en traitement d'image, la détection de contours. Cet opérateur rejoint les performances des opérateurs classiques

lorsqu'il y a absence de bruit, mais s'avère plus performant en présence d'un bruit important en particulier grâce au paramètre de résolution λ . Notre opérateur constituant de fait une famille, il s'agit d'étudier l'influence respective de tous ces paramètres. De futurs travaux concerneront le choix des couples (\top, \perp) , des noyaux spécifiques. Enfin, nous nous intéresserons à l'application de cet opérateur dans un contexte de rejet par ambiguïté en classification.

Références

- [1] I. Bloch et H. Maître, Fuzzy Mathematical Morphology, *Annals of Mathematics and Artificial Intelligence*, Volume 10, Numbers 1-2, pp.55-84, March 1994.
- [2] J. Canny, A Computational Approach to Edge Detection, *IEEE Trans. on Pattern Analysis and Machine Intelligence* Volume 8, pp. 6979–6986, 1986.
- [3] C. Frélicot, L. Mascarilla et M. Berthier, A new cluster validity index for fuzzy clustering based on Combination of dual triples, *Proc. IEEE International Conference on Fuzzy systems*, Vancouver, Canada, 2006.
- [4] R.C. Gozalez et E.W. Richard, *Digital Image Processing*, Prentice-Hall Inc., 2002.
- [5] M. Grabisch et M. Schmitt, Mathematical morphology, order filters and fuzzy logic. *Int. Joint Conf. of the 4th IEEE Int. Conf. on Fuzzy Systems and the 2nd Int. Fuzzy Engineering Symposium*, Yokohama, Japan, pp.2103-2108, March 1995.
- [6] M. Grabisch, T. Murofushi et M. Sugeno, Fuzzy measure of fuzzy events defined by fuzzy integrals, *Fuzzy Sets and Systems*, Volume 50, pp. 293–313, 1991.
- [7] E.P. Klement et R. Mesiar (Eds), *Logical, algebraic, analytic and probabilistic aspects of triangular norms*. Elsevier, 2005.
- [8] D. Martin, C. Fowlkes, D. Tal et J. Malik, A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics, *Proc. of 8th International Conference on Computer Vision*, Volume 2, pp. 416–423, July 2001.
- [9] L. Mascarilla, M. Berthier et C. Frélicot, A K-order Fuzzy OR Operator-Application in Pattern Classification with k-order Ambiguity. *Proc. of 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems*, Paris, France, 2006.
- [10] J. Serra, *Image analysis and Mathematical Morphology*. Academic Press London, 1982.
- [11] T. Terano, K. Asai et M. Sugeno, *Fuzzy Systems Theory and Its Applications*. Academic Press, 1992.