

One Image Reduction Algorithm for RGB Color Images

D. Paternain, A. Jurio, M. Pagola, H. Bustince¹ G. Beliakov²

¹Universidad Publica de Navarra. Pamplona, Spain
 {daniel.paternain, aranzazu.jurio, miguel.pagola, bustince}@unavarra.es

²Deakin University. Burwood, Australia
 gleb@deakin.edu.au

Abstract

We investigate the problem of combining or aggregating several color values given in coding scheme RGB. For this reason, we study the problem of averaging values on lattices, and in particular on discrete product lattices. We study the arithmetic mean and the median on product lattices. We apply these aggregation functions in image reduction and we present a new algorithm based on the minimization of penalty functions on discrete product lattices.

Keywords: Color image reduction, Aggregation Operators.

1. Introduction

The need to aggregate several inputs into a single representative output frequently arises in many practical applications. In image processing, it is often necessary to average the values of several neighboring pixels (to reduce the image size or apply a filter), or average pixel values in two different but related images (e.g., in stereovision). When the images are in color, typically coded as discrete RGB, CMY, or HSL values, then it is customary to average the values in the respective channels. It is not immediately clear that this is appropriate, and what are the other ways to average color values.

The objective of this work is to present a new color image reduction algorithm which is based on minimizing a penalty function defined over product lattices and to analyze the stability of the algorithm presented with respect to noise in the images studying the capability of the algorithm to filter impulsive noise in images.

We first recall the problem of image reduction for grayscale images and we justify the importance of penalty functions. We prove that, when we reconstruct a reduced image, the error with respect to the original image may be determined by the reduction method that has been employed.

Next, we extend our study to color images. For this purpose, we study averaging on product lattices (RGB or another color coding scheme is an example of a product lattice). We show that with

an appropriately chosen class of penalties, the resulting penalty-based functions are monotone and idempotent. We also show that the averages over a product lattice are in general different from the Cartesian products of the averages. This has an implication over the methods of color image reduction or color image filtering.

The structure of the paper is as follows. In Section 2 we provide preliminary definitions. In Section 3 we give the definitions of aggregation functions based on penalties defined on product lattices. In Section 4 we present the problem of image reduction in grayscale images and we propose a new algorithm of color image reduction. Conclusions are presented in Section 5

2. Preliminaries

We start recalling some concepts about aggregation functions and penalty functions. Then we introduce some concepts about lattices.

2.1. Aggregation functions

Definition 1 A function $f : [a, b]^n \rightarrow [a, b]$ is called an aggregation function if it is monotone non-decreasing in each variable and satisfies $f(\mathbf{a}) = a$, $f(\mathbf{b}) = b$, with $\mathbf{a} = (a, a, \dots, a)$, $\mathbf{b} = (b, b, \dots, b)$.

Recent books providing a comprehensive overview of aggregation functions include [1, 2, 3, 4].

Definition 2 An aggregation function f is called averaging if it is bounded by the minimum and maximum of its arguments

$$\min(\mathbf{x}) := \min(x_1, \dots, x_n) \leq f(x_1, \dots, x_n) \leq \max(x_1, \dots, x_n) =: \max(\mathbf{x}).$$

It is immediate that averaging aggregation functions are idempotent (i.e., $\forall t \in [a, b] : f(t, t, \dots, t) = t$) and (because of monotonicity) vice versa.

Well known examples of averaging functions are the arithmetic mean and the median. It is known that the arithmetic means and the median are solutions to simple optimization problems, in which a measure of disagreement between the inputs is minimized, see [2, 5, 6, 7, 8]. The main motivation is the

following. Let \mathbf{x} be the inputs and y be the output. If all the inputs coincide $x = x_1 = \dots, x_n$, then the output is $y = x$, and we have a unanimous vote. If some input $x_i \neq y$, then we impose a ‘‘penalty’’ for this disagreement. The larger the disagreement, and the more inputs disagree with the output, the larger (in general) is the penalty. We look for an aggregated value which minimizes the penalty.

Definition 3 Let $P : [a, b]^{n+1} \rightarrow \mathfrak{R}$ be a penalty function with the properties

- i) $P(\mathbf{x}, y) \geq 0$ for all \mathbf{x}, y ;
- ii) $P(\mathbf{x}, y) = 0$ if all $x_i = y$;
- iii) $P(\mathbf{x}, y)$ is quasiconvex in y for any \mathbf{x} .

The penalty based function is

$$f(\mathbf{x}) = \arg \min_y P(\mathbf{x}, y),$$

if y is the unique minimizer, and $y = \frac{a+b}{2}$ if the set of minimizers is the interval $[a, b]$.

In [6] it was shown that any averaging aggregation function can be represented as a penalty based function.

Example: The arithmetic mean is the solution to

$$\text{minimize}_y \sum_{i=1}^n (x_i - y)^2$$

whereas the median is a solution to

$$\text{minimize}_y \sum_{i=1}^n |x_i - y|.$$

In this work we will deal with penalty based functions defined on discrete lattices, rather than the interval $[a, b]$.

2.2. Lattices

Definition 4 Let L be a set. A lattice $\mathcal{L} = (L, \leq, \wedge, \vee)$ is a poset with the partial order \leq on L , and meet and join operations \wedge, \vee , if every pair of elements from L has both meet and join.

Definition 5 Let P be a poset. A chain in P is a totally ordered subset of P . The length of a chain is its cardinality minus one.

Proposition 1 Let $\mathcal{L}_1 = (L_1, \leq_1, \wedge_1, \vee_1)$ and $\mathcal{L}_2 = (L_2, \leq_2, \wedge_2, \vee_2)$ be two lattices. The Cartesian product $\mathcal{L}_1 \times \mathcal{L}_2 = (L_1 \times L_2, \leq, \wedge, \vee)$ with \leq defined by

$$(x_1, y_1) \leq (x_2, y_2) \Leftrightarrow x_1 \leq_1 x_2 \text{ and } y_1 \leq_2 y_2,$$

and

$$(x_1, y_1) \wedge (x_2, y_2) = (x_1 \wedge_1 x_2, y_1 \wedge_2 y_2),$$

$$(x_1, y_1) \vee (x_2, y_2) = (x_1 \vee_1 x_2, y_1 \vee_2 y_2).$$

is a lattice.

In this work we will represent each color in an image as a finite chain \mathcal{C} (with the length of each chain typically being 256). In this way, each pixel in a color image can be seen as a Cartesian product of finite chains. We note that all finite chains of the same length are isomorph to each other, and hence we can represent them as non-negative integers $0, 1, \dots, K$, and elements of product lattices as tuples $x = (x_1, x_2, \dots, x_m)$, $x_i \in Z_+ = \{0, 1, 2, \dots\}$.

Definition 6 Let f_1, f_2 be two aggregation functions defined on sets X_1 and X_2 respectively. The Cartesian product of aggregation functions is $f = f_1 \times f_2 : X_1 \times X_2 \rightarrow Y_1 \times Y_2$ defined by

$$f(x_1, x_2) = (f_1(x_1), f_2(x_2)).$$

3. Penalty functions on discrete product lattices

In this section we define aggregation functions as penalty based aggregation functions defined on lattices.

Definition 7 Let $\mathcal{L} = (L, \leq, \wedge, \vee)$ be a product of finite chains. The distance between $x, y \in \mathcal{L}$ is defined as the length of a chain \mathcal{C} with the least element $a = x \wedge y$ and the greatest element $b = x \vee y$ minus 1,

$$d(x, y) = \text{length}(\mathcal{C}) - 1.$$

This distance is called the geodesic distance, since it corresponds to the smallest number of edges between vertices x to y in the covering graph of L .

We note that all the chains with the least element a and the greatest element b on a product lattice in Definition 7 have the same length. This definition is equivalent to the following

$$d(x, y) = \sum_{i=1}^m d_i(x_i, y_i) = \sum_{i=1}^m |x_i - y_i|,$$

where d_i is the distance in the i -th chain in the product of m chains.

Let \mathcal{L} be a product of finite chains. Consider n elements $x_1, \dots, x_n \in \mathcal{L}$, that need to be averaged. Let the penalty function be $P : Z_+^n \rightarrow \mathfrak{R}$. The penalty based function on \mathcal{L} is f given by

$$f(x_1, \dots, x_n) = \mu = \arg \min_{y \in \mathcal{L}} P(\mathbf{x}, y)$$

It is easy to see that $f(x_1, \dots, x_n)$ is an averaging (and hence idempotent) function.

A special case of penalty based functions was considered in [9], called dissimilarity functions, where the penalty P is given by

$$P(\mathbf{x}, y) = \sum_{i=1}^n K(x_i - y), \quad (1)$$

where K is a convex function with the unique minimum $K(0) = 0$. In this case the penalty based function is monotone, i.e., an aggregation function. By adapting this definition to our case we have

Theorem 1 Let $x_1, \dots, x_n \in \mathcal{L}$. The function $f : \mathcal{L}^n \rightarrow \mathcal{L}$ given by

$$\begin{aligned} f(x_1, \dots, x_n) &= \mu = \arg \min_{y \in \mathcal{L}} P(\mathbf{x}, y) = \\ &= \arg \min_{y \in \mathcal{L}} \sum_{i=1}^n K(d(x_i, y)) \end{aligned}$$

is an averaging aggregation function on a product lattice.

We provide definitions for the arithmetic mean and the median in the form (1), so Theorem 1 applies.

Definition 8 Let K be:

1. $K(x) = x^2$, and hence $P(\mathbf{x}, y) = \sum_{i=1}^n d(x_i, y)^2$. Then the resulting penalty based aggregation function is the arithmetic mean on discrete product lattices.
2. $K(x) = |x|$, and hence $P(\mathbf{x}, y) = \sum_{i=1}^n d(x_i, y)$. Then the resulting penalty based aggregation function is the median on discrete product lattices.

Considering Definition 8 we have that the arithmetic mean is the solution of

$$\text{minimize}_{y \in \mathcal{L}} \sum_{i=1}^n \left(\sum_{j=1}^m |x_{ij} - y_j| \right)^2, \quad (2)$$

where x_{ij} denotes the j -th component of the i -th tuple $x_i \in \mathcal{L}$. We note that this problem is convex in y . We also note that the solution is *different* from the Cartesian product of the means. Considering now the median, we have the problem

$$\text{minimize}_y \sum_{i=1}^n \sum_{j=1}^m |x_{ij} - y_j| = \sum_{j=1}^m \sum_{i=1}^n |x_{ij} - y_j|. \quad (3)$$

The minimum is achieved at $y = (\text{Med}(x_{\cdot,1}), \dots, \text{Med}(x_{\cdot,m}))$, i.e., the cartesian product of the medians.

4. Image reduction

Image reduction consists in reducing the dimension of the image while keeping as much information as possible. Image reduction can be used to accelerate computations on an image, or just to reduce the cost of its storage or transmission.

There exist several methods for image reduction in the literature. Some of them consider the image to be reduced in a global way [10, 11]. Other widely used methods act locally over pieces (blocks) of the image [12, 13, 14]. The division of the image in blocks of small size allows one to design simple reduction algorithms. In this work we focus on the latter.

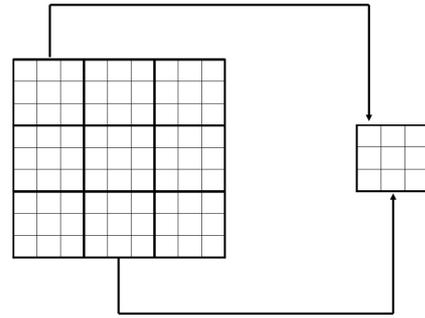


Figure 1: Scheme of the reduction algorithm

In this work, we consider an image of $N \times M$ pixels as a set of $N \times M$ elements arranged in rows and columns. Each element of a grayscale image is represented by x_{ij} with $i \in \{1, \dots, N\}$ and $j \in \{1, \dots, M\}$. The element x_{ij} has a value between 0 and $L - 1$. If we consider color image in the RGB reference system, each element of the image is denoted by $\mathbf{x}_{ij} = (x_{Rij}, x_{Gij}, x_{Bij})$. Each color component will also have a value between 0 and $L - 1$.

We first consider image reduction algorithms in grayscale images and we study the relation between these algorithms and penalty functions. Then, we propose a new image reduction algorithm for color images.

4.1. Image reduction in grayscale images

Consider the following local image reduction algorithm: given an image Q of dimension $N \times M$:

1. Divide the image Q into disjoint blocks of dimension $n \times n$. If N or M are not multiples of n eliminate the smallest number of rows and/or columns to satisfy this condition.
 2. Choose an averaging function f .
 3. FOR each block in Q DO
 - 3.1. Calculate $f(x_{11}, \dots, x_{1n}, \dots, x_{nn})$.
 - 3.2. Place the result in the corresponding pixel of the reduced image (see Figure 1).
- END FOR

Reduction algorithm for grayscale images.

In Figure 2 we show three reduced images obtained from the original image (a) using the following aggregation functions in Step 2: the geometric mean (b1), the arithmetic mean (b2) and the median (b3).

One of the most frequently method to determine the best reduction consists in magnify the image (reduced) to the original dimension and measure the error between the reconstructed and the original image.

For simplicity, we consider the following reconstruction method: for each pixel of the reduced image build a new block of dimension $n \times n$ whose elements have the same value as that pixel.



Figure 2: Original image and reductions

	Image (b1)	Image (b2)	Image (b3)
MSE	224.0065	209.23	237.44
MAE	7.89	7.88	7.42

Table 1: MSE and MAE between reconstructed and original images

Next we show that once the reduction and magnification methods are fixed, the difference between the original and the reduced (and then magnified) image may be determined by the aggregation function used in the reduction algorithm.

We measure the error in the reconstructed images by using the following expressions to compare two images Q, Q' of dimension $N \times M$:

1. Mean Squared Error: $MSE(Q, Q') = \frac{1}{N \times M} \sum_{i=1}^N \sum_{j=1}^M (Q_{ij} - Q'_{ij})^2$
2. Mean Absolute Error: $MAE(Q, Q') = \frac{1}{N \times M} \sum_{i=1}^N \sum_{j=1}^M |Q_{ij} - Q'_{ij}|$

Notice that from the results in Table 1 we have that:

1. If we take MSE, then the best reduction is obtained using the arithmetic mean.
2. If we take MAE, then the best reduction is obtained using the median.

4.2. Image reduction in color images

In this section we present a new color image reduction algorithm based on minimization of penalty functions that are not built as Cartesian products of the corresponding aggregation functions. The idea is to find an approximate solution to the penalty minimization problem chosen from a small subset of alternatives.

The color image reduction algorithm is based on fixing a number of different aggregation functions,

and selecting the one which minimizes a penalty function P for each block of the image:

1. Divide the image Q in disjoint blocks of dimension $n \times n$. If N or M are not multiples of n eliminate the necessary number of rows and/or columns to satisfy this condition.
2. Choose a penalty function P .
3. Take k averaging aggregation functions Ag_1, \dots, Ag_k .
4. FOR each block in Q DO

- 4.1. Apply to each pixel in each block (in the three channels R, G and B) k aggregation functions, as follows:

$$\mathbf{y}_{Ag_1} = (y_{RAg_1}, y_{GAg_1}, y_{BAg_1}) = \begin{pmatrix} Ag_1(x_{Rij}), Ag_1(x_{Gij}), Ag_1(x_{Bij}) \\ i=1..n, j=1..n \end{pmatrix}$$

...

$$\mathbf{y}_{Ag_k} = (y_{RAg_k}, y_{GAg_k}, y_{BAg_k}) = \begin{pmatrix} Ag_k(x_{Rij}), Ag_k(x_{Gij}), Ag_k(x_{Bij}) \\ i=1..n, j=1..n \end{pmatrix}$$

- 4.2. Calculate the penalties $P_i = P(\mathbf{x}, \mathbf{y}_{Ag_i})$ for each \mathbf{y}_{Ag_i}
- 4.3. Assign the value \mathbf{y}_{Ag_i} with the smallest penalty to the corresponding pixel of the reduced image.

END FOR

Algorithm 1

In Figure 3 we illustrate Algorithm 1 on two color images in RGB (images (a) and (b)) in the same setting as in the Example ???. In Table 2 and 3 we show the frequency of choosing each of the aggregation functions. Notice that the biggest percentage corresponds to taking the arithmetic mean \mathbf{y}_{arith} .

Remark 1 *Observe that if all the values of the color components are the same, we can take any averaging function, because they are all idempotent (column Any in Table 2 and 3).*

Min	Geom	Arith	Med	Max	Any
0.00%	8.97%	73.83%	17.15%	0.00%	0.05%

Table 2: Frequency of choosing aggregation functions by Algorithm 1 in Image (a)

Min	Geom	Arith	Med	Max	Any
0.06%	8.89%	71.94%	19.02%	0.09%	0.00%

Table 3: Frequency of choosing aggregation functions by Algorithm 1 in Image (b)

4.2.1. Reaction to noise

We know that impulsive noise can be frequent in practice. We want to analyze the behaviour of Algorithm 1 when images are altered by this type of

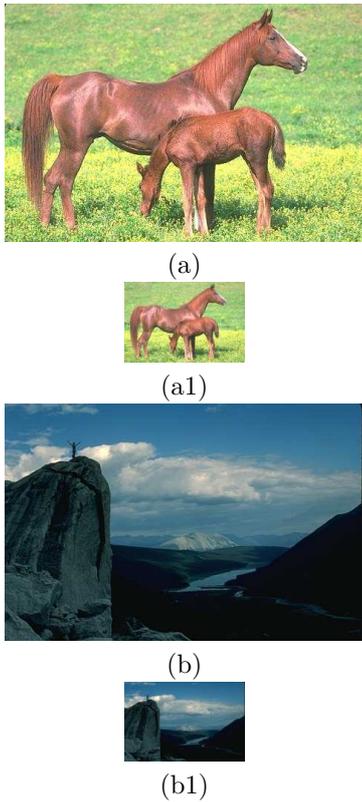


Figure 3: Color image reduction by Algorithm 1

noise. For the experiment we take the same images shown in Figure 3. We add salt and pepper noise with a density of 5, 10, 20 and 30 % of corrupted pixels.

In Table 4 we repeat the experiment of Tables 2 and 3 for these new images. Notice that as the amount of impulsive noise increases, the frequency of choosing the median as the best aggregation functions also increases. This behaviour makes the reduced images to decrease the amount of noise. As the median is taken in the majority of the blocks of the image, Algorithm 1 allows to discard the noise (the median is not affected by the extremal values taken by corrupted pixels).

This change in the frequency of the aggregation functions is shown in Figure 5. On the horizontal axis we show the percentage of pixels affected by noise. On the vertical axis we show the percentage of times that each aggregation function is selected by Algorithm 1. The larger the impulsive noise, the more often the median is selected instead the arithmetic mean.

We show the result of applying Algorithm 1 to real color images in Figure 4. In first row we show original images with noise. In second and third row we show the images obtained applying Algorithm 1 and a simple reduction algorithm of subsampling. Observe that the quality of images (a1) and (b1) is very good if we compare with the quality of images (a2) and (b2).

Moreover, the main advantage of Algorithm 1 is that it makes unnecessary to use an ad-hoc filter

prior to the image reduction in order to eliminate salt and pepper noise.

		<i>Min</i>	<i>Geom</i>	<i>Arith</i>	<i>Med</i>	<i>Max</i>
(a)	5%	0.00%	15.44%	52.59%	31.92%	0.05%
(b)	5%	0.11%	9.47%	61.59%	28.60%	0.23%
(a)	10%	0.02%	9.55%	41.59%	48.81%	0.03%
(b)	10%	0.15%	6.19%	45.30%	48.20%	0.16%
(a)	20%	0.01%	2.46%	30.09%	67.41%	0.03%
(b)	20%	0.12%	6.19%	45.30%	48.20%	0.06%
(a)	30%	0.01%	0.50%	27.10%	72.33%	0.06%
(b)	30%	0.06%	1.05%	22.62%	76.27%	0.00%

Table 4: Frequency of choosing aggregation functions by Algorithm 1 when images are affected by salt and pepper noise

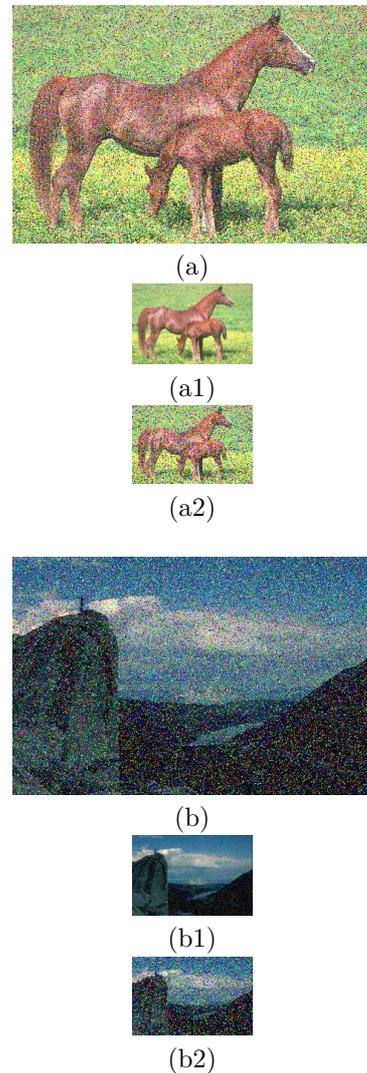


Figure 4: Reduction of images with impulsive noise (20% of pixels)

5. Conclusions

We have studied the importance of aggregation functions and penalty functions in image reduction. We have shown that when we fix the reduction and

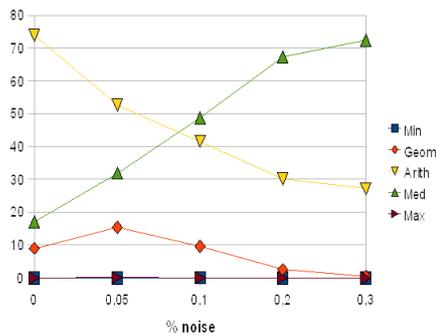


Image (a)

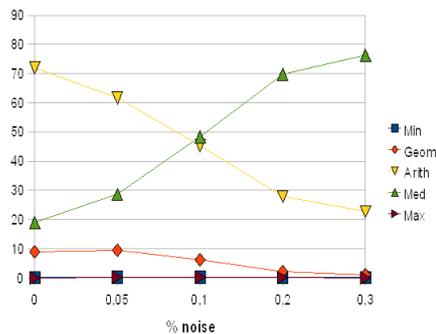


Image (b)

Figure 5: Frequency of aggregation functions as a function of the intensity of the salt and pepper noise.

magnification algorithm in grayscale images, the quality of the reduced image is determined by the aggregation function used in the reduction process.

We have investigated the problem of aggregation color values in RGB. In this way, we have studied aggregation functions in product lattices. In this context, we have presented a new image reduction algorithm based on aggregation by means of penalty functions. We have shown that the proposed algorithm is a very efficient filter for impulsive noise.

As future research, we want to compare our method with other reduction algorithms in the literature. Moreover, we want to study the effectiveness of our algorithm when filtering other kind of noise, as gaussian noise or speckle noise.

Acknowledgement This work has been partially supported by the National Science Foundation of Spain, reference TIN2010-15055

References

- [1] G. Beliakov, A. Pradera, and T. Calvo. *Aggregation Functions: A Guide for Practitioners*. Springer, Heidelberg, Berlin, New York, 2007.
- [2] Y. Torra, V. Narukawa. *Modeling Decisions. Information Fusion and Aggregation Operators*. Springer, Berlin, Heidelberg, 2007.
- [3] T. Calvo, G. Mayor, and R. Mesiar, editors. *Aggregation Operators. New Trends and Appli-*

cations. Physica-Verlag, Heidelberg, New York, 2002.

- [4] M. Grabisch, J.-L. Marichal, R. Mesiar, and E. Pap. *Aggregation Functions*. Cambridge University press, Cambridge, 2009.
- [5] C. Gini. *Le Medie*. Unione Tipografico-Editoriale Torinese, Milan (Russian translation, Srednie Velichiny, Statistika, Moscow, 1970), 1958.
- [6] T. Calvo and G. Beliakov. Aggregation functions based on penalties. *Fuzzy Sets and Systems*, 161:1420–1436, 2010.
- [7] T. Calvo, R. Mesiar, and R. Yager. Quantitative weights and aggregation. *IEEE Trans. on Fuzzy Systems*, 12:62–69, 2004.
- [8] R. Yager and A. Rybalov. Understanding the median as a fusion operator. *Int. J. General Syst.*, 26:239–263, 1997.
- [9] R. Mesiar. Fuzzy set approach to the utility, preference relations, and aggregation operators. *Europ. J. Oper. Res.*, 176:414–422, 2007.
- [10] I. Perfilieva. Fuzzy transforms: Theory and applications. *Fuzzy Sets and Systems*, 157:993–1023, 2006.
- [11] H. Nobuhara, K. Hirota, S. Sessa, and W. Pedrycz. Efficient decomposition methods of fuzzy relation and their application to image decomposition. *Applied Soft Computing*, 5:399–408, 2005.
- [12] F. Di Martino, V. Loia, I. Perfilieva, and S. Sessa. An image coding/decoding method based on direct and inverse fuzzy transforms. *International Journal of Approximate Reasoning*, 48:110–131, 2008.
- [13] V. Loia and S. Sessa. Fuzzy relation equations for coding/decoding processes of images and videos. *Information Sciences*, 171:145–172, 2005.
- [14] D. Paternain, H. Bustince, J. Sanz, M. Galar, and C. Guerra. Image reduction with interval-valued fuzzy sets and OWA operators. In J.P. Carvalho, D. Dubois, U. Kaymak, and J.M.C. Sousa, editors, *Proc. of the 13th IFSA World Congress and 6th EUSFLAT Conference*, pages 754–759, Lisbon, Portugal, 2009.