New measures for comparing matrices and their application to image processing

Mikel Sesma-Sara[a,b,c], Laura De Miguel[a,b,c], Miguel Pagola[a,b,c], Ana Burusco[a,b,c], Radko Mesiar[d,e], Humberto Bustince[a,b,c,*]

[a] *Departamento de Automática y Computación, Universidad Pública de Navarra, Campus Arrosadia s/n, 31006, Pamplona, Spain*
[b] *Institute of Smart Cities, Universidad Pública de Navarra, 31006, Pamplona, Spain*
[c] *Laboratory, Navarrabiomed, Complejo Hospitalario de Navarra (CHN), Universidad Pública de Navarra (UPNA), IdiSNA. Irunlarrea 3, 31008 Pamplona, Spain*
[d] *Department of Mathematics and Descriptive Geometry, Faculty of Civil Engineering, Slovak University of Technology, Radlinského 11, Bratislava, Slovakia*
[e] *Institute for Research and Applications of Fuzzy Modelling, University of Ostrava, 30. dubna 22, Ostrava 1, Czech Republic*

Abstract

In this work we present the class of matrix resemblance functions, i.e., functions that measure the difference between two matrices. We present two construction methods and study the properties that matrix resemblance functions satisfy, which suggest that this class of functions is an appropriate tool for comparing images. Hence, we present a comparison method for grayscale images whose result is a new image, which enables to locate the areas where both images are equally similar or dissimilar. Additionally, we propose some applications in which this comparison method can be used, such as defect detection in industrial manufacturing processes and video motion detection and object tracking.

Keywords: Matrix resemblance functions; Restricted equivalence functions; Inclusion grades; Fuzzy mathematical morphology; Defect detection; Motion detection.

1. Introduction

Measuring how similar or dissimilar two images are is a problem that is far from being closed. There exist many instances of similarity measures and indices [1–3], however there is no standard measure for comparing two images. Moreover, most techniques perform a pixel-wise comparison, which does not take into account the impact that the surrounding of a pixel has when deciding whether the images that are being compared are more or less similar. Another problem with the usual comparison methods is that the result is usually given by a number, which need not be representative in many cases.

*Corresponding author

Email addresses: mikel.sesma@unavarra.es (Mikel Sesma-Sara), laura.demiguel@unavarra.es (Laura De Miguel), miguel.pagola@unavarra.es (Miguel Pagola), burusco@unavarra.es (Ana Burusco), mesiar@math.sk (Radko Mesiar), bustince@unavarra.es (Humberto Bustince)

Many industrial processes make use of image comparison techniques to guarantee certain quality standards. For instance, in the manufacturing process of printed circuit boards (PCB), all the products are compared with an image of an ideal PCB in order to detect any potential defect (see [4]). Another example of the usage of image comparison techniques is video motion detection. It is possible to detect objects that are moving in a video by adequately comparing its frames. Similarly, image comparison is also used for tamper detection as in [5]. All these instances of possible applications for image comparison techniques perform better when the information from the neighbourhoods of pixels is taken into account and they especially benefit from the result of the comparison being a new image.

In order to address these problems, in this work we present a method to compare images, but instead of carrying out a comparison pixel by pixel, we compare neighbourhoods of pixels, i.e., sets consisting of a central pixel and the ones that are adjacent to it. Thus, we include in the comparison the information that can be retrieved from the neighbourhood of each pixel, rather than considering just the pixel itself. Moreover, to avoid representing the difference with a number, the result of the method presented in this paper provides a new image for an outcome, which we call *comparison image*.

To develop a comparison method that preserves the aforementioned features we define a class of functions, the class of matrix resemblance functions, that are adequate to carry out a comparison between the neighbourhoods of two pixels, which ultimately are nothing other than matrices. Along with the definition, we present two different construction methods for this kind of functions. The first one is based on restricted equivalence functions [6] and the second one on inclusion grades [7, 8]. Since our aim is to present a comparison method, we also study some properties that are usually demanded to comparison methods in order to be proper similarity measures (see [2, 3, 6]).

The structure of the paper is as follows: first, we present some preliminary concepts that help making the paper self-contained. In Section 3 the concept of matrix resemblance function is introduced, and in Section 4 two construction methods and some examples are presented. Section 5 exhibits the relation between matrix resemblance function and the erosion operator from fuzzy mathematical morphology. In Section 6, a summary of the properties that image comparison measures ought to satisfy is presented and the conditions under which matrix resemblance functions fulfil them are shown. In Section 7, we present a study of the particular cases where in the construction of matrix resemblance functions an aggregation function and a n-dimensional overlap function are used. In Section 8 an algorithm to compare images based on matrix resemblance functions is introduced and in Section 9 some illustrative examples of this image comparison method are presented. In section 10 we include three fields in which our algorithm could be applied: tamper detection, defect detection in PCB manufacturing processes and a method to compare videos which can be applied to object motion detection and tracking.

2. Preliminaries

A fuzzy set $\mathcal{A}$ on a universe $X \neq \emptyset$ is a mapping $\mathcal{A} : X \to [0,1]$. Given a point $x \in X$, $\mathcal{A}(x)$ refers to the membership degree of the point x in the fuzzy set $\mathcal{A}$. In the case of grayscale images, X would be the set of pixels and $\mathcal{A}(x)$ the intensity of the pixel x.

We consider the neighbourhood of a pixel to be a square set of pixels that are surrounding the central one, so they can be thought of as square matrices with values in the unit interval. We denote the set of this kind of $n \times n$ matrices by $\mathcal{M}_n([0,1])$.

A fuzzy negation is a generalization of the negation in classical logic.

Definition 2.1. A function $c : [0,1] \to [0,1]$ is called a fuzzy negation if $c(0) = 1$, $c(1) = 0$ and c is decreasing. Additionally, c is said to be strict if it is continuous and strictly decreasing. A negation c is called strong negation if it is involutive, i.e., $c(c(x)) = x$ for all $x \in [0,1]$.

Example 2.2. The function $c_z : [0,1] \to [0,1]$ given by $c_z(x) = 1 - x$ is a strong negation. It was given by Zadeh in [9].

Denoting $FS(X)$ the set of all fuzzy sets defined on the universe $X \neq \emptyset$, if $\mathcal{A} \in FS(X)$, we call c-complement of $\mathcal{A}$ to the fuzzy set given by the membership function $\mathcal{A}_c(x) = c(\mathcal{A}(x))$, where c is a fuzzy negation.

A restricted equivalence function [6], or a REF, is a function that enables a comparison between two numbers in the unit interval.

Definition 2.3. Let c be a strong negation. A function $REF : [0,1]^2 \to [0,1]$ is called a restricted equivalence function with respect to c if it satisfies the following conditions:

(REF1) $REF(x,y) = REF(y,x)$ for all $x, y \in [0,1]$;

(REF2) $REF(x,y) = 1$ if and only if $x = y$;

(REF3) $REF(x,y) = 0$ if and only if $\{x,y\} = \{0,1\}$;

(REF4) $REF(x,y) = REF(c(x), c(y))$ for all $x, y \in [0,1]$;

(REF5) For all $x, y, z \in [0,1]$, if $x \leq y \leq z$, then $REF(x,y) \geq REF(x,z)$ and $REF(y,z) \geq REF(x,z)$.

Example 2.4. $REF(x,y) = 1 - |x - y|$ is a restricted equivalence function with respect to the strong negation c_z.

An implication operator is a generalization of the implication in classical logic. It is defined as follows:

Definition 2.5. A function $I : [0,1]^2 \to [0,1]$ is called implication operator if it satisfies the following conditions:

(I1) If $x \leq z$, then $I(x,y) \geq I(z,y)$ for all $y \in [0,1]$;

(I2) If $y \leq t$, then $I(x,y) \leq I(x,t)$ for all $x \in [0,1]$;

(I3) $I(0,x) = 1$ for all $x \in [0,1]$;

(I4) $I(x,1) = 1$ for all $x \in [0,1]$;

(I5) $I(1,0) = 0$.

The following are some additional conditions that are frequently demanded to implication operators:

(I6) $I(1,x) = x$ for all $x \in [0,1]$;

(I7) $I(x, I(y,z)) = I(y,(x,z))$;

(I8) $I(x, y) = 1$ if and only if $x \leq y$;

(I9) $I(x, 0) = c(x)$ is a strong negation;

(I10) $I(x, y) \geq y$;

(I11) $I(x, x) = 1$;

(I12) $I(x, y) = I(c(y), c(x))$ with c a strong negation;

(I13) I is a continuous function.

Example 2.6. The function $I_L(x, y) = \min(1, 1 - x + y)$ is called the Lukasiewicz implication and it satisfies all conditions (I1)-(I13) when considering c_z as strong negation. Conversely, a function that satisfies (I1)-(I13) for a strong negation c, is an isomorphic transformation of I_L, such that the related isomorphism φ generates the strong negation c as $c(x) = \varphi^{-1}(1 - \varphi(x))$ (see [6]).

A fuzzy inclusion grade, inclusion degree, or subsethood measure [8, 10, 11], is a function that indicates how included a fuzzy set is in another. There are three main axiomatizations for this concept; the one given by Kitainik [12] in 1987, the one by Sinha and Dougherty [7] in 1993, and the one by Young [13] in 1996. In 1999, Fan et al. [14] made some modifications to Young's axioms. We have chosen the axiomatization given by Sinha and Dougherty due to the fact that the second axiom allows to link this work to fuzzy mathematical morphology.

Given two fuzzy sets $\mathcal{A}$, $\mathcal{B} \in FS(X)$, we set the fuzzy sets $\mathcal{A} \vee \mathcal{B} \in FS(X)$ and $\mathcal{A} \wedge \mathcal{B} \in FS(X)$, given by $(\mathcal{A} \vee \mathcal{B})(x) = \max(\mathcal{A}(x), \mathcal{B}(x))$ and $(\mathcal{A} \wedge \mathcal{B})(x) = \min(\mathcal{A}(x), \mathcal{B}(x))$ for all $x \in X$, respectively.

Definition 2.7. A function $\sigma : FS(X) \times FS(X) \to [0, 1]$ is called an inclusion grade in the sense of Sinha and Dougherty if it satisfies the following axioms:

(IG1) $\sigma(\mathcal{A}, \mathcal{B}) = 1$ if and only if $\mathcal{A} \leq \mathcal{B}$ in Zadeh's sense[1];

(IG2) $\sigma(\mathcal{A}, \mathcal{B}) = 0$ if and only if there exists x_i such that $\mathcal{A}(x_i) = 1$ and $\mathcal{B}(x_i) = 0$;

(IG3) If $\mathcal{B} \leq \mathcal{C}$, then $\sigma(\mathcal{A}, \mathcal{B}) \leq \sigma(\mathcal{A}, \mathcal{C})$ for all $\mathcal{A} \in FS(X)$;

(IG4) If $\mathcal{B} \leq \mathcal{C}$, then $\sigma(\mathcal{C}, \mathcal{A}) \leq \sigma(\mathcal{B}, \mathcal{A})$ for all $\mathcal{A} \in FS(X)$;

(IG5) $\sigma(\mathcal{A}, \mathcal{B}) = \sigma(\pi(\mathcal{A}), \pi(\mathcal{B}))$, considering π a permutation of the elements of X and denoting $\pi(\mathcal{A})$ and $\pi(\mathcal{B})$ the sets in which the membership degrees are permuted by π, i.e., $\pi(\mathcal{A})(x) = \mathcal{A}(\pi(x))$;

(IG6) $\sigma(\mathcal{A}, \mathcal{B}) = \sigma(\mathcal{B}_c, \mathcal{A}_c)$, where c is a strong negation;

(IG7) $\sigma(\mathcal{B} \vee \mathcal{C}, \mathcal{A}) = \min(\sigma(\mathcal{B}, \mathcal{A}), \sigma(\mathcal{C}, \mathcal{A}))$ for all $\mathcal{A}, \mathcal{B}, \mathcal{C} \in FS(X)$;

(IG8) $\sigma(\mathcal{A}, \mathcal{B} \wedge \mathcal{C}) = \min(\sigma(\mathcal{A}, \mathcal{B}), \sigma(\mathcal{A}, \mathcal{C}))$ for all $\mathcal{A}, \mathcal{B}, \mathcal{C} \in FS(X)$;

[1] A fuzzy set $\mathcal{A}$ in a universe X is included in another fuzzy set $\mathcal{B}$ of the same universe in Zadeh's sense if and only if $\mathcal{A}(x) \leq \mathcal{B}(x)$ for every $x \in X$.

(IG9) $\sigma(\mathcal{A}, \mathcal{B} \vee \mathcal{C}) \geq \max(\sigma(\mathcal{A}, \mathcal{B}), \sigma(\mathcal{A}, \mathcal{C}))$ for all $\mathcal{A}, \mathcal{B}, \mathcal{C} \in FS(X)$.

Subsequently, Burillo et al. [15] proved (IG3) and (IG9) to be equivalent.

Example 2.8. Let $X \neq \emptyset$ be finite. The function $\sigma : FS(X) \times FS(X) \to [0,1]$ given by $\sigma(\mathcal{A}, \mathcal{B}) = \inf_x \{I_L(\mathcal{A}(x), \mathcal{B}(x))\}$ is an inclusion grade in the sense of Sinha and Dougherty.

Let us recall the definition of a n-ary aggregation function, i.e., an aggregation function with n arguments.

Definition 2.9. A n-ary aggregation function is a mapping $f : [0,1]^n \to [0,1]$ such that

(i) $f(0, \ldots, 0) = 0$,

(ii) $f(1, \ldots, 1) = 1$, and

(iii) f is increasing with respect to each component.

3. Matrix resemblance functions

3.1. Definition

Neighbourhoods of pixels can be represented as square matrices, so in order to compare them, matrix comparison techniques are needed. Hence the following definition.

Definition 3.1. A function $\Psi : \mathcal{M}_n([0,1]) \times \mathcal{M}_n([0,1]) \to [0,1]$ is called a matrix resemblance function if it satisfies the following properties:

(MRF1) $\Psi(A, B) = 1$ if and only if $A = B$;

(MRF2) $\Psi(A, B) = 0$ if and only if there exist i and j such that $\{a_{ij}, b_{ij}\} = \{0, 1\}$;

(MRF3) $\Psi(A, B) = \Psi(B, A)$ for all $A, B \in \mathcal{M}_n([0,1])$.

Example 3.2. The function $\Psi(A, B) = \prod_{\substack{i=1 \\ j=1}}^{n} \left(1 - (a_{ij} - b_{ij})^2\right)$ is a matrix resemblance function.

The first and third conditions of the definition of matrix resemblance functions are readily justified as they are natural for a matrix comparison operator. The second property is based on the erosion operator of mathematical morphology. In Section 5 we present the relation between matrix resemblance functions and the erosion operator from fuzzy mathematical morphology.

It sometimes is useful for matrix resemblance functions to have some sort of monotonicity and hence we could add a fourth condition to Definition 3.1:

(MRF4) If $A \leq B \leq C$, then $\Psi(A, C) \leq \Psi(A, B)$ and $\Psi(A, C) \leq \Psi(B, C)$,

where $A \leq B \leq C$ means $a_{ij} \leq b_{ij} \leq c_{ij}$ for all $i, j \in \{1, \ldots, n\}$.
In Section 6.2 we study the effect of this property to matrix resemblance functions. However, we have decided to leave this property out of the axiomatization since our intention is to generalise this concept to be able to compare other non-ordered structures.

Given an arbitrary matrix resemblance function, it is possible to construct another using two additional functions as in the next proposition.

Proposition 3.3. *Let $\phi, \eta : [0,1] \to [0,1]$ be two functions such that $\phi(0) = \eta(0) = 0$, $\phi(1) = \eta(1) = 1$, $\eta(x) \in (0,1)$ for all $x \in (0,1)$, and ϕ is injective. If $\Psi : \mathcal{M}_n([0,1])^2 \to [0,1]$ is a matrix resemblance function, then the mapping $\Psi_{\phi,\eta} : \mathcal{M}_n([0,1])^2 \to [0,1]$ given by*

$$\Psi_{\phi,\eta}(A,B) = \eta\left(\Psi(\phi(A),\phi(B))\right),$$

where $\phi(A)_{ij} = \phi(a_{ij})$, is a matrix resemblance function.

Proof.

(MRF1) $\Psi_{\phi,\eta}(A,B) = \eta\left(\Psi(\phi(A),\phi(B))\right) = 1$ if and only if $\Psi(\phi(A),\phi(B)) = 1$ since $\eta(x) = 1$ only if $x = 1$. By the definition of Ψ, $\Psi(\phi(A),\phi(B)) = 1$ if and only if $\phi(A) = \phi(B)$, which holds if and only if $A = B$, since ϕ is injective.

(MRF2) Since $\eta(x) = 0$ only if $x = 0$, $\Psi_{\phi,\eta}(A,B) = 0$ if and only if $\Psi(\phi(A),\phi(B)) = 0$. This happens if and only if there exist i and j such that $\{\phi(a_{ij}), \phi(b_{ij})\} = \{0,1\}$. Since ϕ is injective, $\{\phi(a_{ij}), \phi(b_{ij})\} = \{0,1\}$ if and only if $\{a_{ij}, b_{ij}\} = \{0,1\}$.

(MRF3) $\Psi_{\phi,\eta}(A,B) = \eta\left(\Psi(\phi(A),\phi(B))\right) = \eta\left(\Psi(\phi(B),\phi(A))\right) = \Psi_{\phi,\eta}(B,A)$.

□

Corollary 3.4. *Let $\phi : [0,1] \to [0,1]$ be an automorphism, i.e., a continuous strictly increasing function such that $\phi(0) = 0$ and $\phi(1) = 1$, and let Ψ be a matrix resemblance function, then the function $\Psi_\phi = \Psi_{\phi,\phi^{-1}}$ is a matrix resemblance function.*

Example 3.5. Consider the matrix resemblance function Ψ as in Example 3.2 and let ϕ be the automorphism given by $\phi(x) = x^2$. Thus, $\Psi(A,B) = \sqrt{\prod_{\substack{i=1\\j=1}}^{n} \left(1 - (a_{ij}^2 - b_{ij}^2)^2\right)}$ is a matrix resemblance function.

In the same vein, given a set of m matrix resemblance functions, it is possible to obtain another by aggregating them as in the next proposition.

Proposition 3.6. *Let $m \geq 2$ and $F : [0,1]^m \to [0,1]$ be an aggregation function with neither zero divisors, nor one divisors. If $\Psi_1, \ldots, \Psi_m$ are matrix resemblance functions, then the mapping $\Psi = F(\Psi_1, \ldots, \Psi_m) : \mathcal{M}_n([0,1])^2 \to [0,1]$ given by*

$$\Psi(A,B) = F(\Psi_1, \ldots, \Psi_m)(A,B) = F(\Psi_1(A,B), \ldots, \Psi_m(A,B)),$$

is a matrix resemblance function.

Proof.

(MRF1) $\Psi(A,B) = F(\Psi_1(A,B), \ldots, \Psi_m(A,B)) = 1$ if and only if there exists $i \in \{1, \ldots, m\}$ such that $\Psi_i(A,B) = 1$, since F has not one divisors. Without loss of generality, let us suppose that $\Psi_1(A,B) = 1$, which, by (MRF1), is equivalent to $A = B$.

(MRF2) $\Psi(A,B) = F(\Psi_1(A,B), \ldots, \Psi_m(A,B)) = 0$ if and only if there exists $i \in \{1, \ldots, m\}$ such that $\Psi_i(A,B) = 0$, due to the lack of zero divisors of F. In either case, by (MRF2), the former holds if and only if there exist $k, j \in \{1, \ldots, n\}$ such that $\{a_{kj}, b_{kj}\} = \{0,1\}$.

(MRF3) $\Psi(A,B) = F(\Psi_1(A,B),\dots,\Psi_m(A,B)) = F(\Psi_1(B,A),\dots,\Psi_m(B,A)) = \Psi(B,A)$.

□

Remark 3.7. In particular, averaging aggregation functions ([16]) do not have either one divisors or zero divisors. Therefore aggregating m matrix resemblance functions with an averaging aggregation function produces a matrix resemblance function. As a note, we recall that the monotonicity of aggregation functions implies that the averaging behaviour is equivalent to idempotency.

Example 3.8. The arithmetic mean of the two matrix resemblance functions from Example 3.2 and Example 3.5 is a matrix resemblance function.

The next result provides information about the structure of the set $\mathcal{F}_n$ of matrix resemblance functions for a fixed n. Let us define an order $\leq$ over the set $\mathcal{F}_n$ by $\Psi_1 \leq \Psi_2$ if $\Psi_1(A,B) \leq \Psi_2(A,B)$ for all $A, B \in \mathcal{M}_n([0,1])$. This is a partial order as it is induced from the order of $[0,1]$.

Proposition 3.9. *Let $n \in \mathbb{N}$. $(\mathcal{F}_n, \leq)$ is a non-complete lattice with neither a maximal nor a minimal element.*

Proof. The set of matrix resemblance functions for a fixed n is a partially ordered set with $\leq$ and given $\Psi_1, \Psi_2 \in \mathcal{F}_n$ we can define the operations $\sqcup$ and $\sqcap$ by:

$$(\Psi_1 \sqcup \Psi_2)(A,B) = \max(\Psi_1,\Psi_2)(A,B) = \max\left(\Psi_1(A,B),\Psi_2(A,B)\right),$$
$$(\Psi_1 \sqcap \Psi_2)(A,B) = \min(\Psi_1,\Psi_2)(A,B) = \min\left(\Psi_1(A,B),\Psi_2(A,B)\right),$$

for all $A, B \in \mathcal{M}_n([0,1])$. Now, $\Psi_1 \sqcup \Psi_2, \Psi_1 \sqcap \Psi_2 \in \mathcal{F}_n$. Indeed,

(MRF1)

$$\begin{aligned}(\Psi_1 \sqcup \Psi_2)(A,B) = 1 &\iff \max\left(\Psi_1(A,B),\Psi_2(A,B)\right) = 1 \\ &\iff \Psi_1(A,B) = 1 \text{ or } \Psi_2(A,B) = 1 \iff A = B.\end{aligned}$$

(MRF2)

$$\begin{aligned}(\Psi_1 \sqcup \Psi_2)(A,B) = 0 &\iff \max\left(\Psi_1(A,B),\Psi_2(A,B)\right) = 0 \\ &\iff \Psi_1(A,B) = 0 \text{ and } \Psi_2(A,B) = 0 \\ &\iff \text{there exist } i \text{ and } j \text{ such that } \{a_{ij}, b_{ij}\} = \{0,1\}.\end{aligned}$$

(MRF3)

$$\begin{aligned}(\Psi_1 \sqcup \Psi_2)(A,B) &= \max\left(\Psi_1(A,B),\Psi_2(A,B)\right) \\ &= \max\left(\Psi_1(B,A),\Psi_2(B,A)\right) = (\Psi_1 \sqcup \Psi_2)(B,A).\end{aligned}$$

The case of $\Psi_1 \sqcap \Psi_2$ is analogous.

Furthermore, the supremum and the infimum of all matrix resemblance functions are, respectively,

$$\Psi_{sup}(A,B) = \begin{cases} 0 & \text{if } \exists\, i,j \text{ s.t. } \{a_{ij}, b_{ij}\} = \{0,1\}, \\ 1 & \text{otherwise,} \end{cases}$$

and

$$\Psi_{inf}(A,B) = \begin{cases} 1 & \text{if } A = B, \\ 0 & \text{otherwise,} \end{cases}$$

and since neither is a matrix resemblance function, $(\mathcal{F}_n, \leq)$ is a non-complete lattice. □

4. Constructions

Once the definition of matrix resemblance functions is set, we need an algebraic expression to work with them and study their properties. In this paper we provide two construction methods, the first one being based on restricted equivalence functions and the second one on inclusion grades.

Both methods make use of a function $F : [0,1]^N \to [0,1]$ that satisfies a set of specific properties. We enlist these properties and refer to them as (F1), (F2) and (F3).

(F1) $F(x_1, \ldots, x_N) = 1$ if and only if $x_i = 1$ for every $1 \leq i \leq N$,

(F2) $F(x_1, \ldots, x_N) = 0$ if and only if there exists $1 \leq j \leq N$ such that $x_j = 0$,

(F3) $F\left((x_i)_{i=1}^N\right) = F\left((x_{\pi(i)})_{i=1}^N\right)$ for every permutation π of $\{1, \ldots, N\}$.

The first two are the properties proposed in [17] for aggregation operators.

Example 4.1. The minimum, the product or the geometric mean are well-known examples of such a function F.

Remark 4.2. Observe that although all functions in the previous example are in fact aggregation functions, we do not require F to be monotone and hence it need not be an aggregation function. The case in which F is an aggregation function and other particular instances of F are further studied in Section 7.

Example 4.3. Let $F : [0,1]^2 \to [0,1]$ be such that

$$F(x,y) = \begin{cases} 1, & \text{if } x = y = 1; \\ 0, & \text{if } xy = 0; \\ 0.2, & \text{if } x = y = 0.9; \\ 0.5 & \text{otherwise.} \end{cases}$$

Thus, F verifies the conditions (F1)-(F3), but is not an aggregation function since it is not increasing.

4.1. First construction method

The next theorem constitutes the first construction method for matrix resemblance functions that we present in this work.

Theorem 4.4. *Let β be a function satisfying (REF1)-(REF3) and $H : [0,1]^{n^2} \to [0,1]$ a function satisfying (F1) and (F2), then the mapping $\Psi : \mathcal{M}_n([0,1])^2 \to [0,1]$ given by*

$$\Psi(A,B) = \underset{\substack{i=1\\j=1}}{\overset{n}{H}} (\beta(a_{ij}, b_{ij})), \tag{1}$$

where $\underset{\substack{i=1\\j=1}}{\overset{n}{H}} (\beta(a_{ij}, b_{ij})) = H(\beta(a_{11}, b_{11}), \ldots, \beta(a_{nn}, b_{nn}))$, is a matrix resemblance function.

Proof.

(MRF1)

$$\Psi(A,B) = \underset{\substack{i=1\\j=1}}{\overset{n}{H}} (\beta(a_{ij}, b_{ij})) = 1 \underset{(F1)}{\Longleftrightarrow} \beta(a_{ij}, b_{ij}) = 1 \text{ for all } i,j \underset{(REF2)}{\Longleftrightarrow} a_{ij} = b_{ij} \text{ for all } i,j.$$

(MRF2)

$$\Psi(A,B) = \underset{\substack{i=1\\j=1}}{\overset{n}{H}} (\beta(a_{ij}, b_{ij})) = 0 \underset{(F2)}{\Longleftrightarrow} \exists i,j \text{ s.t. } \beta(a_{ij}, b_{ij}) = 0 \underset{(REF3)}{\Longleftrightarrow} \exists i,j \text{ s.t. } \{a_{ij}, b_{ij}\} = \{0,1\}.$$

(MRF3)

$$\Psi(A,B) = \underset{\substack{i=1\\j=1}}{\overset{n}{H}} (\beta(a_{ij}, b_{ij})) \underset{(REF1)}{=} \underset{\substack{i=1\\j=1}}{\overset{n}{H}} (\beta(b_{ij}, a_{ij})) = \Psi(B,A).$$

□

Remark 4.5. If Ψ is a matrix resemblance function constructed by the pair (β, H) with the previous method, then the matrix resemblance function $\Psi_{\phi,\eta}$ obtained from the application of Proposition 3.3 is generated by $(\beta\phi, \eta \circ H)$, where $\beta\phi(x,y) = \beta(\phi(x), \phi(y))$.

Example 4.6. An example of a matrix resemblance function as in (1) can be found in Example 3.2 considering H the product and $\beta(x,y) = 1 - (x-y)^2$, i.e.,

$$\Psi(A,B) = \prod_{i,j=1}^{n} \left(1 - (a_{ij} - b_{ij})^2\right). \tag{2}$$

If we applied this matrix resemblance function to the matrices

$$A = \begin{pmatrix} 0.1 & 0.9 & 0.7 \\ 0.1 & 0.7 & 0.1 \\ 0.8 & 0.2 & 0.2 \end{pmatrix}, \quad B = \begin{pmatrix} 0.6 & 0.7 & 0.3 \\ 0.3 & 0.6 & 0.7 \\ 0.6 & 0.7 & 0.9 \end{pmatrix},$$

we would obtain $\Psi(A,B) = 0.1351$; and understanding each value as the gray level of a pixel, we can represent the result as in Figure 1.

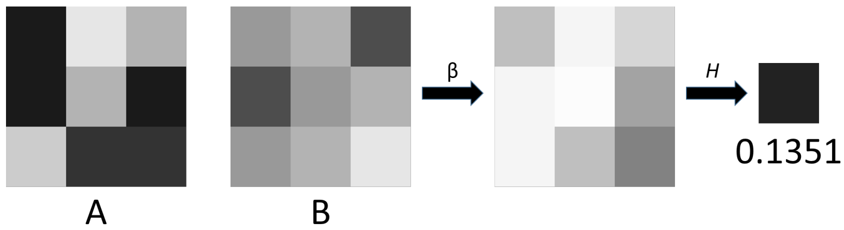


Figure 1: Representation of the application of Ψ (as in (2)).

4.2. Second construction method

The following result provides the second construction method for matrix resemblance functions:

Theorem 4.7. *Let $\sigma : \mathcal{M}_n([0,1])^2 \to [0,1]$ be a function that satisfies (IG1) and (IG2) and let $M : [0,1]^2 \to [0,1]$ be a function satisfying (F1)-(F3), then the mapping $\Psi : \mathcal{M}_n([0,1])^2 \to [0,1]$ given by*

$$\Psi(A,B) = M\left(\sigma(A,B), \sigma(B,A)\right), \tag{3}$$

is a matrix resemblance function.

Proof.

(MRF1)

$$\begin{aligned}\Psi(A,B) = M\left(\sigma(A,B), \sigma(B,A)\right) = 1 &\underset{(F1)}{\Longleftrightarrow} \sigma(A,B) = \sigma(B,A) = 1\\ &\underset{(IG1)}{\Longleftrightarrow} A \leq B \text{ and } B \leq A \Longleftrightarrow A = B.\end{aligned}$$

(MRF2)

$$\begin{aligned}\Psi(A,B) = M\left(\sigma(A,B), \sigma(B,A)\right) = 0 &\underset{(F2)}{\Longleftrightarrow} \sigma(A,B) = 0 \text{ or } \sigma(B,A) = 0\\ &\Longleftrightarrow \exists i,j \text{ such that } a_{ij} = 1 \text{ and } b_{ij} = 0 \text{ or } a_{ij} = 0 \text{ and } b_{ij} = 1\\ &\Longleftrightarrow \exists i,j \text{ s.t. } \{a_{ij}, b_{ij}\} = \{0,1\}.\end{aligned}$$

(MRF3)

$$\Psi(A,B) = M\left(\sigma(A,B), \sigma(B,A)\right) \underset{(F3)}{=} M\left(\sigma(B,A), \sigma(A,B)\right) = \Psi(B,A).$$

□

Example 4.8. Considering $\sigma(A,B) = \inf_{i,j}\{I_L(a_{ij}, b_{ij})\}$ and M the minimum, the function

$$\Psi(A,B) = \min\left(\inf_{i,j}\{I_L(a_{ij}, b_{ij})\}, \inf_{i,j}\{I_L(b_{ij}, a_{ij})\}\right) \tag{4}$$

is a matrix resemblance function as in Theorem 4.7. Moreover, applying this Ψ to the matrices from Example 4.6, we obtain 0.3 (see Figure 2).

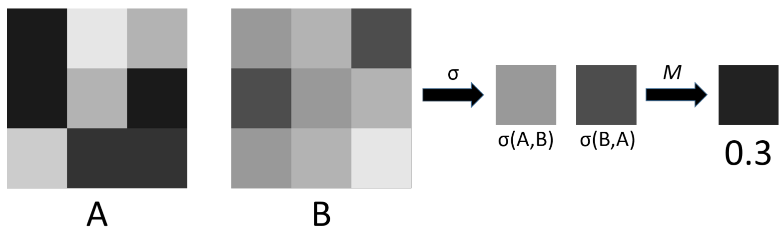


Figure 2: Representation of the application of Ψ (as in (4)).

Remark 4.9. The matrix resemblance function in Example 4.8 can be built using either construction methods. Indeed,

$$\begin{aligned}
\Psi(A,B) &= \min\left(\inf_{i,j}\{I_L(a_{ij}, b_{ij})\}, \inf_{i,j}\{I_L(b_{ij}, a_{ij})\}\right)\\
&= \min\left(\inf_{i,j}\{\min(1, 1-a_{ij}+b_{ij})\}, \inf_{i,j}\{\min(1, 1-b_{ij}+a_{ij})\}\right)\\
&= \inf_{i,j}\{\min(1, 1-a_{ij}+b_{ij}), \min(1, 1-b_{ij}+a_{ij})\}\\
&= \inf_{i,j}\{\min(1, 1-a_{ij}+b_{ij}, 1-b_{ij}+a_{ij})\}\\
&= \inf_{i,j}\{1-|a_{ij}-b_{ij}|\}\\
&= \min_{i,j}\left(1-|a_{ij}-b_{ij}|\right),
\end{aligned}$$

which is a function constructed by the first method using H the minimum and $\beta(x,y) = 1-|x-y|$, the restricted equivalence function from Example 2.4.

4.3. Relation between both constructions

In this section we study the cases where both constructions are equivalent, i.e., whether it is possible to reach the expression given by one of the constructions from the other.

Let us start recalling two theorems that characterize the restricted equivalence functions and the inclusion grades in the sense of Sinha and Dougherty. The first can be found in [6] (Theorem 7).

Theorem 4.10. *A function $REF : [0,1]^2 \to [0,1]$ is a restricted equivalence function if and only if there exists a function $I : [0,1]^2 \to [0,1]$ satisfying (I1), (I8), (I12) and for all $x, y \in [0,1]$:*

$$I(x,y) = 0 \text{ if and only if } x = 1 \text{ and } y = 0, \tag{5}$$

such that

$$REF(x,y) = \min\left(I(x,y), I(y,x)\right). \tag{6}$$

The second theorem, the characterization of inclusion grades, corresponds to Theorem 5.4 of [18].

Theorem 4.11. *Let X be a finite universe. A mapping $\sigma : FS(X) \times FS(X) \to [0,1]$ satisfies all Sinha-Dougherty axioms if and only if there exists a fuzzy implicator I satisfying (I8), (I12) and* (5) *for all $x, y \in [0,1]$, such that for all $\mathcal{A}$ and $\mathcal{B}$ in $FS(X)$:*

$$\sigma(\mathcal{A}, \mathcal{B}) = \inf_{x \in X} I\left(\mathcal{A}(x), \mathcal{B}(x)\right). \tag{7}$$

Note that one of the conditions in Theorem 4.11 is that X is finite, and since we are working with finite dimensional matrices, we are assuming that to be the case. Hence, in the preceding theorem the infimum is actually the minimum.

Proposition 4.12. *Let $I : [0,1]^2 \to [0,1]$ be a function. The requested conditions to the function I in Theorem 4.10 and Theorem 4.11 are equivalent.*

Proof. Let $I : [0,1]^2 \to [0,1]$ be a function that satisfies all the requirements in Theorem 4.10, i.e., (I1), (I8), (I12), and (5). Now, since (I1) is clear, it suffices to show that I satisfies conditions (I2)-(I5).

(I2) Straightforward from (I1) and (I12).

(I4) Straightforward from (I8).

(I3) Straightforward from (I4) and (I12).

(I5) Straightforward from (5).

Some of the preceding dependencies can be found in [19].

The converse is immediate since every implication I satisfies (I1), and the remaining conditions coincide.

□

As a consequence of the aforementioned result, one can see that both constructions are similar in the case of β being a restricted equivalence function and σ an inclusion grade. However, a study on the functions H and M is still required to show when they are actually equivalent. Recall that H is a function with n^2 inputs that satisfies (F1) and (F2), and M is a function with 2 inputs that satisfies the three conditions (F1)-(F3).

By Theorem 4.10, we know that given Ψ constructed by means of a restricted equivalence function, there exists a function I such that:

$$\Psi(A,B) = \underset{\substack{i=1 \\ j=1}}{\overset{n}{H}} \left(\min\left(I(a_{ij}, b_{ij}), I(b_{ij}, a_{ij})\right)\right) \tag{8}$$

Similarly, a matrix resemblance function can also be constructed using inclusion grades, in which case by Theorem 4.11 there exists a function I such that:

$$\Psi(A,B) = M\left(\min_{i,j}(I(a_{ij},b_{ij})), \min_{i,j}(I(b_{ij},a_{ij}))\right). \tag{9}$$

Now, the result in Proposition 4.12 indicates that in the case functions H and M verify that for two sequences $(x_i)_{i=1}^N, (y_i)_{i=1}^N \subset [0,1]^N$ the following equation holds

$$M(\min((x_i)_{i=1}^N), \min((y_i)_{i=1}^N)) = H((\min(x_i,y_i))_{i=1}^N), \tag{10}$$

then both constructions will be equivalent. In the next theorem we show an instance of such functions H and M.

Theorem 4.13. *Let H be the function minimum for n^2 arguments and M the function minimum for two arguments. Then the first construction with a restricted equivalence function, as in* (8)*, and the second with an inclusion grade , as in* (9)*, are equivalent.*

Proof. Since such functions H and M satisfy (10), given a matrix resemblance function as in (8) we can obtain the same as in (9). The converse is analogous. □

But there are other examples of H and M that do not satisfy (10) and hence they produce different matrix resemblance functions for each construction:

Example 4.14. Let H be the geometric mean for N arguments, M the geometric mean for 2 arguments, $(x_i)_{i=1}^N = (0.4)_{i=1}^N$ and $(y_i)_{i=1}^N = (0.7)_{i=1}^N$. Thus,

$$H((\min(x_i,y_i))_{i=1}^N) = H((0.4)_{i=1}^N) = 0.4$$
$$M(\min((x_i)_{i=1}^N), \min((y_i)_{i=1}^N)) = M(0.4, 0.7) = \sqrt{0.28} \neq 0.4$$

Theorem 4.13 ensures that if H and M are the minimum (with the corresponding arity) and a matrix resemblance function is constructed in terms of a restricted equivalence function, then it can also be constructed in terms of an inclusion grade, and viceversa.

The following two results expose that the minimum is the only idempotent function that satisfies (F1)-(F3) and (10). Recall that a function $F : [0,1]^N \to [0,1]$ is said to be idempotent if $F(x,\ldots,x) = x$ for all $x \in [0,1]$.

Theorem 4.15. *Let $H : [0,1]^N \to [0,1]$ be a function satisfying (F1) and (F2), and let $M : [0,1]^2 \to [0,1]$ be a function that satisfies (F1)-(F3). Let $d : [0,1] \to [0,1]$ be the diagonal section of H, i.e., $d(x) = H(x,\ldots,x)$. If H and M satisfy* (10)*, then $M(x,y) = d(\min(x,y))$.*

Proof. Let H be a function that satisfies (F1), (F2) and M a function that satisfies (F1)-(F3), such that (10) is satisfied. Let $x_0, y_0 \in [0,1]$ such that $x_0 \leq y_0$. Then,

$$H((\min(x_0,y_0))_{i=1}^N) = H((x_0)_{i=1}^N) = d(x_0),$$
$$M(\min((x_0)_{i=1}^N), \min((y_0)_{i=1}^N)) = M(x_0,y_0),$$

and, since H and M satisfy (10), $M(x_0,y_0) = d(x_0)$.

Thus, since x_0 and y_0 are arbitrarily taken, for every $x,y \in [0,1]$ such that $x \leq y$, it holds that $M(x,y) = d(x)$. Hence $M(x,y) = d(\min(x,y))$. □

Theorem 4.16. *Let $H : [0,1]^N \to [0,1]$ be an idempotent function satisfying (F1) and (F2), and let $M : [0,1]^2 \to [0,1]$ be a function that satisfies (F1)-(F3). Then H and M satisfy* (10) *if and only if H and M are the N-ary and 2-ary minimum, respectively.*

Proof. Let H be an idempotent function that satisfies (F1), (F2) and M a function that satisfies (F1)-(F3), such that (10) is satisfied.

By Theorem 4.15, it holds that $M(x,y) = d\,(\min(x,y))$, where d is the diagonal section of H. Since H is idempotent, it holds that $d(x) = H(x,\ldots,x) = x$. Hence M coincides with the 2-ary minimum.

Now, by (10), for any two sequences $(x_i)_{i=1}^N, (y_i)_{i=1}^N \subset [0,1]^N$, the value of $H((\min(x_i,y_i))_{i=1}^N)$ must coincide with the minimum of all the inputs $\{x_1,\ldots,x_N,y_1,\ldots,y_N\}$, i.e.,

$$H((\min(x_i,y_i))_{i=1}^N) = \min\left(x_1,\ldots,x_N,y_1,\ldots,y_N\right). \tag{11}$$

Assume that H is not the N-ary minimum, then there exists a sequence $(\lambda_i)_{i=1}^N \subset [0,1]^N$ such that $H(\lambda_1,\ldots,\lambda_N) \neq \min(\lambda_1,\ldots,\lambda_N)$. Thus, if we take $(\gamma_i)_{i=1}^N \subset [0,1]^N$ such that $\min(\lambda_i,\gamma_i) = \lambda_i$ for $1 \leq i \leq N$ we reach that

$$H((\min(\lambda_i,\gamma_i))_{i=1}^N) \neq \min(\lambda_1,\ldots,\lambda_N) = \min\left(\lambda_1,\ldots,\lambda_N,\gamma_1,\ldots,\gamma_N\right),$$

which contradicts (11).

The converse implication is immediate. □

5. Matrix resemblance functions and fuzzy Mathematical Morphology

Mathematical morphology is a theory for processing images based on their form and structure. It was originally developed for binary images [20, 21], but later on it was generalized as fuzzy mathematical morphology for grayscale images [22–25].

In this section we study the relation between the class of matrix resemblance functions and the theory of fuzzy mathematical morphology. This theory has four main operations to transform an image via an structuring element: dilation, erosion, opening and closing. The mentioned relation comes from the second construction method for matrix resemblance functions, as, under certain assumptions, inclusion grades in the sense of Sinha and Dougherty are related to the erosion operator from fuzzy mathematical morphology.

Generally, the erosion operator ε is defined as an operator that commutes with the infimum, i.e.,

$$\varepsilon(\inf Y) = \inf_{y\in Y} \varepsilon(y).$$

In particular, given X a finite universe and $\mathcal{A},\mathcal{B} \in FS(X)$, an expression of erosion operator with respect to an structuring element $\mathcal{B}$, $\varepsilon(\mathcal{A},\mathcal{B}) \in FS(X)$, is given by $\varepsilon(\mathcal{A},\mathcal{B})(z) = \inf_{x\in X}\{I_L(\mathcal{B}(x-z),\mathcal{A}(x))\}$, as in [23, 24]. In this case, the fuzzy erosion operator coincides with an inclusion grade in the sense of Sinha and Dougherty, applied to a set translated by z, i.e., $\varepsilon(\mathcal{A},\mathcal{B})(z) = \sigma(\mathcal{B}_z,\mathcal{A})$.

This can be translated to the framework of matrix resemblance functions. Let $k \in \mathbb{N}$ and let

$A, B \in \mathcal{M}_{2k+1}([0,1])$, if we consider the following indexation for the elements of a matrix:

$$\begin{pmatrix} (-k,-k) & \dots & (-k,0) & \dots & (-k,k) \\ \vdots & & \vdots & & \vdots \\ (0,-k) & \dots & \mathbf{(0,0)} & \dots & (0,k) \\ \vdots & & \vdots & & \vdots \\ (k,-k) & \dots & (k,0) & \dots & (k,k) \end{pmatrix},$$

$(0,0)$ refers to the central element of the matrix (or central pixel of the neighbourhood) and, thus, we get

$$\Psi(A,B) = M(\sigma(A,B), \sigma(B,A)) = M(\varepsilon(B,A)((0,0)), \varepsilon(A,B)((0,0))),$$

which relates the fuzzy erosion operator with matrix resemblance functions.

6. Some properties of matrix resemblance functions

There exist some properties that are expected for comparison measures to satisfy (see [2, 3, 6]). Bustince et al. [3] proposed a set of properties that should be met by any global comparison measure for images. Some of these properties are straightforward from Definition 3.1. Namely, comparison measures are normally asked for *symmetry*, i.e., the difference between two images ought not to depend on the order in which they are compared. Matrix resemblance functions satisfy this condition due to (MRF3). Another property is that a comparison measure should yield that the images are equal if and only if they are exactly equal pixel-wise, which happens for matrix resemblance functions because of (MRF1). This last condition is stronger than the property *reflexivity* in [3]. Additionally, it is often required that the comparison measure between a binary image (in black and white) and its complement is 0, and (MRF2) ensures that.

This subsection goes over some of these usually demanded features and studies in which cases matrix resemblance functions fulfil them.

6.1. Invariance under permutation

As it is mentioned in [7], a permutation of the inputs can be used for modelling certain domain transformations; such as shifts, rotations and reflections. Thus, since a function that measures the similarity between two images ought to provide the same results when comparing images which have been transformed by any of the aforementioned operators, we study the conditions under which a matrix resemblance function Ψ is invariant under permutation.

We say that a matrix resemblance function Ψ is invariant under permutation if $\Psi(A,B) = \Psi(\pi(A), \pi(B))$, for all $A, B \in \mathcal{M}_n([0,1])$ and all permutations π of the set of indices.

Invariance under such permutation would mean that, as far as it concerns to the result of the comparison, it is the same to compare two images or their transformations; either their shifts, their rotations, or their reflections.

In the case of matrix resemblance functions constructed as in the first method, we reach the following result:

Lemma 6.1. *Let $n \in \mathbb{N}$ and $\lambda_1, \dots, \lambda_n \in [0,1]$. There exists a function $\beta : [0,1]^2 \to [0,1]$ satisfying (REF1)-(REF3) such that for all $1 \leq i \leq n$, there exist $x, y \in [0,1]^2$ such that $\beta(x,y) = \lambda_i$.*

Proof. The cases of $\lambda_i = 0$ and $\lambda_i = 1$ for some i are trivial, since $\beta(0,1) = 0$ and $\beta(x_0, x_0) = 1$ for any $x_0 \in [0,1]$.

Let us suppose that for all $1 \leq i \leq n$, $0 \neq \lambda_i \neq 1$. Thus, we can define a function β as follows:

$$\beta(x,y) = \begin{cases} 0, & \text{if } \{x,y\} = \{0,1\}, \\ 1, & \text{if } x = y, \\ \lambda_1, & \text{if } \{x,y\} = \{\frac{1}{n}, 1\}, \\ \lambda_2, & \text{if } \{x,y\} = \{\frac{2}{n}, 1\}, \\ \vdots & \vdots \quad \vdots \\ \lambda_{n-1}, & \text{if } \{x,y\} = \{\frac{n-1}{n}, 1\}, \\ \lambda_n, & \text{otherwise.} \end{cases}$$

□

Theorem 6.2. *Let*

$$\Psi(A,B) = \underset{\substack{i=1\\j=1}}{\overset{n}{H}} (\beta(a_{ij}, b_{ij}))$$

be a matrix resemblance function as in (1)*. Then* Ψ *is invariant under permutation if and only if* H *satisfies (F3).*

Proof. Let β be a function as in (1) and let us assume Ψ is invariant under permutation, i.e., $\Psi(A,B) = \Psi(\pi(A), \pi(B))$, for all $A, B \in \mathcal{M}_n([0,1])$ and all permutations π.

Now, let us suppose there exists a permutation τ of $\{1, \ldots, n^2\}$ such that

$$H(x_1, \ldots, x_{n^2}) \neq H(x_{\tau(1)}, \ldots, x_{\tau(n^2)}).$$

By Lemma 6.1 every x_i is an image of some function β. Thus, we can choose matrices A and B to be such that

$$\begin{aligned} \beta(a_{11}, b_{11}) &= x_1, \\ \beta(a_{12}, b_{12}) &= x_2, \\ &\vdots \\ \beta(a_{nn}, b_{nn}) &= x_{n^2}. \end{aligned}$$

Thus, we reach the following:

$$\begin{aligned} \Psi(A,B) &= \underset{\substack{i=1\\j=1}}{\overset{n}{H}} (\beta(a_{ij}, b_{ij})) \\ &= \overset{n^2}{\underset{i=1}{H}} (x_1, \ldots, x_{n^2}) \\ &\neq \overset{n^2}{\underset{i=1}{H}} (x_{\tau(1)}, \ldots, x_{\tau(n^2)}) \\ &= \Psi(\tau(A), \tau(B)), \end{aligned}$$

which contradicts the fact that Ψ is invariant under permutation. Hence H satisfies (F3).

The converse implication is straightforward.

□

In the case of constructing Ψ with the second construction method, we obtain the next result.

Proposition 6.3. *Let* Ψ *be a matrix resemblance function*

$$\Psi(A,B) = M\left(\sigma(A,B), \sigma(B,A)\right),$$

as in (3). *If* σ *satisfies (IG5), then* Ψ *is invariant under permutation.*

Proof. Let Ψ be as in (3) and σ satisfying (IG5). Then

$$\Psi(A,B) = M\left(\sigma(A,B), \sigma(B,A)\right) \underset{(IG5)}{=} M\left(\sigma(\pi(A),\pi(B)), \sigma(\pi(B),\pi(A))\right) = \Psi(\pi(A),\pi(B)),$$

for all $A, B \in \mathcal{M}_n([0,1])$ and all permutations π. □

Nevertheless, the converse does not hold. For instance, let us consider the function M defined as:

$$M(x,y) = \begin{cases} 1, & \text{if } x = y = 1; \\ 0, & \text{if } xy = 0; \\ 0.5 & \text{otherwise,} \end{cases}$$

and let σ be:

$$\sigma(A,B) = \begin{cases} 0, & \text{if } \exists 1 \leq i,j \leq n \text{ s.t. } a_{ij} = 0 \text{ and } b_{ij} = 1; \\ 1, & \text{if } A \leq B \text{ in Zadeh's sense;} \\ b_{11} - a_{11} & \text{otherwise.} \end{cases}$$

Then, σ does not satisfy (IG5) and yet Ψ is invariant under permutation. Indeed, let $A, B \in \mathcal{M}_n([0,1])$, we have three different cases:

- The case where $\Psi(A,B) = 1$, which implies that $A = B$ and therefore $\pi(A) = \pi(B)$ for all π. Hence $\Psi(A,B) = \Psi(\pi(A),\pi(B))$.
- The case where $\Psi(A,B) = 0$. We can assume $\sigma(A,B) = 0$, then for all permutations π it holds that $\sigma(\pi(A),\pi(B)) = 0$ and hence

$$\Psi(A,B) = M\left(\sigma(A,B), \sigma(B,A)\right) = 0 = M\left(\sigma(\pi(A),\pi(B)), \sigma(\pi(B),\pi(A))\right) = \Psi(\pi(A),\pi(B)).$$

- The case where $0 < \Psi(A,B) < 1$. In this situation, it holds that $0 < \sigma(A,B)\sigma(B,A) < 1$, which means that $0 < \sigma(\pi(A),\pi(B))\sigma(\pi(B),\pi(A)) < 1$ and hence

$$\Psi(A,B) = M\left(\sigma(A,B), \sigma(B,A)\right) = 0.5 = M\left(\sigma(\pi(A),\pi(B)), \sigma(\pi(B),\pi(A))\right) = \Psi(\pi(A),\pi(B)).$$

Clearly, Ψ is invariant under any permutation but σ does not satisfy (IG5).

6.2. Monotonicity

It is natural to ask that a comparison measure's result decreases when comparing an image with another that is darker and darker (or clearer). Similarly, the result should be higher when we compare an image with another that is more akin to itself. In the case of matrix resemblance functions, that monotonicity property is represented by (MRF4). The property *Reaction to lightening and darkening* from [3] is a consequence of this property. We present here some conditions to ensure monotonicity for each construction.

Proposition 6.4. *Consider*

$$\Psi(A,B) = \underset{\substack{i=1\\j=1}}{\overset{n}{H}} (\beta(a_{ij}, b_{ij}))$$

as in (1) *with H increasing. If β satisfies (REF5), then Ψ satisfies (MRF4). Moreover, if H is strictly increasing in* $(0,1]^{n^2}$*, then the converse holds.*

Proof. Let H be increasing and suppose that β satisfies (REF5), i.e., for all $x, y, z \in [0,1]$, if $x \le y \le z$, then $\beta(x,y) \ge \beta(x,z)$ and $\beta(y,z) \ge \beta(x,z)$.

Consider $A, B, C \in \mathcal{M}_n([0,1])$ such that $A \le B \le C$. Since $a_{ij} \le b_{ij} \le c_{ij}$ for all i, j, then $\beta(a_{ij}, c_{ij}) \le \beta(a_{ij}, b_{ij})$ and since H is increasing,

$$\Psi(A,C) = \underset{\substack{i=1\\j=1}}{\overset{n}{H}} (\beta(a_{ij}, c_{ij})) \le \underset{\substack{i=1\\j=1}}{\overset{n}{H}} (\beta(a_{ij}, b_{ij})) = \Psi(A,B).$$

Similarly, it holds that $\beta(a_{ij}, c_{ij}) \le \beta(b_{ij}, c_{ij})$ and thus,

$$\Psi(A,C) = \underset{\substack{i=1\\j=1}}{\overset{n}{H}} (\beta(a_{ij}, c_{ij})) \le \underset{\substack{i=1\\j=1}}{\overset{n}{H}} (\beta(b_{ij}, c_{ij})) = \Psi(B,C).$$

Now, for H strictly increasing and Ψ satisfying (MRF4), suppose that there exist $x \le y \le z$ such that $\beta(x,y) < \beta(x,z)$ or $\beta(y,z) < \beta(x,z)$.

Thus, consider the constant matrix A with x in all its entries, B with y in all its entries and C with z in all its entries. Clearly $A \le B \le C$, but

$$\Psi(A,C) = \underset{\substack{i=1\\j=1}}{\overset{n}{H}} (\beta(x,z)) > \underset{\substack{i=1\\j=1}}{\overset{n}{H}} (\beta(x,y)) = \Psi(A,B),$$

or

$$\Psi(A,C) = \underset{\substack{i=1\\j=1}}{\overset{n}{H}} (\beta(x,z)) > \underset{\substack{i=1\\j=1}}{\overset{n}{H}} (\beta(y,z)) = \Psi(B,C),$$

which contradicts (MRF4). Hence β satisfies (REF5). □

In particular, Ψ satisfies the monotonicity condition when we consider H to be an aggregation function that fulfills (F1), (F2) and we take β a restricted equivalence function.

The following result regards to the second construction method.

Theorem 6.5. *Let Ψ be such that*

$$\Psi(A,B) = M\left(\sigma(A,B), \sigma(B,A)\right),$$

as in (3)*, with M increasing. If σ satisfies, for all $A, B, C \in \mathcal{M}_n([0,1])$ such that $A \leq B \leq C$, the following conditions:*

(a) $\sigma(C, A) \leq \sigma(C, B)$, and

(b) $\sigma(C, A) \leq \sigma(B, A)$;

then Ψ satisfies (MRF4). Besides, if $M(1, x) = x$, the converse also holds.

Proof. Consider $A, B, C \in \mathcal{M}_n([0,1])$ such that $A \leq B \leq C$. Thus, since M is increasing,

$$\begin{aligned}\Psi(A, C) &= M\left(\sigma(A, C), \sigma(C, A)\right) \underset{(IG1)}{=} M\left(1, \sigma(C, A)\right)\\ &\underset{(b)}{\leq} M\left(1, \sigma(B, A)\right) \underset{(IG1)}{=} M\left(\sigma(A, B), \sigma(B, A)\right) = \Psi(A, B),\end{aligned}$$

and

$$\begin{aligned}\Psi(A, C) &= M\left(\sigma(A, C), \sigma(C, A)\right) \underset{(IG1)}{=} M\left(1, \sigma(C, A)\right)\\ &\underset{(a)}{\leq} M\left(1, \sigma(C, B)\right) \underset{(IG1)}{=} M\left(\sigma(B, C), \sigma(C, B)\right) = \Psi(B, C).\end{aligned}$$

Now, suppose that Ψ satisfies (MRF4) and $M(1, x) = x$. Let $A, B, C \in \mathcal{M}_n([0,1])$ such that $A \leq B \leq C$. Then

$$\begin{aligned}\sigma(C, A) &= M\left(1, \sigma(C, A)\right) \underset{(IG1)}{=} M\left(\sigma(A, C), \sigma(C, A)\right) = \Psi(A, C)\\ &\underset{(MRF4)}{\leq} \Psi(B, C) = M\left(\sigma(B, C), \sigma(C, B)\right) \underset{(IG1)}{=} M\left(1, \sigma(C, B)\right) = \sigma(C, B),\end{aligned}$$

and

$$\begin{aligned}\sigma(C, A) &= M\left(1, \sigma(C, A)\right) \underset{(IG1)}{=} M\left(\sigma(A, C), \sigma(C, A)\right) = \Psi(A, C)\\ &\underset{(MRF4)}{\leq} \Psi(A, B) = M\left(\sigma(A, B), \sigma(B, A)\right) \underset{(IG1)}{=} M\left(1, \sigma(B, A)\right) = \sigma(B, A).\end{aligned}$$

□

Remark 6.6. If in the construction of Ψ as in Theorem 4.7 we consider M to be a t-norm, we can assure that Ψ satisfies (MRF4) if and only if σ satisfies (a) and (b).

Corollary 6.7. *Let Ψ be as in* (3) *with M increasing. If σ satisfies (IG3) and (IG4), then Ψ satisfies (MRF4).*

Proof. If σ satisfies (IG3) and (IG4), then it satisfies (a) and (b) and we are under the conditions of Theorem 6.5. □

6.3. Comparing the complements: $\Psi(A, B) = \Psi(A_c, B_c)$

Some comparison measures between two images are required to produce the same result when applied to their c-complements (see Property 4 in [3]). In the case of matrix resemblance functions, this translates into studying under what conditions we get the following equality:

$$\Psi(A,B) = \Psi(A_c, B_c),$$

where c is a strong negation and A_c and B_c denote the matrices $(c(a_{ij}))_{i,j=1}^{n}$ and $(c(b_{ij}))_{i,j=1}^{n}$ respectively.

Proposition 6.8. *Let Ψ be a matrix resemblance function as in* (1) *and β satisfy the additional property (REF4). Then it holds that $\Psi(A,B) = \Psi(A_c, B_c)$.*

Proof.

$$\Psi(A_c, B_c) = \underset{\substack{i=1\\j=1}}{\overset{n}{H}} (\beta(c(a_{ij}), c(b_{ij}))) \underset{(REF4)}{=} \underset{\substack{i=1\\j=1}}{\overset{n}{H}} (\beta(a_{ij}, b_{ij})) = \Psi(A,B).$$

□

The converse of the preceding result does not hold in general. Consider, for instance,

$$\beta(x,y) = \begin{cases} 1, & \text{if } x = y; \\ 0, & \text{if } \{x,y\} = \{0,1\}; \\ 0.4, & \text{if } \{x,y\} = \{0.2, 0.3\}; \\ 0.6 & \text{otherwise,} \end{cases} \tag{12}$$

which satisfies (REF1)-(REF3), and

$$H(x_1, \ldots, x_N) = \begin{cases} 1, & \text{if } x_i = 1 \text{ for all } 1 \leq i \leq N; \\ 0, & \text{if } \exists i \text{ s.t. } x_i = 0; \\ 0.5 & \text{otherwise,} \end{cases} \tag{13}$$

which satisfies (F1) and (F2). Therefore, by Theorem 4.4, $\Psi(A,B) = \underset{\substack{i=1\\j=1}}{\overset{n}{H}} (\beta(a_{ij}, b_{ij}))$ is a matrix resemblance function.

Moreover, let us show that Ψ satisfies $\Psi(A,B) = \Psi(A_c, B_c)$ for all $A, B \in \mathcal{M}_n([0,1])$. Indeed, let $A, B \in \mathcal{M}_n([0,1])$. Firstly, it holds that

$$\begin{aligned} \Psi(A,B) = 0 &\iff \text{there exist } i,j \text{ such that } \{a_{ij}, b_{ij}\} = \{0,1\} \\ &\iff \text{there exist } i,j \text{ such that } \{c(a_{ij}), c(b_{ij})\} = \{0,1\} \\ &\iff \Psi(A_c, B_c) = 0. \end{aligned} \tag{14}$$

Secondly, it holds that

$$\Psi(A,B) = 1 \iff A = B \iff A_c = B_c \iff \Psi(A_c, B_c) = 1. \tag{15}$$

Now, due to the definition of function H, there are only three different cases:

- $\Psi(A,B) = 0$: By (14), it holds that $\Psi(A,B) = \Psi(A_c, B_c) = 0$ for all $A, B \in \mathcal{M}_n([0,1])$.
- $\Psi(A,B) = 1$: By (15), it holds that $\Psi(A,B) = \Psi(A_c, B_c) = 1$ for all $A, B \in \mathcal{M}_n([0,1])$.
- $\Psi(A,B) = 0.5$: Since (14) and (15) cover all the situations in which $\Psi(A_c, B_c) = 0$ and $\Psi(A_c, B_c) = 1$, the only possibility is $\Psi(A_c, B_c) = 0.5$.

Consider now the usual negation $c_z(x) = 1 - x$ and the constant matrices $A = (0.2)^n_{i,j=1}$ and $B = (0.3)^n_{i,j=1}$. Thus, $A_{c_z} = (0.8)^n_{i,j=1}$ and $B_{c_z} = (0.7)^n_{i,j=1}$.

In this manner,

$$\beta(a_{ij}, b_{ij}) = \beta(0.2, 0.3) = 0.4 \neq 0.6 = \beta(0.8, 0.7) = \beta(c_z(a_{ij}), c_z(b_{ij})).$$

In the case of constructing Ψ as in Theorem 4.7, we obtain the following result.

Proposition 6.9. *Let Ψ be a matrix resemblance function as in* (3) *and σ satisfy (IG6). Then it holds that $\Psi(A, B) = \Psi(A_c, B_c)$.*

Proof.

$$\Psi(A_c, B_c) = M(\sigma(A_c, B_c), \sigma(B_c, A_c)) \underset{(IG6)}{=} M(\sigma(B, A), \sigma(A, B)) = \Psi(B, A) = \Psi(A, B).$$

□

The converse of Proposition 6.9 does not hold. Indeed, we can consider M to be the function in (13) for $N = 2$ and

$$\sigma(A, B) = \inf_{i,j}\{\min(1, 1 - a_{ij}^2 + b_{ij}^2)\}, \tag{16}$$

which satisfies (IG1) and (IG2). By Theorem 4.7, $\Psi(A, B) = M(\sigma(A, B), \sigma(B, A))$ is a matrix resemblance function.

Note that, as in the preceding counterexample, Ψ satisfies $\Psi(A, B) = \Psi(A_c, B_c)$ for all $A, B \in \mathcal{M}_n([0, 1])$ since the aforementioned three possible cases coincide.

Now, if we consider the matrices

$$A = \begin{pmatrix} 0.3 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 0.2 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix},$$

then

$$\sigma(A, B) = 0.95 \neq 0.85 = \sigma(B_c, A_c).$$

6.4. Shift invariance

This property, that we have called shift invariance, is related to constant enlightening and darkening of an image. It is sometimes required that a comparison measure gives the same result whenever we compare two images and the same two images constantly enlightened or darkened in the same amount, i.e., adding the same positive or negative amount to each pixel without exceeding the allowed range. This is equivalent to examining when

$$\Psi(A + \lambda J_n, B + \lambda J_n) = \Psi(A, B)$$

holds, where J_n is the $n \times n$ constant matrix with 1 in every entry and provided $a_{ij} + \lambda, b_{ij} + \lambda \in [0, 1]$ for $1 \leq i, j \leq n$, i.e., for all $\lambda \in [0, 1 - \max(a_{ij}, b_{ij})]$.

In the case of the first construction we attain the next result after a simple computation.

Proposition 6.10. *Let Ψ be a matrix resemblance function as in* (1) *and β satisfy $\beta(x + \lambda, y + \lambda) = \beta(x, y)$ for all $\lambda \in [0, 1 - \max(a_{ij}, b_{ij})]$. Then, $\Psi(A + \lambda J_n, B + \lambda J_n) = \Psi(A, B)$ for all $A, B \in \mathcal{M}_n([0, 1])$ and for all $\lambda \in [0, 1 - \max(a_{ij}, b_{ij})]$.*

Nevertheless, the converse does not hold. For example, consider β as in (12) and H as in (13). Thus, Ψ is a matrix resemblance function and let us show that $\Psi(A+\lambda J_n, B+\lambda J_n) = \Psi(A,B)$ for all $A, B \in \mathcal{M}_n([0,1])$ and for all $\lambda \in [0, 1-\max(a_{ij}, b_{ij})]$. Let $A, B \in \mathcal{M}_n([0,1])$. Firstly, it holds that

$$\Psi(A,B) = 0 \Longleftrightarrow \Psi(A+\lambda J_n, B+\lambda J_n) = 0 \quad \text{for all } \lambda \in [0, 1-\max(a_{ij}, b_{ij})] \tag{17}$$

Indeed, if $\Psi(A,B) = 0$, then there exist i, j such that $\{a_{ij}, b_{ij}\} = \{0,1\}$. Therefore $\max(a_{ij}, b_{ij}) = 1$ and hence $\lambda = 0$, which implies $\Psi(A+\lambda J_n, B+\lambda J_n) = 0$.
On the other hand, if $\Psi(A+\lambda J_n, B+\lambda J_n) = 0$, then there exist i, j such that $\{a_{ij}+\lambda, b_{ij}+\lambda\} = \{0,1\}$. Therefore $\lambda = 0$. This means that $\Psi(A,B) = 0$ if and only if $\Psi(A+\lambda J_n, B+\lambda J_n) = 0$.

Secondly, for all $\lambda \in [0, 1-\max(a_{ij}, b_{ij})]$, it holds that

$$\Psi(A,B) = 1 \Longleftrightarrow A = B \Longleftrightarrow A+\lambda J_n = B+\lambda J_n \Longleftrightarrow \Psi(A+\lambda J_n, B+\lambda J_n) = 1. \tag{18}$$

Now, due to the definition of function H, there are only three different cases:

- $\Psi(A,B) = 0$: By (17), $\Psi(A+\lambda J_n, B+\lambda J_n) = \Psi(A,B) = 0$ for all $\lambda \in [0, 1-\max(a_{ij}, b_{ij})]$.
- $\Psi(A,B) = 1$: By (18), $\Psi(A+\lambda J_n, B+\lambda J_n) = \Psi(A,B) = 1$ for all $\lambda \in [0, 1-\max(a_{ij}, b_{ij})]$.
- $\Psi(A,B) = 0.5$: Since (17) and (18) cover all the situations in which $\Psi(A+\lambda J_n, B+\lambda J_n) = 0$ and $\Psi(A+\lambda J_n, B+\lambda J_n) = 1$, it holds that $\Psi(A_c, B_c) = 0.5$ for all $\lambda \in [0, 1-\max(a_{ij}, b_{ij})]$.

Now, if we consider the matrices $A = (0.2)_{i,j=1}^n$ and $B = (0.3)_{i,j=1}^n$ and $\lambda = 0.1$, we get

$$\beta(a_{ij}, b_{ij}) = \beta(0.2, 0.3) = 0.4 \neq 0.6 = \beta(0.3, 0.4) = \beta(a_{ij}+\lambda, b_{ij}+\lambda).$$

Similarly, if we construct Ψ in the manner of Theorem 4.7, we obtain the following.

Proposition 6.11. *Let Ψ be a matrix resemblance function as in (3). If $\sigma(A+\lambda J_n, B+\lambda J_n) = \sigma(A,B)$ for all $\lambda \in [0, 1-\max(a_{ij}, b_{ij})]$, then $\Psi(A,B) = \Psi(A+\lambda J_n, B+\lambda J_n)$ for all $\lambda \in [0, 1-\max(a_{ij}, b_{ij})]$.*

Once again, the converse implication does not hold. If we consider M as the function in (13) for $N = 2$, which also satisfies (F3), σ as in (16), it holds that Ψ is a matrix resemblance function that is shift invariant, as in the preceding counterexample.

Now, consider $\lambda = 0.2$, and the constant matrices $A = (0.6)_{i,j=1}^n$ and $B = (0.3)_{i,j=1}^n$. Thus,

$$\sigma(A,B) = \inf_{i,j}\{\min(1, 1-0.6^2+0.3^2)\} = 1-0.6^2+0.3^2 = 0.73$$
$$\sigma(A+\lambda J_n, B+\lambda J_n) = \inf_{i,j}\{\min(1, 1-(0.6+0.2)^2+(0.3+0.2)^2)\} = 1-0.8^2+0.5^2 = 0.61.$$

6.5. Homogeneity and migrativity

Homogeneity and migrativity are two properties related to how a perturbation in both or one of the images, respectively, affects on the result. A function is said to be homogeneous of order $k > 0$ if when each argument is multiplied by a factor $\lambda > 0$, then the result is multiplied by λ^k. In the context of image comparison, multiplying an image by a factor equates to enlightening or darkening the image proportionally and the fact that matrix resemblance functions were homogeneous would mean that

$$\Psi(\lambda A, \lambda A) = \lambda^k \Psi(A,A) = \lambda^k,$$

but by (MRF1), $\Psi(\lambda A, \lambda A) = 1$, hence matrix resemblance functions are not homogeneous operators.

As for migrativity, the property of α-migrativity for a class of binary functions was introduced in [26] and it was studied for aggregation operators in [27]. Given $\alpha \in (0, 1)$, a function $F : [0, 1]^2 \to [0, 1]$ is said to be α-migrative if $F(\alpha x, y) = F(x, \alpha y)$ for all $\alpha \geq 0$ such that $(\alpha x, \alpha y) \in [0, 1]^2$. In the case of matrix resemblance functions, a MRF Ψ is said to be α-migrative if $\Psi(\alpha A, B) = \Psi(A, \alpha B)$ for some $\alpha \geq 0$ such that αA, $\alpha B \in \mathcal{M}_n([0, 1])$. However, matrix resemblance functions do not satisfy this property due to (MRF2). Indeed, consider A and B such that there exist i, j such that $a_{ij} = 0$ and $b_{ij} = 1$. Thus, $\Psi(\alpha A, B) = 0$ and $\Psi(A, \alpha B)$ need not be equal to 0. Therefore matrix resemblance functions are not migrative operators.

6.6. Additivity

Some similarity measures fulfill a property known as additivity (see [28]). In general, a comparison measure between two fuzzy sets $m : FS(X)^2 \to \mathbb{R}$ is said to be additive if there exists a function $h : [0, 1]^2 \to \mathbb{R}$ so that m can be decomposed in the following way:

$$m(A, B) = \sum_{x \in X} h(A(x), B(x)).$$

In that case h is said to be the additive generator of m.

However, matrix resemblance functions are not additive. Indeed, if they were, there would exist a function h such that

$$\Psi(A, B) = \sum_{i,j=1}^{n} h(a_{ij}, b_{ij}). \tag{19}$$

Now, consider A to be the 3×3 constant matrix with the value 0.8 in all its entries and set $B = A$.

Thus, $1 = \Psi(A, B) = \sum_{i,j=1}^{3} h(0.8, 0.8)$ and hence $h(0.8, 0.8) = \frac{1}{9}$. But if we modify the values a_{11} and b_{11} to be 1 and 0 respectively, then $0 = \Psi(A, B) = \sum_{i,j=1}^{n} h(a_{ij}, b_{ij})$ and therefore $h(0.8, 0.8) = 0$, a contradiction.

7. Special cases of functions H and M

In this section we study some special cases of the functions H in the first construction and M in the second. Recall that H is a function with n^2 inputs and M is a function with 2 inputs. Additionally, for the first construction H is not required to satisfy condition (F3), although, as seen in Section 6.1, it is convenient for a proper comparison measure as it ensures invariance under permutation.

7.1. H and M aggregation functions

The first two conditions of n-ary aggregation functions (see Definition 2.9) are trivially satisfied by any function that verifies (F1) and (F2), since the latter are more restrictive.

The third condition of aggregation functions is the one about monotonicity. Aggregation functions are increasing with respect to each component and functions H and M do not need to be. Nevertheless, in Section 6.2 we show that if H and M satisfy an additional monotonicity condition, i.e., H and M are increasing, we are under the conditions to apply Proposition 6.4 and Theorem 6.5, respectively. Therefore, in the cases H and M are aggregation functions, by the mentioned results, we know in which cases we have monotone matrix resemblance functions, which is an important property for an image comparison measure.

Moreover, in both constructions these functions are intended to aggregate the values resulting of applying the function β, in the first case, and the function σ, in the second. Hence, it seems natural to use an aggregation function for that purpose. However, increasingness is not strictly required, as in some situations it might be better to use some functions H and M that are not monotone.

7.2. H and M n-dimensional overlap functions

Overlap functions are a particular instance of aggregation functions that are symmetric and continuous. They were first introduced as bivariate functions in [29], and were later generalized to the n-dimensional setting in [30].

Since overlap functions verify (F1)-(F3) conditions, as well as a monotonicity and a continuity condition, they belong to an adequate family of functions that can be used as H and M in both constructions of matrix resemblance functions; as a n^2-ary function in the first case and a bivariate one in the second. Let us start with the definition of n-dimensional overlap functions.

Definition 7.1. A function $G : [0,1]^n \to [0,1]$ is said to be a n-dimensional overlap function if it satisfies (F1)-(F3) and:

(i) G is increasing with respect to each component,

(ii) G is a continuous function.

Example 7.2. The minimum, the product and the geometric mean are examples of n-dimensional overlap functions.

Overlap functions satisfy an increasingness condition, which is important for the image comparison property studied in Section 6.2. Moreover, overlap functions are continuous and hence, their use in the construction of matrix resemblance functions can lead to obtaining continuous matrix resemblance functions. Continuity can be considered a desirable property for comparison measures as it ensures a certain degree of robustness, i.e., comparing two images and the same images having been slightly altered produce similar results due to the continuity of the comparison operator.

Let us further study the cases in which a matrix resemblance function Ψ is continuous. In the case of the first construction with H an overlap function, if β is continuous then so is Ψ, as it can be seen of a composition of continuous functions.

However, the converse does not hold. Indeed, let the function $H : [0,1]^N \to [0,1]$ be defined as:

$$H(x_1, \ldots, x_N) = \max\left(\min(x_1, \ldots, x_N, 0.2), \frac{0.56\,\min(x_1, \ldots, x_N) - 0.07}{0.49}\right), \tag{20}$$

which is an overlap function, and let $\beta : [0,1]^2 \to [0,1]$ be:

$$\beta(x,y) = \begin{cases} 0.2, & \text{if } \{x,y\} = \{0.15, 0.9\}, \\ 1 - |x-y|, & \text{otherwise.} \end{cases} \tag{21}$$

Thus, one easily verifies that β is not continuous and yet the matrix resemblance function constructed as $\Psi(A,B) = \underset{\substack{i=1\\j=1}}{\overset{n}{H}}(\beta(a_{ij}, b_{ij}))$ is continuous. This is due to the fact that β has the discontinuity in the area where H is constant (see Figure 3 for a graphical representation of H in the two dimensional case).

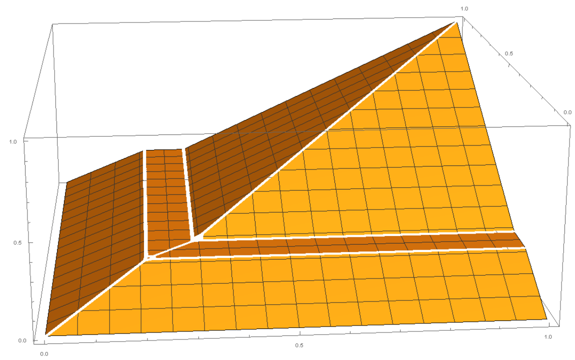


Figure 3: Graphical representation of the function H in (20) for $N = 2$.

However, if H is an idempotent overlap function, we can characterize the continuity of Ψ in terms of the continuity of β:

Theorem 7.3. *Let Ψ be a matrix resemblance function as in* (1) *with H an idempotent overlap function. Then Ψ is continuous if and only if β is continuous.*

Proof. Let H be an idempotent n^2-dimensional overlap function and $\Psi(A,B) = \underset{\substack{i=1\\j=1}}{\overset{n}{H}}(\beta(a_{ij}, b_{ij}))$ a continuous matrix resemblance function and let us assume that β is not continuous.

Recall that a real function f of N arguments is continuous if for any sequence $(x_{ij})_{i=1}^{N}$ such that $\lim_{j\to\infty} x_{ij} = y_i$, then $\lim_{j\to\infty} f(x_{1j}, \ldots, x_{Nj}) = f(y_1, \ldots, y_N)$.

Thus, since β is not continuous there exists a sequence (x_k, y_k) such that $\lim_{k\to\infty}(x_k, y_k) = (x, y)$ for some $x, y \in [0,1]$, but $\lim_{k\to\infty} \beta(x_k, y_k) \neq \beta(x, y)$.

Now, consider A_k and B_k the constant matrices with x_k and y_k in all their entries, respectively. Thus, $\lim_{k\to\infty}(A_k, B_k) = (A, B)$, where A is the constant matrix with x in all its entries and B is the constant y matrix.

Since Ψ is continuous, it holds that $\lim_{k\to\infty} \Psi(A_k, B_k) = \Psi(A, B)$. But since H is idempotent, it holds that

$$\Psi(A_k, B_k) = \underset{\substack{i=1\\j=1}}{\overset{n}{H}}(\beta(x_k, y_k)) = \beta(x_k, y_k), \text{ and}$$

$$\Psi(A, B) = \underset{\substack{i=1\\j=1}}{\overset{n}{H}}(\beta(x, y)) = \beta(x, y).$$

Which contradicts the fact that $\lim_{k\to\infty} \beta(x_k, y_k) \neq \beta(x, y)$. Therefore β is continuous.

The converse implication is immediate, since Ψ is the composition of continuous functions. □

8. Image comparison algorithm

In this section we present a method to measure the difference, or similarity, between two grayscale images. The following algorithm is underpinned by the concept of matrix resemblance functions as a method to compare neighbourhoods of pixels. The reason for a neighbourhood-based comparison as an alternative of proceeding pixel-wise is that we wish to take into account the visual impact that an alteration has in its proximity (see [2]).

One of the main contributions of this paper is that instead of obtaining a number for a result, we get another image, that we will call comparison image and will allow us to set similarity regions, which means that not only will we get a global idea of how similar the images are, but also we will be able to extract areas in the images where they are equally similar or equally different.

The final step of the algorithm is to cluster the comparison image in a variable number of similarity regions, depending on our purpose. For that we use a variation of the k-means algorithm [31, 32]. The k-means algorithm is a well-known clustering algorithm that divides a set of data into k groups, being k fixed beforehand. The first step is to set k centroids, one for each group, and then classify each datum in the class of the closest centroid. The next step is to recalculate the centroids as the arithmetic mean of the data that belong to each group, then it proceeds to redistribute all the data according to the closeness to the new centroids. The process continues until the groups remain unchanged for two consecutive iterations. When using this algorithm for image segmentation, it is usual that the closeness of the pixels to each centroid is computed based on the intensity of each pixel, but the variation of the algorithm that we use also takes into account their spatial distribution.

Our comparison algorithm takes two images of the same size and returns two other images, the comparison image and the clustered image by the k-means algorithm. Then, for each pixel of the first input image and the corresponding one in the second, it considers their neighbourhood and compares them using a matrix resemblance function, then sets the number resulting from that local comparison to the pixel from the comparison image in the position that is being considered. Once the loop is finished, the user must decide the number of clusters for the algorithm to apply the clustering technique to the comparison image and get the one divided in similarity regions.

Algorithm 1 Image comparison measure algorithm

Input: Two images (of the same size) to compare: A and B
Output: C the comparison image and SC the image clustered in similarity regions
for each pixel in A **do**
    Consider its neighbourhood in A and the corresponding neighbourhood in B
    Compare both neighbourhoods using a matrix resemblance function
    Define in C a pixel in that position and whose value is the result of the comparison
end for
Show the comparison image C
Ask for the number k of clusters needed
Perform the spatial k-means clustering algorithm with k clusters and save it in SC

Once we have the comparison image C and the clustered image SC, we are able to visually inspect which areas are more similar and which more different. Since matrix resemblance functions give results closer to 1 when matrices are similar, the regions that are more akin will appear clearer in the comparison image and the more different regions will be darker.

Furthermore, in the clustered image SC, for each cluster we can compute the arithmetic mean of the values from C that are in that cluster and get a number that expresses a similarity measure in each region. In this way the image is divided in zones and we provide a local similarity measure.

9. Illustrative examples

In this section we present some examples that illustrate the algorithm proposed in Section 8.

These examples show the advantages of considering a new image rather than a number as the result of a comparison measure. In this way it is possible to extract location information, such as where both images are more different and how different they are in that area. Besides, a number can lead us to confusion when using it as a measure to compare two images; if we obtain a number that is close to 1 we might think that both images are nearly identical, nevertheless, as we will see in the following examples, this is not always the case. The choice of the parameter k in the final part of the algorithm is made according to the nature of each example to better illustrate the different regions of similarity.

For the examples we have considered the matrix resemblance function of Example 4.8, which can be obtained using either construction method given in this paper, and the images that we use can be found in http://decsai.ugr.es/cvg/index2.php.

9.1. Example 1

In the first example, Figure 4, we compare an image with another which is the result of enlightening and darkening the division in three areas of the first (see Figure 4.a and 4.b). The result of applying Algorithm 1 are the images in Figure 4.c (the comparison image) and in Figure 4.d (the result of applying the k-means clustering algorithm to the comparison image).

a *b* *c* *d*

Figure 4: *a:* Original picture, *b:* Locally enlightened and darkened picture, *c:* Comparison image and *d:* Result of the k-means clustering algorithm with $k = 3$.

In the comparison image we can see the regions where the images are more similar (the lighter regions) and the ones where the images are more different (the darker ones). In this case, the fact that the bottom part of Figure 4.c is the brightest denotes that it is the region where Figures 4.a and 4.b are the most similar. Similarly, the fact that the top left part of Figure 4.c is the darkest denotes that Figures 4.a and 4.b are the least similar on the top left part. Furthermore, if we computed the arithmetic mean of the pixels of the comparison image we would get a global image comparison measure, i.e., the result of the comparison would be a number. However, that number in the case of Figure 4 would be 0.7737 and this number does not provide too much information. Using the whole comparison image as a result, we are able to distinguish the zones where both pictures are more similar and computing the arithmetic mean to each of the three regions obtained by the k-means algorithm we obtain a local measure for each region of similarity (see Figure 5).

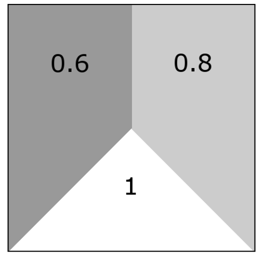


Figure 5: Comparison measure of each similarity region.

9.2. Example 2

Another possibility is to compare an image with another in which noise has been added. For the next example we use the image in Figure 6.a and we compare it to Figure 6.b, a new image in which we have added different intensities of Gaussian noise to different areas.

In this case, it is difficult to accurately tell, in plain sight, the regions where each intensity of Gaussian noise has been applied. However, the comparison image that results from the Algorithm shows that images Figure 6.a and Figure 6.b are more similar in the centre.

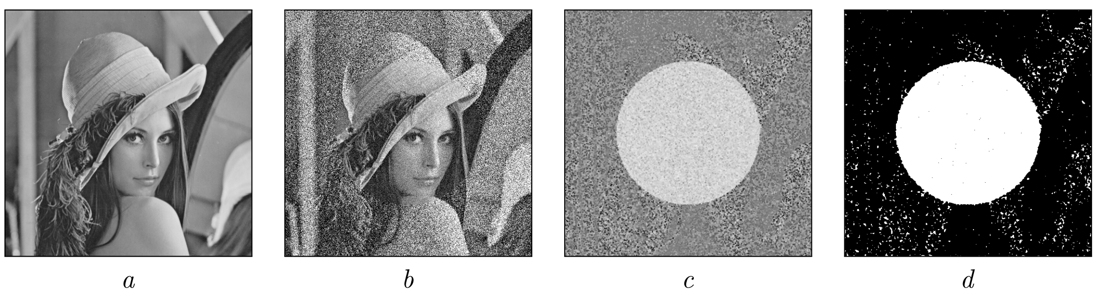


Figure 6: *a:* Original picture, *b:* Picture with 2 different Gaussian noises ($\sigma = 0.1$ and $\sigma = 0.01$), *c:* Comparison image and *d:* Result of the k-means clustering algorithm with $k = 2$.

The mean pixel intensity of the comparison image is 0.604 and if we compute as before the arithmetic mean of each similarity region given by the k-means segmentation, we reach the results that appear in Figure 7.

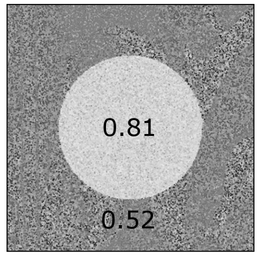


Figure 7: Comparison measure of each similarity region.

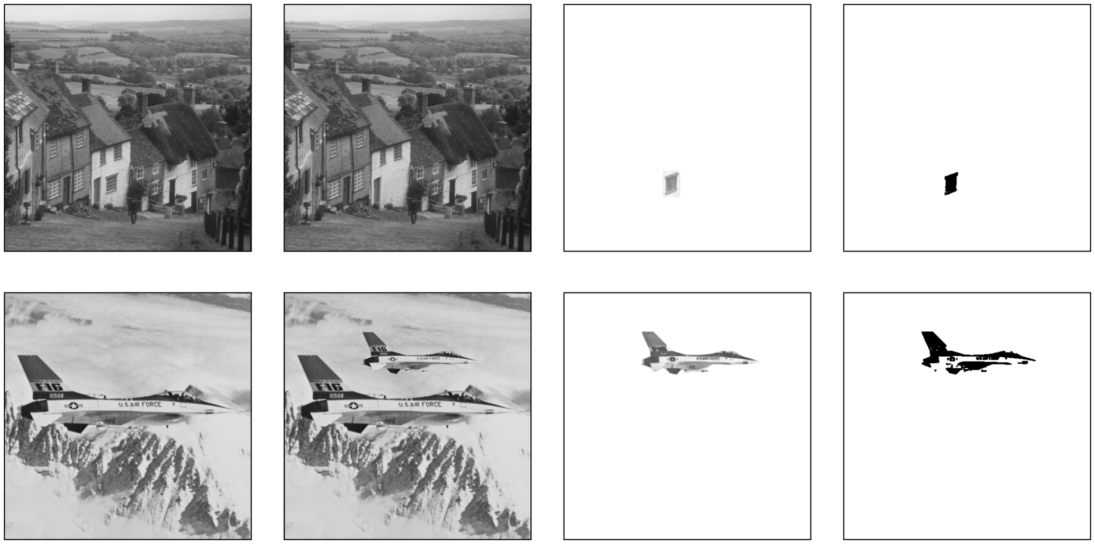
The difference between the average of the values in each region indicates that the k-means algorithm segments the comparison image to obtain well defined similarity regions, areas where the images we compare are equally similar or dissimilar.

10. Three possible applications

In this section we expose some possible applications for the previously presented MRF-based algorithm. Besides the application as an image comparison measure as such, this method has also potential applications in such fields as pattern matching, vision information retrieval, tamper and damaged areas detection for image reconstruction algorithms, defect detection in industrial processes, video motion detection and object tracking, etc. In this section we present some examples for the last three.

10.1. Tamper and damaged areas detection

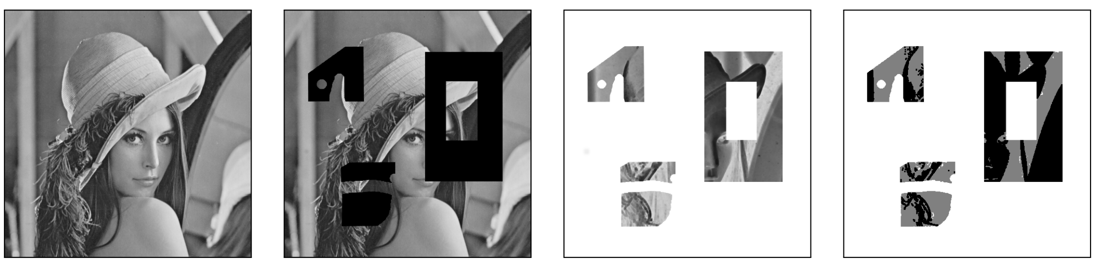
One possible application of the proposed comparison method is tamper detection [5, 33]. In Figure 8 we show an example, similar to the ones given in [33], of a tampered image with image synthesis attacks. As it appears in Figure 8.d, the algorithm successfully locates the tampered areas.



a *b* *c* *d*

Figure 8: *a:* Original images, *b:* Tampered images, *c:* Comparison images and *d:* Results of the k-means clustering algorithm with $k = 2$.

Additionally, this algorithm could be used for image reconstruction techniques as it successfully locates the damaged areas. The next example (Figure 9) is similar to the one that can be found in [34], it consists of an image and a damaged version of it.



a *b* *c* *d*

Figure 9: *a:* Original image. *b:* Damaged image. *c:* Comparison images. *d:* Results of the k-means clustering algorithm with $k = 3$.

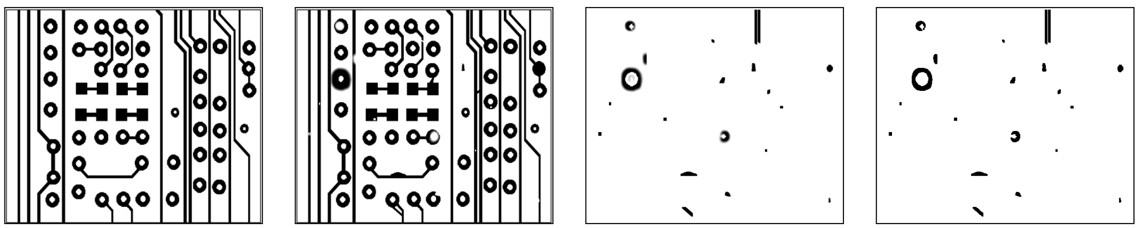
10.2. Defect detection in industrial processes

The aim of this section is to present some examples of the performance of our method for defect detection in PCBs, showing the applicability of our algorithm in this field.

Visual inspection systems play a crucial role in manufacturing processes, as they benefit in the goal of having a 100% rate of defect-free products. In this section we focus on the case of defect detection in the assembly of printed circuit boards (PCB). PCBs are a basic component in any electronic device and therefore it is important that they do not have any defects to ensure the proper performance of the device in question. Defects in PCBs are sorted in two types, functionals and cosmetic [35]. There are several proposals of automatic optical inspection systems to detect either kind of defects in the manufacturing production of PCBs [36, 37].

The use of our algorithm based on matrix resemblance functions as a PCB inspection algorithm would be categorised as a referential approach [4], as it would be a model-based technique.

In Figure 10 one can see an example, as the one in [37], of defect detection and location using Algorithm 1 to compare a well assembled PCB image with a defective one.



a *b* *c* *d*

Figure 10: *a:* Image of good PCB patterns, *b:* Image of defected PCB patterns, *c:* Comparison image and *d:* Results of the k-means clustering algorithm with $k = 2$.

The defects that appear in Figure 10.b are detected and located by our algorithm (Figure 10.d).

10.3. Video motion detection and object tracking

The image comparison method presented in this work could also be used for videos, specifically for motion detection or object tracking, i.e., locating an object that is moving in a video.

In this subsection, we show how the comparison algorithm with matrix resemblance functions could be applied to object tracking in a video. The idea is to use the algorithm to compare two videos instead of two images; a video in which an object moves and another in which either the object does not appear or it is still.

The first step to carry a comparison between two videos, that have exactly the same duration, is to extract their frames. Thus, we have a series of images and the amount of pixels to manage is the result of adding all the pixels from each frame. In this case, we consider that the neighbourhood of a pixel (i, j, k), i.e., the pixel in the row i, column j of the frame k, is formed by the adjacent pixels in the same frame k, the corresponding ones in the previous frame $k - 1$ and those from the following frame $k + 1$. So, a neighbourhood of a pixel in a video can be seen as a 3-D matrix and two neighbourhoods can be compared using matrix resemblance functions as before. In Figure 11 we show a representation of an instance of a neighbourhood of a pixel in a video. Note that, since the definition of matrix resemblance function can be straightforwardly generalized to the case of 3 dimensional matrices, the same algorithm can be used.

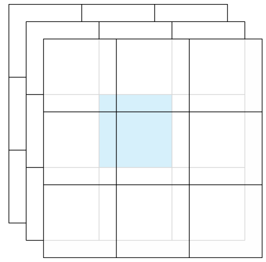


Figure 11: $3 \times 3 \times 3$ neighbourhood of a pixel in a video. The blue square represents the pixel.

Let us now present an example of the usage of matrix resemblance functions to detect a person who is crossing a street. We use a video from a human motion database[2] that is described in [38]. In order to reproduce a video in which the street is empty, we build a new one consisting in a copy of a frame in which the street is empty from the original video.

The video of the following example is a conversion to grayscale of a movie-clip from the 2002 film *About a boy* directed by Paul and Chris Weitz. In Figure 12 a glimpse of the results are shown. It is apparent that a extraction of the object in motion, the pedestrian in this case, is achieved and that it is reasonably determined by the k-means algorithm.

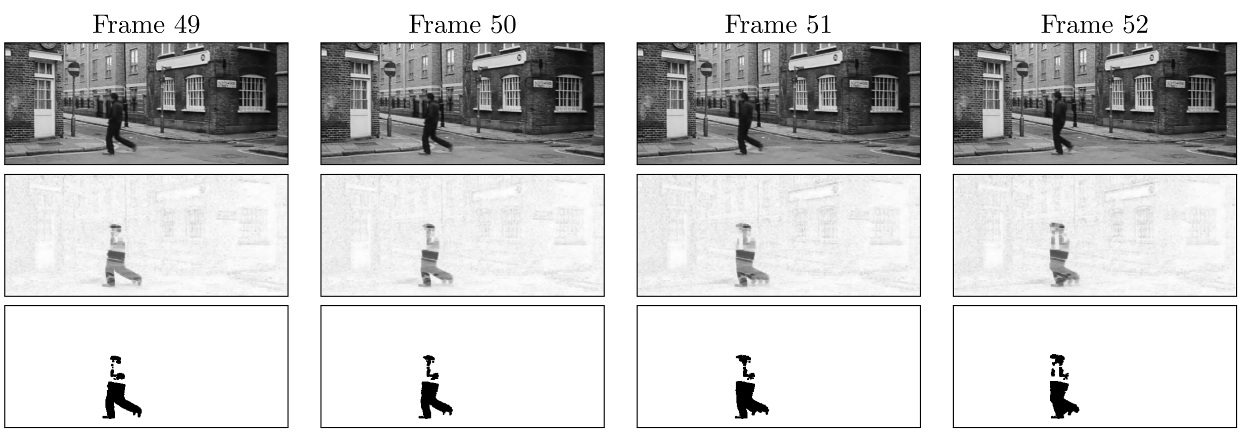


Figure 12: *First row:* Grayscale conversion of the original frames from the clip. *Second row:* Results of comparing the clips. *Third row:* Results of the k-means algorithm for $k = 2$.

11. Conclusions

In this paper a method for comparing images is presented which not only considers the information provided by each pixel, but also the impact that the surrounding of each pixel has in the comparison. For that purpose, the concept of matrix resemblance function is introduced and two

[2] http://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/

construction methods are presented. Furthermore, since the result of the comparison is a new image, we are able to identify areas in which both images are equally similar and equally dissimilar. Due to this fact, the comparison method presented in this paper is versatile when it comes to possible applications. We have seen that the method could yield good results when applied to tamper detection, location of defect detection in manufacturing processes and video motion detection and object tracking.

Acknowledgments

This work is partially supported by the research services of *Universidad Publica de Navarra* and by the project TIN2016-77356-P (AEI/FEDER, UE). R. Mesiar is supported by Slovak grant APVV-14-0013, and by Czech Project LQ1602 "IT4Innovations excellence in science".

References

[1] X. Wang, B. De Baets, and E. Kerre. A comparative study of similarity measures. *Fuzzy Sets and Systems*, 73(2):259 – 268, 1995.

[2] D. Van der Weken, M. Nachtegael, and E. E. Kerre. Using similarity measures and homogeneity for the comparison of images. *Image and Vision Computing*, 22(9):695 – 702, 2004.

[3] H. Bustince, M. Pagola, and E. Barrenechea. Construction of fuzzy indices from fuzzy DI-subsethood measures: Application to the global comparison of images. *Information Sciences*, 177(3):906 – 929, 2007.

[4] P. S. Malge and R. S. Nadaf. PCB defect detection, classification and localization using mathematical morphology and image processing tools. *International Journal of Computer Applications*, 87(9):40 – 45, 2014.

[5] C. S. Hsu and S. F. Tu. Image tamper detection and recovery using adaptive embedding rules. *Measurement*, 88:287 – 296, 2016.

[6] H. Bustince, E. Barrenechea, and M. Pagola. Restricted equivalence functions. *Fuzzy Sets and Systems*, 157(17):2333 – 2346, 2006.

[7] D. Sinha and E. R. Dougherty. Fuzzification of set inclusion: Theory and applications. *Fuzzy Sets and Systems*, 55(1):15 – 42, 1993.

[8] H. Bustince. Indicator of inclusion grade for interval-valued fuzzy sets. Application to approximate reasoning based on interval-valued fuzzy sets. *International Journal of Approximate Reasoning*, 23(3):137 – 209, 2000.

[9] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338 – 353, 1965.

[10] H. Bustince, E. Barrenechea, M. Pagola, and F. Soria. Weak fuzzy S-subsethood measures: Overlap Index. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 14(05):537–560, 2006.

[11] H. Bustince, V. Mohedano, E. Barrenechea, and M. Pagola. Definition and construction of fuzzy DI-subsethood measures. *Information Sciences*, 176(21):3190 – 3231, 2006.

[12] L. M. Kitainik. *Fuzzy Inclusions and Fuzzy Dichotomous Decision Procedures*, pages 154 – 170. Springer Netherlands, Dordrecht, 1987.

[13] V. R. Young. Fuzzy subsethood. *Fuzzy Sets and Systems*, 77(3):371 – 384, 1996.

[14] J. Fan, W. Xie, and J. Pei. Subsethood measure: new definitions. *Fuzzy Sets and Systems*, 106(2):201 – 209, 1999.

[15] P. Burillo, N. Frago, and R. Fuentes. Inclusion grade and fuzzy implication operators. *Fuzzy Sets and Systems*, 114(3):417 – 429, 2000.

[16] G. Beliakovand H. Bustince and T. Calvo. *A practical guide to averaging functions*, volume 329. Springer, 2016.

[17] H. Bustince, J. Montero, E. Barrenechea, and M. Pagola. Semiautoduality in a restricted family of aggregation operators. *Fuzzy Sets and Systems*, 158(12):1360 – 1377, 2007.

[18] C. Cornelis, C. Van der Donck, and E. Kerre. Sinha-Dougherty approach to the fuzzification of set inclusion revisited. *Fuzzy Sets Syst.*, 134(2):283 – 295, 2003.

[19] H. Bustince, P. Burillo, and F. Soria. Automorphisms, negations and implication operators. *Fuzzy Sets and Systems*, 134(2):209 – 229, 2003.

[20] J.P. Serra. *Image analysis and mathematical morphology.* Number 1 in Image Analysis and Mathematical Morphology. Academic Press, 1982.

[21] H.J.A.M. Heijmans and C. Ronse. The algebraic basis of mathematical morphology I. dilations and erosions. *Computer Vision, Graphics, and Image Processing*, 50(3):245 – 295, 1990.

[22] S. R. Sternberg. Grayscale morphology. *Computer Vision, Graphics, and Image Processing*, 35(3):333 – 355, 1986.

[23] D. Sinha and E. R. Dougherty. Fuzzy mathematical morphology. *Journal of Visual Communication and Image Representation*, 3(3):286 – 302, 1992.

[24] I. Bloch and H. Maitre. Fuzzy mathematical morphologies: A comparative study. *Pattern Recognition*, 28(9):1341 – 1387, 1995.

[25] M. Nachtegael and E. E. Kerre. Connections between binary, gray-scale and fuzzy mathematical morphologies. *Fuzzy Sets and Systems*, 124(1):73 – 85, 2001.

[26] F. Durante and P. Sarkoci. A note on the convex combinations of triangular norms. *Fuzzy Sets and Systems*, 159(1):77 – 80, 2008.

[27] H. Bustince, J. Montero, and R. Mesiar. Migrativity of aggregation functions. *Fuzzy Sets and Systems*, 160(6):766 – 777, 2009.

[28] I. Couso and L. Sánchez. Additive similarity and dissimilarity measures. *Fuzzy Sets and Systems*, 2016. In Press.

[29] H. Bustince, J. Fernandez, R. Mesiar, J. Montero, and R. Orduna. Overlap functions. *Nonlinear Analysis: Theory, Methods & Applications*, 72(3–4):1488 – 1499, 2010.

[30] D. Gómez, J. Tinguaro Rodríguez, J. Montero, H. Bustince, and E. Barrenechea. n-dimensional overlap functions. *Fuzzy Sets and Systems*, 287:57 – 75, 2016. Theme: Aggregation Operations.

[31] V. Faber. Clustering and the continuous k-means algorithm. *Los Alamos Science*, 22:138–144, 1994.

[32] J. A. Hartigan and M. A. Wong. Algorithm AS 136: A K-means clustering algorithm. *Applied Statistics*, pages 100–108, 1979.

[33] C. S. Hsu and S. F. Tu. Probability-based tampering detection scheme for digital images. *Optics Communications*, 283(9):1737 – 1743, 2010.

[34] I. Perfilieva and P. Vlašánek. Image reconstruction by means of F-transform. *Knowledge-Based Systems*, 70:55 – 63, 2014.

[35] W. Y. Wu, M. J. Wang, and C. M. Liu. Automated inspection of printed circuit boards through machine vision. *Computers in Industry*, 28(2):103 – 111, 1996.

[36] K. Kamalpreet and K. Beant. PCB defect detection and classification using image processing. *International Journal of Emerging Research in Management &Technology*, 3(8), 2014.

[37] N. Dave, V. Tambade, B. Pandhare, and S. Saurav. PCB defect detection using image processing and embedded system. *International Research Journal of Engineering and Technology*, 3(5), 2016.

[38] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.