mathematics

MDPI

Article

New Model of Heteroasociative Min Memory Robust to Acquisition Noise

Julio César Salgado-Ramírez [1,*], Jean Marie Vianney Kinani [2,*], Eduardo Antonio Cendejas-Castro [3], Alberto Jorge Rosales-Silva [4], Eduardo Ramos-Díaz [5] and Juan Luis Díaz-de-Léon-Santiago [6,*]

1 Ingeniería Biomédica, Universidad Politécnica de Pachuca (UPP), Zempoala 43830, Mexico
2 Ingeniería Mecatrónica, Instituto Politécnico Nacional-UPIIH, Pachuca 07738, Mexico
3 Escuela de Ingeniería y Ciencias, Tecnologico de Monterrey, Pachuca 42083, Mexico; eduardo.cendejas@tec.mx
4 Sección de Estudios de Posgrado e Investigación, Instituto Politécnico Nacional-ESIME Zacatenco, Mexico City 07738, Mexico; arosaless@ipn.mx
5 Ingeniería en Sistemas Electrónicos y de Telecomunicaciones, Universidad Autónoma de la Ciudad de México, Mexico City 09790, Mexico; eduardo.ramos@uacm.edu.mx
6 Centro de Investigación en Computación, Instituto Politécnico Nacional, Mexico City 07700, Mexico
* Correspondence: csalgado@upp.edu.mx (J.C.S.-R.); jkinani@ipn.mx (J.M.V.K.); jdiaz@cic.ipn.mx (J.L.D.-d.-L.-S.)

Abstract: Associative memories in min and max algebra are of great interest for pattern recognition. One property of these is that they are one-shot, that is, in an attempt they converge to the solution without having to iterate. These memories have proven to be very efficient, but they manifest some weakness with mixed noise. If an appropriate kernel is not used, that is, a subset of the pattern to be recalled that is not affected by noise, memories fail noticeably. A possible problem for building kernels with sufficient conditions, using binary and gray-scale images, is not knowing how the noise is registered in these images. A solution to this problem is presented by analyzing the behavior of the acquisition noise. What is new about this analysis is that, noise can be mapped to a distance obtained by a distance transform. Furthermore, this analysis provides the basis for a new model of min heteroassociative memory that is robust to the acquisition/mixed noise. The proposed model is novel because min associative memories are typically inoperative to mixed noise. The new model of heteroassocitative memory obtains very interesting results with this type of noise.

Keywords: associative memories; noise; kernel; Fast Distance Transform

Citation: Salgado-Ramírez, J.C.; Vianney Kinani, J.M.; Cendejas-Castro, E.A.; Rosales-Silva, A.J.; Ramos-Díaz, E.; Díaz-de-Léon-Santiago, J.L. New Model of Heteroasociative min Memory Robust to Acquisition Noise. *Mathematics* **2022**, *10*, 148. https://doi.org/10.3390/math10010148

Academic Editor: Xinsong Yang

Received: 6 December 2021
Accepted: 2 January 2022
Published: 4 January 2022

1. Introduction

The human brain is quite intriguing when it comes to learning and remembering its environment. By just hearing a sound, smelling an aroma, seeing an image, touching a texture, among other things, our brain, through its neurons, may associate external information with something that has been learned and consequently indicate how to act according to that information. The most surprising thing is that very little information acquired by our senses is enough for our brain to fully remember the learned information. For example, the brain learns the characteristics of a person's face in broad daylight and is able to remember the same characteristics learned from that face on a poorly lit night; this means that the brain needs minimal but sufficient information to remember. It should be noted that, if the brain does not have enough minimum information, it ends up being unable to remember; which implies that, if this same minimum information needed by the brain is affected by other types of information, like noise, regardless of whether it is something already learned, it cannot be remembered. This fact has inspired some researchers, in the area of pattern recognition, to come up with some models that simulate the behavior of the brain. One of those models is associative memories [1–15]. In the development of associative memory algorithms, one pushes towards those algorithms

with greater learning capacity, better performance, and above all, they must be robust to different types of noise [13,16–21]. Thanks to the versatility of associative memories, they have been implemented to solve pattern recognition problems in different areas such as robotics [22–24], medicine [25–27], among others [20,28–38] . One should point out that the first associative memory presented in the state of the art, that is, the Lernmatrix [1], is the basis of the associative classifier $LM\tau 9$ and has proven to be such a competitive classifier with most used algorithms in machine learning [39,40]. Owing to their versatility, associative memories have continued to be the subject of research for the last decade [20,22–33,35–37,41–50].

Generally, an associative memory is a process that aims to recover or fully remember patterns, from input patterns which may be altered by some type of noise. Associative memory can be exemplified as a black box that receives an x pattern as input, processes it, and generates a y pattern as a result, as shown in Figure 1.

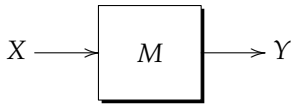


Figure 1. Associative memory as black box.

The term "fully recall" means that the resulting pattern is identical to the pattern that was learned by associative memory beforehand. The relationship between the input pattern x and the output pattern y is defined as an ordered pair (x, y), and both are column vectors. Hence, the associative memory must be able to learn a set of ordered pairs of patterns and retrieve the output patterns from the input patterns. Memory M is defined as:

$$\{(x^{\omega}, y^{\omega}) \mid \omega \in \{1, 2, \ldots, p\}\} \tag{1}$$

where p indicates the cardinality of the displayed set. The finite set of patterns in expression (1) is called fundamental set of patterns and its elements, fundamental patterns, which can be input or output.

To refer to some element of an x pattern or to that of a y pattern, the following notation will be adopted, that is, x_j^{ω} or y_j^{ω}, where j is the index of the pattern element position and ω is the ordered pair index.

According to Figure 1, M is the learning matrix or the associative memory, and it will contain the encoded information of the fundamental set after it has learned the pattern; it will also be operated in a certain way with the previously learned pattern x, where x can be altered with some kind of noise; and as a result a corresponding output pattern y is expected.

Associative memories consist of two phases–learning and recalling phases. The learning phase consists of finding the necessary operator (s), so that, in some way, the relationship that exists between the input and output patterns is encoded, and through this encoding the learning matrix M is generated. On the other hand, the recalling phase consists of finding the necessary operator or operators along with sufficient conditions to generate an output pattern; that is, once the matrix M has been formed, an input pattern x that was previously learned is presented, then, M is operated with the needed operator or operators under certain circumstances together with the x pattern, thereby generating output pattern y.

In this paper, an input pattern altered by noise will be represented as $\widetilde{x}$. For example, the expression $\widetilde{x}^{\omega}$ represents the input pattern x^{ω} altered by noise.

Typically, associative memories are classified into auto and heteroassociative memories. An associative memory is said to be autoassociative if $x^{\mu} = y^{\mu}\ \forall \mu \in \{1, 2, \ldots, p\}$ and a memory is hetero-associative if $\exists \mu \in \{1, 2, \ldots, p\}$ for which $x^{\mu} \neq y^{\mu}$.

Associative memories differ in the form of algebra they use; for instance, min and max-memories, as their names imply, work in the so-called minimax algebra, that is, minimal

and maximal algebra [16,17,19,21,34,35,45], whereas other types of memories work in real algebra [1–14].

In this article, we will focus on associative memories based on minimax algebra such as morphological and $\alpha\beta$ memories, which are memories that meet the cited conditions [16,17,19,21,34,35,45].

1.1. Morphological Associative Memories

There are two types of morphological associative memories: the $\bigvee$ (*max* $\bigvee$) memories represented by M and the $\bigwedge$ (*min* $\bigwedge$) memories represented by W. Both memories work for heteroassociative and autoassociative modes. The fundamental set for morphological associative memories is:

$$\{(x^\mu, y^\mu) | \mu = 1, 2, \ldots, p\}$$

$$A \in \mathbb{R}, \quad x^\mu = \begin{pmatrix} x_1^\mu \\ x_2^\mu \\ \vdots \\ x_n^\mu \end{pmatrix} \in A^n \quad \text{y} \quad y^\mu = \begin{pmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_m^\mu \end{pmatrix} \in A^m \tag{2}$$

Two new operations between arrays are defined in terms of the $+$, $\bigvee$ and $\bigwedge$ operations [16,17] in order to express the learning and recall phases of the morphological associative memories.

Let D be a matrix $[d_{ij}]_{m\times r}$ and H, a matrix $[h_{ij}]_{r\times n}$ whose terms are integer numbers.

Definition 1. *Maximum product of D and H denoted by $C = D \bigtriangledown H$, is a matrix $[c_{ij}]_{m\times n}$ whose ij-th component c_{ij} is defined as follows:*

$$c_{ij} = \bigvee_{k=1}^{r} \left(d_{ik} + h_{kj} \right) \tag{3}$$

Definition 2. *The minimum product of D and H, denoted by $C = D \bigtriangleup H$, is a matrix $[c_{ij}]_{m\times n}$ whose ij-th component c_{ij} is defined as follows:*

$$c_{ij} = \bigwedge_{k=1}^{r} \left(d_{ik} + h_{kj} \right) \tag{4}$$

The learning phase of morphological $\bigvee$ memories consists of two stages:

1. In each of the p associations (x^μ, y^μ), Equation (4) is applied to build memory $y^\mu \bigtriangleup (-x^\mu)^t$ of dimension $m \times n$, where the negated transpose of the input pattern x^μ is defined as $(-x^\mu)^t = (-x_1^\mu, -x_2^\mu, \ldots, -x_n^\mu)$. This expression may be elaborated as follows:

$$y^\mu \bigtriangleup (-x^\mu)^t = \begin{pmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_m^\mu \end{pmatrix} \bigtriangleup \left(-x_1^\mu, -x_2^\mu, \ldots, -x_n^\mu \right)$$

$$y^\mu \bigtriangleup (-x^\mu)^t = \begin{pmatrix} y_1^\mu - x_1^\mu & y_1^\mu - x_2^\mu & \cdots & y_1^\mu - x_j^\mu & \cdots & y_1^\mu - x_n^\mu \\ y_2^\mu - x_1^\mu & y_2^\mu - x_2^\mu & \cdots & y_2^\mu - x_j^\mu & \cdots & y_2^\mu - x_n^\mu \\ \vdots & \vdots & & \vdots & & \vdots \\ y_i^\mu - x_1^\mu & y_i^\mu - x_2^\mu & \cdots & y_i^\mu - x_j^\mu & \cdots & y_i^\mu - x_n^\mu \\ \vdots & \vdots & & \vdots & & \vdots \\ y_m^\mu - x_1^\mu & y_m^\mu - x_2^\mu & \cdots & y_m^\mu - x_j^\mu & \cdots & y_m^\mu - x_n^\mu \end{pmatrix} \tag{5}$$

2. Equation (5) is applied to the p matrices to obtain the morphological memory M, as shown in Equation (6).

$$M = \bigvee_{\mu=1}^{p} \left[y^{\mu} \bigwedge (-x^{\mu})^{t} \right] = \left[m_{ij} \right]_{m \times n}$$

$$m_{ij} = \bigvee_{\mu=1}^{p} \left(y_i^{\mu} - x_j^{\mu} \right) \tag{6}$$

The recall phase consists of applying the minimum product as shown in Equation (4) between memory M and the input pattern x^{ω}, where $\omega \in \{1, 2, \ldots, p\}$, in order to obtain a column vector of dimension m, as shown in Equation (8).

$$y = M \bigwedge x^{\omega} \tag{7}$$

Note that, the i-th component of the vector y is:

$$y_i = \bigwedge_{j=1}^{n} \left(m_{ij} + x_j^{\omega} \right) \tag{8}$$

The learning phase for morphological $\bigwedge$ memories consists of two stages:

1. In each of the p associations (x^{μ}, y^{μ}), Equation (7) is applied to build memory $y^{\mu} \nabla (-x^{\mu})^{t}$ of dimension $m \times n$, where the negated transpose of the input pattern x^{μ} is defined as $(-x^{\mu})^{t} = (-x_1^{\mu}, -x_2^{\mu}, \ldots, -x_n^{\mu})$. This expression may be expanded as:

$$y^{\mu} \nabla (-x^{\mu})^{t} = \begin{pmatrix} y_1^{\mu} \\ y_2^{\mu} \\ \vdots \\ y_m^{\mu} \end{pmatrix} \nabla \left(-x_1^{\mu}, -x_2^{\mu}, \ldots, -x_n^{\mu} \right)$$

$$y^{\mu} \nabla (-x^{\mu})^{t} = \begin{pmatrix} y_1^{\mu} - x_1^{\mu} & y_1^{\mu} - x_2^{\mu} & \cdots & y_1^{\mu} - x_j^{\mu} & \cdots & y_1^{\mu} - x_n^{\mu} \\ y_2^{\mu} - x_1^{\mu} & y_2^{\mu} - x_2^{\mu} & \cdots & y_2^{\mu} - x_j^{\mu} & \cdots & y_2^{\mu} - x_n^{\mu} \\ \vdots & \vdots & & \vdots & & \vdots \\ y_i^{\mu} - x_1^{\mu} & y_i^{\mu} - x_2^{\mu} & \cdots & y_i^{\mu} - x_j^{\mu} & \cdots & y_i^{\mu} - x_n^{\mu} \\ \vdots & \vdots & & \vdots & & \vdots \\ y_m^{\mu} - x_1^{\mu} & y_m^{\mu} - x_2^{\mu} & \cdots & y_m^{\mu} - x_j^{\mu} & \cdots & y_m^{\mu} - x_n^{\mu} \end{pmatrix} \tag{9}$$

2. Equation (4) is applied to the p matrices to obtain the morphological memory W, as shown in Equation (10).

$$W = \bigwedge_{\mu=1}^{p} \left[y^{\mu} \bigtriangledown (-x^{\mu})^{t} \right] = \left[w_{ij} \right]_{m \times n}$$

$$w_{ij} = \bigwedge_{\mu=1}^{p} \left(y_i^{\mu} - x_j^{\mu} \right) \tag{10}$$

The recall phase consists of applying the maximum product as shown in Equation (7) between memory W and the input pattern x^{ω}, where $\omega \in \{1, 2, \ldots, p\}$ so as to obtain a column vector of dimension m, as shown in Equation (12).

$$y = W \bigtriangledown x^{\omega} \tag{11}$$

Note that, the i-th component of the vector y is:

$$y_i = \bigvee_{j=1}^{n} \left(w_{ij} + x_j^{\omega} \right) \tag{12}$$

1.2. $\alpha\beta$ Associative Memories

The $\alpha\beta$ associative memories are fundamentally based on the maximum and minimum of order relationships between the patterns. Both the α operator–used at the learning phase– and the β operator used at the recall phase were defined. These memories work for both the autoassociative and heteroassociative modes [34,35,45].

$\alpha\beta$ memories require two sets to work, that is, the sets A and B which are defined as $A = \{0, 1\}$ y $B = \{0, 1, 2\}$.

The binary $\alpha = A \times A \longrightarrow B$ operation is defined by Table 1.

Table 1. The binary α operation.

x	y	$\alpha(x, y)$
0	0	1
0	1	0
1	0	2
1	1	1

The binary $\beta = B \times A \longrightarrow A$ operation is defined by Table 2.

Table 2. The binary β operation.

x	y	$\beta(x, y)$
0	0	0
0	1	0
1	0	0
1	1	1
2	0	1
2	1	1

The learning phase for $\alpha\beta \bigvee$ memories consists of two stages:

1. In each of the p associations (x^{μ}, y^{μ}), Table 1 is applied to build memory $y^{\mu} \amalg_{\alpha} (x^{\mu})^t$ of dimension $m \times n$, where the transpose of the input pattern x^{μ} is defined as $(x^{\mu})^t = (x_1^{\mu}, x_2^{\mu}, \ldots, x_n^{\mu})$, and the $\amalg_{\alpha}$ operator refers to the order relationship of the alpha operator. This expression develops as shown below:

$$y^{\mu} \amalg_{\alpha} (x^{\mu})^t = \begin{pmatrix} y_1^{\mu} \\ y_2^{\mu} \\ \vdots \\ y_m^{\mu} \end{pmatrix} \amalg_{\alpha} \left(x_1^{\mu}, x_2^{\mu}, \ldots, x_n^{\mu} \right)$$

$$y^{\mu} \amalg_{\alpha} x^t = \begin{pmatrix} \alpha(y_1, x_1) & \alpha(y_1, x_2) & \cdots & \alpha(y_1, x_n) \\ \alpha(y_2, x_1) & \alpha(y_2, x_2) & \cdots & \alpha(y_2, x_n) \\ \vdots & \vdots & & \vdots \\ \alpha(y_m, x_1) & \alpha(y_m, x_2) & \cdots & \alpha(y_m, x_n) \end{pmatrix} \tag{13}$$

2. The $\bigvee$ operator applies to the p matrices obtained from the expression (13) to create the memory V.

$$\mathbf{V} = \bigvee_{\mu=1}^{p} \left[y^{\mu} \amalg_{\alpha} (x^{\mu})^t \right]$$

with the ij-th component of v being:

$$v_{ij} = \bigvee_{\mu=1}^{p} \alpha\left(y_i^\mu, x_j^\mu\right) \tag{14}$$

According to (14), in the operation $\alpha : A \times A \longrightarrow B$, we observed that $v_{ij} \in B, \forall i \in \{1, 2, \ldots, m\}, \forall j \in \{1, 2, \ldots, n\}$.

The recall phase consists of performing the $\mathbf{V} \Cap_\beta x^\omega$, where x^ω, with $\omega \in \{1, 2, \ldots, p\}$, is the input pattern to be recalled and $\mathbf{V}$ is the matrix obtained in (14). As a result, a column vector of dimension m is generated, whose ith-component can be obtained according to expression (15).

$$\left(\mathbf{V} \Cap_\beta x^\omega\right)_i = \bigwedge_{j=1}^{n} \beta\left\{\left[\bigvee_{\mu=1}^{p} \alpha\left(y_i^\mu, x_j^\mu\right)\right], x_j^\omega\right\}$$

$$\left(\mathbf{V} \Cap_\beta x^\omega\right)_i = \bigwedge_{j=1}^{n} \beta\left(v_{ij}, x_j^\omega\right) \tag{15}$$

Likewise, the learning phase for $\alpha\beta \bigwedge$ memories consists of two stages:

1. In each of the p associations (x^μ, y^μ), Table 1 is applied to build memory $y^\mu \Cap_\alpha (x^\mu)^t$ of dimension $m \times n$, where the transpose of the input pattern x^μ is defined as $(x^\mu)^t = (x_1^\mu, x_2^\mu, \ldots, x_n^\mu)$ and the $\Cap_\alpha$ operator refers to the order relationship of the alpha operator. This expression develops as shown below:

$$y^\mu \Cap_\alpha (x^\mu)^t = \begin{pmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_m^\mu \end{pmatrix} \Cap_\alpha \left(x_1^\mu, x_2^\mu, \ldots, x_n^\mu\right)$$

$$y^\mu \Cap_\alpha x^t = \begin{pmatrix} \alpha(y_1, x_1) & \alpha(y_1, x_2) & \cdots & \alpha(y_1, x_n) \\ \alpha(y_2, x_1) & \alpha(y_2, x_2) & \cdots & \alpha(y_2, x_n) \\ \vdots & \vdots & & \vdots \\ \alpha(y_m, x_1) & \alpha(y_m, x_2) & \cdots & \alpha(y_m, x_n) \end{pmatrix} \tag{16}$$

2. The $\bigwedge$ operator applies to the p matrices obtained from the expression (16) so as to create the memory Λ.

$$\Lambda = \bigwedge_{\mu=1}^{p} \left[y^\mu \Cap_\alpha (x^\mu)^t\right]$$

with the ij-th component of λ being:

$$\lambda_{ij} = \bigwedge_{\mu=1}^{p} \alpha\left(y_i^\mu, x_j^\mu\right) \tag{17}$$

The recall phase consists of performing the $\mathbf{V} \Cup_\beta x^\omega$, where x^ω, having $\omega \in \{1, 2, \ldots, p\}$, is the input pattern to be recalled and $\mathbf{V}$ is the matrix obtained in (17). As a result, a column vector of dimension m is generated, and its ith-component can be obtained according to expression (18).

$$\left(\mathbf{V} \Cup_\beta x^\omega\right)_i = \bigvee_{j=1}^{n} \beta\left\{\left[\bigwedge_{\mu=1}^{p} \alpha\left(y_i^\mu, x_j^\mu\right)\right], x_j^\omega\right\}$$

$$\left(\mathbf{V} \Cup_\beta x^\omega\right)_i = \bigvee_{j=1}^{n} \beta\left(v_{ij}, x_j^\omega\right) \tag{18}$$

1.3. Noise

Noise has played a crucial role in the statistical analysis of data and associative memories have been no exception [16,17,19,34,35,45]. For instance, in the case of image processing; image quality depends upon various factors, such as noise, temperature, and light. Given that noise is always present in images, it is necessary to reduce or eliminate it so as to insure reliable analysis [51]. Noise usually arises from the image acquisition and transmission processes [52] and its removal has always been a field of interest in image processing. There are recent and very important advances in filtering different types of noise for digital image processing [22,51–55], and these have been modeled in order to meticulously study them and thus control their effects [56].

Noise is typically modeled as a Gaussian distribution because the image acquisition and transmission–which are the dominant sources of noise–are continuous processes that involve physical phenomena like thermal agitation and the discrete nature of light which are responsible for the random nature of noise; and according to the famous Central Limit Theorem, the sum of a large number of random variables of any type, essentially noise, will always tend toward a Gaussian distribution [51,52,54]. As we mentioned earlier, noise is an element that alters information and makes it difficult to process, hence the importance of knowing its behavior. Regardless of its type, noise can be classified as either additive, subtractive, or mixed (salt and pepper). To exemplify this fact, Figure 2 intuitively shows these types of noise.

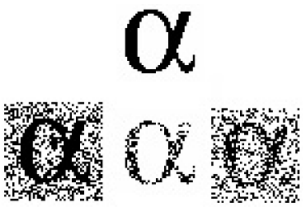


Figure 2. Additive noise, subtractive noise, and mixed noise, respectively.

In the case of associative memories based on minmax algebra, noise behavior is such an important factor. For example, min memories are inoperative with additive noise while max memories are inoperative with subtractive noise, nevertheless, both memories do not operate with mixed noise. Therefore, noise is such an interesting topic for minmax algebra-based memories. The objective of these memories is to address the behavior of the mixed noise because through this information one may accordingly build a kernel model that recovers the original pattern [16,17,21].The kernel model consists of finding out, in some way, a Z kernel that complies with $Z \subseteq X$, where X is the input pattern and as a condition, Z must not contain any kind of noise. For the above reason, the authors of the morphological memories affirm that the choice of Kernel is an open problem [16,21]. Note that, the kernel model is functional for associative $\alpha\beta$ memories as well.

Now, the learning and recall phases of the kernel model for associative memories in min and max algebra will be presented.

1. Learning phase: The diagram in Figure 3 shows the learning phase of the kernel model. As seen in the figure, the input pattern X enters a process that obtains $Z \subseteq X$, then, Z is autoassociatively learned with memory M; furthermore, Z is heteroassociatively learned with output pattern Y but this time with memory W.
2. Recall phase: Figure 4 shows the process followed when applying the recall phase in the kernel model. Given $\widetilde{X}$ as the mixed noise-distorted version of the learned pattern X, $\widetilde{X}$ is presented to memory M_{ZZ} and Z is recalled, immediately afterwards, Z is presented to Memory W_{ZY} and as a result the output pattern Y is recalled.

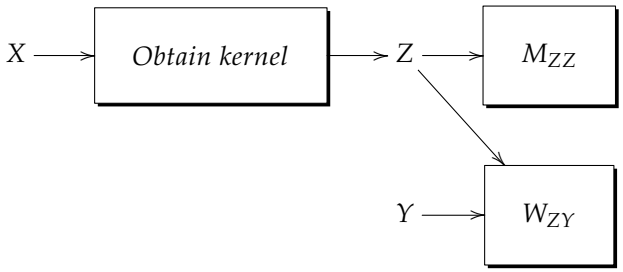


Figure 3. kernel model learning phase.

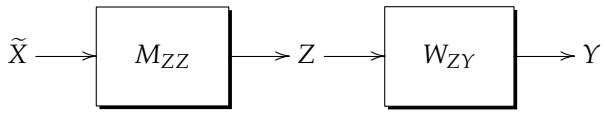


Figure 4. kernel model recall phase.

1.4. Fast Distance Transform (FDT)

A Distance Transform (DT) measures the distance in pixels from a pixel to the edge of the region of interest regardless the considered direction [57–63]. The DT allows finding useful geometric information within images; having this information, regions of interest can be found for many applications, including medicine etc. [57,60]. A widely used metric for the DT is the Euclidean distance for the extraction of geometric information [59,61,62], in fact, the algorithm of the Euclidean distance transform has been parallelized so that it may be processed even faster [59]. As one can see, TD is a very useful tool in image processing for data extraction. Another transform that is even faster than DT is the Fast Distance Transform (FDT) and it indeed generates interesting results [63]. Now, let us turn our attention to the FDT because it will be used for noise modeling.

The FDT algorithm consists of 2 steps [63], namely:

1. Read each pixel in the binary image from top to bottom and from left to right, then, each pixel $c \in R$, where R is the region of interest, is assigned as presented in Equation (19). Algorithm 1 illustrates the pseudocode of this same Equation (19).

$$\delta(c) = 1 + min(\delta(p_j) : p_j \in E) \tag{19}$$

E is one of the following sets shown in Figure 5. Only the points assigned in E are used in the first part of the transformation.

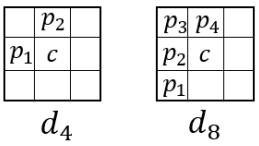


Figure 5. d_4 and d_8 metrics for the first step.

2. Read the binary image from bottom to top and from right to left, then, each pixel $c \in R$, where R is the region of interest, is assigned as shown in Equation (20). Algorithm 2 illustrates the pseudocode of this same Equation (20).

$$\delta(c) = min\{\delta(c), 1 + min\{\delta(p_i) : p_i \in D\}\} \tag{20}$$

D is one of the sets shown in Figure 6. Note that, only the points assigned in D are used in the first part of the transformation.

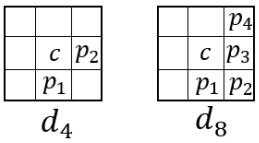


Figure 6. d_4 and d_8 metrics for the second step.

Figure 7 illustrates the result of the two steps of the FDT.

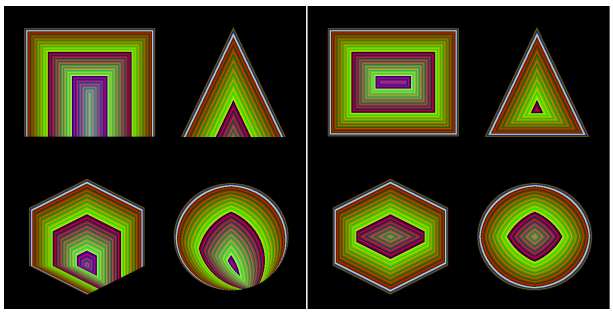


Figure 7. Result of the two steps of the FDT.

Algorithm 1 FDT algorithm first step with the d_8 metrics.

```
for y = 1 to binaryImage.height −1 do
    for x = 1 to binaryImage.width −1 do
        a = binaryImage.getPixel(x,y);
        if a = 1 then
            b = binaryImage.getPixel(x,y−1);
            c = binaryImage.getPixel(x−1,y);
            d = binaryImage.getPixel(x−1,y−1);
            e = binaryImage.getPixel(x−1,y−1);
            if b < a then
                a = b;
            end
            if c < a then
                a = c;
            end
            if d < a then
                a = d;
            end
            if e < a then
                a = e;
            end
            a = a + 1;
            binaryImage.setPixel(x,y,a);
        end
    end
end
```

Algorithm 2 FDT algorithm second step with the d_8 metrics.

```
for y = binaryImage.height −1 to 1 do
    for x = binaryImage.width −1 to 1 do
        a = binaryImage.getPixel(x,y);
        if a = 1 then
            b = binaryImage.getPixel(x+1,y);
            c = binaryImage.getPixel(x,y+1);
            d = binaryImage.getPixel(x+1,y+1);
            e = binaryImage.getPixel(x−1,y+1);
            if b < a then
                a = b;
            end
            if c < a then
                a = c;
            end
            if d < a then
                a = d;
            end
            if e < a then
                a = e;
            end
            a = a + 1;
            if binaryImage.getPixel(x,y) < a then
                a = binaryImage.getPixel(x,y);
            end
            binaryImage.setPixel(x,y,a);
        end
    end
end
δ = binaryImage;
```

2. Materials and Methods

This section details the procedure followed to generate a kernel suitable for hetero-associative memories based on minmax algebra which are on their turn used to recall patterns altered by mixed noise. Furthermore, it presents the modeling of noise by means of the FDT.

2.1. Noise

When working with pattern recognition from an image perspective, it is assumed that the noise is distributed in the function domain,however, when working from a signal perspective, it is assumed that the noise is distributed in the range of the function and this is evidenced by the signal-to-noise ratio [51,55]; in fact, it makes more sense to talk about the signal-to-noise ratio than about the noise distribution in the domain, because where there is noise there must be a signal that carries it. In this work we assume that noise exists and is distributed both in the range and in the domain of the function, thus, the following hypothesis is presented: The noise is concentrated where the information exists and its distribution is proportional to the signal amplitude.

Since the noise is distributed where there is signal, the proposed model is based on signals distribution along the equipotential lines to the signal in their domain, that is, based on the distance transform.

It is assumed that each data acquisition device has its own probability distribution; consequently, a kernel based on the Fast Distance Transform can be created to minimize the effect of the data acquisition process.

To determine the noise distribution generated by acquisition devices in binary images, the following process was performed:

1. Print the binary image on paper.
2. Scan the image obtained from step 1, generating a new digital image.
3. Compare the new digital image with the original one and store the percentage difference.
4. Print the new digital image obtained in step 2.
5. Repeat steps 2 to 4, 15 times with 80 different images (40 binary images and 40 gray -scale images).

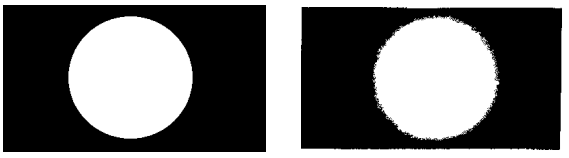
The characteristics of the binary images used to determine the noise distribution in the acquisition devices are as follow:

- 420×420 with 64 dpi resolution.
- 600×800 with 180 dpi resolution
- 542×700 with 96 dpi resolution.

To determine the percentage of variation between the scanned images and their originals, a pixel-by-pixel comparison was carried out. In case the pixels of both images had a different value in the coordinate (x, y), that implied the existence of a variation and this had to be taken into account; then, using a rule of three, the percentage of variation in the scanned image is calculated and corresponds to the acquired noise.

Figure 8 shows four among seven scanned images of a circle and highlights the similarity between images, however, there are very marked variations in the edges of the images; note how the frame is affected and also the contour of the circles; this is clearly seen in Figure 8d. This same image is compared to both the original image and to the scanned image number 7. The differences between images are green-colored. This result is significant because it allows us to conclude that: the noise that affects a binary image obtained from an electronic acquisition device is distributed at the edges; that is, the acquisition noise in a binary image arises, grows and distributes in a structured manner where there are significant gradient changes, besides being a mixed noise.

The images in Figure 8 reveal that, when performing the scanning process, the images obtained are subject to slight changes in scale (due to the configuration of the points per pixel -ppp- in the scanner) and in rotation (due to the placement of the printed image in the scanner bed); however, despite these technical drawbacks, when performing the aforementioned process, it can be guaranteed that: the further away the pixel is from the edge, the less likely it will be affected,and the higher the probability of being affect would be otherwise.



(**a**) Original image. (**b**) Scanned image 3.

(**c**) Scanned image 7. (**d**) Image difference.

Figure 8. Appearance of the 7-scan process images.

Now, we will describe the algorithms that allow the simulation of the acquisition noise distribution. Knowing the noise distribution allows us to determine the probability that it affects a certain pattern in specific parts, because this permits us to determine which parts

of the binary image that may serve as kernel. Algorithm 3 shows the process to obtain the acquisition noise probability distribution to binary images.

Algorithm 3 Noise probability distribution algorithm for binary images.

```
Obtain δ1(BinaryImage) and δ2(ScannedBinaryImage)

for x ← 0 to BinaryImage.width do
    for y ← 0 to BinaryImage.height do
        a ← BinaryImage(x, y)
        b ← ScannedBinaryImage(x, y)
        if a ≠ b then
            c ← δ1(x, y)
            his1[c] + +

for x ← 0 to BinaryImageNegative.width do
    for y ← 0 to BinaryImageNegative.height do
        a ← BinaryImageNegative(x, y)
        b ← ScannedBinaryImageNegative(x, y)
        if a ≠ b then
            c ← δ2(x, y)
            his2[c] + +

//Noise probability is generated
s1 ← s2 ← 0
for i ← 0 to his1.length do
    s1 ← s1 + his1[i]
    s2 ← s2 + his2[i]
s3 ← s1 + s2

for i ← 0 to his1.length do
    h1[i] ← his1[i]/s3
    h2[i] ← his2[i]/s3
```

$\delta 1$ represents the positive distances while $\delta 2$ represents the negative distances and there is no 0 distance. In the plots that will be shortly presented in the following sections, $\delta 1$ and $\delta 2$ depict both the x and y-axis along with their corresponding histogram frequencies of $\delta 1$ and $\delta 2$. Note that noise is distributed at the edges.

Since the noise distribution–as a function of the FDT has already been obtained–it is possible to simulate the noise by applying Algorithm 4.

Now we will proceed to detail the obtainment of the acquisition noise in gray-scale images as well as its simulation. The process to obtain the distribution of mixed noise in gray-scale images is described in Algorithm 5.

Since the distribution of the acquisition noise in gray-scale images has already been obtained, it is now possible to simulate it and Algorithm 6 shows how this is carried out.

So far the probability distribution of the acquisition noise has been obtained by means of the proposed image scanning process. Now, we will proceed to formally define noise.

Definition 3. *Let f be a function from P to A, that is, $f : P \to A$, the function affected by the noise is expressed by:*

$$f^* = f + r = \pi(t) + \psi(\tau(f)) + \kappa(P) \tag{21}$$

where:

- $\pi(t)$ *is a time-dependent random function of t and independent from* f.
- $\psi(\tau(f))$ *is a random function depending on a measure* τ *taken from the obtained data.*
- $\kappa(P)$ *is a p-dependent random function of* $p \in f$, *P-domain of the noisy information.*

$\pi(t)$ represents the transmission noise and is independent from the transmitted information. $\psi(\tau(f))$ is the acquisition noise that is based on a measure τ. $\kappa(P)$ is known as geometric noise.

Noise can be depicted as shown in Figure 9. Since $\pi(t)$ has been extensively treated in pattern recognition, which includes associative memories in minmax algebra, it will be left out of this paper. Moreover, $\kappa(P)$, which is the geometric noise introduced by the acquisition devices, will not be considered either. It is assumed that both $\pi(t)$ and $\kappa(p)$ are 0; therefore, the noise to be considered is phi $\psi(\tau(f))$.

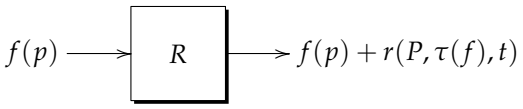


Figure 9. Noise scheme.

Algorithm 4 Mixed noise simulation algorithm for binary images.

Obtain $\delta 1(BinaryImage)$ and $\delta 2(BinaryImageNegative)$
$hh1 \leftarrow histogram(\delta 1)$
$hh2 \leftarrow histogram(\delta 2)$
$noisyImage \leftarrow BinaryImage$

/* Last noisy distance is assigned*/
$dd \leftarrow lastDistance$
for $d \leftarrow 0$ ***to*** dd **do**
    /* h1 represents vector of the noise probability by distance of region interest*/
    $pr \leftarrow h1[d] \times hh1[d]$ /* The pixel number to modify is obtained*/
    $c \leftarrow 0$
    while $c \leq pr$ **do**
        $x \leftarrow random(0, BinaryImage.width)$
        $y \leftarrow random(0, BinaryImage.height)$
        $a = \delta 1(x, y)$
        if $a = d$ **then**
            $noisyImage(x, y) \leftarrow 0$
            $c \leftarrow c + 1$
for $d \leftarrow 0$ ***to*** dd **do**
    // h2 represents vector of the noise probability by distance of complement
    $pr \leftarrow h2[d] \times hh2[d]$ /* The pixel number to modify is obtained*/
    $c \leftarrow 0$
    while $c \leq pr$ **do**
        $x \leftarrow random(0, BinaryImage.width)$
        $y \leftarrow random(0, BinaryImage.height)$
        $a = \delta 2(x, y)$
        if $a = d$ **then**
            $noisyImage(x, y) \leftarrow 1$
            $c \leftarrow c + 1$

Algorithm 5 Algorithm that obtains the probability distribution of the acquisition noise.

Input: grayScaleImage, scannedNoiseImage
Output: mixed noise distribution vectors. $p1, p2$

for $x \leftarrow 0$ ***to*** *grayScaleImage.width-1* **do**
  for $y \leftarrow 0$ ***to*** *grayScaleImage.height-1* **do**
    $a \leftarrow grayScaleImage(x, y)$
    $b \leftarrow scannedNoiseImage(x, y)$
    $a \leftarrow a - b$
    if $a > 0$ **then**
      $h1[a] \leftarrow h1[a] + 1$
    else if $a < 0$ **then**
      $h2[abs(a)] \leftarrow h2[abs(a)] + 1$

$$s3 \leftarrow \sum_{i=0}^{n-1} h1_i + \sum_{i=0}^{n-1} h2_i$$

for $i \leftarrow n - 1$ ***to*** 0 **do**
  $p2[i] = h2[i]/s3$

for $i \leftarrow 0$ ***to*** $n - 1$ **do**
  $p1[i] = h1[i]/s3$

return $p1, p2$

Algorithm 6 image with simulated mixed noise.

Input: A finite set $p1$ of additive noise probability distribution
Input: A finite set $p2$ of subtractive noise probability distribution
Output: noiseImage

$f1 \leftarrow$ concatenate $p1$ with $p2$
$d \leftarrow p1.length + p2.length$
$s \leftarrow 0, j \leftarrow 0, i \leftarrow 0;$

while $i < d$ *and* $j < d$ **do**
  if $s < i$ **then**
    $s \leftarrow s + f1[j++] \times d$
  else
    $D[i++] \leftarrow j - 256$

for $x \leftarrow 0$ ***to*** $x < OriginalImage.width$ **do**
  for $y \leftarrow 0$ ***to*** $x < OriginalImage.height$ **do**
    $r \leftarrow random(0, d)$
    $a \leftarrow OriginalImage.getPixel(x, y)$
    $a \leftarrow a + D[r]$
    $noiseImage.setPixel(x, y) \leftarrow a$

return *noiseImage*

Definition 4. *The probability that a point $p \in P$ is affected by noise r since its distance measure $\tau(p) = i$ is expressed as:*

$$P_r(p|\tau(p) = i) \tag{22}$$

where $\tau(p)$ represents a particular distance taken from the FDT affected by noise and obtained from $\psi(\tau(f))$.

Lemma 1. *Let $P_r[(p|\tau(p) = d_1) \cap (p|\tau(p) = d_2)] = 0$ if $d_1 \neq d_2$*

Proof. By contradiction; suppose that, $P_r[(p|\tau(p) = d_1) \cap (p|\tau(p) = d_2)] \neq 0$, then, there is a noise event in p with $\tau(p) = d_1$ and $\tau(p) = d_2$, but τ is a measure, therefore it is a mapping and does not have different values. □

Corollary 1. *$P_r(p|\tau(p)) = d_1$ and $P_r(p|\tau(p)) = d_2$ are independent events.*

Proof. Direct consequence of the Lemma 1. Since τ is a distance measure, the probability that an event in p will affect the noise at this distance is unique. However, the only way to affect a different distance is through another noise probability event; therefore, $P_r(p|\tau(p)) = d_1$ is independent from $P_r(p|\tau(p)) = d_2$. □

Corollary 2. $P_r[(p|\tau(p) = d_1) \cup (p|\tau(p) = d_2)] = P_r[(p|\tau(p) = d_1) + (p|\tau(p) = d_2)]$.

Proof. Corollary 1 showed that $(p|\tau(p) = d_1)$ is an independent event from $(p|\tau(p) = d_2)$; that is, the probabilities that a noise event in p will affect two different distances at different times are different; this indicates that the union of the two probabilities is the sum of both probabilities; therefore,

$P_r[(p|\tau(p) = d_1) \cup (p|\tau(p) = d_2)] = P_r[(p|\tau(p) = d_1) + (p|\tau(p) = d_2)]$ □

Lemma 2. $p_r\left(\bigcup_{d=d_1}^{d_2}(p|\tau(p) = d)\right) = \sum_{d=d_1}^{d_2} P_r(p|\tau(p) = d)$.

Proof. By Lemma 1 and Corollary 2 we have:

$p_r\left(\bigcup_{d=d_1}^{d_2}(p|\tau(p) = d)\right) = \sum_{d=d_1}^{d_2} P_r(p|\tau(p) = d)$. □

Theorem 1. $P_r(p|d_1 \leq \tau(p) \leq d_2) = \sum_{d=d_1}^{d2} P_r(p|\tau(p) = d)$.

Proof.

$$P_r(p|d_1 \leq \tau(p) \leq d_2) = P_r(\bigcup_{d=d_1}^{d_2}(p|\tau(p) = d).$$

Then by Lemma 2 we have:

$P_r(p|d_1 \leq \tau(p) \leq d_2) = \sum_{d=d_1}^{d_2} P_r(p|\tau(p) = d) = \bigcup_{d=d_1}^{d_2}(p|\tau(p) = d)$. □

Corollary 3. $P_r(p| - d_1 \leq \tau(p) \leq d_1) = \sum_{d=d_1}^{d_2} P_r(p|\tau(p) = d)$

Proof. Direct consequence of Theorem 1 with $d_1 = -d_1$ and $d_2 = d_1$. □

Corollary 4. $P_r^-(p|d_1 \leq \tau(p) \leq d_2) = 1 - \sum_{d=d_1}^{d_2} P_r(p|\tau(p) = d)$, *where P_r^- refers to the complementary probability to P_r.*

Proof.

$$\begin{aligned} 1 &= \sum_{d=-\infty}^{\infty} P_r(p|\tau(p)=d) \\ &= \sum_{d=-\infty}^{d_1-1} P_r(p|\tau(p)=d) + \sum_{d=d_1}^{d_2} P_r(p|\tau(p)=d) + \sum_{d=d_2+1}^{\infty} P_r(p|\tau(p)=d) \\ 1 &= P_r(p|\tau(p)<d_1 \vee \tau(p)>d_2) + P_r(p|d_1 \leq \tau(p) \leq d_2) \end{aligned}$$

therefore:
$P_r^-(p|d_1 \leq \tau(p) \leq d_2) = 1 - \sum_{d=d_1}^{d_2} P_r(p|\tau(p)=d)$. □

Lemma 3. *$P_r(p|r$ is additive$) = \sum_{d=-1}^{-\infty}(p|\tau(p)=d)$*

Proof. By definition, additive noise exists in the complement of the region; therefore $\tau(p)<0$ and
$P_r(p|r \text{ is additive}) = P_r(\bigcup_{d=-1}^{-\infty}(p|\tau(p)=d)) = \sum_{d=-1}^{-\infty}(p|\tau(p)=d)$. □

Lemma 4. *$P_r(p|r$ is subtractive$) = \sum_{d=1}^{\infty}(p|\tau(p)=d)$.*

Proof. By definition, subtractive noise exists in the region; therefore $\tau(p)>0$ and
$P_r(p|r \text{ is subtractive}) = P_r(\bigcup_{d=1}^{\infty}(p|\tau(p)=d)) = \sum_{d=1}^{\infty}(p|\tau(p)=d)$. □

2.2. Optimal Kernel Based on FDT

Given a function $\psi(\tau)$ that represents the acquisition noise distribution, let us assume the noise is distributed from the edges to their surroundings, Theorem 1 and Corollary 3 show that it is possible to find a range of distances from d_1 to d_2 of that distribution where the probability of having the noise affect this region is high. The kernel is constructed from the hypothesis that noise is distributed along the edges and it is enough to make erosions in order to eliminate the range between the distances d_1 and d_2 obtained from Theorem 1 and still preserve the remaining distances. When performing the erosions, the probable remaining noise can be seen as additive noise (if the kernel is built with the distances of the complement of the region) and for this type of noise, the max memories are robust; and if the kernel distances left over by erosion are those of the complement, then the noise is subtractive, and for this type of noise, min memories are the ones that are robust. It is assumed that when dealing with the complement of the region that will make up the kernel, singular features are preserved so they may be distinguishable from other patterns, thus reducing the risk that one or more patterns may be memorized by becoming a subset or superset of others.

Remark 1. *The erosion term used in this paper does not refer specifically to the morphological operation of erosion that is defined in the mathematical morphology, it does also refer to the arithmetic operation of subtraction performed between two gray levels of an image.*

Definition 5. *Given a function $\psi(\tau)$ and the distances d_1 (the distance that is likely to be affected by noise in the region complement) and d_2 (the distance that is likely to be affected by noise in the region) that satisfy $P_r(p|d_1 \leq \tau(p) \leq d_2)$ we will proceed to build the optimal binary kernel as follows:*

1. *Erode up to distance d_2 of $\delta 1$.*
2. *Binarize the eroded $\delta 1$.*
3. *Obtain the complement of the eroded image from step 2.*

Definition 6. *Given a function $\psi(\tau)$ and the distances d_1 and d_2 that were chosen to satisfy $P_r(p|d_1 \leq \tau(p) \leq d_2)$, we will proceed to build the grayscale optimal kernel as follows:*

1. *Erode the image.*
2. *Obtain the complement of the eroded image.*

Based on Theorem 1, Corollaries 1–4 and Lemmas 1–4 along with the acquisition noise distribution function $\psi(\tau)$, it is possible to propose a generic model of heteroasociative memories in minmax algebra that is robust to mixed noise.

Remark 2. *Since morphological and $\alpha\beta$ memories use minimum and maximum operators in the learning and recalling phases, morphological memories will be taken as a basis to make the necessary demonstrations for the creation of the generic model of min heteroasociative memory that is robust to mixed noise.*

The new generic model is defined as follows:
Let A be a matrix $\left[a_{ij}\right]_{m\times r}$ and B a matrix $\left[b_{ij}\right]_{r\times n}$ whose terms are integers.

Definition 7. *The maximum product of A and B, denoted by $C = A \vee B$, is a matrix $\left[c_{ij}\right]_{m\times n}$ whose ij-th component c_{ij} is defined as:*

$$c_{ij} = \bigvee_{k=1}^{r} \left(a_{ik} + b_{kj}\right)$$

2.2.1. Learning Phase

The heteroassociative memory in min algebra meant for the learning phase is constructed as follows:

$$W = \bigwedge_{\varrho=1}^{p} \left[y^{\varrho} \vee \left(-x^{\varrho}\right)^{t}\right] = \left[w_{ij}\right]_{m\times n} \tag{23}$$

$$w_{ij} = \bigwedge_{\varrho=1}^{p} \left(y_i^{\varrho} - x_j^{\varrho}\right) \tag{24}$$

2.2.2. Recall Phase

The recall phase consists of applying the Definition 7 between min memory and the input pattern x^{ϑ}, where $\vartheta \in \{1, 2, \ldots, p\}$ in order to get a column vector of m dimension:

$$y = W \vee x^{\vartheta} \tag{25}$$

where the i-th component of the vector y is:

$$y_i = \bigvee_{j=1}^{n} \left(w_{ij} + x_j^{\vartheta}\right) \tag{26}$$

Remark 3. *Typically, min memories are robust to subtractive noise while max memories are robust to additive noise, but both are inoperative when it comes to mixed noise; for this reason the kernel model arises in an effort to solve the mixed noise problem. The demonstration of the above argument will be omitted in this paper since it is well detailed in [16,45]. We will only focus on demonstrating that through the generation of a kernel based on the noise distribution, one can create a min heteroasociative memory that is robust to mixed noise, which is the original contribution of this paper.*

Theorem 2. *Let $\widetilde{x}^{\vartheta}$ $\forall \vartheta = 1, \ldots, k$ be the distorted version of the pattern x^{ϑ}. $W \vee \widetilde{x}^{\vartheta} = y^{\vartheta}$ will be true if and only if*

$$\widetilde{x}_j^{\vartheta} \leq x_j^{\vartheta} \quad \forall j = 1, \ldots, n \tag{27}$$

for each index row $i \in \{1, \ldots, m\}$ an index column exists $j_i \in \{1, \ldots, n\}$ such that:

$$\widetilde{x}_{j_i}^{\vartheta} = x_{j_i}^{\vartheta} \vee \left(\bigvee_{\varrho \neq \vartheta} [y_i^{\vartheta} - y_i^{\varrho} + x_{j_i}^{\varrho}] \right) \tag{28}$$

Proof. Suppose $\widetilde{x}^{\vartheta}$ denotes the distorted version of x^{ϑ} and $\forall \vartheta = 1, \ldots, k$, $W \vee \widetilde{x}^{\vartheta} = y^{\vartheta}$, then:

$$y_i^{\vartheta} = (W \vee \widetilde{x}^{\vartheta})_i = \bigvee_{l=1}^{n} (w_{il} + \widetilde{x}_l^{\vartheta}) \geq w_{ij} + \widetilde{x}_j^{\vartheta} \quad \forall i = 1, \ldots, m \text{ and } \forall j = 1, \ldots, n \tag{29}$$

thus,

$$\begin{aligned}
& \widetilde{x}_j^{\vartheta} \leq y_i^{\vartheta} - w_{ij} \; \forall i = 1, \ldots, m \text{ y } \forall j = 1, \ldots, n && (30)\\
\Leftrightarrow \quad & \widetilde{x} \leq \bigwedge_{i=1}^{m} (y_i^{\vartheta} - w_{ij}) \; \forall j = 1, \ldots, n \\
\Leftrightarrow \quad & \widetilde{x}_j^{\vartheta} \leq \bigwedge_{i=1}^{m} [y_i^{\vartheta} - \bigwedge_{\varrho=1}^{k} (y_i^{\varrho} - x_j^{\varrho})] \; \forall j = 1, \ldots, n \\
\Leftrightarrow \quad & \widetilde{x}_j^{\vartheta} \leq \bigwedge_{i=1}^{m} [y_i^{\vartheta} + \bigvee_{\varrho=1}^{k} (x_j^{\varrho} - y_i^{\varrho})] \; \forall j = 1, \ldots, n \\
\Leftrightarrow \quad & \widetilde{x}_j^{\vartheta} \leq \bigwedge_{i=1}^{m} [y_i^{\vartheta} + \bigvee_{\varrho \neq \vartheta} (x_j^{\varrho} - y_i^{\varrho}) \vee (x_j^{\vartheta} - y_i^{\vartheta})] \; \forall j = 1, \ldots, n \\
\Leftrightarrow \quad & \widetilde{x}_j^{\vartheta} \leq \bigwedge_{i=1}^{m} [\bigvee_{\varrho \neq \vartheta} (y_i^{\vartheta} - y_i^{\varrho} + x_j^{\varrho}) \vee x_j^{\vartheta}] \; \forall j = 1, \ldots, n \\
\Leftrightarrow \quad & \widetilde{x}_j^{\vartheta} \leq x_j^{\vartheta} \vee \bigwedge_{i=1}^{m} [\bigvee_{\varrho \neq \vartheta} (y_i^{\vartheta} - y_i^{\varrho} + x_j^{\varrho})] \geq x_j^{\vartheta} \quad \forall j = 1, \ldots, n \\
\Leftarrow \quad & \widetilde{x}_j^{\vartheta} \leq x_j^{\vartheta}
\end{aligned}$$

This shows that the inequality obtained in (27) is sufficient for $\widetilde{x}_j^{\vartheta}$ to be recovered. then,

$$\widetilde{x}_j^{\vartheta} \leq x_j^{\vartheta} \vee [\bigvee_{\varrho \neq \vartheta} (y_i^{\vartheta} - y_i^{\varrho} + x_j^{\varrho})] \; \forall j = 1, \ldots, n \text{ and } \forall i = 1, \ldots, m. \tag{31}$$

Now, suppose the set obtained in (31) does not contain the equivalence for $i = 1, \ldots, m$, i.e., it is assumed that there are indices of row $i \in \{1, \ldots, m\}$ such that:

$$\widetilde{x}_j^{\mu} < x_j^{\vartheta} \vee \left[\bigvee_{\varrho \neq \vartheta} (y_i^{\vartheta} - y_i^{\varrho} + x_j^{\varrho}) \right] \; \forall j = 1, \ldots, n \tag{32}$$

then

$$\begin{aligned}(W \vee \widetilde{x}^{\vartheta})_i &= \bigvee_{j=1}^{n} (w_{ij} + \widetilde{x}_j^{\vartheta}) \\ &< \bigvee_{j=1}^{n} \left[w_{ij} + x_j^{\vartheta} \vee \Big(\bigvee_{\varrho \neq 1} [y_i^{\vartheta} - y_i^{\varrho} + x_j^{\varrho}] \Big) \right] \\ &= \bigvee_{j=1}^{n} \left[w_{ij} + \Big(\bigvee_{\varrho=1}^{k} [y_i^{\vartheta} - y_i^{\varrho} + x_j^{\varrho}] \Big) \right] \\ &= \bigvee_{j=1}^{n} [w_{ij} + y_i^{\vartheta} - \bigwedge_{\varrho=1}^{k} (y_i^{\varrho} - x_j^{\varrho})] \\ &= \bigvee_{j=1}^{n} [w_{ij} + y_i^{\vartheta} - w_{ij}] \\ &= y_i^{\vartheta}. \end{aligned} \tag{33}$$

Thus, $(W \vee \widetilde{x}^{\vartheta})_i < y_i^{\vartheta}$ which contradicts the hypothesis that $W \vee \widetilde{x}^{\vartheta} = y^{\vartheta}$. This indicates that for each row index there must be a column index of j_i that satisfies (28).

The opposite will now be proofed. Suppose that

$$\widetilde{x}_j^{\vartheta} \leq x_j^{\vartheta} \vee \bigwedge_{i=1}^{m} \Big(\bigvee_{\varrho \neq \vartheta} [y_i^{\vartheta} - y_i^{\varrho} + x_j^{\varrho}] \Big) \; \forall j = 1, \ldots, n \tag{34}$$

for the first part of the proof, the inequality is true if and only if

$$\widetilde{x}_j^{\vartheta} \leq y_i^{\vartheta} - w_{ij} \; \forall i = 1, \ldots, m \; y \; \forall j = 1, \ldots, n \tag{35}$$

or, equivalently, if and only if

$$\begin{aligned} & w_{ij} + \widetilde{x}_j^{\vartheta} \leq y_i^{\vartheta} \; \forall j = 1, \ldots, m \; y \; \forall i = 1, \ldots, n \\ \Leftrightarrow \quad & \bigvee_{j=1}^{n} (w_{ij} + \widetilde{x}_j^{\vartheta}) \leq y_i^{\vartheta} \; \forall i = 1, \ldots, m \\ \Leftrightarrow \quad & (W_{XY} \vee \widetilde{X})_i \leq y_i^{\vartheta} \; \forall i = 1, \ldots, m \end{aligned} \tag{36}$$

this implies that $W_{XY} \vee \widetilde{x}_{\vartheta} \leq y_{\vartheta} \; \forall \vartheta = 1, \ldots, k$ therefore, if it is proven that $W_{XY} \vee \widetilde{x}^{\vartheta} \geq y_{\vartheta} \; \forall \vartheta = 1, \ldots, k$, then as a result $W_{XY} \vee \widetilde{x}^{\vartheta} = y^{\vartheta} \; \forall \vartheta$. Now, let $\vartheta \in \{1, \ldots, k\}$ and $i \in \{1, \ldots, m\}$ be arbitrarily chosen, then

$$
\begin{aligned}
(W_{XY} \vee \widetilde{x}^{\vartheta})_i &= \bigvee_{j=1}^{n} (w_{ij} + \widetilde{x}_j^{\vartheta}) \\
&\geq w_{ij_i} + \widetilde{x}_{j_i}^{\vartheta} \\
&= w_{ij_i} + \left(x_{j_i}^{\vartheta} \vee \bigvee_{\varrho \neq 1} [y_i^{\vartheta} - y_i^{\varrho} + x_{j_i}^{\varrho}] \right) \\
&= w_{ij_i} + \bigvee_{\varrho=1}^{k} [y_i^{\vartheta} - y_i^{\varrho} + x_{j_i}^{\varrho}] \\
&= w_{ij_i} + y_i^{\vartheta} - \bigwedge_{\varrho=1}^{k} (y_i^{\varrho} - x_{j_i}^{\varrho}) \\
&= w_{ij_i} + y_i^{\vartheta} - w_{ij_i} \\
&= y_i^{\vartheta}
\end{aligned} \tag{37}
$$

This shows that $W_{XY} \vee \widetilde{x}^{\vartheta} \geq y^{\vartheta}$. □

Remark 4. *Expression* (27) *shows that the new min heteroasociative memory model is robust to mixed noise and is directly related to acquisition noise.*

Theorem 3. *The new min heteroasociative memory model is robust to mixed noise in a parameterized way by d within $\psi(\tau)$ and it is true that $E(d) \geq 1 - \sum_{d}^{\infty} P_r(p|\tau(p) = i)$ for $d > d_1$ where $E(d)$ is the probability of success in the complete recall of altered patterns with mixed noise.*

Proof. Lemma 4 shows that subtractive noise is located on the positive side of the $\psi(\tau)$ curve which is expressed as $P_r(p|r \text{ being subtractive}) = \sum_{d=1}^{\infty}(p|\tau(p) = d)$. It has been determined that the noise is distributed across the edges. However, by performing pattern erosion and obtaining the complement, the model gets robust to mixed noise from a distance d_1 where $d_1 < 0$. Thus, the probability of success in the recall of those patterns affected by mixed noise in this new model of associative memories W is expressed as $1 - \sum_{i=d_1}^{\infty} P_r(p|\tau(p) = i$ where $d_1 < 0$. On the other hand, Theorem 2 showed that it is a sufficient condition for the patterns recovery if the following condition is fulfilled, i.e., $\widetilde{x}_j^{\vartheta} \leq x_j^{\vartheta} \quad \forall j = 1, \ldots, n$ and expression (28) guarantees that for each row index i there must be a column index of j_i so that a complete recall may occur; this implies that the W heteroassociative memory model, as such, is robust to high percentages of mixed noise; therefore, it can be demonstrated that:

$$
E(d) \geq 1 - \sum_{d}^{\infty} P_r(p|\tau(p) = i) \text{ for } d > d_1 \tag{38}
$$

□

Corollary 5. *The probability of full recall of the new heteroassociative memory W is* 0*, if and only if, when parameterizing by d, $\sum_{-\infty}^{-d} P_r(p|\tau(p) = i)$ holds.*

Proof. Direct consequence of Theorem 3: Given that $E(d) \geq 1 - \sum_{d}^{\infty} P_r(p|\tau(p) = i)$ and d is negative, the same expression can be presented as $E(d) \geq 1 - \sum_{-d}^{\infty} P_r(p|\tau(p) = i)$. However, the complement of $E(d)$ is expressed as $\sum_{-\infty}^{-d} P_r(p|\tau(p) = i)$, which indicates that if the above is true, then, there is a 100% probability that the memory will fail. □

Corollary 6. *The new heteroassociative memory model W with mixed noise, may fail in full pattern recall if the noise is sufficient enough to turn X^{ϑ} pattern into a subset of another pattern X^{γ} where $\vartheta \neq \gamma$.*

Proof. Direct consequence of not complying with expression (28). □

The Corollary 6 is of utmost importance, since it is sufficient to have row index i in memory W that does not contain a column index j of X for the recall to be incomplete. This implies that in case $\widetilde{y}^{\varrho} \subset X^{\vartheta} \cup X^{\gamma}$, where $\vartheta \neq \gamma$, is not fulfilled, the pattern shall contain subtractive noise.

2.2.3. New Generic Model of Min Heteroassociative Memories Robust to Mixed Noise

Given an acquisition noise distribution function $\psi(\tau)$, where it is highlighted that the noise is distributed over distances close to distance 0, i.e., by the edges; and taking Theorem 3 as a reference, then, we will proceed to propose another novel model of min heteroassociative memory that is robust to mixed noise.

Learning phase.

1. Obtain $Z \subset X$ by $\psi(\tau)$ and Theorem 3.
2. Obtain the Z complement (Z^c).
3. Obtain the Y complement (Y^c).
4. Perform the learning process with $W_{Z^cY^c}$.

Figure 10 shows the learning process of the new model of min heteroasociative memory.

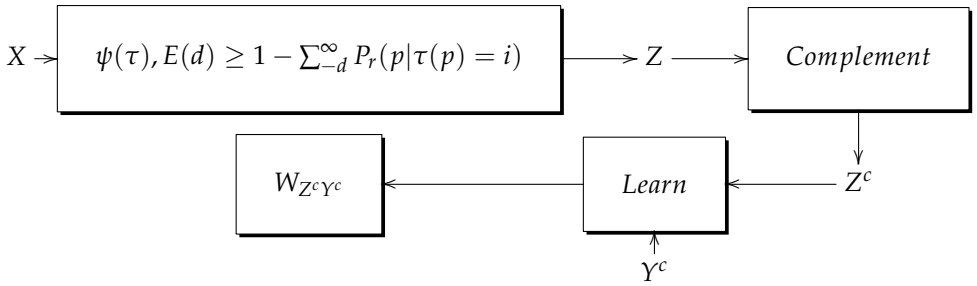


Figure 10. Learning process of the new model of min heteroassociative memories.

Recall phase.

1. Obtain the $\widetilde{X}$ complement.
2. Perform the recall process with memory $W_{Z^cY^c}$.
3. Obtain the Y^c complement.

Figure 11 graphically shows the recall process of the new min heteroassociative memories model that is robust to mixed noise

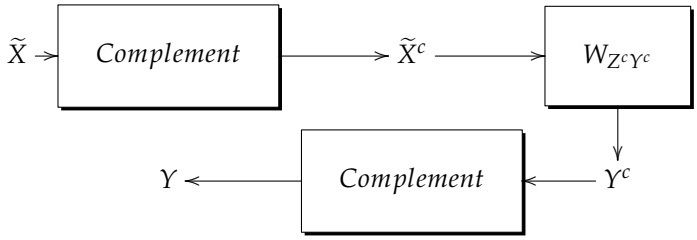


Figure 11. Recall process of the new model of min heteroassociative memories.

3. Results

In this section we will show the results of the acquisition noise distribution along with the acquisition noise simulation algorithms in binary and grayscale images, and finally the behavior of the new min heteroassociative memory model for morphological memories and $\alpha\beta$.

3.1. Acquisition Noise Distribution

3.1.1. Acquisition Noise Distribution in Binary Images

The final acquisition noise distribution function is obtained by averaging the 40 noise distributions obtained from the process described in Section 2.1. Table 3 shows the distribution of acquisition noise in binary images.

Negative distances shown in Figure 12 represent the subtractive noise while the positive distances make up the additive noise. This indicates that additive noise occurs with greater intensity than subtractive noise encountered in acquisition devices. Furthermore, it can be observed that the noise occurs at the edges and the greater the distance, the lower the noise effect. Also, the highest noise concentration is found at distances 1 and −1, and approximately 50% of noise is concentrated at the edges.

Table 3 shows 3 columns. The Distances column indicates that the noise distribution is from distance −20 to distance 20. The Absolute frequency column indicates the number of pixels that are affected at each distance in the range of −20 to 20. The Relative frequency column indicates the probability that noise will affect that distance. The sum of the Relative frequency column is 1 which indicates 100% of the noise affects the binary image.

According to Table 3 and Figure 12 one can observe that the acquisition noise is presented and distributed along the edge and, on top of this, it grows in a structured way as it moves away from the edge.

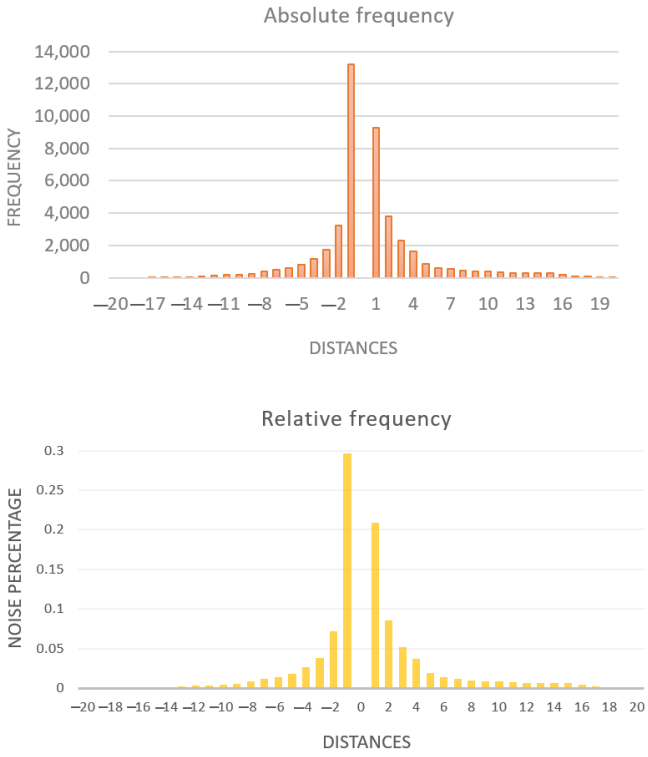


Figure 12. Absolute and relative frequency distributions of noise acquisition in binary images.

Table 3. Acquisition noise distribution table in binary images.

Distance	Frequency	Probability	Distance	Frequency	Probability
−20	0	0.0	1	9292	0.2084483
−19	0	0.0	2	3826	0.08582901
−18	0	0.0	3	2301	0.051618546
−17	3	0.00000729928	4	1649	0.03699217
−16	10	0.000022433093	5	830	0.018619467
−15	18	0.000040379568	6	619	0.013886085
−14	42	0.000094218994	7	535	0.012001705
−13	79	0.0017722144	8	445	0.009982727
−12	123	0.0027592704	9	391	0.008771339
−11	162	0.003634161	10	382	0.008569442
−10	194	0.00435202	11	338	0.0075823856
−9	238	0.0053390763	12	288	0.006460731
−8	397	0.008905938	13	275	0.0061691008
−7	512	0.011485743	14	270	0.006056935
−6	595	0.013347691	15	266	0.0059672026
−5	823	0.018462436	16	197	0.0044193193
−4	1172	0.026291585	17	95	0.0021311438
−3	1712	0.038405456	18	56	0.0012562532
−2	3212	0.072055094	18	56	0.0012562532
−1	13,204	0.29620656	19	24	0.000053839426
0	0	0	20	2	0.0000044866185

Table 3 shows the percentages of noise probability by distance. It is possible to simulate the acquisition noise in binary images and the result is shown in Figure 13. One can see that the noise is distributed at the edge.

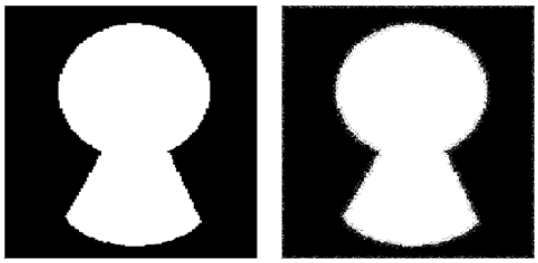


Figure 13. Binary image with simulated acquisition noise.

Acquisition noise grows and is distributed proportionally according to the size of the image, for example, in scaled images, noise is presented and distributed in the same way, but at different scales, keeping a relationship of growth against distribution. To confirm this, we experimented with images of size 120 × 120 that are scales of the images used in the present article, with the same noise distribution, but affecting less distance.

3.1.2. Acquisition Noise Distribution in Grayscale Images

By applying the Algorithm 5, the acquisition noise distribution in grayscale images was generated. This distribution is presented numerically in Tables A1 and A2 and graphically in Figure 14. Tables A1 and A2 show 3 columns, namely, the Distances column which represents the distances affected by the acquisition noise; the Frequency column which represents the number of pixels per distance affected by the noise; and the Probability column that represents the probability that noise will occur at the corresponding distance, whose sum is 1, that is, 100% noise should affect the grayscale image.

The final noise distribution in grayscale images is the result of averaging the 40 noise distributions generated by the process described in Section 2.1. One can see that the noise is distributed from distance −199 to distance 199 and there is no distance 0.

Remark 5. *Additive noise for binary and grayscale images is located in negative distances of graphs and tables while subtractive noise is located in positive distances.*

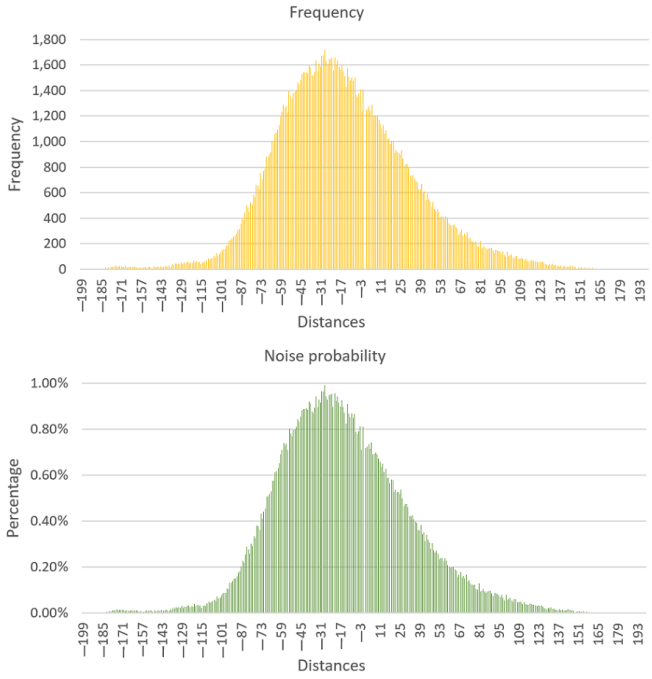


Figure 14. Process generating the noise distribution function.

Tables A1 and A2 define the distribution function of acquisition noise in grayscale images and by using these two tables in Algorithm 6, one can simulate this type of noise.

Figure 15 compares the noise distribution of the scanned image against that of the image with simulated noise. One can see that the distributions are very similar. The difference between these two images is that the image with simulated noise was generated having the same distribution as the one proposed in this article while the scanned image is just an image that was scanned once. Another difference is that the scanned image has geometric noise, that is, the noise in the form of a texture that was added by the acquisition device during the image scanning. It should be noted that each acquisition device has its own geometric noise which is different from other acquisition devices. However, the simulated acquisition noise distribution allows us to ensure that the acquisition noise is very close to the original.

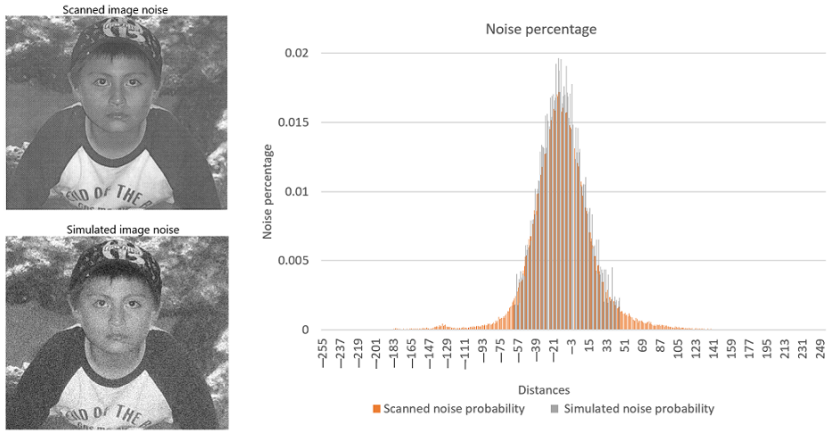


Figure 15. Scanned image vs simulated image noise distributions.

Having the results shown in the acquisition noise simulation, the images that will be used as patterns with the new min heteroassociative memory model will be altered by the acquisition noise simulation algorithms proposed in this article.

3.2. New Model of Min Heteroassociative Memory

This section presents the results of experiments conducted in order to show the effectiveness of the new min heteroassociative memory model. The experiments were carried out on a Dell XPS 8700 with Intel Core i7 processor and 16 Gigabytes of RAM. The images used for memory learning and retrieval were both binary and grayscale. The size of the images was 50×50, 80×80 and 120×120.

The process of learning and recalling patterns consisted of first learning the whole fundamental set with a kernel formed from distance i, then once this was learned, one pattern was chosen at a time, in order to be recalled 1000 times, thus resulting in the percentage of effectiveness in the complete patterns recall.

The construction of a kernel with sufficient conditions for pattern recall is based on the probability that noise is distributed around the edges, that is, the more we move away from distance 0, be it towards the left or right, the better the kernel. As mentioned earlier in Section 3.1.1, the acquisition noise grows and distributes proportionally according to the size of the image; and in scaled images, noise is generally presented and distributed in the same way, but at different scales, keeping the relationship of growth against distribution.

The main feature of the original kernel model is that the kernel must be free from noise so as to have a full recall. For this reason, our kernel construction was based upon the acquisition noise distribution proposed in this article.

Acquisition noise grows or decreases depending on the size of the image; to corroborate this, binary images were scaled from 420×420 to 50×50, that is, a reduction of 88%. When performing the process described in Section 2.1, it was observed that the distances affected by noise in the 50×50 images, on average were up to distance 2. Based on the above, to calculate the distances affected by the noise that was simulated in images used in this experiment, the operation $df = \frac{t_2}{t_1} \times d_{max}$ was defined, where df is the final distance that could be affected by noise. $t1$ represents the size of the original image, t2 being the size of the scaled image and $dmax$, the maximum distance in the noise distribution table; note that for binary images it corresponds to the maximum distance of Table 4, while for grayscale images, it is the maximum distance of Table A2. To illustrate this, we take $df = \frac{50}{420} = 0.12 \times 20 = 2.4$; therefore, one can conclude that, in this very image, noise will have no effect starting from distance 3. Taking df as a reference, images used in this experiment will probably no longer be affected by acquisition noise starting from distances that were shown in Table 4.

Table 4. Probable distances where the acquisition noise does not affect.

Binary Image			Grayscale Image		
Original Size	**New Size**	df	**Original Size**	**New Size**	df
420×240	50×50	3	420×240	50×50	24
420×240	80×80	4	420×240	80×80	38
420×240	120×120	6	420×240	120×120	56

Experiment characteristics:

- 6 fundamental sets, 3 with binary images and 3 with grayscale images. Figure 16 shows the fundamental sets appearance.
- The images of fundamental set 1 and 2 are of size 50×50, those of set 3 and 4 are 80×80, while those of set 5 and 6 are 120×120.
- Table 4 shows how far away the kernel will be created.
- 1000 recall process per fundamental set.

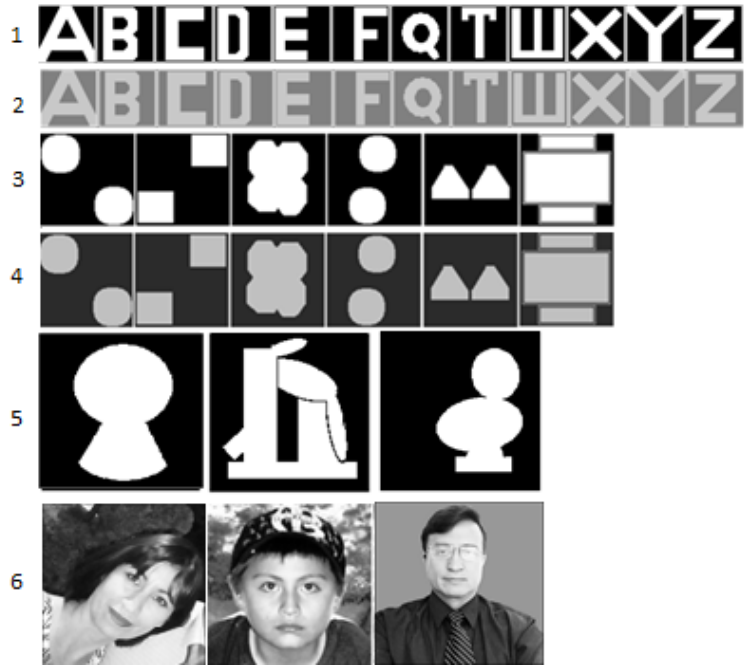


Figure 16. Fundamental sets.

The processes of building the kernels are shown in Figures 17 and 18. The morphological erosion used in the kernel construction is shown in Figure 18.

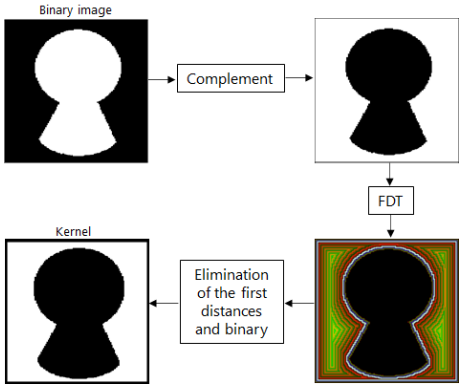


Figure 17. Operations to build binary kernels.

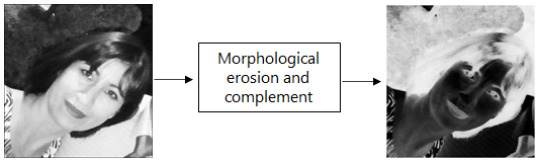


Figure 18. Operations to build grayscale kernels.

Applying the new model of min heteroasociative memory described in Section 2.2.3 along with the 6 fundamental sets illustrated above yields the results shown in Tables 5–10.

Table 5 shows the performance of fundamental set 1 of Figure 16. Fundamental set 1 is composed of binary images of size 50×50. It is noted that, if the patterns do not contain noise, both the original kernel model and the proposed new model have a 100% recall, that is, the two models are efficient for noiseless patterns. The original kernel model is efficient from distance 3 and this makes sense because by removing the first distances from the FDT, the kernel ends up being built outside the region that affects the acquisition noise, and for this same reason the new model proves to be efficient as well. However, the proposed new

model has full recalls from distance 1 even though the kernel may be affected by acquisition noise. The question is why, given the fact that the kernel is supposed to be noiseless. The answer is as follows:

Since it was determined that the distances affected by noise in 50×50 binary images are 1 and 2, it is noted in Table 3 that the sum of the probabilities of noise affecting these two distances is 37%, the new model proposes that the min heteroasociative memory is able to recover those patterns affected by mixed noise based on a noise distribution function; therefore, the new model indicates that, for a 50×50 image patterns, at least one should expect a 63% effectiveness in full pattern recall. The percentages in Table 4 represent how many times each pattern was fully recalled after 1000 attempts. For the new model on distance 1, recovery percentage per pattern is greater than 63%, so one can ensure that the new model could recall patterns where the original model could not. In the case of distance 2, as distance 1 was eliminated in kernel building, and according to Table 3, the probability of acquisition noise affecting distance 2 is 7%, hence the new model expects memory shall recall patterns by at least 93%, and Table 5 proves that to be true.

Table 5. Performance of the proposed model vs. original model in 50×50 binary images.

Original Model		**Morphological**		$\boldsymbol{\alpha\beta}$	
Pattern	$d = 3$	$d = 1$	$d = 2$	$d = 1$	$d = 2$
A	100%	77.00%	100%	76.90%	100%
B	100%	70.10%	100%	70.60%	100%
C	100%	93.00%	100%	93.00%	100%
D	100%	74.20%	100%	74.30%	100%
E	100%	78.00%	100%	77.90%	100%
F	100%	79.10%	100%	79.00%	100%
Q	100%	71.70%	100%	71.90%	100%
T	100%	79.00%	100%	79.00%	100%
W	100%	71.20%	100%	71.20%	100%
X	100%	70.00%	100%	70.20%	100%
Y	100%	70.10%	100%	70.00%	100%
Z	100%	69.80%	100%	69.50%	100%

Table 6 shows the performance of fundamental set 2 of Figure 16. Fundamental set 2 is composed of grayscale images of size 50×50. According to Table A2, the sum of probabilities that the noise will affect up to distance 20 is about 14%, thus, the probability that noise will have no effect is about 86%. These probability percentages are significant, because they corroborate that the proposed new model works as expected and this is proven in Table 6. The proposed model performance in pattern recall is greater than 86% from distance 18 and is 100% from distance 22. Original kernel model recalls from distance 24. The above proves that the proposed new model has better efficiency than original model.

Table 7 shows the performance of fundamental set 3 of Figure 16. Fundamental set 3 is composed of binary images of size 80×80. One can see that in Table 7 original kernel model has a performance of 100% from distance 4, and the performance is 0% before this distance. The proposed model has a higher performance of 69.90% in pattern recall from distance 1 and a 100% recall from distance 3. These results show that the new model is consistent, that is, it still shows better efficiency than original model.

Table 8 shows the performance of fundamental set 4 of Figure 16. Fundamental set 4 is composed of grayscale images of size 80×80. The images of fundamental set 4 are bigger than those of fundamental set 2, so the noise is higher and it affects more distances. For this reason, the original model has 100% performance from distance 38 onward. Obviously, if the kernel is noiseless, both models have 100% efficiency. The new model has complete recall percentages greater than 75% from distance 10, and from distance 30, its complete recall is 100%.

Table 6. Performance of the proposed model vs. original model in 50 × 50 grayscale images.

Original Model Kernel			**New Model**		
Pattern	No noise	$d = 24$	No noise	$d = 18$	$d = 22$
A	100%	100%	100%	86.10	100%
B	100%	100%	100%	87.00	100%
C	100%	100%	100%	92.20	100%
D	100%	100%	100%	90.10	100%
E	100%	100%	100%	86.30	100%
F	100%	100%	100%	87.20	100%
Q	100%	100%	100%	86.50	100%
T	100%	100%	100%	88.10	100%
W	100%	100%	100%	86.20	100%
X	100%	100%	100%	89.50	100%
Y	100%	100%	100%	91.30	100%
Z	100%	100%	100%	86.00	100%

Table 7. Performance of the proposed model vs. original model in 80 × 80 binary images.

Original Model		**Morphological**			**$\alpha\beta$**		
Pattern	$d = 4$	$d = 1$	$d = 2$	$d = 3$	$d = 1$	$d = 2$	$d = 3$
1	100%	78.10%	97.10%	100%	78.00%	97.00	100%
2	100%	75.20%	95.60%	100%	75.10%	95.60	100%
3	100%	77.10%	97.10%	100%	77.20%	97.20	100%
4	100%	78.40%	97.30%	100%	78.40%	97.20	100%
5	100%	69.90%	98.20%	100%	70.00%	98.40	100%
6	100%	80.00%	99.00%	100%	80.00%	99.00	100%

Table 8. Performance of the proposed model vs. original model in 80 × 80 grayscale images.

Original Model Kernel			**New Model**			
Pattern	No noise	$d = 38$	No noise	$d = 10$	$d = 20$	$d = 30$
1	100%	100%	100%	77.60	89.00	100%
2	100%	100%	100%	79.10	89.20	100%
3	100%	100%	100%	75.20	87.10	100%
4	100%	100%	100%	80.00	91.00	100%
5	100%	100%	100%	78.10	90.10	100%
6	100%	100%	100%	82.90	92.20	100%

Table 9 shows the performance of fundamental set 5 of Figure 16. Fundamental set 5 is composed of binary images of size 120 × 120. Table 9 shows the same behavior in patterns formed with binary images, and the bigger the size of the images, the greater the noise. As for the original model, it can recall 100% of the pattern if the kernel has no noise and this happens from distance 6. The new model has a performance greater than 77% from distance 1, and it can fully recall all learned patterns starting from distance 3.

Table 9. Performance of the proposed model vs. original model in 120 × 120 binary images.

	Original Model	**Morphological**					**$\alpha\beta$**
Pattern	$d = 6$	$d = 1$	$d = 2$	$d = 3$	$d = 1$	$d = 2$	$d = 3$
1	100%	80.00%	96.70%	100%	80.10%	96.60	100%
2	100%	77.80%	96.50%	100%	77.80%	95.80	100%
3	100%	81.70%	98.90%	100%	82.90%	98.20	100%

Table 10 shows the performance of fundamental set 6 of Figure 16. Fundamental set 6 is composed of grayscale images of size 120 × 120. Table 10 confirms what was commented

on the results of Table 10, but in this case for the fundamental set 6 which is made up of grayscale images. The original model completely recalls from distance 56 while the new model recalls from distance 10 with a performance greater than 77%. Again, it is proven that the new model has a better performance when compared to the original model.

Table 10. Performance of the proposed model vs original model in 120×120 grayscale images.

Original Model Kernel			**New Model**			
Pattern	No noise	$d = 56$	No noise	$d = 10$	$d = 30$	$d = 35$
1	100%	100%	100%	78.10	95.90	100%
2	100%	100%	100%	77.20	96.10	100%
3	100%	100%	100%	78.30	91.10	100%

We must provide high certainty that the proposed kernel is a subset of one and only one set for the recall phase to be complete. If the proposed kernel is made up of few elements, then the probability that the kernel is a subset of other sets is high, and as a result, the associative memory will fail. The more patterns the associative memory learns, the greater the probability that the kernel is a subset of another pattern. That is why the proposed new model is so important. The proposed model uses elements from those patterns that are certainly affected by noise, and it does not delete them as the original model does, however, there are chances of failing some pattern recoveries. The results show that the new model preserves distances deleted by the original model, with an efficiency greater than 70% in those distances where the original model is totally inoperative.

4. Discussion

For an associative memory to work as designed it must ensure that noise does not affect its operation. Knowing how acquisition noise presents itself will make associative memories know how to treat it. Associative memories in minmax algebra guarantee that if they meet the conditions of their design, they then have an infinite capacity to learn and recall.

This article showed that the acquisition noise has a Gaussian distribution where the maximum point is near distance 0, and these distances were obtained through a distance transform. Moreover, the probability that noise can affect these distances was obtained. These results allowed kernels to be generated with sufficient conditions for the original and new model to completely recall patterns affected by mixed noise.

A new model of min heteroassociative memory that is robust to mixed noise was also proposed. We affirm that the proposed new model is better than the original kernel model for two reasons: first, as shown in Figures 3 and 4, the original kernel model involves min memory and max memory for the learning and recall phases and it doesn't show how to get the Z kernel. Figure 10 shows that the new min heteroassociative memory model consists of only one memory, i.e., min memory, associated with a function that determines the distance from where Z will be formed, thus ensuring that there will be full pattern recalls. One can conclude that, the proposed new model is faster in its execution and proposes conditions that also satisfies the original model. Second reason, the original kernel model is deterministic, hence it sets where it will fail and where it will be efficient. The proposed new model is probabilistic, which gives it the advantage of recalling completely patterns where the original model cannot; and in those cases where the original model recalls completely, the new model guarantees a 100% complete recalls as well. As one can see, the new model surpasses the original model.

5. Conclusions

In this paper, it was proven that where there is information, there is acquisition noise, and this noise grows and distributes in a structured way. It was possible to associate the distribution of the acquisition noise to a distance transform so as to form kernels that satisfy sufficient and necessary conditions for the associative memories in minmax algebra to fully

recall the learned patterns. Taking the above as a reference, bases were laid for a new model of min associative memory that is robust to mixed noise, i.e., robust to acquisition noise, and this model proved to surpasses the original model.

Author Contributions: Conceptualization, J.C.S.-R. and J.L.D.-d.-L.-S.; validation, E.A.C.-C., A.J.R.-S. and E.R.-D.; formal analysis, J.C.S.-R., J.M.V.K. and J.L.D.-d.-L.-S.; writing—original draft preparation, J.C.S.-R.; writing—review and editing, A.J.R.-S., E.R.-D., J.M.V.K., E.A.C.-C. and J.C.S.-R.; visualization, E.A.C.-C., A.J.R.-S. and E.R.-D.; supervision, J.C.S.-R. and J.M.V.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We thank the Universidad Politécnica de Pachuca, Tecnológico de Monterrey and CONACYT for the support provided.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Acquisition Noise Distribution Table in Grayscale Images

Table A1. Acquisition noise distribution table in grayscale images.

Distance	Frequency	Probability	Distance	Frequency	Probability
−189	3	0.00000173	−94	231	0.001332764
−188	3	0.00000173	−93	242	0.001396229
−187	2	0.00000115	−92	257	0.001482772
−186	3	0.00000173	−91	263	0.001517389
−185	4	0.00000231	−90	276	0.001592393
−184	5	0.00000288	−89	305	0.00175971
−183	6	0.00000346	−88	314	0.001811636
−182	12	0.00000692	−87	353	0.002036648
−181	6	0.00000346	−86	397	0.002290508
−180	16	0.00000923	−85	383	0.002209734
−179	9	0.00000519	−84	442	0.002550137
−178	21	0.00012116	−83	500	0.002884771
−177	18	0.000103852	−82	480	0.00276938
−176	20	0.000115391	−81	449	0.002590524
−175	30	0.000173086	−80	524	0.00302324
−174	20	0.000115391	−79	505	0.002913618
−173	23	0.000132699	−78	581	0.003352104
−172	23	0.000132699	−77	568	0.0032771
−171	25	0.000144239	−76	661	0.003813667
−170	18	0.000103852	−75	654	0.00377328
−169	17	0.00000981	−74	627	0.003617502
−168	22	0.00012693	−73	752	0.004338695
−167	15	0.00000865	−72	711	0.004102144
−166	14	0.00000808	−71	767	0.004425238
−165	19	0.000109621	−70	788	0.004546399
−164	18	0.000103852	−69	880	0.005077196
−163	19	0.000109621	−68	878	0.005065657
−162	17	0.00000981	−67	898	0.005181048
−161	21	0.00012116	−66	918	0.005296439
−160	13	0.00000750	−65	998	0.005758002
−159	10	0.00000577	−64	999	0.005763772
−158	13	0.00000750	−63	1060	0.006115714
−157	8	0.00000462	−62	1071	0.006179179

Table A1. *Cont.*

Distance	Frequency	Probability	Distance	Frequency	Probability
−156	12	0.00000692	−61	1091	0.00629457
−155	15	0.00000865	−60	1130	0.006519582
−154	18	0.000103852	−59	1198	0.006911911
−153	11	0.00000635	−58	1230	0.007096536
−152	19	0.000109621	−57	1284	0.007408091
−151	18	0.000103852	−56	1265	0.00729847
−150	10	0.00000577	−55	1278	0.007373474
−149	7	0.00000404	−54	1228	0.007084997
−148	22	0.00012693	−53	1393	0.008036971
−147	16	0.00000923	−52	1353	0.00780619
−146	20	0.000115391	−51	1334	0.007696568
−145	17	0.00000981	−50	1381	0.007967737
−144	16	0.00000923	−49	1389	0.008013893
−143	24	0.000138469	−48	1404	0.008100436
−142	16	0.00000923	−47	1463	0.008440839
−141	19	0.000109621	−46	1446	0.008342757
−140	23	0.000132699	−45	1483	0.00855623
−139	22	0.00012693	−44	1527	0.00881009
−138	13	0.00000750	−43	1540	0.008885094
−137	28	0.000161547	−42	1540	0.008885094
−136	20	0.000115391	−41	1544	0.008908172
−135	39	0.000225012	−40	1534	0.008850477
−134	36	0.000207703	−39	1596	0.009208188
−133	42	0.000242321	−38	1583	0.009133184
−132	39	0.000225012	−37	1518	0.008758164
−131	43	0.00024809	−36	1511	0.008717777
−130	39	0.000225012	−35	1548	0.00893125
−129	54	0.000311555	−34	1634	0.009427431
−128	40	0.000230782	−33	1554	0.008965867
−127	49	0.000282708	−32	1612	0.009300501
−126	48	0.000276938	−31	1585	0.009144723
−125	54	0.000311555	−30	1671	0.009640904
−124	59	0.000340403	−29	1673	0.009652443
−123	45	0.000259629	−28	1719	0.009917842
−122	54	0.000311555	−27	1632	0.009415892
−121	47	0.000271168	−26	1608	0.009277423
−120	69	0.000398098	−25	1644	0.009485126
−119	51	0.000294247	−24	1647	0.009502435
−118	64	0.000369251	−23	1655	0.009548591
−117	53	0.000305786	−22	1555	0.008971637
−116	59	0.000340403	−21	1659	0.009571669
−115	39	0.000225012	−20	1600	0.009231266
−114	51	0.000294247	−19	1635	0.0094332
−113	60	0.000346172	−18	1586	0.009150493
−112	54	0.000311555	−17	1560	0.009000485
−111	73	0.000421177	−16	1603	0.009248575
−110	85	0.000490411	−15	1552	0.008954328
−109	75	0.000432716	−14	1511	0.008717777
−108	82	0.000473102	−13	1429	0.008244675
−107	90	0.000519259	−12	1578	0.009104336
−106	101	0.000582724	−11	1505	0.00868316
−105	91	0.000525028	−10	1479	0.008533152
−104	125	0.000721193	−9	1504	0.00867739
−103	108	0.00062311	−8	1472	0.008492765
−102	117	0.000675036	−7	1502	0.008665851
−101	133	0.000767349	−6	1366	0.007881194
−100	150	0.000865431	−5	1349	0.007783111

Table A1. *Cont.*

Distance	frequency	Probability	Distance	frequency	Probability
−99	151	0.000871201	−4	1374	0.00792735
−98	155	0.000894279	−3	1409	0.008129284
−97	184	0.001061596	−2	1232	0.007108075
−96	183	0.001055826	−1	1408	0.008123514
−95	228	0.001315455	0	0	0

Table A2. Acquisition noise distribution table in grayscale images.

Distance	Frequency	Probability	Distance	Frequency	Probability
1	1245	0.007183079	95	135	0.000778888
2	1256	0.007246544	96	120	0.000692345
3	1274	0.007350396	97	137	0.000790427
4	1239	0.007148462	98	113	0.000651958
5	1285	0.007413861	99	96	0.000553876
6	1197	0.006906141	100	137	0.000790427
7	1200	0.00692345	101	102	0.000588493
8	1209	0.006975376	102	107	0.000617341
9	1198	0.006911911	103	116	0.000669267
10	1166	0.006727285	104	94	0.000542337
11	1139	0.006571508	105	82	0.000473102
12	1103	0.006363804	106	104	0.000600032
13	1126	0.006496504	107	104	0.000600032
14	1064	0.006138792	108	83	0.000478872
15	1086	0.006265722	109	81	0.000467333
16	1016	0.005861854	110	87	0.00050195
17	1023	0.005902241	111	81	0.000467333
18	979	0.005648381	112	67	0.000386559
19	1005	0.005798389	113	82	0.000473102
20	1003	0.00578685	114	69	0.000398098
21	911	0.005256052	115	64	0.000369251
22	933	0.005382982	116	62	0.000357712
23	914	0.005273361	117	73	0.000421177
24	912	0.005261822	118	69	0.000398098
25	897	0.005175279	119	61	0.000351942
26	933	0.005382982	120	64	0.000369251
27	867	0.005002192	121	56	0.000323094
28	807	0.00465602	122	53	0.000305786
29	821	0.004736794	123	60	0.000346172
30	822	0.004742563	124	52	0.000300016
31	797	0.004598325	125	59	0.000340403
32	734	0.004234843	126	48	0.000276938
33	727	0.004194457	127	38	0.000219243
34	737	0.004252152	128	35	0.000201934
35	710	0.004096374	129	29	0.000167317
36	689	0.003975214	130	39	0.000225012
37	679	0.003917519	131	36	0.000207703
38	626	0.003611733	132	43	0.00024809
39	625	0.003605963	133	29	0.000167317
40	667	0.003848284	134	35	0.000201934
41	596	0.003438647	135	26	0.000150008
42	608	0.003507881	136	27	0.000155778
43	542	0.003127091	137	20	0.000115391
44	591	0.003409799	138	29	0.000167317
45	552	0.003184787	139	20	0.000115391
46	532	0.003069396	140	23	0.000132699
47	486	0.002803997	141	19	0.000109621

Table A2. *Cont.*

Distance	Frequency	Probability	Distance	Frequency	Probability
48	526	0.003034779	142	18	0.000103852
49	474	0.002734763	143	18	0.000103852
50	459	0.00264822	144	30	0.000173086
51	470	0.002711684	145	26	0.000150008
52	449	0.002590524	146	22	0.00012693
53	403	0.002325125	147	18	0.000103852
54	413	0.002382821	148	16	0.00000923
55	415	0.00239436	149	6	0.00000346
56	388	0.002238582	150	12	0.00000692
57	414	0.00238859	151	10	0.00000577
58	397	0.002290508	152	13	0.00000750
59	360	0.002077035	153	10	0.00000577
60	346	0.001996261	154	8	0.00000462
61	341	0.001967414	155	13	0.00000750
62	346	0.001996261	156	6	0.00000346
63	347	0.002002031	157	9	0.00000519
64	327	0.00188664	158	8	0.00000462
65	320	0.001846253	159	4	0.00000231
66	271	0.001563546	160	7	0.00000404
67	292	0.001684706	161	6	0.00000346
68	310	0.001788558	162	3	0.00000173
69	266	0.001534698	163	4	0.00000231
70	279	0.001609702	164	5	0.00000288
71	263	0.001517389	165	2	0.00000115
72	290	0.001673167	166	2	0.00000115
73	240	0.00138469	167	5	0.00000288
74	246	0.001419307	168	5	0.00000288
75	223	0.001286608	169	1	0.00000577
76	215	0.001240451	170	2	0.00000115
77	211	0.001217373	171	1	0.000000577
78	215	0.001240451	172	3	0.00000173
79	181	0.001044287	173	2	0.00000115
80	179	0.001032748	174	1	0.000000577
81	222	0.001280838	175	2	0.00000115
82	161	0.000928896	176	0	0
83	167	0.000963513	177	0	0
84	183	0.001055826	178	2	0.00000115
85	164	0.000946205	179	2	0.00000115
86	154	0.000888509	180	0	0
87	161	0.000928896	181	1	0.000000577
88	168	0.000969283	182	0	0
89	166	0.000957744	183	1	0.000000577
90	147	0.000848123	184	0	0
91	127	0.000732732	185	0	0
92	153	0.00088274	186	0	0
93	149	0.000859662	187	1	0.000000577
94	144	0.000830814			

References

1. Steinbuch, K. Die Lernmatrix. *Kybernetik* **1961**, *1*, 36–45. [CrossRef]
2. Willshaw, D.; Buneman, O.; Longuet-Higgins, H.Non-holographic associative memory. *Nature* **1969**, *222*, 960–962. [CrossRef]
3. Amari, S. Learning patterns and pattern sequences by self-organizing nets of threshold elements. *IEEE Trans. Comput.* **1972**, *C-21*, 1197–1206. [CrossRef]
4. Anderson, J.A. A simple neural network generating an interactive memory. *Math. Biosci.* **1972**, *14*, 197–220. [CrossRef]
5. Kohonen, T. Correlation matrix memories. *IEEE Trans. Comput.* **1972**, *100*, 353–359. [CrossRef]
6. Nakano, K. Associatron-A model of associative memory. *IEEE Trans. Syst. Man Cybern.* **1972**, *SMC-2*, 380–388. [CrossRef]
7. Kohonen, T.; Ruohonen, M. Representation of associated data by matrix operators. *IEEE Trans. Comput.* **1973**, *c-22*, 701–702. [CrossRef]

8. Kohonen, T.; Ruohonen, M. An adaptive associative memory principle. *IEEE Trans. Comput.* **1973**, *c-24*, 444–445. [CrossRef]
9. Anderson, J.A.; Silverstein, J.; Ritz, S.; Jones, R. Distinctive features, categorical perception, and probability learning: Some applications of a neural model. *Psichol. Rev.* **1977**, *84*, 413–451. [CrossRef]
10. Amari, S. Neural theory of association and concept-formation. *Biol. Cybern.* **1977**, *26*, 175–185. [CrossRef]
11. Hopfield, J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA* **1982**, *79*, 2554–2558. [CrossRef]
12. Hopfield, J. Neurons with graded respose have collective computational properties like those of two-state neurons. *Proc. Natl. Acad. Sci. USA* **1984**, *81*, 3088–3092. [CrossRef]
13. Bosch, H.; Kurfess, F. Information storage capacity of incompletely connected associative memories. *Neural Netw.* **1998**, *11*, 869–876. [CrossRef]
14. Karpov, Y.L.; Karpov, L.E.; Smetanin, Y.G. Associative Memory Construction Based on a Hopfield Network. *Program. Comput. Softw.* **2020**, *46*, 305–311. [CrossRef]
15. Ferreyra, A.; Rodríguez, E.; Avilés, C.; López, F. Image retrieval system based on a binary auto-encoder and a convolutional neural network. *IEEE Lat. Am. Trans.* **2020**, *100*, 1–8.
16. Ritter, G.X.; Sussner, P.; Diaz-de-Leon, J. Morphological associative memories. *IEEE Trans. Neural Netw.* **1998**, *9*, 281–293. [CrossRef]
17. Ritter, G.X.; Diaz-de-Leon, J.; Sussner, P. Morphological bidirectional associative memories. *IEEE Neural Netw.* **1999**, *12*, 851–867. [CrossRef]
18. Santana, A.X.; Valle, M. Max-plus and min-plus projection autoassociative morphological memories and their compositions for pattern classification. *Neural Netw.* **2018**, *100*, 84–94.
19. Sussner, P. Associative morphological memories based on variations of the kernel and dual kernel methods. *Neural Netw.* **2003**, *16*, 625–632. [CrossRef]
20. Heusel, J.; Löwe, M.; Vermet, F. On the capacity of an associative memory model based on neural cliques. *Stat. Probab. Lett.* **2015**, *106*, 256–261. [CrossRef]
21. Sussner, P. Observations on morphological associative memories and the kernel method. *Neurocomputing* **2000**, *31*, 167–183. [CrossRef]
22. Kim, H.; Hwang, S.; Park, J.; Yun, S.; Lee, J.; Park, B. Spiking Neural Network Using Synaptic Transistors and Neuron Circuits for Pattern Recognition With Noisy Images. *IEEE Electron Device Lett.* **2018**, *39*, 630–633. [CrossRef]
23. Masuyama, N.; Kiong, C.; Seera, M. Personality affected robotic emotional model with associative memory for human-robot interaction. *Neurocomputing* **2018**, *272*, 213–225. [CrossRef]
24. Masuyama, N.; Islam, N.; Seera, M.; Kiong, C. Application of emotion affected associative memory based on mood congruency effects for a humanoid. *Neural Comput. Appl.* **2017**, *28*, 737–752. [CrossRef]
25. Aldape-Pérez, M.; Yáñez-Márquez, C.; López-Yáñez, I.; Camacho-Nieto, O.; Argüelles-Cruz, A. Collaborative learning based on associative models: Application to pattern classification in medical datasets. *Comput. Hum. Behav.* **2015**, *51*, 771–779. [CrossRef]
26. Aldape-Pérez, M.; Alarcón-Paredes, A.; Yáñez-Márquez, C; López-Yáñez, I,; Camacho-Nieto, O. An Associative Memory Approach to Healthcare Monitoring and Decision Making. *Sensors* **2018**, *18*, 2960. [CrossRef]
27. Njafa, J.P.T.; Engo, S.N. Quantum associative memory with linear and non-linear algorithms for the diagnosis of some tropical diseases. *Neural Netw.* **2018**, *97*, 1–10. [CrossRef]
28. Yong, K.; Pyo, G.; Sik, D.; Ho, D.; Jun, B.; Ryoung, K.; Kim, J. New iris recognition method for noisy iris images. *Pattern Recognit. Lett.* **2012**, *33*, 991–999.
29. Peng, X.; Wen, J.; Li, Z.;Yang, G. Rough Set Theory Applied to Pattern Recognition of Partial Discharge in Noise Affected Cable Data. *IEEE Trans. Dielectr. Electr. Insul.* **2017**, *24*, 147–156. [CrossRef]
30. Zhu, Z.; You, X.; Philip, Z. An adaptive hybrid pattern for noise-robust texture analysis. *Pattern Recognit.* **2015**, *48*, 2592–2608. [CrossRef]
31. Li, Y.; Li, J.; Duan, S.; Wang, L.; Guo, M. A reconfigurable bidirectional associative memory network with memristor bridge. *IEEE Neurocomputing* **2021**, *454*, 382–391. [CrossRef]
32. Knoblauch, A. Neural associative memory with optimal bayesian learning. *Neural Comput.* **2011**, *23*, 1393–1451. doi: 10.1162/NECO_a_00127. [CrossRef]
33. Rendeiro, D.; Sacramento, J.; Wichert, A. Taxonomical associative memory. *Cogn. Comput.* **2014**, *6*, 45–65. s12559-012-9198-4. [CrossRef]
34. Acevedo-Mosqueda, M.; Yáñez-Márquez, C.; López-Yáñez, I. Alpha-Beta bidirectional associative memories: Theory and applications. *Neural Process. Lett.* **2007**, *26*, 1–40. [CrossRef]
35. Acevedo, M.E.; Yáñez-Márquez, C.; Acevedo, M.A. Bidirectional associative memories: Different approaches. *ACM Comput. Surv.* **2013**, *45*, 1–30. [CrossRef]
36. Luna-Benoso, B.; Flores-Carapia, R.; Yáñez-Márquez, C. Associative memories based on cellular automata: An application to pattern recognition. *Appl. Math. Sci.* **2013**, *7*, 857–866. [CrossRef]
37. Cleofas-Sánchez, L.; Sánchez, J.S.; García, V.; Valdovinos, R.M. Associative learning on imbalanced environments: An empirical study. *Expert Syst. Appl.* **2016**, *54*, 387–397. [CrossRef]

38. Mustafa, A. Probabilistic binary similarity distance for quick binary image matching. *IET Image Process.* **2018**, *12*, 1844–1856. [CrossRef]
39. Velázquez-Rodríguez, J.L.; Villuendas-Rey, Y.; Camacho-Nieto, O.; Yáñez-Márquez, C. A novel and simple mathematical transform improves the perfomance of Lernmatrix in pattern classification. *Mathematics* **2020**, *8*, 732. [CrossRef]
40. Reyes-León, P.; Salgado-Ramírez, J.C; Velázquez-Rodríguez, J.L. Application of the Lernmatrix tau[9] to the classifi-cation of patterns in medical datasets. *Int. J. Adv. Trends Comput. Sci. Eng.* **2020**, *9* 8488–8497. [CrossRef]
41. Gamino, A.; Díaz-de-León, J. A new method to build an associative memory model. *IEEE Lat. Am. Trans.* **2021**, *19*, 1692–1701. [CrossRef]
42. Yiannis, B. A new method for constructing kernel vectors in morphological associative memories of binary patterns. *Comput. Sci. Inf. Syst.* **2011**, *8*, 141–166. [CrossRef]
43. Esmi, E.; Sussner, P.; Bustince, H.; Fernández, J. Theta-Fuzzy Associative Memories (Theta-FAMs). *IEEE Trans. Fuzzy Syst.* **2015**, *23*, 313–326. [CrossRef]
44. Tarkov, M.S. Application of emotion affected associative memory based on mood congruency effects for a humanoid. *Opt. Mem. Neural Netw.* **2016**, *25*, 219–227. [CrossRef]
45. Yáñez-Márquez, C.; López-Yáñez, I.; Aldape-Pérez, M.; Camacho-Nieto, O.; Argüelles-Cruz, A.; Villuendas-Rey, Y. Theoretical Foundations for the Alpha-Beta Associative Memories: 10 Years of Derived Extensions, Models, and Applications. *Neural Process. Lett.* **2018**, *48*, 811–847. [CrossRef]
46. Estevão, E.; Sussner, P.; Sandri, S. Tunable equivalence fuzzy associative memories. *Fuzzy Sets Syst.* **2016**, *292*, 242–260.
47. Li, L.; Pedrycz, W.; Qu, T.; Li, Z. Fuzzy associative memories with autoencoding mechanisms. *Knowl.-Based Syst.* **2020**, *191*, 105090. [CrossRef]
48. Starzyk, J.A.; Maciura, Ł.; Horzyk, A. Associative Memories With Synaptic Delays. *J. Assoc. Inf. Syst.* **2020**, *21*, 7. [CrossRef]
49. Lindberg, A. Developing Theory Through Integrating Human and Machine Pattern Recognition. *IEEE Trans. Neural Networks Learn. Syst.* **2019**, *31*, 331–344. [CrossRef]
50. Feng, N.; Sun, B. On simulating one-trial learning using morphologicalneural networks. *Cogn. Syst. Res.* **2019**, *53*, 61–70. [CrossRef]
51. Ahmad, K.; Khan J; Salah, M. A comparative study of Different Denoising Techniques in Digital Image Processing. In Proceedings of the IEEE 2019 8th International Conference on Modeling Simulation and Applied Optimization, Manama, Bahrain, 15–17 April 2019; doi:10.1109/ICMSAO.2019.8880389. [CrossRef]
52. Fan, Y.; Zhang, L.; Guo, H.; Hao, H.; Qian, K. Image Processing for Laser Imaging Using Adaptive Homomorphic Filtering and Total Variation. *Photonics* **2020**, *7*, 30. [CrossRef]
53. Lu, C.; Chou, T. Denoising of salt-and-pepper noise corrupted image using modified directional-weighted-median filter. *Pattern Recognit. Lett.* **2012**, *33*, 1287–1295. [CrossRef]
54. Xiao, Y.; Zeng, J.; Michael, Y. Restoration of images corrupted by mixed Gaussian-impulse noise via l1–l0 minimization. *Pattern Recognit.* **2011**, *44*, 1708–1720. [CrossRef]
55. Chervyakov, N.; Lyakhov, P.; Kaplun, D.; Butusov, D.; Nagornov, N. Analysis of the Quantization Noise in Discrete Wavelet Transform Filters for Image Processing. *Electronics* **2018**, *7*, 135. [CrossRef]
56. Gonzalez, R.; Woods, R. *Digital Image Processing*, 3rd ed.; Pearson: Upper Saddle River, NJ, USA, 2008; pp. 331–345.
57. Kipli, K.; Hoque, M.; Lim, L.; Afendi, T.; Kudnie, S.; Hamdi, M. Retinal image blood vessel extraction and quantification with Euclidean distance transform approach. *IET Image Process.* **2021**, *14*, 3718–3724. [CrossRef]
58. Duy, D.; Dovletov, G.; Pauli, J. A Differentiable Convolutional Distance Transform Layer for Improved Image Segmentation. *Pattern Recognit.* **2021**, *12544*, 432–444. [CrossRef]
59. Elizondo, J.; Ramirez, J.; Barron, J.; Diaz, A.; Nuño, M.; Saldivar, V. Parallel Raster Scan for Euclidean Distance Transform. *Symmetry* **2020**, *12*, 1808. [CrossRef]
60. Hill, B.; Baldock, R. Constrained distance transforms for spatial atlas registration. *BMC Bioinform.* **2015**, *16*, 90. [CrossRef]
61. Elizondo, J.; Parra, E.; Ramirez, J. The Exact Euclidean Distance Transform: A New Algorithm for Universal Path Planning. *Int. J. Adv. Robot. Syst.* **2013**, *10*, 266. [CrossRef]
62. Torelli, J.; Fabbri, R.; Travieso, G.; Martinez, B. A A high performance 3d exact euclidean distance transform algorithm for distributed computing. *Int. J. Pattern Recognit. Artif. Intell.* **2010**, *24*, 897–915. [CrossRef]
63. Bautista, S.; Salgado, J.; Gomez, A.; Tellez, A.; Ortega, R.; Jimenez, J.; Cadena, A. Image segmentation with fast distance transform (FDT) and morphological skeleton in microalgae Raceway culture systems applications. *Rev. Mex. Ing. Quim.* **2021**, *20*, 885–898. [CrossRef]