

# Image Fusion Based on Principal Component Analysis and Slicing Image Transformation

Leonardo Acho<sup>1,\*</sup> and Pablo Buenestado<sup>1,\*\*</sup>

<sup>1</sup>Department of Mathematics, Universitat Politècnica de Catalunya-BarcelonaTech (EEBE), 08034 Barcelona, Spain.

**Abstract.** Image fusion deals with the ability to integrate data from image sensors at different instants when the source information is uncertain. Although there exist many techniques on the subject, in this paper, we develop two originative techniques based on principal component analysis and slicing image transformation to efficiently fuse a small set of noisy images. For instance, in neural data fusion, this approach requires a considerable number of corrupted images to efficiently produce the desired outcome and also requiring a considerable computing time because of the dynamics involved in the fusion data process. In our approaches, the computation time is considerably smaller. This results appealing to increasing feasibility, for instance, in remote sensing or wireless sensor network. Moreover, and according to our numerical experiments, when our methods are compared against the neural data fusion algorithm, they present better performance.

## 1 Introduction

On one hand, data fusion consists in measurements integration from different sensors at different time instants when the original information is uncertain [1–3]. As a result, the fused data are more valuable for situation awareness understanding by completing notable information. Nowadays, and because of the availability of high technology and faster computers, suitable discrete dataset has been applied in many research activities such as medical image, machine vision, remote sensing, vehicle driving, robotics, secure communication design, pattern recognition, diagnosis, job shop scheduling, etc. [1, 3–11]. Generally speaking, image multi-sensor measurements data fusion refers to the acquisition, processing and synergistic combination of information gathered by various knowledge sources to provide a better understanding of a phenomenon. It is worth noticing that data fusion presents a notable difficulty when the available data is incomplete, inconsistent or imprecise; and additionally if the information is corrupted by external additive noise [1, 3, 12–14].

Additionally, data fusion not only consists in using integration measurements based on sensors but also integration of decisions, conflicts facts, and human intelligent assessments, [15–17]. Even the same human brain uses data fusion to make an inference regarding the surrounding environment by realizing data fusion coming from the sight, smell, hearing, taste, and touch [18]. Moreover, in the multi-sensor data fusion theory, there are many related data fusion techniques. For instance, the well known Kalman filter can be viewed as a data

fusion algorithm [18, 19]. Fuzzy logic [20, 21], genetic algorithms [22, 23], and wavelet analysis [24] can also be sorted as artificial intelligence techniques for data fusion. Statistical methods and Bayesian theory both can be invoked for data fusion too [25, 26]. Just to name a few. As a complementary information on it, see, for instance, [27]. Finally, it is interesting to highlight that there exist other definitions for data fusion: ‘multilevel, multifaceted process handling the automatic detection, association, correlation, estimation, and combination of data and information from several sources’ [27].

On the other hand, for instance, the design of new fault detection systems based on image or signal processing is an important and challenging task in many engineering applications, such as chemical processes, nuclear engineering, automotive system, wind turbines, and so on [4, 12, 28]. Where, basically, the main motivation on designing new monitoring processes is to maintain safe and proper operation of these plants. In recent literature, the design of fault detection systems can be classified into two main categories (see, for instance, [12]): model-based and data-based methods, where some of the model-based techniques invoke the statistical hypothesis testing technique for monitoring and diagnosis [12], among many other ideas. Nerveless, in these and other applications of image processing require a pre-processing stage (like, for instance, noise filtering, data fusion, noise analysis, and so on) to go further in diagnosis and/or control. Therefore, the typical pre-preprocessing stage consists of a kind of noise data filtering -data fusion may be employed as a kind of noise filtering although data fusion may also mitigate intrinsic sensor uncertainties too-. Moreover, among the different today's image processing

\*e-mail: [leonardo.acho@upc.edu](mailto:leonardo.acho@upc.edu)

\*\*e-mail: [pablo.buenestado@upc.edu](mailto:pablo.buenestado@upc.edu)

methods (see, for instance [29–34] and references there in), we are going to specially focus on the principal component analysis (PCA) along with slicing image transformation. The principal motivation to invoke PCA is to separate redundant information from the contaminated data [12, 35–38]. And slicing image transformation (or bit-plane slicing) is commonly employed to extract significant amount of visual entropy from a given image [39]. Hence, the main objective of this paper is to analyze image data fusion by invoking the previous cited methods and to grant new numerical algorithms for noisy image data fusion under the situation of limited access to image sensors information. It is worth noticing that the data fusion by using Lagrangian network converges to its optimal fusion if we have access to a huge number of discrete measurement information. In resume, by taking into account that image data fusion can be realized by doing a proper linear combination of the acquired image samples at certain instant of times [1, 2, 40–43], the main objective of this paper is to present two novel numerical techniques on data fusion by using PCA and slicing image transformation. To support our designs, several numerical scenarios are prepared. According to our numerical experiments, our approaches present beneficial performance because they are efficient on separating the useful image data from the noisy information by showing both of them separately. The performance evaluation of our processes were carried out by utilizing the structural similarity index measurement (SSIM). See [44]. In resume, so far, and to our best knowledge, the PCA technique in combination to the slicing image transformation has not been related. Specially, and according to the SSIM index, our second approach presents better performance. This fact is illustrated in our numerical experiments specially when the least significant bit-plane images are incorporated to the image fusion process.

The rest of this paper is organized as follows. Section 2 gives the standard theory and the main statement on image data fusion by using Lagrangian networks in continuous-time domain. In Section 3, we grant our main contribution on PCA image data fusion. The related numerical experiments are depicted in Section 4. Results and discussion are shown in Section 5. In this section, we also test the robustness of our design by proposing another example by using three captured frames coming from a film and noisily corrupted, among other examples. Finally, some concluding remarks are stated in Section 6.

**Notations.-** Throughout this paper, symbols in boldface format represent vectors or matrices, the un-boldface style for scalar variables, and  $T$  denotes the transpose.  $E[\cdot]$  represents the expected value of a random variable,  $\nabla f(x)$  is the gradient of the given scalar function  $f(x)$ , and  $\nabla^2 f(x)$  is the Hessian matrix of the stated function  $f(x)$ .

## 2 Image multi-sensor data system modeling

Consider an image multi-sensor system with  $K$  ( $K \geq 2$ ) image sensors and processed at the pixel detail. Let the  $k$ -th image sensor measurement be given by [1]:

$$x_k(t) = a_k s(t) + n_k(t), \quad (k = 1, \dots, K; t = 1, \dots, N), \quad (1)$$

where  $a_k$  is the sensor scaling coefficient -or sensor gain-,  $N$  is the number of sensor measurements,  $s(t)$  is the mother signal, and  $n_k(t)$  represents the additive white Gaussian noise at the  $k$ th sensor with zero mean value. Recall that noise is a random variation on the image density [39]. Principal sources of Gaussian noise in digital images arise during the acquisition of the image by the sensor due to, for instance, poor illumination, high temperature, and the noise induced by the electronic circuit system. Furthermore,  $s(t)$  and  $n_k(t)$  are mutually independent random processes. See Fig. 1. Hence, in the mentioned scheme, the pixel by pixel image processing system is adopted. Then, by defining  $\mathbf{a} = [a_1, \dots, a_K]^T$ ,  $\mathbf{x}(t) = [x_1(t), \dots, x_K(t)]^T$  (called the *non-mean deviation* observation vector), and  $\mathbf{n}(t) = [n_1(t), \dots, n_K(t)]^T$ , the above system can be represented as follows:

$$\mathbf{x}(t) = \mathbf{a}s(t) + \mathbf{n}(t). \quad (2)$$

The principal objective of data fusion consists to finding a weighting vector  $\mathbf{w} = [w_1, \dots, w_K]^T$  such that

$$f(\mathbf{w}) = E[(z(t) - s(t))^2], \quad (3)$$

be minimized with

$$z(t) = \mathbf{w}^T \mathbf{x}(t). \quad (4)$$

In this way,  $z(t)$  represents the optimal fused data so that the uncertainty (for instance, the set of additive noises) is optimally attenuated. See Fig. 1. Hence,  $z(t)$  is in fact a linear combination of the sensor measurements. In this way, the corresponding optimal 'weighting' vector  $\mathbf{w}$  linearly combine the pixels coming from the images to be processed in the restricted class of linear optimization theory.

The above data fusion statement is also equivalent to [1]:

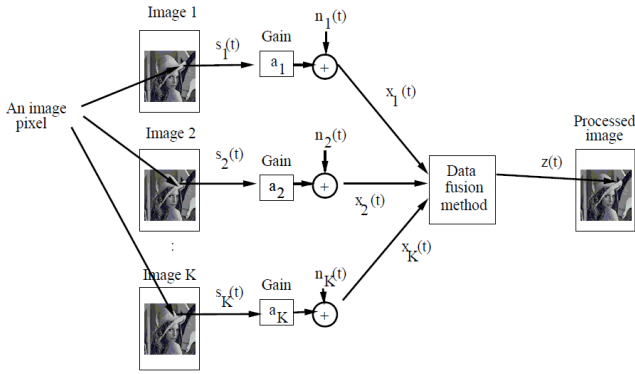
$$\begin{aligned} &\text{minimize} && f_1(\mathbf{w}) = \mathbf{w}^T \mathbf{R} \mathbf{w} \\ &\text{subject to} && \mathbf{a}^T \mathbf{w} = 1, \end{aligned} \quad (5)$$

where

$$\mathbf{R} = \frac{1}{N} \sum_{t=1}^N \mathbf{x}(t) \mathbf{x}^T(t) = \frac{1}{N} \mathbf{X} \mathbf{X}^T \quad (6)$$

being  $\mathbf{X} = [\mathbf{x}(1) \ \mathbf{x}(2) \ \dots \ \mathbf{x}(N)]$  the *non-mean-deviation* observation matrix.

Then, by solving the above optimization problem, we give a solution to the data fusion statement. Obviously, it is



**Figure 1.** Processing image data fusion technique based on  $K$  image sensors at the pixel level.

assumed that the probabilistic density model of the noise is known - as previously mentioned, the Gaussian case in our case-. Moreover, the obtained optimal fused signal  $z(t)$  is an unbiased estimation of the mother signal  $s(t)$  [1]. A solution to the higher up optimization problem can be realized by invoking the next Lagrangian network [1, 2]:

$$\frac{d}{dt} \begin{bmatrix} \mathbf{w} \\ y \end{bmatrix} = - \begin{bmatrix} \nabla f_1(\mathbf{w}) - \mathbf{a}y \\ \mathbf{a}^T \mathbf{w} - 1 \end{bmatrix}. \quad (7)$$

It is worth noticing that another option to the previous optimization problem is as follows [1]:

$$\begin{aligned} &\text{minimize} && f_2(\mathbf{w}) = \mathbf{w}^T(\alpha)\mathbf{R}\mathbf{w} = \alpha f_1(\mathbf{w}) \\ &\text{subject to} && \mathbf{a}^T \mathbf{w} = 1, \end{aligned} \quad (8)$$

being its corresponding Lagrangian network:

$$\frac{d}{dt} \begin{bmatrix} \mathbf{w} \\ y \end{bmatrix} = - \begin{bmatrix} \alpha \nabla f_1(\mathbf{w}) - \mathbf{a}y \\ \mathbf{a}^T \mathbf{w} - 1 \end{bmatrix}, \quad (9)$$

where  $\alpha$  is a given positive constant. In fact, by manipulating the value of  $\alpha$ , the convergence speed of the Lagrangian dynamics is also manipulated (see, for instance, [40]).  $y = y(t)$  is the Lagrangian multiplier. Moreover, the next important statistical properties are cited [1]:

- $E[z(t)] = E[s(t)]$
- $E[\mathbf{x}(t)\mathbf{x}^T] = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N \mathbf{x}(t)\mathbf{x}^T(t)$  with probability one (w.p. 1).

Lately, the optimal fusion solution is  $z^*(t) = (\mathbf{w}^*)^T \mathbf{x}(t)$  where [1]:

$$\mathbf{w}^* = \lim_{N \rightarrow \infty} \arg \min_{\mathbf{w}^T \mathbf{a} = 1} f_1(\mathbf{w}) = \lim_{N \rightarrow \infty} \arg \min_{\mathbf{w}^T \mathbf{a} = 1} f_2(\mathbf{w}), \text{ (w. p.1)} \quad (10)$$

From the previous properties, we can observe that the optimal solution is obtainable if we have access to an enough huge number of images to be processed.

In summary, we have the following result [2, 40]:

*Theorem 1:* Assume that  $\nabla^2 f_1(\mathbf{w})$  is positive definite. Then, the Lagrangian network in (7) (or (9)) is stable

in the Lyapunov sense and is globally convergent to an equilibrium point of (7) (or (9)), which corresponds a unique optimal solution of the stated minimization problem.

Next, we will comment a key important reflexion to justify the use of PCA as an alternative technique on data fusion. According to [1], the optimization problem stated in (5) follows after invoking

$$E[(\mathbf{w}^T \mathbf{a} - 1)s(t)] = 0, \quad (11)$$

which is true if  $\mathbf{w}^T \mathbf{a} = 1$ . But it is also true if  $E[s(t)] = 0$ . This realism can be obtained if we produce the corresponding *mean-deviation* observation matrix from the *non-mean-deviation* observation matrix. In fact, this is realized by the PCA technique. Hence, by producing the mean-deviation matrix, we arrive to a sub-optimal data fusion by just using PCA. And because of this, Lagrangian dynamic is not need anymore. Obviously, this brings a potential improvement in computer calculations. In other words, by invoking (6) and extracting its noise information (by employing, for instance, the PCA method), we are solving the un-constraint sub-optimal case to the cited image fusion objective. Finally, and because of data fusion by using Lagrangian networks assumes additive Gaussian noise in the sensor signals, this assumption is not restricted.

### 3 Introduction to principal component analysis

In signal processing, the principal component analysis can be efficiently utilized as a mathematical tool able to separate noisy data into two sets. One containing its filtered version, and in the other an estimation of the noisy entropy affecting the original clean data. To resume this mathematical method, let

$$[\mathbf{X}_1 \ \mathbf{X}_2 \ \dots \ \mathbf{X}_N] \in \mathbb{R}^{k \times N} \quad (12)$$

be the matrix observation, where  $\mathbf{X}_1, \dots, \mathbf{X}_N \in \mathbb{R}^k$  are the *non-mean-deviation* observation vectors. Then, its sample mean matrix is given by

$$\mathbf{M} = \frac{1}{N}(\mathbf{X}_1 + \dots + \mathbf{X}_N) \in \mathbb{R}^{k \times 1}. \quad (13)$$

The *mean-deviation* matrix is then stated as:

$$\mathbf{B} = [\hat{\mathbf{X}}_1 \ \hat{\mathbf{X}}_2 \ \dots \ \hat{\mathbf{X}}_N] \in \mathbb{R}^{k \times N} \quad (14)$$

where

$$\hat{\mathbf{X}}_k = \mathbf{X}_k - \mathbf{M}, \quad k = 1, \dots, N. \quad (15)$$

Finally, the PCA covariance matrix is defined as:

$$\mathbf{S} = \frac{1}{N-1} \mathbf{B}\mathbf{B}^T \in \mathbb{R}^{k \times k}. \quad (16)$$

The essential PCA procedure consists in finding a orthogonal matrix  $\mathbf{P}$  such that

$$P^T S P = D, \quad (17)$$

where  $D$  is a diagonal matrix with the eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_k$  of  $S$  in its diagonal arranged so that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_k \geq 0$ .

Consider  $P = [u_1 \ u_2 \ \dots \ u_k]$ , where  $u_1, u_2, \dots, u_k$  are the corresponding unit eigenvectors of the related eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_k$ . Consequently, the first principal component is  $u_1 = [f_1, f_2, \dots, f_k]^T$  and, for instance, the  $K$  one yields  $u_K = [K_1, K_2, \dots, K_k]^T$ , and so on. Lately, a filtering image fusion is obtained by realizing the following linear combination:

$$NI_1 = f_1 I_{m_1} + f_2 I_{m_2} + f_k I_{m_k}, \quad (18)$$

and the next represents the estimation noisy image data affecting the measured image information:

$$NI_2 = K_1 I_{m_1} + K_2 I_{m_2} + K_k I_{m_k}, \quad (19)$$

where  $I_{m_1}, I_{m_2}, \dots, I_{m_k}$  are the noisy images to be processed and invoked to produce the PCA covariance matrix. To note that the standard representations in (24) and (25) represent a linear vector combination being  $f_1, f_2$ , and  $f_3$ ; and  $K_1, K_2$ , and  $K_3$  the weighting elements.

## 4 Numerical experiments

By discomposing an image into its bit planes, see Figure 2, results useful for analyzing the relative entropy of the number of bits used to quantize an image [39]. The reconstruction is realized by multiplying the pixel intensity of the  $n$ th image plane by the constant  $2^{n-1}$ . This corresponds to converting the  $n$ th significant binary bit into its decimal format. On the other hand, by using the two more significant bit planes outcomes the most relevant image information. We use them in our next algorithms. For easy reference and without loss of generality, we invoke three noisy images.

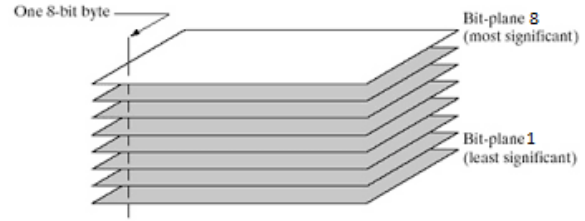
### 4.1 First approach on data fusion based on PCA

In this subsection we are going to present our first approach on data fusion based on PCA and slicing image transformation. Given three noise images,  $L_1, L_2$  and  $L_3$ , we obtain their two bit planes slicing transformation  $L_a, L_b$  and  $L_c$  (we use the *bitget* Matlab command) to obtain the first and third principal components by following the PCA algorithm previously commented:

$$NIa = -(f_1 L_1 + f_2 L_2 + f_k L_3), \quad (20)$$

$$NI = \frac{255}{\max(\max(NIa))} NIa, \quad (21)$$

and



**Figure 2.** Slicing bit planes representation of an 8-bit image.

$$NIa2 = -(K_1 L_1 + K_2 L_2 + K_k L_3), \quad (22)$$

$$NI2 = \frac{255}{\max(\max(NIa2))} NIa2. \quad (23)$$

Then  $NI$  and  $NI2$  represent the obtained fusion image (a filtered image version) and the noisy image estimation, respectively. We decide to use the slicing transformation in order to illustrate the potentiality of the method to be used, for instance, in compressed images. Moreover, the negative sign in (20) is inserted to avoid image inversion and followed by a mathematical normalization (in equation (21)) to avoid image saturation. For comparison objective, we also realize the average filtering:

$$AverFilt = \frac{1}{3}(L_1 + L_2 + L_3). \quad (24)$$

Finally, for performance evaluation, we use the structural similarity index (we invoke the *ssim* Matlab command) between the corresponding processed image and the original one. See [44]. For sake of simplicity, Appendix A.1 shows the corresponding Matlab code. By using the original image depicted in Figure 3, and its noise corrupted images shown in Figure 4 (in these figures, the noisy images are corrupted by uncorrelated Gaussian noise of zero mean and variance 0.01), numerical experiment results are shown in Figures 5-6. A second round of numerical experiments is repeated by using the noisy images now affected by the uncorrelated Poisson noise and shown in Figure 7. Figures 8-9 show the agreeing simulation results.

### 4.2 Second approach on data fusion based on PCA

In this subsection we will present our second approach on data fusion based on PCA and slicing image transformation. Given three noise images,  $L_1, L_2$  and  $L_3$ , we again obtain their two bit planes slicing transformation  $L_a, L_b$  and  $L_c$  to obtain the first and third principal components by using the PCA method. Then, we propose:

$$NIa = -(f_1 L_a + f_2 L_b + f_k L_c), \quad (25)$$

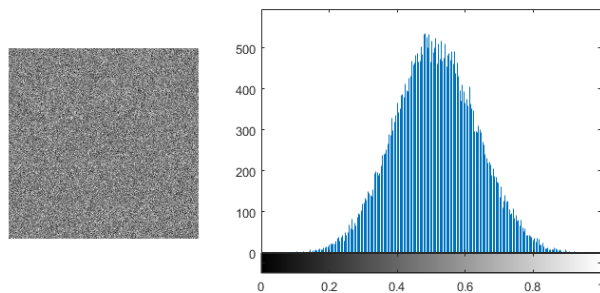
$$NI = \frac{255}{\max(\max(NIa))} NIa, \quad (26)$$

and





**Figure 3.** Original image of Lena (200X200 pixels).



**Figure 6.** The noisy image estimation affecting our original image and its histogram.



**Figure 4.** The uncorrelated Gaussian noisy images for processing.



**Figure 7.** The uncorrelated Poisson noisy images for processing.



**Figure 5.** The corresponding processed images. Left: by using the average filtering ( $SSIM=0.4174$ ). Right: by invoking our first approach ( $SSIM=0.4091$ )



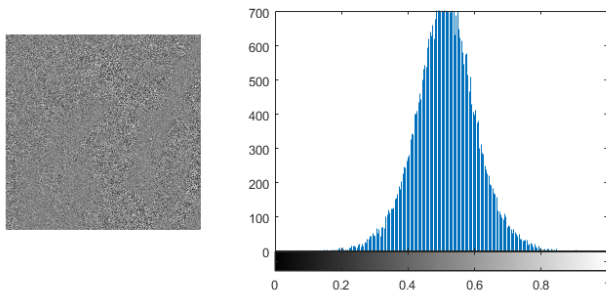
**Figure 8.** The corresponding processed images. Left: by using the average filtering ( $SSIM=0.6834$ ). Right: by invoking our first approach ( $SSIM=0.6720$ ).

$$NIa2 = -(K_1L_a + K_2L_b + K_kL_c), \quad (27)$$

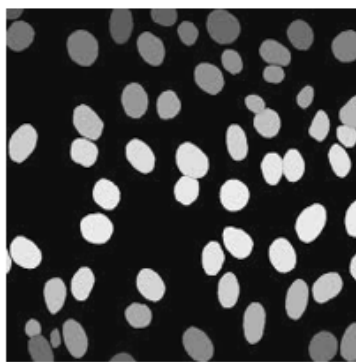
$$NI2 = \frac{255}{\max(\max(NIa))} NIa2. \quad (28)$$

Once again,  $NI$  and  $NI2$  represent the obtained fusion image (a filtered image version) and the noisy image estimation, respectively. Over again, we decide to employ the slicing transformation in order to illustrate the potentiality

of the method to be used in compressed images. Then, the negative sign in (25) is inserted to avoid image inversion and followed by a mathematical normalization (in equation (26)) to avoid image saturation. For comparison objective, once more we realize the related average filtering:



**Figure 9.** The noisy image estimation affecting our original image and its histogram.



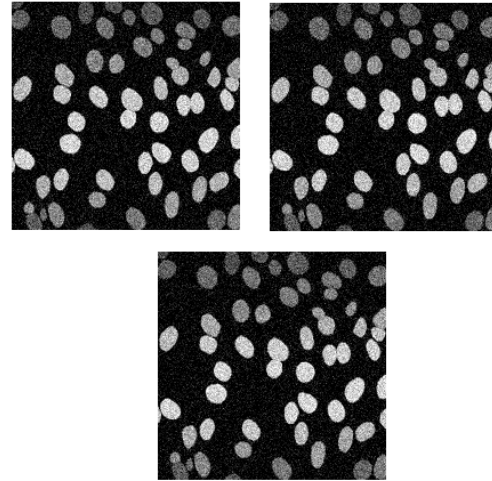
**Figure 10.** The original clean image (200X200 pixels).

$$AverFilt = \frac{1}{3}(L_a + L_b + L_c). \quad (29)$$

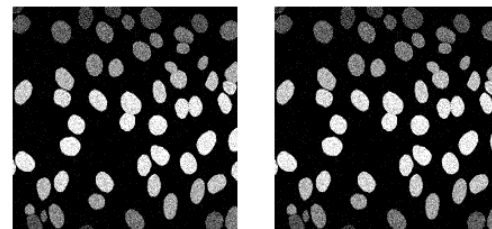
Lately, for performance evaluation, we keep using the structural similarity index between the corresponding processed image and the original one. Appendix A.2 shows the corresponding Matlab code. By using the original image depicted in Figure 10, and its noise corrupted images show in Figure 11 (in these figures, the noisy images are corrupted by uncorrelated Gaussian noise of zero mean and variance 0.01), numerical experiment results are shown in Figures 12-13. A second round of numerical experiments is repeated by using the noisy images now affected by the uncorrelated Poisson noise and shown in Figure 14. Figures 15-16 show the agreeing simulation results.

## 5 Results and Discussion

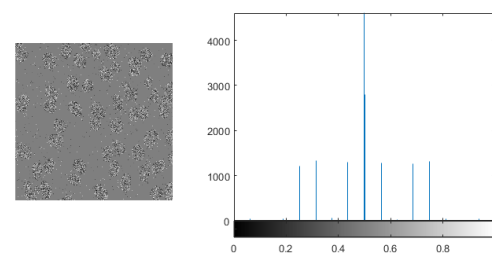
Nonlinear convex programming by using Lagrangian networks in the sense of [2, 41, 42, 45–48] is stated in continuous-time domain. One of its main application is linked to the data fusion when linear equality constraints is imposed to the optimization objective. However, and taking into account that digital devices are usually employed for physical realization, translating the above statement into discrete-time domain requires special modification due to the next facts:



**Figure 11.** The uncorrelated Gaussian noisy images for processing.



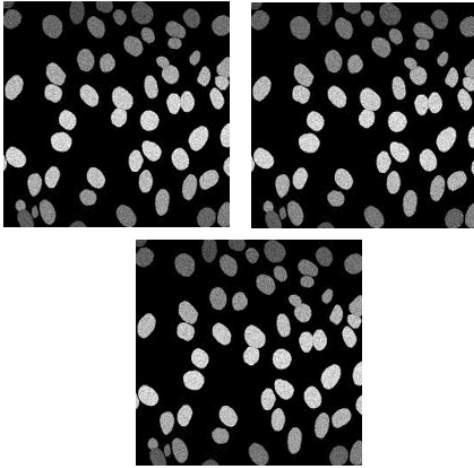
**Figure 12.** The corresponding processed images. Left: by using the average filtering ( $SSIM=0.4462$ ). Right: by invoking our second approach ( $SSIM=0.4731$ ).



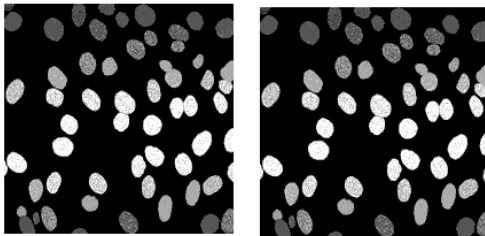
**Figure 13.** The noisy image estimation affecting our original image and its histogram.

- Stability of discrete-time systems depend on the sampling-rate of the employed digital device.
- Stability of the Lagrangian network dynamics will depend on the positive definiteness of  $\nabla^2 f_1(w)$ .
- Positive definiteness of  $\nabla^2 f_1(w)$  will depend on the available data to process and captured in  $R$ .
- The available captured data to produce  $R$  may cause that the Lagrangian network dynamics be un-convergent.

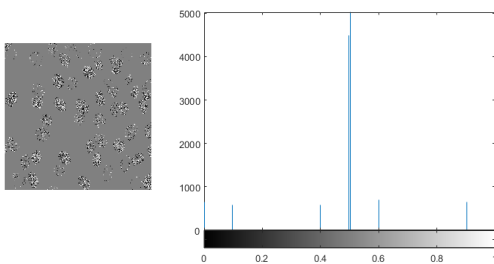
From the above writing, data fusion by using Lagrangian networks results inappropriate if the number sensors is



**Figure 14.** The uncorrelated Poisson noisy images for processing.



**Figure 15.** The corresponding processed images. Left: by using the average filtering ( $SSIM=0.5433$ ). Right: by invoking our second approach ( $SSIM=0.5787$ ).



**Figure 16.** The noisy image estimation affecting our original image and its histogram.

small. Hence, our approaches are simpler and effective due to the following facts:

- They produce a fused image.
- They provide an estimation of the noise affecting our image sensors.

The second one results attractive because by knowing an estimation of the noise we can figure out the reason of it. For instance, the speckle noise can be induced by a kind of radiation, and so on.

On the other hand, in some applications, the set of image to be processed arrives from the same image sensor, which means that these image samples can be considered perfectly *aligned*. But if different image sensors are used, or if the scene presents slightly time moving agents, then a *perfect* alignment among the image pixels is not true anymore. With this issue in mind, we prepare the following experiment. From a film, three different screen captures were obtained (see Figure 17). These images, in fact, capture a slightly moving environment. Moreover, and because the captures were realized by using the classical selecting, copying and pasting method into an image file, and then adding, for instance, the uncorrelated Gaussian noise, the spatial variation among them is also presented. The numerical results by using, for example, our first approach, are shown in Figures 18-19. Clearly, this experiment shows the robustness of our approach.

To amplify our numerical examples, we slightly modify our second approach by changing the slicing images  $L_a$ ,  $L_b$ , and  $L_c$  by (see Appendix A.2):

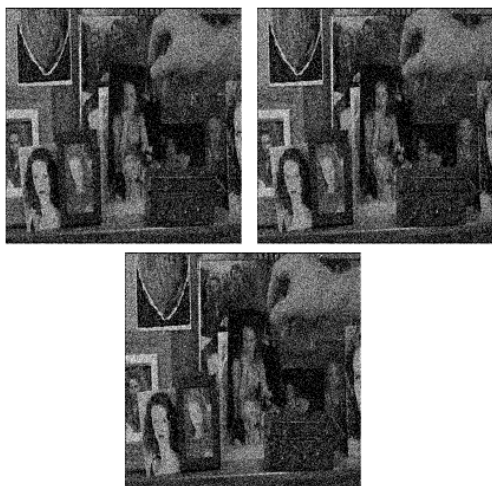
```

La=
1*double(bitget(uint8(L1),1))+...
1*2*double(bitget(uint8(L1),2))+...
1*64*double(bitget(uint8(L1),7))+...
128*double(bitget(uint8(L1),8));
Lb=
1*double(bitget(uint8(L2),1))+...
1*2*double(bitget(uint8(L2),2))+...
1*64*double(bitget(uint8(L2),7))+...
128*double(bitget(uint8(L2),8));
Lc=
1*double(bitget(uint8(L3),1))+...
1*2*double(bitget(uint8(L3),2))+...
1*64*double(bitget(uint8(L3),7))+...
128*double(bitget(uint8(L3),8));
    
```

On this way, and by adding the least significant slicing images, we increase noise information to the data fusion procedure. The obtained results by employing the clean image given in Figure 20, and its noise samples shown in Figure 21, the received results are shown in Figures 22 and 23. By reading the structural similarity measurement (SSIM), our approach present better performance. Moreover, the corresponding histogram to the third principal component gives important information about the noise information in the images. For instance, the average gray scale of the image is most affected by the related noise. To conclude our numerical realizations, Figure 25 shows the corresponding first and third principal components by invoking our second approach given in Appendix A.2 to the noisy images shown in Figure 24. We select this experiment to illustrate the ability of the slicing process to highlight the influence of the noise on the rocked instead of the image background. This realism follows because of algorithm.

In addition, from the revised literature on data fusion methods stated in the Introduction section, our design

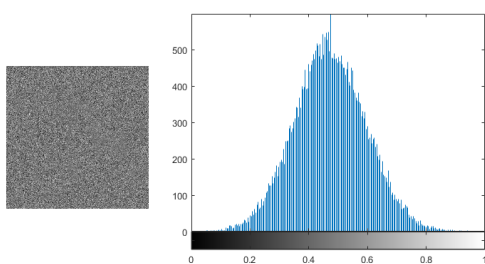




**Figure 17.** Noisy frames taken from a film.

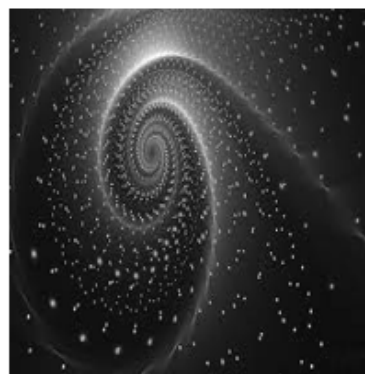


**Figure 18.** The obtained image fusion by using our first approach.

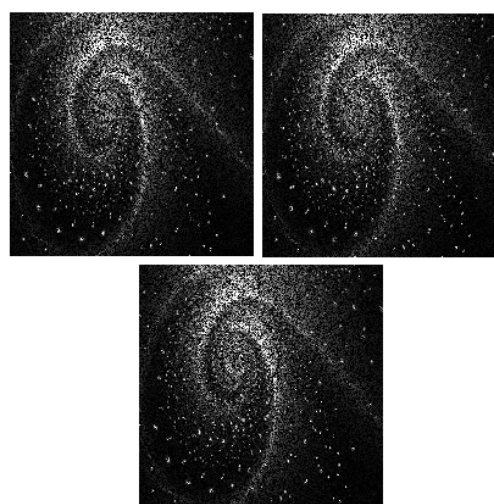


**Figure 19.** The noisy image estimation affecting our original image and its histogram.

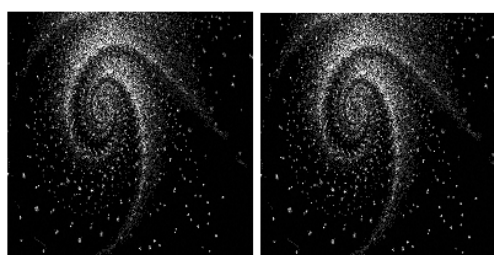
has the following gains. For instance, in comparison to the method given in [18], our propose does not require clustering nor update prediction rules. In contrast to the Fuzzy logic technique shown in [20, 21], our strategic does not involve implementing inference rules, implication, aggregation, and defuzzification processes. These tasks are no always easy to realize, especially to write down the corresponding input and output membership functions. On the other hand, the genetic algorithms granted in [22, 23] basically needs a kind of intelligent



**Figure 20.** The sample clean image (200X200 pixels).



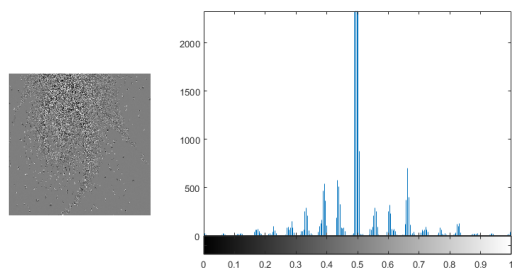
**Figure 21.** The uncorrelated Speckle noisy images with multiplicative noise 0.7 for processing.



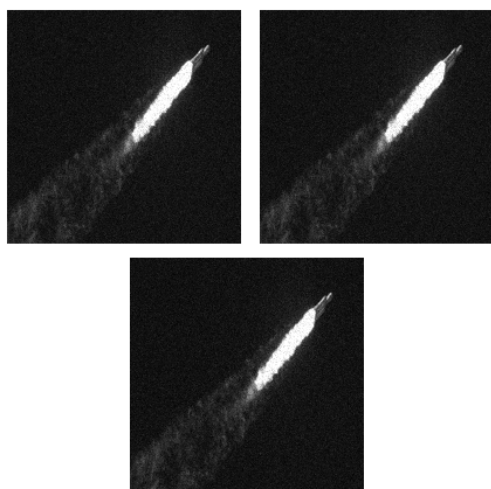
**Figure 22.** The corresponding processed images. Left: by using the average filtering ( $SSIM=0.2609$ ). Right: by invoking our second approach ( $SSIM=0.2805$ )

exploitation of a random search to solve optimization part. This may consume a notable machine computation time. Lastly, the statistical and Bayesian designs given in [25, 26] essentially require decision rules based on probabilistic loss functions requiring a training process. In our approaches, we just follow a deterministic solution by employing PCA. In our opinion, it is easier to programing. On the other hand, image processing by using wavelets transformation requires, as in the Fourier theory, the next

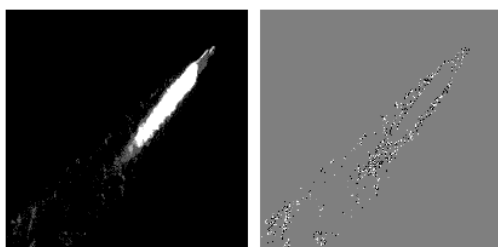




**Figure 23.** The noisy image estimation affecting our original image and its histogram.



**Figure 24.** The uncorrelated Poisson noisy images for processing.



**Figure 25.** The obtained image fusion by using our second approach: the related first and third principal components.

basic steps [39]:

- Compute a 2-D wavelet transform of an image.
- Alter the transform.
- Compute the inverse transform.

Hence, to essential capture the philosophy design by using this technique, we first involve some basic knowledge on Fourier transform [39]. Therefore, we can say that our design and the wavelets transformation tool are two completely different approaches and each one has their own advantages and disadvantages depending on the



**Figure 26.** Data fusion by using the Lena noisy images and the Lagrangian neuronal method stated in [1].

application because data fusion has the main aim to obtain information of greater quality from the supplied image information, where the exact definition of "greater quality" will *specially* depend upon the application [49]. Furthermore, by using the data fusion stated in [1], the obtained image fusion by using the Lena Gaussian noisy images is shown in Figure 26. It is clear that this technique is un-convergent due to the small number of images to process. To highlight that the computation time to obtain this result was around 75 minutes by using a common personal computer. Hence the PCA technique is substantially faster.

To close this section, although there are some PCA solutions to image fusion, our main approach presents an alternative resolution. For instance, in [50], the proposed PCA algorithm is used in combination with the intensity-hue-saturation transform. But this technique was not conceived as an image fusion method to attenuate the external noise effect on the images to be fused. In [51], some other PCA procedures are commented to integrate the geometric detail of a given high-resolution panchromatic image. Finally, in [52], more PCA techniques to image applications are documented. However, the mixing combination of PCA with the slicing image transformation has not been related.

## 6 Conclusions

In this paper we have presented two data fusion methods based on PCA and slicing image transformation. According to our numerical experiments, our approaches are able to separate the noise from the fused image. This results useful to study, for instance, the environmental noise affecting our sensors. In this way, we can detect, for instance, some kind of external radiation. Moreover, in our approaches, the computation time is considerably

## A.1 First approach

```

clear all;
figure(7);imshow(mat2gray(NI2));
A1=double(rgb2gray(imread('file_name.tif')));hist(mat2gray(NI2));
A=imresize(A1,[200,200]); %%%%-----Image size
figure(1);imshow(mat2gray(A));
%%%%%%%% Different sample noisy images to analyze
%%%%%%%% by selecting one set of the
%%%%%%%% below shown
% %%%% 1) Gaussian noise:
% L1=double(imnoise(uint8(A),'gaussian',0,0.01));
% figure(2);imshow(mat2gray(L1));
% L2=double(imnoise(uint8(A),'gaussian',0,0.01));
% figure(3);imshow(mat2gray(L2));
% L3=double(imnoise(uint8(A),'gaussian',0,0.01));
% figure(4);imshow(mat2gray(L3));
%%%%%%%% 2) Salt and pepper noise:
% L1=double(imnoise(uint8(A),'salt & pepper',0.05));
% figure(2);imshow(mat2gray(L1));
% L2=double(imnoise(uint8(A),'salt & pepper',0.05));
% figure(3);imshow(mat2gray(L2));
% L3=double(imnoise(uint8(A),'salt & pepper',0.05));
% figure(4);imshow(mat2gray(L3));
%%%%%%%% 3) Speckle noise:
% L1=double(imnoise(uint8(A),'speckle',0.5));
% figure(2);imshow(mat2gray(L1));
% L2=double(imnoise(uint8(A),'speckle',0.5));
% figure(3);imshow(mat2gray(L2));
% L3=double(imnoise(uint8(A),'speckle',0.5));
% figure(4);imshow(mat2gray(L3));
%%%%%%%% 4) Poisson noise:
L1=double(imnoise(uint8(A),'poisson'));
figure(2);imshow(mat2gray(L1));
L2=double(imnoise(uint8(A),'poisson'));
figure(3);imshow(mat2gray(L2));
L3=double(imnoise(uint8(A),'poisson'));
figure(4);imshow(mat2gray(L3));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Creation of the mean-deviation observation matrix by using slicing transformation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[I,J]=size(A);i=1;
La=1*64*double(bitget(uint8(L1),7))+128*double(bitget(uint8(L1),8));
Lb=1*64*double(bitget(uint8(L2),7))+128*double(bitget(uint8(L2),8));
Lc=1*64*double(bitget(uint8(L3),7))+128*double(bitget(uint8(L3),8));
for r=1:I
    for c=1:J
        X(1,i)=La(r,c);
        X(2,i)=Lb(r,c);
        X(3,i)=Lc(r,c);
        i=i+1;
    end
end
M_1=0;M_2=0;M_3=0;
for k=1:i-1
    M_1=M_1+X(1,k);
    M_2=M_2+X(2,k);
    M_3=M_3+X(3,k);
end
M_1=M_1/(i-1);
M_2=M_2/(i-1);
M_3=M_3/(i-1);
i=1;
for r=1:I
    for c=1:J
        B(1,i)=X(1,i)-M_1;
        B(2,i)=X(2,i)-M_2;
        B(3,i)=X(3,i)-M_3;
        i=i+1;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Creation of the covariance matrix S
S=(1/(i-2))*B*B';
%%%%%%%% Conception of the SVD of S by using the standard procedure
[U,D,V]=svd(S);
%%%%%%%% Image fusion by using the first principle component vector,
NIa=-(U(1,1)*L1+U(2,1)*L2+U(3,1)*L3);
%%%%%%%% then its normalization to avoid image saturation
NI=NIa*255/max(max(NIa));
%%%%%%%% and noise extraction by using the third principle component vector,
NIa2=-(U(1,3)*L1+U(2,3)*L2+U(3,3)*L3);
NI2=NIa2*255/max(max(NIa2));
figure(5);imshow(mat2gray(NI));
%%%%%%%% Average filter for contrasting our numerical experiments
AverFilt=(L1+L2+L3)/3;figure(6);imshow(mat2gray(AverFilt));
%%%%%%%% The corresponding structural similarity parameters
SSIM1=ssim(A,NI)
SSIM2=ssim(A,AverFilt)
    
```

## A.2 Second approach

```

clear all;
A1=double(rgb2gray(imread('file_name','jpg')));
A=imresize(A1,[200,200]); %%%%<----Image size
figure(1);imshow(mat2gray(A));
%%%%%%%% Different sample noisy images to analyze by selecting one set of the
%%%%%%%% below shown
% %%%% 1) Gaussian noise:
% L1=double(imnoise(uint8(A),'gaussian',0,0.01));
%figure(2);imshow(mat2gray(L1));
% L2=double(imnoise(uint8(A),'gaussian',0,0.01));
%figure(3);imshow(mat2gray(L2));
% L3=double(imnoise(uint8(A),'gaussian',0,0.01));
%figure(4);imshow(mat2gray(L3));
%%%%%%%% 2) Salt and pepper noise:
% L1=double(imnoise(uint8(A),'salt & pepper',0.05));figure(2);imshow(mat2gray(L1));
% L2=double(imnoise(uint8(A),'salt & pepper',0.05));figure(3);imshow(mat2gray(L2));
% L3=double(imnoise(uint8(A),'salt & pepper',0.05));figure(4);imshow(mat2gray(L3));
%%%%%%%% 3) Speckle noise:
% L1=double(imnoise(uint8(A),'speckle',0.5));
%figure(2);imshow(mat2gray(L1));
% L2=double(imnoise(uint8(A),'speckle',0.5));
%figure(3);imshow(mat2gray(L2));
% L3=double(imnoise(uint8(A),'speckle',0.5));
%figure(4);imshow(mat2gray(L3));
%%%%%%%% 4) Poisson noise:
L1=double(imnoise(uint8(A),'poisson'));
figure(2);imshow(mat2gray(L1));
L2=double(imnoise(uint8(A),'poisson'));
figure(3);imshow(mat2gray(L2));
L3=double(imnoise(uint8(A),'poisson'));
figure(4);imshow(mat2gray(L3));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Creation of the mean-deviation observation matrix by using slicing transformation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[I,J]=size(A);i=1;
La=1*64*double(bitget(uint8(L1),7))+128*double(bitget(uint8(L1),8));
Lb=1*64*double(bitget(uint8(L2),7))+128*double(bitget(uint8(L2),8));
Lc=1*64*double(bitget(uint8(L3),7))+128*double(bitget(uint8(L3),8));
for r=1:I
    for c=1:J
        X(1,i)=La(r,c);
        X(2,i)=Lb(r,c);
        X(3,i)=Lc(r,c);
        i=i+1;
    end
end
M_1=0;M_2=0;M_3=0;
for k=1:i-1
    M_1=M_1+X(1,k);
    M_2=M_2+X(2,k);
    M_3=M_3+X(3,k);
end
M_1=M_1/(i-1);
M_2=M_2/(i-1);
M_3=M_3/(i-1);
i=1;
for r=1:I
    for c=1:J
        B(1,i)=X(1,i)-M_1;
        B(2,i)=X(2,i)-M_2;
        B(3,i)=X(3,i)-M_3;
        i=i+1;
    end
end
%%%%%%%% Creation of the covariance matrix S
S=(1/(i-2))*B*B';
%%%%%%%% Conception of the SVD of S by using the standard procedure
[U,D,V]=svd(S);
%%%%%%%% Image fusion by using the first principle component vector,
NIa=-(U(1,1)*La+U(2,1)*Lb+U(3,1)*Lc);
%%%%%%%% then its normalization to avoid image saturation
NI=NIa*255/max(max(NIa));
%%%%%%%% and noise extraction by using the third principle component vector,
NIa2=-(U(1,3)*La+U(2,3)*Lb+U(3,3)*Lc);
NI2=NIa2*255/max(max(NIa2));
figure(5);imshow(mat2gray(NI));
%%%%%%%% Average filter for contrasting our numerical experiments
AverFilt=(La+Lb+Lc)/3;figure(6);imshow(mat2gray(AverFilt));
%%%%%%%% The corresponding structural similarity parameters
SSIM1=ssim(A,NI)
SSIM2=ssim(A,AverFilt)
figure(7);imshow(mat2gray(NI2));
figure(8);imhist(mat2gray(NI2));
    
```



smaller in comparison to other image fusion techniques. This results appealing to increasing feasibility, for instance, in remote sensing or wireless sensor network. Finally, we have used a neuronal Lagrangian technique to develop image fusion by using the given Gaussian noisy Lena images, and the obtained result was un-convergent. Therefore, our essential approach based on PCA is better.

## 7 Conclusions

In this paper, we have developed a controller based on *K*-means clustering theory. In our philosophy design, the controller continuously observes the plant output and when the data measurement is located outside of the desired cluster size located around the closed-loop system equilibrium point, it is forced to be in it. Moreover, the obtained controller is robust against un-vanishing perturbation and nonlinearity effects on the overall closed-loop system such as saturation, slew-rate limit, limit bandwidth frequency operation, and so on. Finally, and from the academic point of view, the employed base to reach our controller approach seems interesting.

## Acknowledgments

This work was partially supported by the Spanish Ministry of Economy, Industry and Competitiveness under grant DPI2015-64170-R (MINECO/FEDER).

## A Program lines by using Matlab

The following codes were written for Matlab Release 2017a from Mathworks.

## References

- [1] Y. Xia, H. Leung, E. Bossé, IEEE Transactions on Neural Networks **13**, 320 (2002)
- [2] Y. Xia, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications **50**, 818 (2003)
- [3] P.K. Varshney, Electronics & Communication Engineering Journal **9**, 245 (1997)
- [4] R.C. Luo, M.H. Lin, R.S. Scherp, IEEE Journal on Robotics and Automation **4**, 386 (1988)
- [5] S. Constantinou, M.S. Pattichis, E. Micheli-Tzanakou, *Medical imaging fusion applications: An overview*, in *Signals, Systems and Computers, 2001. Conference Record of the Thirty-Fifth Asilomar Conference on* (IEEE, 2001), Vol. 2, pp. 1263–1267
- [6] B. Steux, C. Laugeau, L. Salesse, D. Wautier, *Fade: A vehicle detection and tracking system featuring monocular color vision and radar data fusion*, in *Intelligent Vehicle Symposium, 2002. IEEE* (IEEE, 2002), Vol. 2, pp. 632–639
- [7] M.A. Bueno-Ibarra, L. Acho et al., Optical Engineering **44**, 063601 (2005)
- [8] M.A. Bueno, J. Álvarez-Borrego, L. Acho, M.C. Chávez-Sánchez, Optical Engineering **44**, 093201 (2005)
- [9] S. Alférez, A. Merino, L. Bigorra, L. Mujica, M. Ruiz, J. Rodellar, American journal of clinical pathology **143**, 168 (2015)
- [10] P.B. Luh, X. Zhao, Y. Wang, L.S. Thakur, IEEE Transactions on Robotics and Automation **16**, 78 (2000)
- [11] G. Qi, J. Wang, Q. Zhang, F. Zeng, Z. Zhu, Future Internet **9**, 61 (2017)
- [12] F. Harrou, M.N. Nounou, H.N. Nounou, M. Madakyaru, Journal of loss prevention in the process industries **26**, 129 (2013)
- [13] Y. Xia, H. Leung, Information Sciences **277**, 808 (2014)
- [14] P. Miao, Y. Shen, Y. Li, L. Bao, Neurocomputing **177**, 120 (2016)
- [15] D. Ciunzo, A. Buonanno, M. D’Urso, F.A. Palmieri, *Distributed classification of multiple moving targets with binary wireless sensor networks*, in *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on* (IEEE, 2011), pp. 1–8
- [16] F. Palmieri, D. Ciunzo, Proc. of 20th Work. on Neural Networks, WIRN, Frontiers in Artificial Intelligence and Applications **226**, 107 (2011)
- [17] T. Wimalajeewa, P.K. Varshney, IEEE Signal Processing Letters **21**, 994 (2014)
- [18] M.M. Alyannezhadi, A.A. Pouyan, V. Abolghasemi, Journal of Electrical Systems and Information Technology **4**, 269 (2017)
- [19] N.A. Carlson, IEEE Transactions on Aerospace and Electronic Systems **26**, 517 (1990)
- [20] J.A. Stover, D.L. Hall, R.E. Gibson, IEEE Transactions on Industrial Electronics **43**, 403 (1996)

- [21] F. Russo, G. Ramponi, *Ieee transactions on instrumentation and measurement* **43**, 288 (1994)
- [22] Z. Wang, K.s. Leung, J. Wang, *Fuzzy Sets and Systems* **102**, 463 (1999)
- [23] W. Liu, Y. Lu, J.S. Fu, *IEEE Transactions on Aerospace and electronic Systems* **38**, 601 (2002)
- [24] L. Xu, J.Q. Zhang, Y. Yan, *IEEE transactions on instrumentation and measurement* **53**, 1539 (2004)
- [25] H. Nguyen, N. Cressie, A. Braverman, *Journal of the American Statistical Association* **107**, 1004 (2012)
- [26] M. Guerriero, L. Svensson, P. Willett, *IEEE Transactions on Signal Processing* **58**, 3417 (2010)
- [27] B. Khaleghi, A. Khamis, F.O. Karray, S.N. Razavi, *Information Fusion* **14**, 28 (2013)
- [28] I. Hwang, S. Kim, Y. Kim, C.E. Seah, *IEEE Transactions on Control Systems Technology* **18**, 636 (2010)
- [29] K. He, J. Sun, X. Tang, *IEEE transactions on pattern analysis and machine intelligence* **35**, 1397 (2013)
- [30] D. Van De Ville, M. Nachtegael, D. Van der Weken, E.E. Kerre, W. Philips, I. Lemahieu, *IEEE transactions on fuzzy systems* **11**, 429 (2003)
- [31] A. Achim, E.E. Kuruoglu, J. Zerubia, *IEEE Transactions on Image Processing* **15**, 2686 (2006)
- [32] O. Fialka, M. Cadik, *FFT and convolution performance in image filtering on GPU*, in *Information Visualization, 2006. IV 2006. Tenth International Conference on* (IEEE, 2006), pp. 609–614
- [33] R. Lukac, *Multidimensional Systems and Signal Processing* **15**, 169 (2004)
- [34] P.K. Saha, J.K. Udupa, *IEEE Transactions on Medical Imaging* **20**, 1140 (2001)
- [35] J. Shlens, preprint:1404.1100 (2014)
- [36] T. Bass, *Communications of the ACM* **43**, 99 (2000)
- [37] R.O. Duda, P.E. Hart, D.G. Stork, *Pattern classification* (Wiley, New York, 1973)
- [38] D.C. Lay, *Linear algebra and its applications*, 2016
- [39] C. Rafael Gonzalez, R. Woods, Pearson Education (2002)
- [40] L. Acho Zuppa, *Settling-time improvement in global convergence lagrangian networks*, in *ICINCO 2005* (2005), pp. 315–318
- [41] F. Li, *Neurocomputing* **73**, 2266 (2010)
- [42] S. Zhang, A. Constantinides, *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* **39**, 441 (1992)
- [43] Z. Zhu, G. Qi, Y. Chai, P. Li, *Applied Sciences* **7**, 161 (2017)
- [44] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, *IEEE transactions on image processing* **13**, 600 (2004)
- [45] M. Forti, P. Nistri, *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* **50**, 1421 (2003)
- [46] E.K. Chong, S. Hui, S.H. Zak, *IEEE Transactions on Automatic Control* **44**, 1995 (1999)
- [47] Y. Xia, G. Feng, J. Wang, *Neural Networks* **17**, 1003 (2004)
- [48] G. Costantini, R. Perfetti, M. Todisco, *IEEE Transactions on Neural Networks* **19**, 1804 (2008)
- [49] L. Alparone, L. Wald, J. Chanussot, C. Thomas, P. Gamba, L.M. Bruce, *IEEE Transactions on Geoscience and Remote Sensing* **45**, 3012 (2007)
- [50] C. He, Q. Liu, H. Li, H. Wang, *Procedia Engineering* **7**, 280 (2010)
- [51] Y. Zhang, *Photogramm. Eng. Remote Sens* **70**, 657 (2004)
- [52] C. Pohl, J.L. Van Genderen, *International journal of remote sensing* **19**, 823 (1998)