

# CALCULATION OF THE SIMILARITY RATE BETWEEN IMAGES BASED ON THE LOCAL MINIMA PRESENT THEREIN

K. Hourany<sup>1,2</sup>, F. Benmeddour<sup>2</sup>, E. Moulin<sup>2</sup>, J. Assaad<sup>2</sup> and Y. Zaatari<sup>1</sup>

<sup>1</sup>Applied Physics Laboratory, Lebanese University, Campus Fanar, BP 90656 Jdeidet, Lebanon

<sup>2</sup>IEMN UMR CNRS 8520, University of Valenciennes and Hainaut-Cambresis, F59313 Valenciennes cedex 9, France  
yzaatar@gmail.com

(Received 4 May 2016 - Accepted 6 June 2016)

## ABSTRACT

**Hourany, K., Benmeddour, F., Moulin, E., Assaad, J. and Zaatari, Y. Calculation of the similarity rate between images based on the local minima present therein. 2016. Lebanese Science Journal, 17(2): 177-192.**

*Image processing is a very vast field that includes both IT and applied mathematics. It is a discipline that studies the improvement and transformations of digital images hence permitting the improvement of the quality of these images and the extraction of information. The comparison of digital images is a paramount issue that has been discussed in several researches because of its various applications especially in the field of control and surveillance such as the Structural Health Monitoring using acoustic waves. The IT support of the images serves especially for comparing them notably in distinguishing differences between these images and quantifying them automatically. In this study we will present an algorithm, allowing us to calculate the similarity rate between images based on the local minima present therein. This algorithm is divided into two main parts. In the first part we will explain how to extract the local minima from an image and in the second part we will show how to calculate the similarity rate between two images.*

**Keywords:** image processing, local minima, similarity rate, digital images comparison

## INTRODUCTION

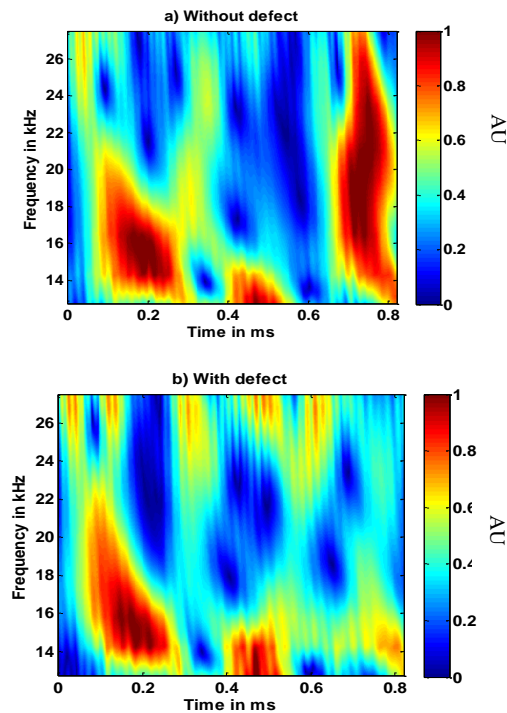
The comparison of images has been used in many domain, especially in the Structural Health monitoring (Abou Leyla, 2008) for the detection of defects in a structure using cross-correlation of ambient noise field.

Figure 1 represent an example of two images (frequency-time images) corresponding to the cross-correlation of noise signals received on two receivers placed on a pipeline in two cases, with and without defect. The difference between these images leads to conclude that

<http://dx.doi.org/10.22453/LSJ-017.2.177192>

National Council for Scientific Research – Lebanon 2016©  
lsj.cnrs.edu.lb/vol-17-no-2-2016/

there is a defect presented in the pipeline so we need an algorithm to automatically measure the similarity between these images.



**Figure 1. Frequency-time images corresponding to a structure with and without defect.**

The similarity measures between images are used in many domains such as content-based image retrieval (Smeulders, 2000), image classification (Baudrier, 2004), and quality assessment (Zhou, 2004). In the literature, there are different methods to make the comparison between images. These methods are divided into two main categories: firstly, methods based on the comparison of image descriptors (Bigand, 2010); in fact, those descriptors represent information on the color, texture and shape extracted from the image. Secondly, methods based on the direct comparison of the images without extracting characteristics; notably we may mention the Hausdorff distance (Huttenlocher, 1993) for example. Most of these methods are cumbersome and require significant computing time together with a large memory space.

Among recent research based on the measure of similarity between images, we can also cite the work of Silvia M. Ojeda (Ojeda, 2012). Ojeda proposes to use the codispersion coefficient to define a measure of similarity between images, which has been widely used in spatial statistics to quantify the association between two spatial processes.

In this paper, we will describe a comparison algorithm to test the degree of resemblance between images, having an advantage in computation time and memory space since only specific points (or pixels) of the image will be extracted and used for the comparison. These points are none other than the local minima present in the image. Figure 2 represents the local minima extracted from the images illustrated in figure 1 and represented by dots.

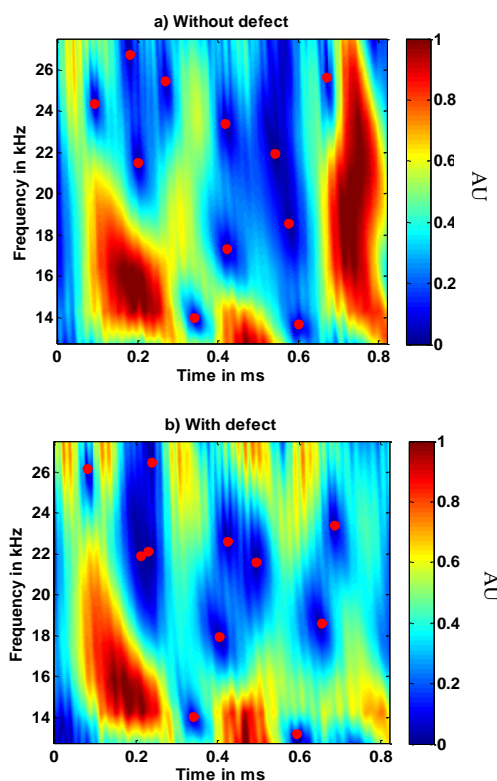


Figure 2. Frequency-time images with the extracted local minima.

### EXTRACTION OF LOCAL MINIMA OF AN IMAGE

For a better understanding of the algorithm, we consider a simple image of sixteen pixels defined by the matrix [I] whose indices are represented by ‘i’ (i = 1, 2, 3, 4) and ‘j’ (j = 1, 2, 3, 4) designating rows and columns respectively.

$$[I] = \begin{bmatrix} 2 & 2 & 2 & 1.1 \\ 4 & 3 & 1.2 & 1.3 \\ 2 & 2 & 2 & 5 \\ 1 & 1.1 & 1.2 & 1.3 \end{bmatrix} \quad (1.1)$$

This matrix corresponds to the image given in the figure 3, obtained by the “images c” command in Matlab.

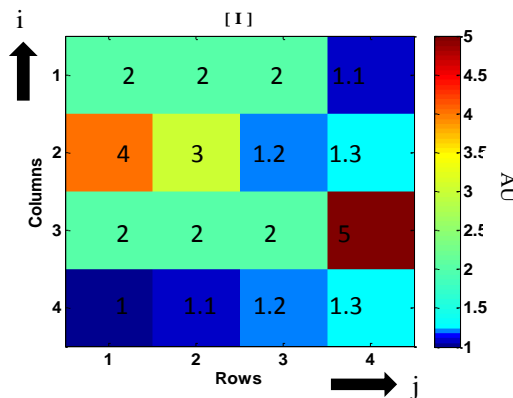


Figure 3. Image of the matrix [I].

Once the image of [I] is drawn, we sort it from the smallest to the largest value and we extract the first minimal values as if a threshold has been associated to this image. We choose to extract the first two values and we classify them in a vector named  $\mathbf{x}$ ; but since the value ‘1.1’ is repeated twice, we take both of them into account.

$$\mathbf{x} = [1 \ 1.1 \ 1.1] = [I_{4,1} \ I_{4,2} \ I_{1,4}]. \quad (1.2)$$

Later we will see which of the values of  $\mathbf{x}$ , are those which form a zone in the image corresponding to [I]. In other words these are the values which are in neighboring rows and columns in [I], so that the difference between their indices (i and / or j) is equal to ‘1’. In our case, we have two zones which are shown in Figure 4.

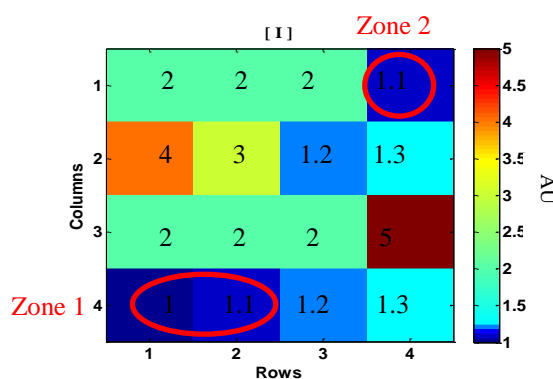


Figure 4. Representation of the two zones.

For this purpose, we extract the indices ‘i’ and ‘j’ of the elements of  $\mathbf{x}$  in the matrix [I] and we classify them in a matrix [Q] where the vectors  $\mathbf{r}$  and  $\mathbf{c}$  correspond to the rows ‘i’ and columns ‘j’ respectively.

$$[Q] = \begin{Bmatrix} \mathbf{r} \\ \mathbf{c} \end{Bmatrix} = \begin{bmatrix} 4 & 4 & 1 \\ 1 & 2 & 4 \end{bmatrix}. \tag{1.3}$$

Then we calculate the absolute value of the difference between each element of  $\mathbf{r}$  and all the other elements of the same vector. We do the same for the vector  $\mathbf{c}$ . The corresponding difference for  $\mathbf{r}$  will be classified in a matrix named [A] and the corresponding difference for  $\mathbf{c}$  will be classified in a matrix named [B].

$$\begin{bmatrix} 4-4 & 4-4 & 4-1 \\ 4-4 & 4-4 & 4-1 \\ 1-4 & 1-4 & 1-1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 3 \\ 0 & 0 & 3 \\ 3 & 3 & 0 \end{bmatrix} = [A]. \tag{1.4}$$

$$\begin{bmatrix} 1-1 & 1-2 & 1-4 \\ 2-1 & 2-2 & 2-4 \\ 4-1 & 4-2 & 4-4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 3 \\ 1 & 0 & 2 \\ 3 & 2 & 0 \end{bmatrix} = [B]. \tag{1.5}$$

Once [A] and [B] are calculated, we will calculate the sum of these matrices in order to be able to determine the pixels of the image corresponding to [I] having a difference equal to ‘1’ between their indices. Let the matrix [N] be the sum of [A] and [B].

$$\underbrace{\begin{bmatrix} 0 & 0 & 3 \\ 0 & 0 & 3 \\ 3 & 3 & 0 \end{bmatrix}}_{[A]} + \underbrace{\begin{bmatrix} 0 & 1 & 3 \\ 1 & 0 & 2 \\ 3 & 2 & 0 \end{bmatrix}}_{[B]} = \underbrace{\begin{bmatrix} 0 & 1 & 6 \\ 1 & 0 & 5 \\ 6 & 5 & 0 \end{bmatrix}}_{[N]}. \tag{1.6}$$

Therefore, the matrix [N] displays the distance between the various elements of the vector  $\mathbf{x}$ . The value ‘1’ having indices as ‘first row’ and ‘second column’ represents the distance between the first element of the vector  $\mathbf{x}$  having as indices ‘i = 4’, ‘j = 1’ and the second element of the vector  $\mathbf{x}$  having as indices ‘i = 4’, ‘j = 2’. It is similar for all other values of [N]. On the other hand, we note that the matrix [N] is symmetric, so we just consider with the upper triangular segment. It can also be noted that the diagonal of the matrix is zero since the latter is the distance between the pixel and itself (or between one element of the vector  $\mathbf{x}$  and itself) so it is necessary to take such element into consideration.

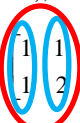
Since the elements of the diagonal of [N] are to be considered, we will associate to this diagonal, a value equal to ‘1’. Then the matrix [N] becomes as illustrated below.

$$[N] = \begin{bmatrix} 1 & 1 & 6 \\ 1 & 1 & 5 \\ 6 & 5 & 1 \end{bmatrix} \tag{1.7}$$

A final step remains to be implemented regarding the extraction of the local minima; that is to determine which elements of  $\mathbf{x}$  form a zone (elements circled in figure 4. Therefore, we extract the indices of the value '1' in [N] and we insert them in a matrix [M] where the vectors  $\mathbf{o}$  and  $\mathbf{p}$ , represent the rows and columns of these indices respectively.


$$[M] = \begin{Bmatrix} \mathbf{o} \\ \mathbf{p} \end{Bmatrix} = \begin{bmatrix} 1 & 1 & 2 & 3 \\ 1 & 2 & 2 & 3 \end{bmatrix} \tag{1.8}$$

If we compare the first column of [M] to other columns of the same matrix, i.e. the second one, we look for any intersection of identical values (i.e. the value '1' is in common between the first two columns), so we join them together.


$$[M] = \begin{bmatrix} 1 & 1 & 2 & 3 \\ 1 & 2 & 2 & 3 \end{bmatrix} \tag{1.9}$$


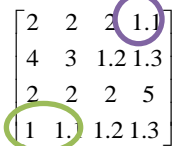
When the two joined columns are compared to the other columns, i.e. the third one, we find a similar intersection; hence we join them again together.

This process will be repeated until the intersections with all the columns of the matrix [M] are completed. Subsequently we note an intersection between the first three columns forming the first zone, whereas the fourth column forms a second zone.

$$[M] = \begin{bmatrix} 1 & 1 & 2 & 3 \\ 1 & 2 & 2 & 3 \end{bmatrix} \tag{1.10}$$


Since, the first zone contains the values '1' and '2' this means that the first and second columns of the matrix [Q] form a first zone. The second zone contains the value '3' which means that the third column of the matrix [Q] forms a second zone. Now if we go back to the initial matrix [I], we can precisely define which values will eventually form a zone since the matrix [Q] is none other than the matrix of the indices 'i' and 'j' of these values represented in the vectors  $\mathbf{r}$ , and  $\mathbf{c}$ .

$$[Q] = \begin{Bmatrix} \mathbf{R} \\ \mathbf{C} \end{Bmatrix} = \begin{bmatrix} 4 & 4 & 1 \\ 1 & 2 & 4 \end{bmatrix} \tag{1.11}$$


$$[I] = \begin{bmatrix} 2 & 2 & 2 & 1.1 \\ 4 & 3 & 1.2 & 1.3 \\ 2 & 2 & 2 & 5 \\ 1 & 1. & 1.2 & 1.3 \end{bmatrix} \tag{1.12}$$


Finally, and after tracing the regrouping of the elements of  $x$  in zones, the local minima to be extracted are actually the minimum value in each zone. In the first zone the value '1' is the local minimum, while in the second one, and since this zone contains only a single value which is '1.1', then this value is the local minimum.

Figure 3 shows the local minima extracted from the image of [I].

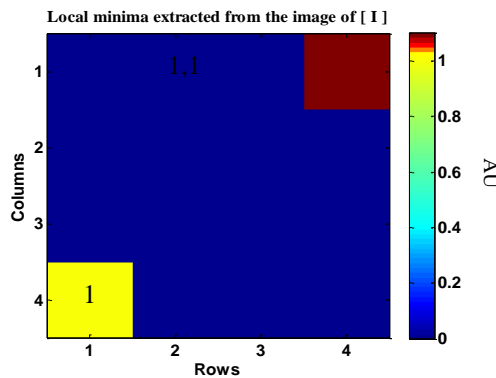


Figure 5. Local minima extracted from the image of [I].

It is noteworthy to observe that the local minima were duly extracted. In this section, we have shown in details how to extract the local minima of a corresponding image of a certain matrix. Our next section will be dedicated to the calculation of the similarity rate between two images based on the local minima extracted from each image.

**CALCULATION OF THE SIMILARITY RATE BETWEEN TWO IMAGES**

Let's consider now the matrix [II] defined as follows.

$$[II] = \begin{bmatrix} 1.1 & 2 & 2 & 2 \\ 1.3 & 1.2 & 3 & 4 \\ 5 & 2 & 2 & 2 \\ 1.1 & 1.2 & 1.3 & 1 \end{bmatrix} \tag{1.13}$$

The corresponding image is represented in the following figure.

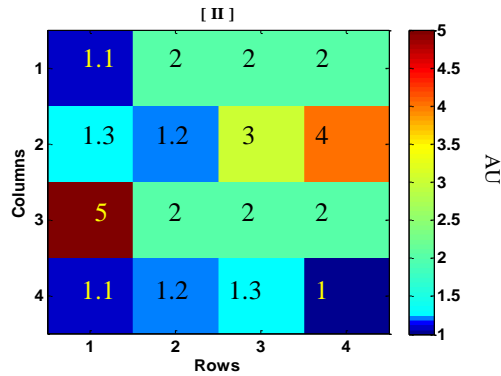


Figure 6. Image of the matrix [II].

In application of the approach explained in the previous section, we obtain figure 7 which shows the extracted local minima.

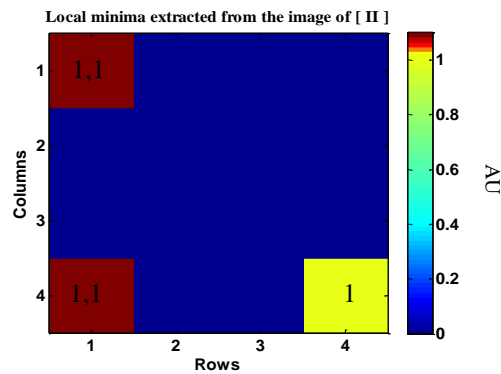
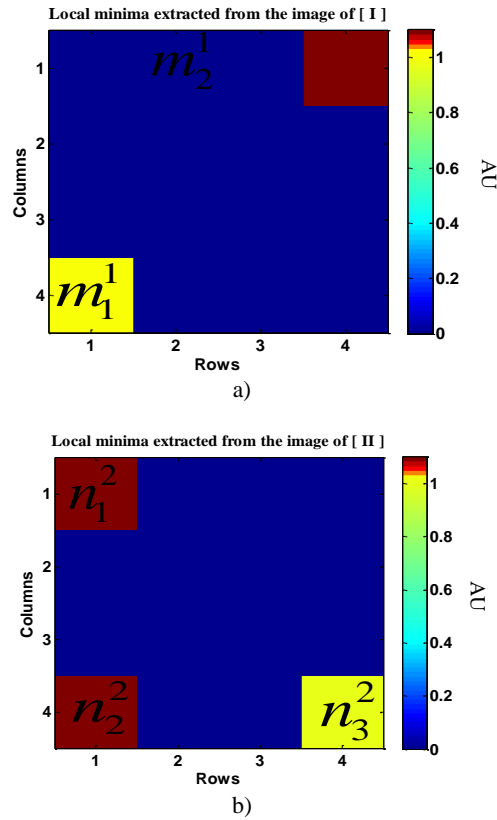


Figure 7. Local minima extracted from the image of [II].

We calculate the Euclidean distance between each local minimum of the image of [I] and those of the image of [II].

Let  $m_z^y$  and  $n_{z'}^{y'}$  be the labels of the local minima of the images of [I] and [II] respectively, where  $y$  and  $y'$  are the image numbers ( $y=1$  for the image of [I] and  $y'=2$  for the image of [II]) and  $z$  and  $z'$  the number of the local minimum in each image ( $z = 1, 2$  for the image of [I] and  $z'= 1, 2, 3$  to the image of [II]). These labels are shown in Figure 8 knowing that  $m_z^y$  and  $n_{z'}^{y'}$  are none other than the values '1' and '1.1' respectively (see Figure 3). It is the same for  $m_z^y$  and  $n_{z'}^{y'}$  and  $n_{z'}^{y'}$  are none other than the values '1.1', '1.1' and '1' respectively (see Figure 5).





**Figure 8. Local minima extracted from the images of a) matrix [I] b) matrix [II].**

The Euclidean distance between each of the local minima of the image of [I] and each of the local minima of the image of [II] is given by the following equation:

$$d_z^{z'} = \sqrt{(c(m_z^y) - c(n_{z'}^{y'}))^2 + (l(m_z^y) - l(n_{z'}^{y'}))^2} \quad (1.14)$$

Where 'l' and 'c' are the indices ('l' for lines and 'c' for columns) of the local minima in the matrices [I] and [II].

The calculated distances are represented by the matrix [Dist] as follows:

$$[\text{Dist}] = \begin{bmatrix} 3 & 0 & 3 \\ 3 & 4.2426 & 3 \end{bmatrix} = \begin{bmatrix} d_1^1 & d_1^2 & d_1^3 \\ d_2^1 & d_2^2 & d_2^3 \end{bmatrix}, \quad (1.15)$$

Where the first value '3' having as indices, first line and first column in the matrix [Dist], represents the distance between the first minimum of [I] and the first minimum of [II]. This value '3' is obtained as follows:

$$\begin{aligned} d_1^1 &= \sqrt{(c(m_1^1) - c(n_1^2))^2 + (l(m_1^1) - l(n_1^2))^2} \\ &= \sqrt{(1-1)^2 + (4-1)^2} = 3 \end{aligned} \quad (1.16)$$

It is the same for all the other values of the matrix [Dist].

Secondly, and after calculating the matrix [Dist], we must calculate the average distance between the local minima of the image of [I] with the ones which are the closest to them in the image of [II]. In the case of matrix [Dist], we search the smallest value, which is the value '0' and we classify it in a vector named **dist1**, then we eliminate the line and the column related to this value in [Dist].

$$[\text{Dist}] = \begin{bmatrix} 3 & 0 & 3 \\ 3 & 4.2426 & 3 \end{bmatrix}. \quad (1.17)$$

$$[\text{Dist}] = \begin{bmatrix} \cancel{3} & \textcircled{0} & \cancel{3} \\ 3 & 4.2426 & 3 \end{bmatrix}. \quad (1.18)$$

$$\mathbf{dist1} = [0]. \quad (1.19)$$

On the other hand, the indices of the value '0' in [Dist] are (1, 2) or first line and second column, so this is the distance between the first minimum of the image of [I] and the second minimum of the image of [II]; Therefore we should also check these minima in the corresponding images as shown in Figure 9.

Our aim being to seek the smallest value again in [Dist], and after removing the previous line and column, we find the value '3'. In case there were several identical values (i.e. that a minimum of an image is equidistant to several minima of another image), the value to be considered in [Dist] shall be the one with the highest value in its vicinity. In our case and knowing that there is only one minimum in the image of [I] then we take any value '3' of the matrix [Dist] and we classify it in **dist1**, then we check again respectively the corresponding line and column of that value and the corresponding minima in the images of each matrix, as we did previously (see Figure 10).

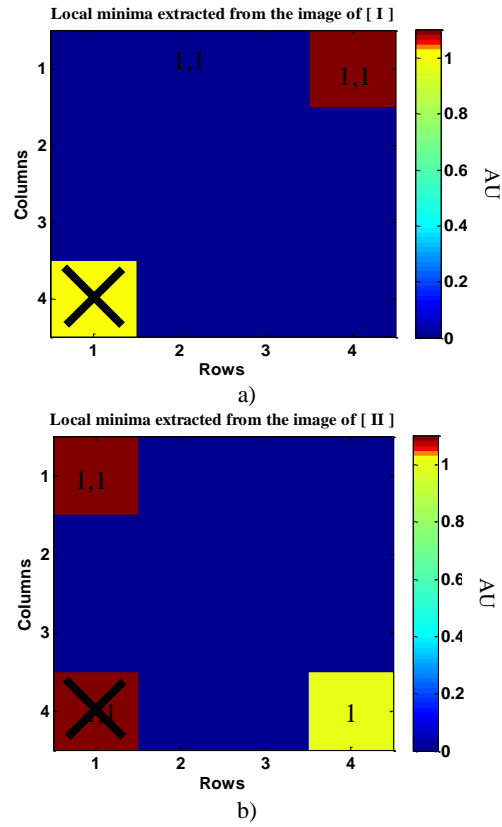
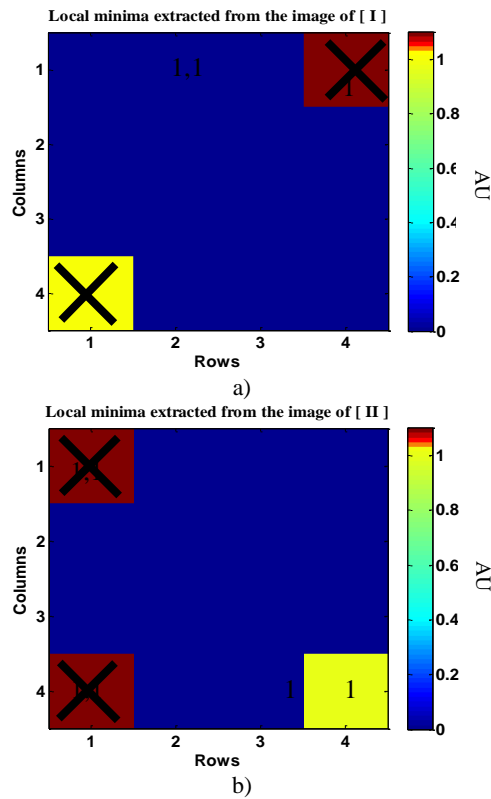


Figure 9. Local minima extracted from the images of a) matrix [I] b) matrix [II].

$$[\text{Dist}] = \begin{bmatrix} 3 & 4.2 & 26 & 3 \\ 3 & 4.2 & 26 & 3 \end{bmatrix} \quad (1.20)$$

$$[\text{Dist}] = \begin{bmatrix} 2 & 4.2 & 26 & 2 \\ 2 & 4.2 & 26 & 2 \end{bmatrix} \quad (1.21)$$

$$\mathbf{dist1} = [0 \ 3]. \quad (1.22)$$



**Figure 10. Local minima extracted from the images of a) matrix [I] b) matrix [II].**

In case there is a point in the image of [II] which does not have a neighbor, we associate to the vector **dist1** a value equal to the distance of the diagonal of the image which is equal to the largest distance obtainable between two local minima. In other words, we associate to the vector **dist1** the value '4.2426'. Finally, we obtain the following vector **dist1**:

$$\mathbf{dist1} = [0 \ 3 \ 4.2426]. \quad (1.23)$$

Once the vector **dist1** is determined, we calculate the average value of the elements of this vector which is the sum the elements divided by the number of these elements. This average is equal to '2.4142', which corresponds to the average of the distances between the local minima of the image of [I] and those which are their neighbors in the image of [II]. The greatest value that the average can consider is the value of the distance of the diagonal of an image; in this case the value is equal to '4.2426'. This means that each minimum of an image is very far from each minimum of another image, so the two images do not look like, hence the similarity rate is equal to:

$$100 - \left( \frac{4.2426}{4.2426} \times 100 \right) = 0\%. \tag{1.24}$$

Whereas the smallest value to be considered as an average is '0', this means that the two images are identical. So the similarity rate is equal to:

$$100 - \left( \frac{0}{4.2426} \times 100 \right) = 100\%. \tag{1.25}$$

In this section, we explained the calculation of the similarity rate between images. It is based on the calculation of the average value of the distances between a local minimum of an image and the one which is adjacent to it in the other image. Next we will give some examples in order to show the variation of the similarity rate upon variation of the local minima.

**VARIATION OF THE SIMILARITY RATE ACCORDING TO THE VARIATION OF THE LOCAL MINIMA**

In this section, four examples are given where the image of the matrix [II] is taken as a reference, and it is compared to four images corresponding to four different matrices where only one minimum (which is 1.1) of each image is moving.

In the first example we compare the image of the matrix [II] to that of the matrix [III] shown as follows:

$$[III] = \begin{bmatrix} 2 & 2 & 1.1 & 2 \\ 4 & 3 & 1.2 & 1.3 \\ 2 & 2 & 2 & 5 \\ 1 & 1.1 & 1.2 & 1.3 \end{bmatrix}. \tag{1.26}$$

The corresponding image of this matrix is shown in Figure 11.

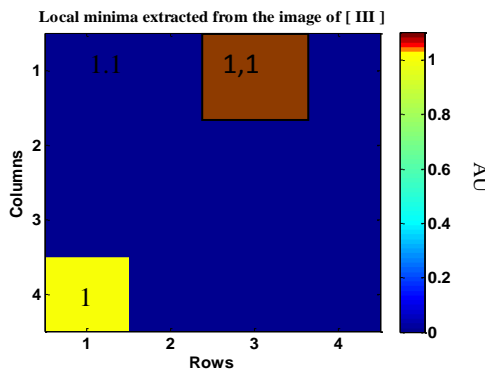


Figure 11. Local minima extracted from the image of [III].

The average distance between the local minima in the image of [III] and those which are their neighbors in the image of [II] is equal to '2.0809'; so the similarity rate is equal to:

$$100 - \left( \frac{2.0809}{4.2426} \times 100 \right) = 50.96\%. \quad (1.27)$$

In the second example, we compare the image of [II] to that of [IV]:

$$[IV] = \begin{bmatrix} 2 & 1.1 & 2 & 2 \\ 4 & 3 & 1.2 & 1.3 \\ 2 & 2 & 2 & 5 \\ 1 & 1.1 & 1.2 & 1.3 \end{bmatrix}. \quad (1.28)$$

The corresponding image of matrix [IV] is shown in Figure 12.

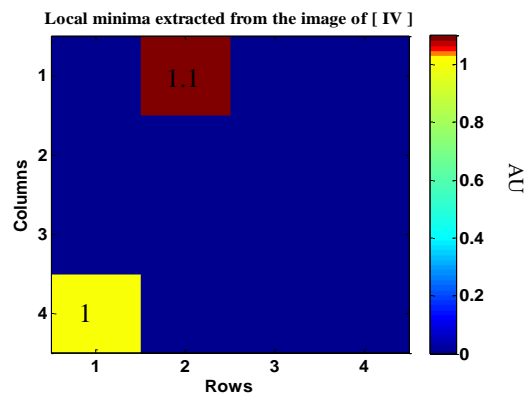


Figure 12. Local minima extracted from the image of [IV].

In this case, the average distance between the local minima of the two images corresponding to the two matrices [IV] and [II] is equal to '1.7475' and the similarity rate is equal to:

$$100 - \left( \frac{1.7475}{4.2426} \times 100 \right) = 58.82\%. \quad (1.29)$$

In the third example, we compare the image of [II] to that of [V]:

$$[V] = \begin{bmatrix} 1.1 & 2 & 2 & 2 \\ 4 & 3 & 1.2 & 1.3 \\ 2 & 2 & 2 & 5 \\ 1 & 1.1 & 1.2 & 1.3 \end{bmatrix}. \quad (1.30)$$

Its corresponding image is shown in Figure 13.

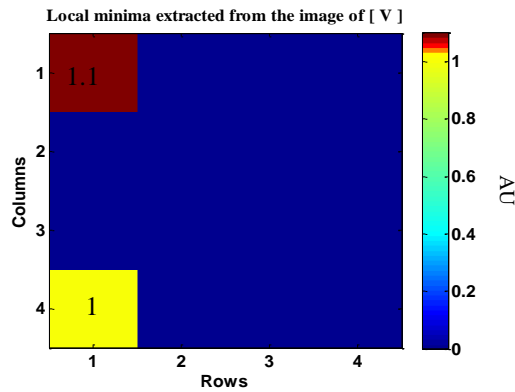


Figure 13. Local minima extracted from the image of [V].

In this case the similarity rate is equal to:

$$100 - \left( \frac{1.4142}{4.2426} \times 100 \right) = 66.67\%. \tag{1.31}$$

Finally, in the last example, we compare the image of [II] to itself. In this case the average distance is equal to '0' and the similarity rate is equal to:

$$100 - \left( \frac{0}{4.2426} \times 100 \right) = 100\%. \tag{1.32}$$

Based on the four examples given above, we can observe that the similarity rate increases when the two images that we compared do look alike. In this section, we explained the calculation of the similarity rate between two images and for such purpose four examples were given in order to show how this rate varies when a minimum is approaching or moving away. Note that this calculation was applied on matrices of size (4 x4) but this calculation would still be valid when using matrices of any size.

### CONCLUSION

This study demonstrates in detail an algorithm to test the degree of resemblance between images, based on local minima present therein. In the first section the method to extract the local minima of the images has been presented, and in the second the manner to calculate the similarity rate between images has been explained. This algorithm will be applied in future works to the detection of defects in a structure.

### ACKNOWLEDGMENT

K. Hourany thanks the Lebanese National Council for Scientific Research (CNRS-L) for the financial assistance.

## REFERENCES

- Abou Leyla, N., Moulin, E., Assaad, J., Grondel, S. and Poussot, P. 2008. Structural health monitoring using cross-correlation of an ambient noise field. *The Journal of the Acoustical Society of America*, 123(5): 3698-3698.
- Baudrier, E., Millon, G., Nicolier, F. and Ruan, S. 2004. A new similarity measure using Hausdorff distance map. ICIP '04. 2004. *IEEE*, 1: 667-672.
- Bigand, A. and Colot, O. 2010. Fuzzy filter based on interval-valued fuzzy sets for image filtering. *Fuzzy Sets and Systems*, 161(1): 96-117.
- Huttenlocher, D.P., Klanderman, G.A. and Rucklidge, W.J. 1993. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9): 850-863.
- Ojeda, S.M., Vallejos, R.O., Lamberti, P.W. 2012. Measure of similarity between images based on the codispersion coefficient. *Journal of Electronic Imaging*, 21(2): 23-19.
- Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A. and Jain, R. 2000. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12): 1349-1380.
- Wang, Z., Bovik, A.C., Sheikh, H.R. and Simoncelli, E.P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600-612.