2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) | 978-1-6654-0690-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/UEMCON53757.2021.9666634

A modified Normalized Cross-Correlation algorithm for Embedded Systems

Ismael Lopez Sanchez*
Embedded Systems Specialization
se729345@iteso.mx

Miguel Angel Rosas Galaviz*
Embedded Systems Specialization
se729226@iteso.mx

Damian Gomez Herrera*
Embedded Systems Specialization
xe729752@iteso.mx

Dr. Luis Rizo Dominguez*
IEEE Member
lrizo@iteso.mx

***Abstract*—Problems related to image matching, such as processing time, performance, and changes in rotation and scale are widely studied. However, creating simple solutions for image matching that can be implemented in an embedded system has been overlooked. Therefore, this paper proposes a modified Normalized Cross-Correlation algorithm for embedded systems aimed at creating a low-cost solution for a cement tile application. The algorithm relies on the concept of template matching and it makes use of cross-correlation, zero-mean normalization, and thresholding for its implementation. Experimental results show that the algorithm is suitable for a cement tile application and appropriate to detect defective cement tiles during an on-site installation or manufacturing process. Finally, it is shown that the algorithm can act as a discriminator to identify quality non-conformance issues.**

Keywords— Normalized Cross-Correlation (NCC) algorithm, template, reference image, embedded system

I. Introduction

Image processing methods have played a significant role in detecting defects to prevent consumers from purchasing defective products. One of these methods involves template matching techniques, which are widely used to compare images and evaluate if a product has been manufactured with the desired quality [1]. The Sum of Absolute Differences (SAD), the Sum of Squared Differences (SSD), and the Normalized Cross-Correlation (NCC), are some of the template matching algorithms that have been proposed to correlate two images [2]. According to [3], NCC algorithms are widely reported in the literature; they are based on correlation measures that detect similarities and they are characterized by their robustness and effectiveness.

In essence, NCC algorithms are used to shift and compare an input image, also known as template, with windows of the same size in a reference image so that different correlation coefficients can be obtained. These correlation coefficients are necessary to quantify the degree of similarity between the template and each image window. By measuring this, it is possible to determine if the template is an acceptable match; hence, defective products can be segregated with the intention of not compromising the quality of a product.

Although there have been numerous studies aimed at improving image matching techniques based on NCC algorithms, they only focus on solving problems related to image matching, such as processing time, performance, and changes in rotation and scale [4,5,6]. Moreover, these studies have overlooked the creation of a simple and affordable solution for micro and small-sized enterprises.

In contrast with large and medium-sized enterprises, micro and small-sized enterprises do not have the same possibilities to spend their budget in state-of-the-art technology. Thus, these companies seek straightforward and stand-alone solutions that can be easily integrated in their manufacturing process without representing a huge investment. Previous research has proposed the implementation of NCC algorithms in workstations, computers, or even using MATLAB, a programming software platform designed for engineers and scientists [7,8]. These NCC algorithms require significant computational resources, which complicate their implementation in a single embedded system [9].

An NCC algorithm has been selected due to its simplicity and effectiveness when it is implemented in an embedded system. Therefore, this paper proposes the implementation of a modified NCC algorithm in an embedded system based on image matching for a low-cost cement tile application.

The rest of the paper is organized as follows. Section II introduces the NCC fundamentals for computing the cross-correlation coefficients. The proposal for the modified NCC algorithm is presented in section III. The algorithm implementation is described in section IV. Lastly, the algorithm validation and the results are given in section V.

II. Normalized Cross-Correlation (NCC)

An NCC algorithm locates a template into a reference image to calculate a correlation coefficient; typically, the template origin is overlapped with the origin of the reference image to set the coordinates of both origins at (0, 0) as shown in Fig. 1. After computing the first correlation coefficient, the template is shifted around the reference image to compute the rest of the correlation coefficients to find the maximum value of the NCC function according to the following equation:

$$NCC(x,y) = \frac{\sum_{j=1}^{m}\sum_{i=1}^{n} I(x+i,y+j)\cdot T(i,j)}{\sqrt{\sum_{j=1}^{m}\sum_{i=1}^{n} I(x+i,y+j)^2}\cdot\sqrt{\sum_{j=1}^{m}\sum_{i=1}^{n} T(i,j)^2}} \quad (1)$$

where x and y denote the coordinates of the reference image of size M x N, $x= 0...N-n$ *and* $y= 0...M-m$, and i and j are the

978-1-6654-0690-1/21/$31.00 ©2021 IEEE

*ITESO, Jesuit University of Guadalajara

shifting variables that represent the distance from the corresponding *x* and *y* position. Moreover, the numerator of (1) denotes the cross-correlation between the template and the reference image, while the two terms on the denominator represent the normalization of the reference image *I,* and of the template *T*.

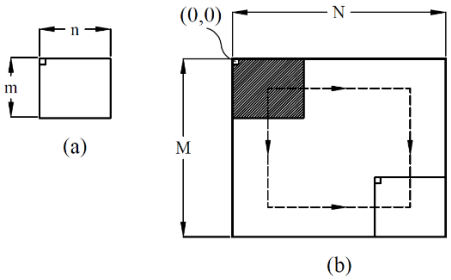

Fig. 1. (a) Template and (b) Reference image.

III. Proposed Modified NCC Algorithm

The proposed algorithm in this paper showed a better performance when the images had the same reference; this was accomplished by sharing the same mean. Thus, a modification of the traditional NCC algorithm with a zero-mean normalization, is proposed. The key elements of the algorithm can be listed as follows: cross-correlation, zero-mean normalization, and thresholding.

A. Cross-Correlation

The algorithm uses the numerator of (1) to calculate the dot product of two numeric arrays: the reference image and the template. In other words, an element-by-element multiplication of the template and the reference image vectors is performed to produce a vector of products and then, these vectors of products are summed up to produce a scalar result, also known as the correlation coefficient.

B. Zero-Mean Normalization

Before the dot product can be used, the two numeric arrays must be normalized. One alternative to normalize a cross-correlation function is to use the denominator of (1). This method is particularly useful when it is implemented in a computer. However, executing this function in an embedded system represents a great challenge because of the numerical operations that must be performed. For this reason, the mean is subtracted from both numeric arrays, which is another method to accomplish a normalized function. Specifically, the mean of the template pixels is subtracted from each of the elements contained in the template numeric array; similarly, the reference image is normalized to zero mean. Hence, the equations to calculate the mean are,

$$\bar{T} = \frac{\sum_{i=j=1,}^{i=n,j=m} T(i,j)}{n \cdot m} \text{ and } \bar{I} = \frac{\sum_{i=j=1,}^{i=N,j=M} I(i,j)}{M \cdot N} \qquad (2)$$

C. Thresholding

Once the pixel values of the images are normalized and the correlation coefficients are calculated, a criterion for the similarity measure must be determined. Thresholding is a simple and efficient approach that is used in image processing to separate regions of interest within an image. Normally, a pixel in an image is compared with a constant threshold and it is replaced with a black pixel if its value is less than the defined constant or with a white pixel if its value is greater than that constant, or vice versa. Thus, the purpose of a thresholding operation is to create a binary image. Since a binary image can be easily related to a pass-fail criterion, thresholding supports the definition of a similarity measure.

Although thresholding is a method that processes a grayscale or color image into a black and white image, the definition of a cutoff value is required to create a binary output. By leveraging on the cross-correlation and the zero-mean normalization, the implementation of the algorithm can be simplified. However, instead of using the correlation between two different images, the autocorrelation function (ACF) can be used,

$$ACF(x,y) = \sum_{j=1}^{M} \sum_{i=1}^{N} T(x+i, y+j) \cdot T(i,j) \qquad (3)$$

The autocorrelation function is an adequate method to determine the cutoff value since the maximum correlation coefficient that can be attained from an image is the resulting value of the dot product from the same image. Therefore, solving for (3) results in the maximum degree of similarity of any image. Consequently, a cutoff threshold can be established by multiplying the autocorrelation coefficient by a constant,

$$Threshold = constant \cdot ACF(x,y) \qquad (4)$$

where the constant lies between 0 and 1. The characterization of the constant value depends on the intensity of the image.

These three elements were used to build the proposed algorithm. While the cross-correlation and the zero-mean normalization are responsible for quantifying the degree of similarity between the template and the reference image, thresholding is the method responsible for dictating if the correlation coefficient value means that a satisfactory match has been found.

IV. Implementation of the Proposed NCC Algorithm

There are four significant aspects that were considered for the implementation.

Firstly, the clear definition of the elements used in the numerator in (1) and (3), as well as relating them to a cement tile application for validation purposes, were essential to specify the implementation approach. Typically, to obtain the correlation coefficients, the template is shifted around the reference image as indicated in (1). Note that if *T* and *I* are interchanged, then the outcome of the cross-correlation (CC) is the same. Thus, in equation (5), *T* and *I* were swapped as follows:

$$CC(x,y) = \sum_{j=1}^{M} \sum_{i=1}^{N} T(x+i, y+j) \cdot I(i,j) \qquad (5)$$

to explicitly state that the reference image is shifted around the template since this is a more convenient approach for the cement tile application and, mathematically speaking, it yields the same output. Consequently, (3) can be expressed as,

$$ACF(x,y) = \sum_{j=1}^{M} \sum_{i=1}^{N} I(x+i, y+j) \cdot I(i,j) \qquad (6)$$

to illustrate that the threshold in (4) is obtained from the reference image *I*.

Once the elements are properly defined, the boundaries for moving the images must be established. As expressed in (5), Fig. 2 shows that the reference image is shifted around the template taking into consideration the size of the image to delimit the region where no calculations are required.

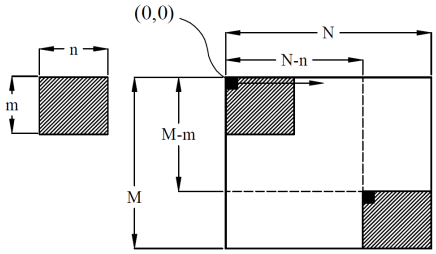

Fig. 2. Shifting the reference image around the template.

Secondly, determining the procedure to compute the algorithm is essential since embedded systems are known for their limited resources. Although this is a valid belief, it is no longer the rule. Current technological developments have led embedded systems to use more than one microprocessor core, allowing tasks to be run in parallel. Therefore, the reference image is shifted around the template by sliding it using four threads as shown in Fig. 3.

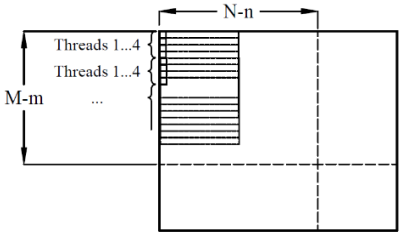

Fig. 3. Parallel computing scheme considering 4 threads.

Thirdly, since an embedded system has memory limitations, an optimization for computing the mean was considered. Hence, instead of using (2) – the traditional algebraic function for the mean, the equation that was used within the implementation is the following recurrence relation:

$$\bar{T}_{k+1} = \left(\frac{k}{k+1}\right)\bar{T}_k + \left(\frac{1}{k+1}\right)T(k+1) \tag{7}$$

where *k* is related to the iteration number.

Lastly, the threshold in (4) is set in order to guarantee the quality of the cement tile. This number is obtained from the autocorrelation function and then it is compared to the outcome of the modified NCC algorithm; defining this approach was the key aspect for stipulating a pass-fail criterion.

The flow chart in Fig. 4 depicts the modified NCC algorithm that was implemented in an embedded system using (5), (6), and (7).

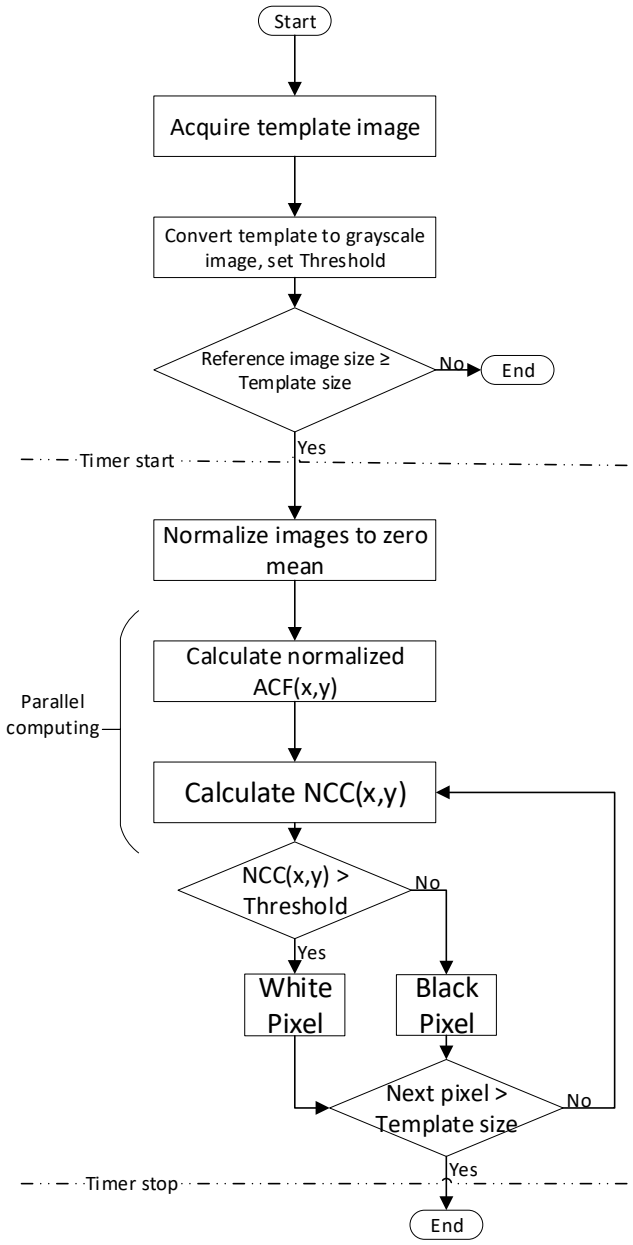

Fig. 4. Flow chart of the modified NCC algorithm.

V. Experimental Results

This section presents the results of the embedded system implementation and testing using the modified NCC algorithm as described in Fig. 4. To demonstrate the effectiveness of the algorithm, the experiments were tested on real grayscale images of a cement tile application in terms of a pass-fail criterion and processing time. Fig. 5(a), Fig. 5(b), Fig. 6(a), Fig. 6(b), and Fig. 8(a) show the reference images and templates that were used during the algorithm validation. All images were taken with the same camera. The experiments were conducted on a RASPBERRY PI 3 Model B, processor Broadcom BCM2711, Quad core Cortex-A72 (ARM v8), and 64-bit SoC @ 1.5GHz.

Fig. 5(a) shows the typical four-image pattern used during the installation process of cement tiles. This means that four

cement tiles are placed in a specific sequence. The algorithm utilizes this four-image pattern as the template since this is the image that must be validated. Fig. 5(b) shows the reference image that is shifted around Fig. 5(a) to determine if an acceptable match can be found. In other words, the reference image is considered the golden sample that dictates the degree of similarity that must be achieved by setting the threshold value according to the autocorrelation function. Fig 5(c) shows the binary image that is obtained after executing the proposed algorithm. The thresholding operation of the algorithm simplifies the visualization of the outcome by indicating in black the region where no matches were found and in white the matches that are acceptable. As shown in Fig. 5(c), the upper right corner was replaced with white pixels demonstrating that Fig. 5(b) was only found in that region. Fig. 5(d) shows the cross-correlation plot of the outcome. Thus, the modified NCC algorithm can identify the reference image within the template and, therefore, this algorithm can also detect quality non-conformance issues.

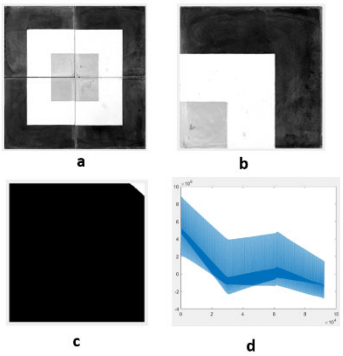

Fig. 5. (a) Template with four-image pattern (b) Square model for reference image (c) Image proccessing output considering a specific threshold (d) Cross-correlation outcome.

A different four-image pattern and distinct template were used during the experiments to verify the robustness of the algorithm. Similar satisfactory results as previously discussed for Fig. 5 were attained. These results are shown in Fig. 6.

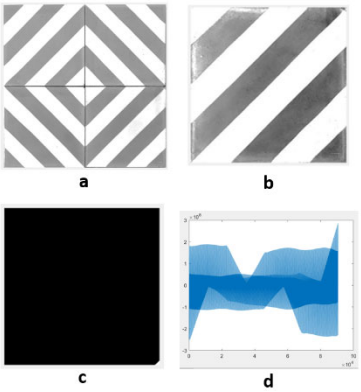

Fig. 6. (a) Template with four-image pattern (b) Diagonal model for reference image (c) Image proccessing output considering a specific threshold (d) Cross-correlation outcome.

The cement tiles are produced using a handmade manufacturing process where only one tile can be produced at a time. The modified NCC algorithm proposed in this paper can also evaluate a single piece. Therefore, this algorithm was subjected to an additional test where a single-image pattern and the same template from Fig. 5(b), were used.

Fig. 7(a) depicts the template as a single-image pattern. Fig. 7(b) shows the reference image. Note that Fig. 7(a) and Fig. 7(b) are images of cement tiles that were manufactured at different points in time. Fig. 7(c) shows the image processing output, which is a single white pixel that corresponds to the correlation coefficient obtained from the algorithm. Hence, the modified NCC algorithm can also detect the degree of similarity of a single-image pattern; specifically, Fig. 7(c) exemplifies a satisfactory match. Moreover, it was observed that the processing time is related to the image size.

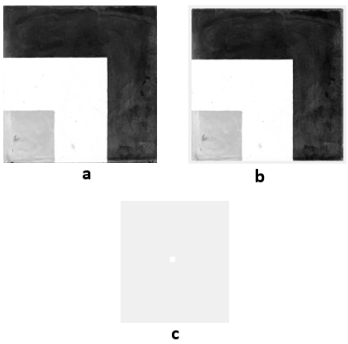

Fig. 7. (a) Template with correct single-image pattern (b) Square model for reference image (c) Image proccessing output considering a specific threshold.

An incorrect single-image pattern, Fig. 8(a), was used to test that the algorithm can determine that no acceptable matches are found. The results of Fig. 8 show that the modified NCC algorithm can also detect quality non-conformance issues on a single-image pattern.

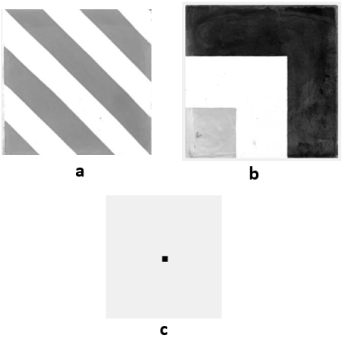

Fig. 8. (a) Template with incorrect single-image pattern (b) Square model for reference image (c) Image proccessing output considering a specific threshold.

In order to evaluate the time performance of the modified algorithm, Table I reports the processing times for two different template sizes and three reference image sizes considering the four-image pattern of Fig. 5 and Fig. 6.

TABLE I. ALGORITHM RESULTS USING A FOUR-IMAGE PATTERN

Sample No.	*Template Size*	*Reference Image Size*	*Execution Time (s)*
1.	480x480	100x100	36.624
2.	480x480	150x150	60.144
3.	480x480	230x230	83.069
4.	320x240	100x100	7.471
5.	320x240	150x150	8.819
6.	320x240	230x230	1.381

Table II shows the execution times for images of various sizes and including the single-image pattern of Fig. 7 and Fig. 8, which implies that the template and the reference image are of the same size.

TABLE II. ALGORITHM RESULTS USING A SINGLE-IMAGE PATTERN

Sample No.	*Template Size*	*Reference Image Size*	*Execution Time (s)*
1.	100x100	100x100	0.007
2.	130x130	130x130	0.012
3.	150x150	150x150	0.014
4.	180x180	180x180	0.011
5.	200x200	200x200	0.011
6.	230x230	230x230	0.016

In accordance with other experiments reported in the literature, reduced processing times are obtained with the modified NCC algorithm. For instance, a study reports an experiment where an image of size 384 x 512 and a template of size 150 x150 was used resulting in a processing time of 346.9846 s [3]. Meanwhile, the execution time for the proposed algorithm with similar image sizes – 480 x 480 and 150x150 – was 60.144 s. In general terms, the processing time was reduced to 17% during the four-image pattern experiment. From the experimental results, it is observed that the processing time is proportional to the number of iterations when shifting the reference image around the template. The execution time is also proportional to the number of pixels.

Furthermore, since the cement tile application relies on a handmade manufacturing process, it implies a slow production rate and the execution times attained from the modified NCC algorithm do not impact negatively on the production line.

VI. CONCLUSION

In this paper, a modified NCC algorithm was implemented in an embedded system using cross-correlation, zero-mean normalization, and thresholding. The experimental results show that the proposed algorithm is suitable for a cement tile application. Moreover, the execution times obtained in the embedded system implementation are appropriate for the manufacturing process.

The proposed algorithm was tested under two different scenarios – the four-image pattern and the single-image pattern – to determine an acceptable match. Consequently, when the threshold is set, it can be used as a discriminator of defective cement tiles during the on-site installation or manufacturing process.

Future research could include an algorithm to detect changes in illumination and color variation. Similarly, edge detection algorithms can be developed on top of the current implementation. Additionally, the proposed algorithm can be implemented in other boards.

ACKNOWLEDGMENTS

The authors thank the cement tile company, Studio Victoria, for manufacturing the tiles used in this study.

REFERENCES

[1] S. Omachi and M. Omachi, "Fast Template Matching With Polynomials," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2139–2149, Aug. 2007.

[2] J. N. Sarvaiya, S. Patnaik, and S. Bombaywala, "Image Registration by Template Matching Using Normalized Cross-Correlation," in *2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies*, Dec. 2009, pp. 819–822.

[3] M. B. Hisham, S. N. Yaakob, R. A. A. Raof, A. B. A. Nazren, and N. M. W. Embedded, "Template Matching using Sum of Squared Difference and Normalized Cross Correlation," in *2015 IEEE Student Conference on Research and Development (SCOReD)*, Dec. 2015, pp. 100–104.

[4] L. Di Stefano, S. Mattoccia, and M. Mola, "An efficient algorithm for exhaustive template matching based on normalized cross correlation," in *12th International Conference on Image Analysis and Processing, 2003.Proceedings.*, Sep. 2003, pp. 322–327.

[5] S.-D. Wei and S.-H. Lai, "Fast Template Matching Based on Normalized Cross Correlation With Adaptive Multilevel Winner Update," *IEEE Transactions on Image Processing*, vol. 17, no. 11, pp. 2227–2235, Nov. 2008.

[6] F. Zhao, Q. Huang, and W. Gao, "Image Matching by Normalized Cross-Correlation," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, May 2006, vol. 2, pp. II–II.

[7] Y. R. Rao, N. Prathapani, and E. Nagabhooshanam, "Application of Normalized Cross Correlation to Image Registration," *IJRET*, vol. 03, no. 5, pp. 12–16, May 2014.

[8] S. Mattoccia, F. Tombari, and L. Di Stefano, "Fast Full-Search Equivalent Template Matching by Enhanced Bounded Correlation," *IEEE Transactions on Image Processing*, vol. 17, no. 4, pp. 528–538, Apr. 2008.

[9] Z. Yang, "Fast Template Matching Based on Normalized Cross Correlation with Centroid Bounding," in *2010 International Conference on Measuring Technology and Mechatronics Automation*, Mar. 2010, vol. 2, pp. 224–227.