International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:11, No:4, 2017

Improved Pattern Matching Applied to Surface Mounting Devices Components Localization on Automated Optical Inspection

Pedro M. A. Vitoriano, Tito. G. Amaral

***Abstract*—**Automated Optical Inspection (AOI) Systems are commonly used on Printed Circuit Boards (PCB) manufacturing. The use of this technology has been proven as highly efficient for process improvements and quality achievements. The correct extraction of the component for posterior analysis is a critical step of the AOI process. Nowadays, the Pattern Matching Algorithm is commonly used, although this algorithm requires extensive calculations and is time consuming. This paper will present an improved algorithm for the component localization process, with the capability of implementation in a parallel execution system.

***Keywords*—**AOI, automated optical inspection, SMD, surface mounting devices, pattern matching, parallel execution.

I. Introduction

ELECTRONIC components continue to decrease in size in the PCB assembly production process, but the risk of defects also increases due the reduced dimensions and denser packing of boards [1]. In those conditions, failure detection has become critical factor for any SMT manufacturing process. AOI systems for PCB have proved to be a viable solution against the human-based inspection method [2].

One of the most important phases of the AOI algorithm is the component position exact extraction, this process is critical for correct analysis of the relative position of the component to the electrical pads, and also for the pads areas extraction to allow correct solder joint analysis [3].

Pattern Matching Algorithm have been commonly used, as a standard for identifying parts on the inspection images, however these algorithms are very heavy in terms of mathematical calculations, affecting the cycle time of the AOI machines.

This paper presents one optimized algorithm for pattern matching developed specifically for SMD localizations that will allow the correct localization of the SMD and their posterior analysis.

Since the SMD components are rectangular shapes, this characteristic is used for optimization of the proposal pattern matching algorithm, reducing the number of calculations interactions, and allowing a greater reduction in terms of time consumed on this process [4].

Pedro M. A. Vitoriano, ESTSetúbal/Institudo Politécnico de Setúbal, Portugal, Setúbal, Portugal (e-mail: pedro.vitoriano@gmail.com; tito).

Tito. G. Amaral, ESTSetúbal/Institudo Politécnico de Setúbal, Portugal, Ceset, Setúbal, (e-mail: Portugal.amaral@estsetubal.ips.pt).

The presented approach as seen in Fig. 1, presents the AOI algorithm system for SMD components localization.

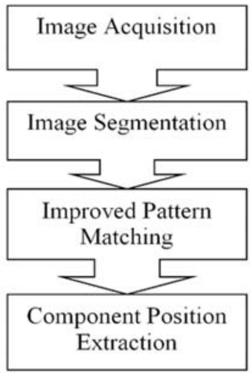

Fig. 1 Flow of the proposed approach

II. Image Acquisition

During the development, more than 400 images were used acquired under SMDs manufacturing lines, equipped with OMRON AOI systems, VT-RNS-II.

The images correspond to two different component types (Resistors and Capacitors), and different component packages were used for validation such as [1206], [0805] and [0603] (Fig. 2).

The algorithms were implemented using National Instruments Labview software, with Vision Development Module.

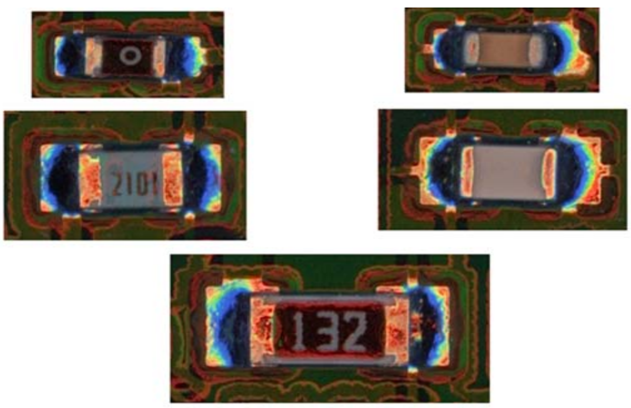

Fig. 2 Sample Image containing SMD [1206, 0805 and 0603] package

III. Image Segmentation

The images binarizations process were based on a previous study, "Automatic Optical Inspection for Surface Mounting Devices with IPC-A-610D compliance" [4].

The binary images were used with positive weight or negative weight to allow a better and improved search results (Fig. 3). In the cases that one single threshold could not segment all search objects, several binary images can be used in this process to improve the localization result [4].

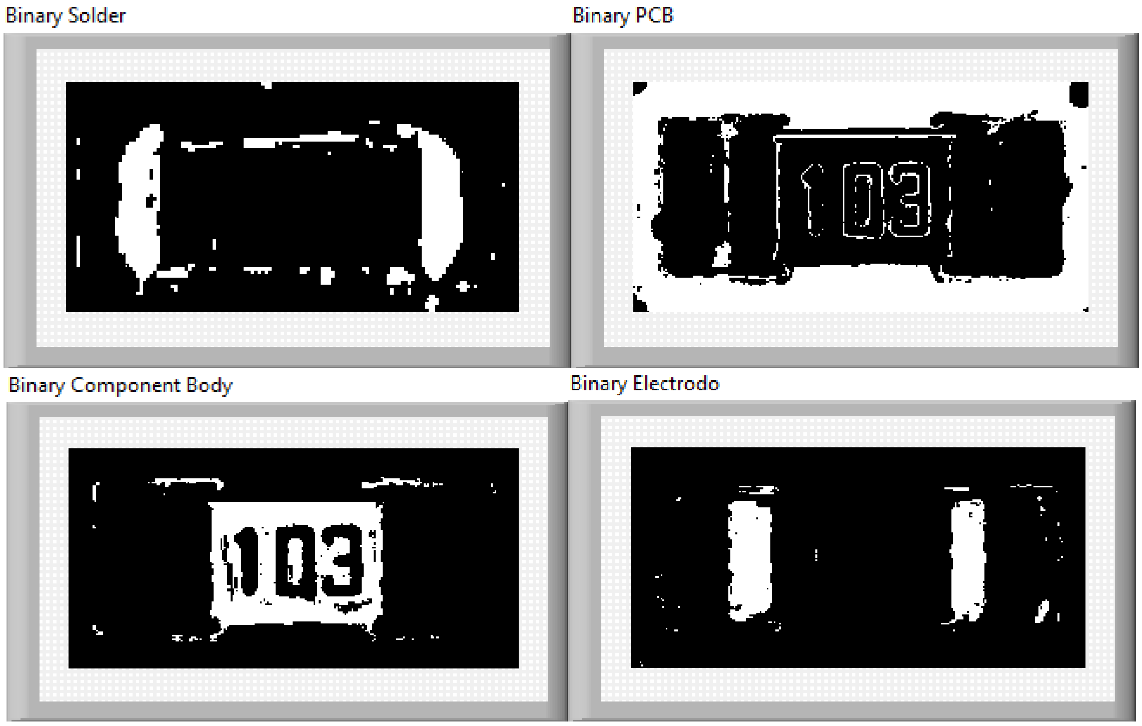

Fig. 3 Image Binarizations [4]

IV. Pattern Matching Algorithm for Component Search Process

Pattern matching algorithm was used to detect objects on images, using a pattern image as a sample for the search process. However, this algorithm shows a very slow performance.

The algorithm is based on the pattern matching (1), where the *Im* is the initial image, *Ipat* the pattern image and X', Y' the size of the pattern image [4].

$$\mathbf{Ipm}_{x,y} = \frac{\sum_{i=0}^{X'} \sum_{y=0}^{Y'} \left| \boldsymbol{Im}_{x+i,y+j} - \boldsymbol{Ipat}_{x+i,y+j} \right|}{\mathbf{X'}.\mathbf{Y'}} \quad (1)$$

For reducing the complexity of image processing due to the color variations that occur on different component lots, the images used were binary images (2), where the ***IBim*** base image is binarized, and ***IBpat***, where the pattern image is binarized.

$$\mathbf{IBpm}_{x,y} = \frac{\sum_{i=0}^{X'} \sum_{y=0}^{Y'} \left| \boldsymbol{IBim}_{x+i,y+j} - \boldsymbol{IBpat}_{x+i,y+j} \right|}{\mathbf{X'}.\mathbf{Y'}} \quad (2)$$

For the mathematical simplification, since the **IBpat**, corresponds practically to a full binarized image, it was possible to obtain the following result (3):

$$\mathbf{IBpm}_{x,y} = \frac{\sum_{i=0}^{X'} \sum_{y=0}^{Y'} \left| \boldsymbol{IBim}_{x+i,y+j} - \mathbf{1} \right|}{\mathbf{X'}.\mathbf{Y'}} \quad (3)$$

The component coordinate is obtained by the following function (4), where $[\boldsymbol{xc} \quad \boldsymbol{yc}]$ will correspond to the XY (in pixels) coordinates, where the component is the base image.

$$[\boldsymbol{xc} \quad \boldsymbol{yc}] = \mathbf{findmin}(\ \mathbf{IBpm}_{x,y}) \quad (4)$$

Looking in detail the (3), it is clear that is possible to achieve another simplification (5).

$$\mathbf{IBpm}_{x,y} = \frac{\sum_{i=0}^{X'} \sum_{y=0}^{Y'} \mathbf{IBim}_{x+i,y+j}}{\mathbf{X'}.\mathbf{Y'}} \quad (5)$$

Based on (5), the component coordinate will not be obtained by finding the minimum of the matrix, as in (4), but the maximum (6).

$$[xc \quad yc] = \mathbf{findmax}(\mathbf{IBpm}_{x,y}) \tag{6}$$

A modification was then implemented to (5). Instead of using a single binary image **IBpm**, to improve the localization process of the component, the results of several binary images IB1, IB2, IB3 and IB4, with positive and negative weight, were used for the localization of search object (7). IB corresponds to the results of the subtraction of two binary Images.

$$\mathrm{IB}_{x,y} = \sum_{i=0}^{X'} \sum_{y=0}^{Y'} \mathrm{IB1}_{x+i,y+j} + \mathrm{IB2}_{x+i,y+j} - \mathrm{IB3}_{x+i,y+j} - \mathrm{IB4}_{x+i,y+j} \tag{7}$$

To simplify the calculations, we will use a single binary image **IB** (8) that was obtained by the sum of the four original binary images (**IB1**, **IB2**, **IB3**, **IB4**), with positive weight value for desired areas and negative weight for non-desired areas [4].

$$\mathbf{IB} = \mathbf{IB1} + \mathbf{IB2} - \mathbf{IB3} - \mathbf{IB4} \tag{8}$$

Based on the previous development (8) it was obtaining the following result (9):

$$\mathbf{IBpm}_{x,y} = \frac{\sum_{i=0}^{X'} \sum_{y=0}^{Y'} \mathbf{IB}_{x+i,y+j}}{\mathbf{X'}.\mathbf{Y'}} \tag{9}$$

Fig. 4 shows the result of the $\mathbf{IBpm}_{x,y}$, where in detail the coordinate of the maximum value will correspond to the $[xc \quad yc]$, the coordinates of the component localization.

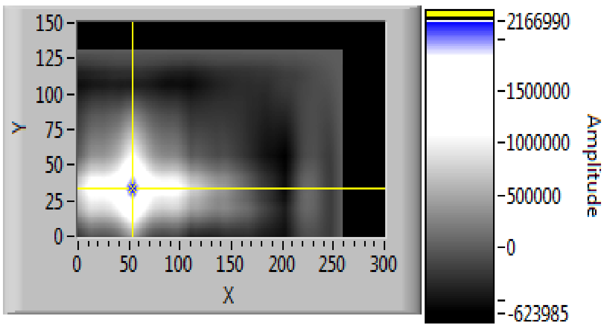

Fig. 4 Component Localization Coordinates

V. Parallelization Algorithm for Component Search Process

Based on the SMD components, the shape is rectangular, and thus we could simplify the calculations. This way instead of calculating the sum of differences of two images, the algorithm will be calculating only the sum of the image within specific X, Y size (10).

$$\mathbf{IBpm}_{x,y} = \frac{\sum_{i=x}^{X'} \sum_{j=y}^{Y'} \mathbf{IB}_{x+i,y+j}}{\mathbf{X'}.\mathbf{Y'}} \tag{10}$$

Pattern Matching is well-known as a very slow algorithm. In SMT Manufacturing production lines, it will be needed to improve the efficiency of the algorithm, reducing the time that takes to make all the calculations.

To improve the execution of these calculations, (10) was split into sub-equations that will be executed sequentially, but in final terms it will allow bringing a high improvement in terms of algorithm calculations and the time needed to achieve the end result, without affecting the overall results.

x=0
repeat
(for each y value)
$\mathbf{IP}_{x,y} = \mathbf{IB}_{x,y} + \mathbf{IB}_{x-1,y}$
x=x+1;
Until x=last (11)

After this algorithm (11) we will obtain the following result for each matrix coordinates.

$$\mathbf{IP1}_{x,y} = \sum_{i=0}^{x} \mathbf{IB}_{i,y} \tag{12}$$

Taking in consideration the X dimension of the SMD component, we will apply algorithm (13).

x=0
repeat
(for each y value)
$\mathbf{IP2}_{x,y} = \mathbf{IP1}_{x+X',y} - \mathbf{IP1}_{x,y}$
x=x+1;
Until x=last-X’ (13)

After algorithm (13) we will obtain the following result for each of the matrix coordinates.

$$\mathbf{IP2}_{x,y} = \sum_{i=x}^{x+X'} \mathbf{IB}_{i,y} \tag{14}$$

After this step we will need to make the same steps for the vertical directions.

y=0
repeat
(for each x value)
$\mathbf{IP3}_{x,y} = \mathbf{IP2}_{x,y} + \mathbf{IP2}_{x,y-1}$
y=y+1;
Until y=last (15)

After (15), the results of **IP3** will be the following:

$$\mathbf{IP3}_{x,y}(\mathbf{x},\mathbf{y}) = \sum_{i=x}^{x+X'} \sum_{j=0}^{y} \mathbf{IB}_{i,j} \tag{16}$$

The next algorithm will execute the vertical subtraction by the Y’s offset, corresponding to the dimension of the SMD component.

y=0
repeat
(for each x value)
$\mathbf{IP4_{x,y} = IP3_{x,y+Y'} - IP3_{x,y}}$ (17)
y=y+1;
Until y=last

After IP4, the results will be the following (17):

$$\mathbf{IP4_{x,y}} = \sum_{i=x}^{x+X'} \sum_{j=y}^{y+Y'} \mathbf{IB_{i,j}} \tag{18}$$

Now it is clear how this sub-calculation allows for the same initial result, but with the advantage of decreased mathematical calculations and the possibility of integration in a GPGPU (General Purpose Graphics Processor Units) or Multicore solution, which allow for an even greater increase of performance.

VI. Experimental Results

The implementation was performed using LabVIEW software from National Instruments. The selection for this software was based on the advantage of automatically executing different threads in separate cores. LabVIEW 2009, and onwards, also enabled the iteration parallelism on Loops that will be essential in this specific case. The algorithms presented in this paper, such as the columns/rows' calculations could be executed independently due to this new iteration parallelism of Loops [5], [6] (Fig. 5).

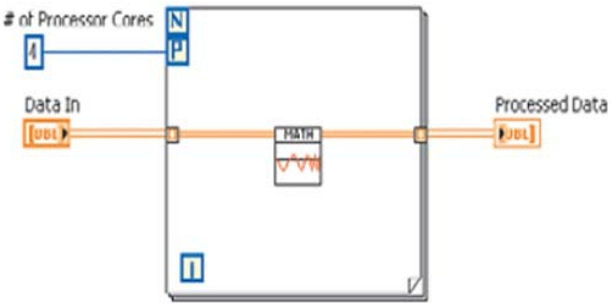

Fig. 5 Example of Parallelism on Loops on LabVIEW [5]

In this work, all the developments and simulations use pre-captured images on manufacturing PCB lines for several different components packages. Fig. 6 shows the view examples of the component extractions.

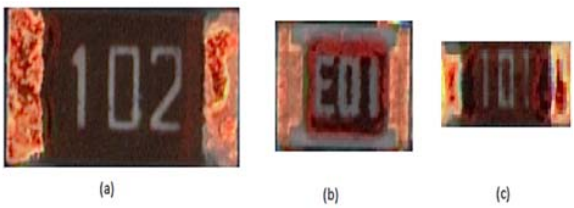

Fig. 6 Example of Component extracted, (a) resistor [1206], (b) resistor [0805], (c) resistor [0603].

The efficiency of the algorithm implemented shows a high-cycle time improvement. The number of calculations interactions required by the standard Pattern Matching Algorithm could be calculated by using (19).

$$\textbf{Pattern Matching Calculations} = \mathbf{(X.Y).(X'.Y')} \tag{19}$$

The number of calculations interactions required by the Improved Pattern Matching Algorithm could be calculated by using (20).

$$\textbf{Improved Pattern Matching Calculations} = \mathbf{4.(X.Y)} \tag{20}$$

The relations of the number of interactions calculations could be achieved by (21), where it could confirm that this new algorithm allows for better results depending on the size of the Pattern Image size.

$$\textbf{Acceleration Ratio} = \frac{\mathbf{(X.Y).(X'.Y')}}{\mathbf{4.(X.Y)}} = \frac{\mathbf{X'.Y'}}{\mathbf{4}} \tag{21}$$

In the case of an image pattern size of 50 X 50 pixels, we will have one improvement which is 625 times faster (Table I).

TABLE I
Comparing Algorithms

Image Size	Pattern Matching (Integer Calculations Operations		
	Standard	Fast	Improvement
Image: 500x250 Pattern :100x50	625M	500K	1250x
Image: 500x250 Pattern :50x50	312.5M	500K	625x

The implementation of parallel executions or by using multiple cores CPU or by using GPU, will allow improving the previous results by the number of cores available.

VII. Conclusions

The work developed allows achieving a high improvement in terms of performance. The cycle time of AOI equipment it a critical point for manufacturing lines and this improvement will permit a higher number of PCBs to be inspected. The Implementation of this algorithm shows a higher reduction in terms of mathematical operations, and could allow even greater improvements in the case of using the Multicore capability of the actual processors, or by using the parallel processing directly on the Graphical Processors.

References

[1] G. Acciani, G. Brunetti and G. Fornarelli, "A Multiple Neural Network System to Classify Solder Joints on Integrated Circuits", International Journal of computational intelligence Research (2006) Vol. 2, No 4, pp 337-348

[2] Tae-Hyeon Kim, Tai-Hoon Cho, Young-Shik Moon and Sung-Han Park, "Visual Inspection System for Classification of Solder Joints", Pattern Recognition 32 (1999), pp 565-575

[3] Kuk Won Ko and Hyung Suck Cho, "Solder Joints Inspection Using Neural Network and Fuzzy Rule-Based Classification", IEEE 98, pp. 1565-1570
[4] Pedro M. A. Vitoriano, Tito G. Amaral, Octávio Páscoa Dias, "Automatic Optical Inspection for Surface Mounting Devices with IPC-A-610D compliance." IEEE 4/11, 978-1-4244-9843-7
[5] Parallel Programming for Everyone – Take Advantage of Multicore CPUs with LabVIEW, (2016, October 20). Retrieved from http://www.ni.com/newsletter/51020/en/
[6] Multicore Programming with LabVIEW Technical Resource Guide, (2016, October 20). Retrieved from ftp://ftp.ni.com/evaluation/labview/ekit/multicore_programming_resource_guide.pdf