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 to the reduced dimensions and denser packing of boards [1]. In those conditions, failure detection has become critical 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/Instituto Politécnico de Setúbal, Portugal, Setúbal, Portugal (e-mail: pedro.vitoriano@gmail.com; tito).

Tito. G. Amaral, ESTSetúbal/Instituto 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.

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.

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

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 \( \text{Im} \) is the initial image, \( \text{Ipat} \) the pattern image and \( X', Y' \) the size of the pattern image [4].

\[
\text{lpm}_{xy} = \frac{\sum_{i=0}^{X'} \sum_{j=0}^{Y'} [(\text{Im}_{x+i,y+j}) - \text{Ipat}_{x+i,y+j}]}{X'.Y'}
\]  

(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 \( \text{IBim} \) base image is binarized, and \( \text{IBpat} \), where the pattern image is binarized.

\[
\text{IBpm}_{xy} = \frac{\sum_{i=0}^{X'} \sum_{j=0}^{Y'} [\text{IBim}_{x+i,y+j} - \text{IBpat}_{x+i,y+j}]}{X'.Y'}
\]  

(2)

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

\[
\text{IBpm}_{xy} = \frac{\sum_{i=0}^{X'} \sum_{j=0}^{Y'} [\text{IBim}_{x+i,y+j} - 1]}{X'.Y'}
\]  

(3)

The component coordinate is obtained by the following function (4), where \([x_c \ y_c]\) will correspond to the XY (in pixels) coordinates, where the component is the base image.

\[
[x_c \ y_c] = \text{findmin}(\text{IBpm}_{xy})
\]  

(4)

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

\[
\text{IBpm}_{xy} = \frac{\sum_{i=0}^{X'} \sum_{j=0}^{Y'} \text{IBim}_{x+i,y+j}}{X'.Y'}
\]  

(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).
A modification was then implemented to (5). Instead of using a single binary image \( IB \), to improve the localization process of the component, the results of several binary images \( IB_1, IB_2, IB_3 \) and \( IB_4 \), 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.

\[
IB_{xy} = \sum_{i=0}^{x'} \sum_{j=0}^{y'} IB_{x+i,y+j} + IB_{2x+i,y+j} - IB_{3x+i,y+j} - IB_{4x+i,y+j}
\]  

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 \( IB_1, IB_2, IB_3, IB_4 \), with positive weight value for desired areas and negative weight for non-desired areas [4].

\[
IB = IB_1 + IB_2 - IB_3 - IB_4
\]  

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

\[
IB_{pm,xy} = \frac{\sum_{i=0}^{x'} \sum_{j=0}^{y'} IB_{x+i,y+j}}{X',Y'}
\]  

Fig. 4 shows the result of the \( IB_{pm,xy} \), where in detail the coordinate of the maximum value will correspond to the \([xc \ yc]\), the coordinates of the component localization.

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.

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).

\[
IB_{pm,xy} = \frac{\sum_{i=0}^{x'} \sum_{j=0}^{y'} IB_{x+i,y+j}}{X',Y'}
\]  

Therefore, the next algorithm will execute the vertical subtraction by the \( y \)’s offset, corresponding to the dimension of the SMD component.

\[
IP_{xy} = IB_{xy} + IB_{x-1,y}
\]  

\[
x=0
\]

\[
\text{repeat}
\]

\[
\text{for each } y \text{ value}
\]

\[
\frac{\sum_{i=0}^{x'} \sum_{j=0}^{y'} IB_{x+i,y+j}}{X',Y'}
\]

\[
x=x+1;
\]

\[
\text{Until } x=\text{last}
\]

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

\[
IP_1_{xy} = \sum_{i=0}^{x} IB_{i,y}
\]  

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

\[
x=0
\]

\[
\text{repeat}
\]

\[
\text{for each } y \text{ value}
\]

\[
IP_2_{xy} = IP_1_{x+1,y} - IP_1_{xy}
\]

\[
x=x+1;
\]

\[
\text{Until } x=\text{last-X'}
\]

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

\[
IP_2_{xy} = \sum_{i=x}^{x} IB_{1,y}
\]  

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

\[
y=0
\]

\[
\text{repeat}
\]

\[
\text{for each } x \text{ value}
\]

\[
IP_3_{xy} = IP_2_{xy} + IP_2_{x,y-1}
\]

\[
y=y+1;
\]

\[
\text{Until } y=\text{last}
\]

After (15), the results of \( IP_3 \) will be the following:

\[
IP_3(x,y) = \sum_{i=x}^{x} \sum_{j=y}^{y} IB_{i,j}
\]  

The next algorithm will execute the vertical subtraction by the \( y \)’s offset, corresponding to the dimension of the SMD component.
After IP4, the results will be the following (17):

$$IP_{4y} = IP_{3x_{y}+y_{i}} - IP_{3x_{y}}$$

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

$$\text{Improved Pattern Matching Calculations} = \frac{4(X.Y)}{4}$$

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.

$$\text{Acceleration Ratio} = \frac{(X.Y)(X'.Y')}{(X.Y)} = \frac{X'.Y'}{4}$$

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

<table>
<thead>
<tr>
<th>Image Size</th>
<th>Pattern Matching (Integer Calculations Operations)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Standard</td>
</tr>
<tr>
<td>Image: 500x250</td>
<td>625M</td>
</tr>
<tr>
<td>Pattern :100x50</td>
<td>312.5M</td>
</tr>
<tr>
<td>Image: 500x250</td>
<td>312.5M</td>
</tr>
</tbody>
</table>

VII. CONCLUSIONS

The work developed allows achieving a high improvement in terms of performance. The cycle time of AOI equipment is 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


