Text Recognition for Information Retrieval in Images of Printed Circuit Boards

Wei Li¹, Stefan Neullens¹, Matthias Breier¹, Marcel Bosling², Thomas Pretz² and Dorit Merhof¹
¹Institute of Imaging and Computer Vision, ²Department of Processing and Recycling
RWTH Aachen University, 52056 Aachen, Germany
wei.li@lfb.rwth-aachen.de

Abstract—In order to achieve an efficient and environment-friendly recycling of printed circuit boards (PCBs), a comprehensive analysis of their material composition is essential. Besides sophisticated chemical and physical methods for a direct material analysis, an indirect method based on information retrieval provides a less costly and more efficient alternative. During the process of information retrieval, PCBs and their components need to be recognized based on their appearance and the corresponding text information. Their material composition is then available through a pre-established database. Therefore, a practical text recognition is necessary for a successful data analysis prior to PCB recycling.

Our paper is focusing on two key aspects of text recognition: binarization and final recognition of text objects using optical character recognition (OCR) engines. For binarization of text contents, a novel local thresholding method using an adaptive window size along with background estimation is presented. Several state-of-the-art algorithms and the proposed method were evaluated for comparing their binarization performance on text objects in PCB images. With respect to a data set containing manually created references, our novel method provides superior results. Furthermore, in contrast to previous work on text recognition, an additional evaluation of available open source OCR engines was conducted to assess technical limitations of OCR applications. We show that the quality of text recognition can be significantly improved if the binarization approach accounts for these technical limitations of OCR software. The presented method and results are expected to provide improved OCR performance also in other applications.

I. INTRODUCTION

While technological developments have been consistently improving the quality of our lives, there are always downsides involved, too. Benefiting from the rapid expansion of consumer electronics (CE), plenty of new electronic products (EPs) are introduced into the market annually. At the same time, due to the short life span of such products and the rapid emergency of new generations, large quantities of electronic waste are produced. Since electronic waste contains at the same time valuable materials, reusable components and toxic substances, an efficient and environment-friendly recycling is desirable. In the European Union (EU), several attempts have already been made to increase the ratio of recycled Waste Electrical and Electronic Equipment (WEEE). For example, the directive 2012/19/EU of the European Parliament and of the Council [1] was adopted for this purpose.

As the core components for realizing technical functionalities, PCBs are always found in EPs. According to e.g. [2,3,4], there is great potential for PCB recycling in consideration of the huge quantities of electronic waste and the both valuable and hazardous compositions contained therein. However, corresponding to the diverse application purposes of EPs, a high variability of PCBs is available. Furthermore, prior knowledge about the PCBs and their components is rarely available for recycling. For these reasons, a comprehensive recycling is currently still very challenging. A survey research of existing recycling technologies and systems for PCBs was conducted by Li et al. [5]. However, due to the complex structure and the high variability, an accurate analysis of material compositions can hardly be obtained. As a result, most recycling systems recover only dominant and commonly available metallic materials, such as copper, iron or gold. A lot of scarce materials used as tuning elements and non-metallic fractions remain in rest scraps which are landfilled or incinerated. A similar observation has also been made by Sohaili et al. [3].

In consideration of these facts and with the objective of a more efficient and more environment-friendly recycling, we proposed an automated processing pipeline for this purpose [6,7,8]. As illustrated in Fig. 1, after collection and preprocessing of WEEE, PCBs are ready for information retrieval. Sensors of different modalities are employed to acquire all relevant information of PCBs. Using the obtained sensor data, sophisticated PCB analysis is conducted in a subsystem consisting of many information processing units, see Fig. 2. With the help of the retrieved information, the optimal configuration of the recycling routine is generated regarding the PCBs to be processed and thereby the pre-defined economical and ecological requirements can be met. Some preliminary work for achieving desired PCB analysis was introduced in [6,7,8]. So far, the realized information retrieval is merely based on...
appearance (color, geometry, etc.) of components and PCBs. In this paper, text recognition is considered in addition and serves as another important source of information for a successful recycling.

Generally, text recognition is an intensively researched area, particularly for digitalization of documents. To extract text contents in printed or hand-written documents, they must be at first distinguished from background. To this end, numerous binarization approaches have been developed. For their simplicity and efficiency, Niblack-inspired binarization methods were introduced [9,10,11,12]. A comparison of their performance on ancient documents was presented in [13]. To obtain improved binarization results, more sophisticated approaches have also been proposed. In [14], Zhang and Yang sought for possible text contents with respect to line thickness and contour properties of characters. Patvardhan et al. [15] employed curvelet transform to deal with non-uniform illumination and noisy background. In [16], a multi-stage binarization and an analysis of connected components were applied to overcome diverse disturbances in indoor and outdoor images. Using morphological closing, a non-parametric binarization was achieved [17]. After binarization of characters, OCR software is then applied to determine the contained text information of detected objects. Besides commercial text recognition engines, there is also open source OCR software freely available which is attractive for research due to more technical transparency and low cost. However, to the best knowledge of the authors, OCR software has always been considered as ready-to-use components without any quantitative evaluation.

To achieve a feasible retrieval of text information in PCB images, we present a novel binarization method which specifically addresses the characteristics of text objects found on PCBs. To further support the overall quality of text recognition, technical limitations of commonly used OCR software were investigated and the output of binarization was adapted to meet them.

The remainder of this paper is organized as follows: the proposed binarization approach is introduced in Section II. Section III provides a detailed description of OCR software limitations. A performance evaluation of different text recognition methods is conducted in Section IV.

II. IMAGE DATA

To assess the performance of thresholding of text objects and subsequent text recognition, a data set consisting of 860 cropped image tiles with text contents is used. A PCB image with exemplarily marked (in red rectangles) image tiles and recognized (in green) text information are presented in Fig. 3 for demonstration. In Fig. 4, further image tiles in the employed database are illustrated to show the great variations of text contents and background on PCBs.

III. METHOD

PCB images are far more complex than images of documents. A broad variety of possible disturbances can be observed during processing, such as non-ideal illumination, noisy background or diverse colors of texts. To obtain a feasible text recognition under these difficult conditions, a sophisticated analysis procedure regarding characteristics of PCB images needs to be considered. To this end, a novel, practical binarization approach consisting of different processing stages is proposed.

A. Non-linear Smoothing & Contrast Improvement

For better reading, there is always a good contrast between texts and background. To represent the contrast information, all color images are first converted into gray images. Since texts can be found not only on substrates, but also on components with different distance values to the camera, a slight blurring of text contours is unavoidable. Furthermore, color fringes caused by chromatic aberrations [18] are also present in
images. In order to restore sharp edges of characters, Perona-Malik diffusion [19], is applied. While significant gradients are preserved and strengthened, regions with low contrast are smoothed.

A further improvement of the image quality can be achieved by enhancing the overall contrast if necessary. As an indicator for low contrast images, the mean absolute difference value \( \delta \) of each image is calculated as

\[
\delta = \frac{1}{M} \sum_{g_i \in G} |g_i - \bar{g}|
\]

where \( M \) is the number of pixels in image \( G \), \( g_i \) is the gray value at pixel position \( p_i \), \( \bar{g} \) is the mean gray value of \( G \). If \( \delta < T_{\text{contrast}} \), contrast improvement is performed as follows

\[
g_{i,\text{new}} = \begin{cases} \exp(g_i), & \text{if } \bar{g} > 0.5, \\ \exp(1) - \exp(1 - g_i), & \text{otherwise.} \end{cases}
\]

### B. Thresholding

To capture the contrast information in complex scenes correctly, local thresholding methods are preferred to global variants. The Niblack-inspired approaches in [9,10,11,12] are generally suitable for this purpose. In our work, the variant in [11] developed by Wolf and Jolion is employed for its practical applicability and efficiency. For an arbitrary pixel \( p_i \) in \( G \), the local threshold \( T_{\text{Wolf},i} \) is determined as

\[
T_{\text{Wolf},i} = (1 - K) \cdot m_i + K \cdot M + K \cdot \frac{r_i}{R} \cdot (m_i - M)
\]

where \( m_i \) and \( r_i \) are the mean value and the standard deviation of pixel gray values in the local neighborhood (window size: \( W \times W \)) of pixel \( p_i \), respectively. \( M \) is the minimal value of the gray image and \( R \) is the maximal value of \( r \) with \( M \leq g_i \) and \( R \geq r_i \) and \( K \) is a constant. Performance of local thresholding methods is strongly dependent on the selection of the window width \( W \). Optimally, \( W \) should be adaptively selected to capture the local contrast information correctly. This is particularly important for text recognition on PCBs since texts with diverse font sizes are available. To this end, we propose an adaptive local thresholding method. Depending on the detected stroke width \( SW \), it selects the window width \( W \) automatically as \( W = 2 \cdot SW + 1 \).

All image tiles with text objects have significant contrast between foreground and background. Therefore, roughly bimodal and quasi-bimodal gray value histograms can be obtained. Otsu’s method [20] is able to give a good initialization for estimating \( SW \). Let \( C_T \) and \( C_B \) denote boundaries of detected text and background areas, respectively. The distance between a pixel \( p_i \) and a boundary \( C \) is defined as \( d(p_i, C) \). The key points \( \{p^k_i\} \) of characters are those pixels of text objects with

\[
\forall \quad p^k_i, \quad p_i \in \mathcal{N}(p^k_i), \quad d(p^k_i, C_B) \geq d(p_j, C_B)
\]

where \( \mathcal{N}(p^k_i) \) is the neighborhood of the pixel \( p^k_i \). In other words, the key points are those text pixels with the maximum distance values to background in their neighborhood \( \mathcal{N} \). Normally, \( \mathcal{N} \) can be taken as the 8-connected neighborhood or

\[
d(p_i, C_B) = \min_{p_j \in \mathcal{N}(p_i)} |x_i - x_j| + |y_i - y_j|
\]

where \( x \) and \( y \) are the 2D pixel coordinates. For demonstration, a capital letter “A” is illustrated in Fig. 5(a). All text pixels are marked in gray (light and dark) squares. Numbers denote the “chessboard” distance values of text pixels to background. Found key points are marked in dark gray. In Fig. 5(b) for example, the occurrence of distance values of all key points is translated into a histogram. The peak arising at the distance value \( d_{SW} = 2 \) reflects the fact that the most key points are two pixels away from background. Since key points are mostly to be found in the middle of line structures, \( SW \) of the letter “A” is \( (2 \cdot d_{SW} - 1) = 3 \) pixels.

### C. Background Estimation

So far, an unsolved problem of our approach is the assumption of dark text contents on bright background which is not necessarily the case for PCBs. To overcome this limitation, a novel, simple but effective feature is employed for distinguishing text objects from background. In our observations, text objects are always consisting of thin line structures. On the other hand, the background areas are frequently huge and thick. Let \( \Omega_T \) and \( \Omega_B \) denote detected text and background areas, respectively. The observations can be mathematically described as

\[
\sum_{p_i \in \Omega_T} d(p_i, C_B) < \sum_{p_i \in \Omega_B} d(p_i, C_T).
\]
components-based analysis is performed and all candidates comprising less than \( l \) pixels are removed where \( l \) is dependent on the stroke width \( SW \) with \( l = (SW)^2/2 \). It is also not seldom to see some characters linked with others through unexpected artifacts, e.g. H-bridges, spur pixels. With morphological operations using appropriate structure elements, these characters can be separated for further analysis.

The last step of post-processing is to smooth the character contours. All edge pixels of detected objects are inspected. Let \(#\{N(p_i) \in \Omega_B\}\) denote the number of pixels belonging to background in the 8-connected neighborhood of an edge pixel \( p_i \), and \(#\{N(p_i) \in \Omega_T\}\) the number of pixels belonging to text, respectively. \( p_i \) is then removed if \(#\{N(p_i) \in \Omega_B\}\) > \(#\{N(p_i) \in \Omega_T\}\).

F. Improvement through Iterations

To improve binarization performance, some of the aforementioned operations are carried out iteratively to obtain better estimations of \( SW \), as well as color values of texts and background. For the readers’ convenience, a block diagram of the iterative processing is shown in Fig. 7. It is worth to mention that all intermediate images should be normalized to have the value range of \([0, 1]\).

G. Evaluation Method

For quantitative evaluation of binarization results, \( F \)-score (\( F_{pixel} \)) combining precision (\( P_{pixel} \)) and recall (\( R_{pixel} \)) is employed. These measures are defined as follows

\[
P_{pixel} = \frac{\#(TP_{pixel})}{\#(TP_{pixel}) + \#(FP_{pixel})},
\]

\[
R_{pixel} = \frac{\#(TP_{pixel})}{\#(TP_{pixel}) + \#(FN_{pixel})},
\]

\[
F_{pixel} = 2 \cdot \frac{P_{pixel} \cdot R_{pixel}}{P_{pixel} + R_{pixel}}
\]

where \(#(TP_{pixel})\), \(#(FP_{pixel})\) and \(#(FN_{pixel})\) denote the number of true text pixels in found text objects after binarization, the number of true background pixels in found

![Fig. 7. Flow diagram of proposed binarization approach.](image-url)
TABLE I
GENERAL PERFORMANCE OF OCR ENGINES ON DOCUMENT IMAGES.

<table>
<thead>
<tr>
<th>OCR engines</th>
<th>precision ($P_{ocr}$)</th>
<th>recall ($R_{ocr}$)</th>
<th>f-score ($F_{ocr}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tesseract</td>
<td>0.934</td>
<td>0.944</td>
<td>0.939</td>
</tr>
<tr>
<td>Cuneiform</td>
<td>0.941</td>
<td>0.944</td>
<td>0.943</td>
</tr>
<tr>
<td>GOCR</td>
<td>0.711</td>
<td>0.831</td>
<td>0.766</td>
</tr>
<tr>
<td>OCRAD</td>
<td>0.851</td>
<td>0.879</td>
<td>0.865</td>
</tr>
</tbody>
</table>

text objects and the number of true text pixels in found background regions, respectively. For multiple images, the overall precision, recall and f-score ($P_{pixel}$, $R_{pixel}$ and $F_{pixel}$) are computed as the mean values of corresponding measures from single image tiles.

IV. OCR ENGINES AND APPLICATION

Currently, there are various software tools developed for the purpose of OCR. But the open source variants are more attractive regarding their low cost (even zero cost) and more technical transparency. In our work, we considered four mostly applied freely available OCR engines: Tesseract-ocr [21], Cuneiform-linux [22], GOCR [23] and OCRAD [24].

To the best of our knowledge, most publications that address text recognition simply take the employed OCR software as off-the-shelf component. It is often assumed that OCR software generally provides satisfactory results which is, however, not always the case due to known or unknown limitations.

To obtain a feasible information retrieval, the necessary requirements for a successful text recognition using different OCR engines are investigated in this work.

A. General Performance

The considered OCR engines are first evaluated for general cases. A set of document images are generated to this end. In documents, texts appear in three commonly used fonts for PCB labeling: “Arial”, “OCR-A” and “OCR-B” with diverse font sizes and shapes (normal, bold, italic). All text contents are ideally represented without any geometric transform (rotation, shearing, etc.).

B. Varying Size of Characters

In digital images, each character consists of a couple of pixels. More details about the character structure can be observed when the character size increases. At the same time, better OCR results should be obtained. To conduct a comprehensive analysis of OCR performance versus the variation of character size, synthetic images of characters (a - z, A - Z, 0 - 9) are generated with different font sizes (size 6 to size 60 of the font “Arial”).

C. Varying Text Orientation

Another essential factor with great impact on OCR performance is the orientation of characters. In practice, a perfect orientation estimation can be hardly achieved in any text recognition. Therefore, OCR engines are expected to tolerate certain variations of the text orientation. Moreover, it is also important to determine the tolerance range. For this purpose, we rotated the document images used for the general evaluation from 0 to 10 degrees using a step size of 1 degree.

D. Evaluation Method and Results for OCR Engines

For quantitative evaluation of OCR performance, f-score ($F_{ocr}$) similar to $F_{pixel}$ in Eq. 9 is employed. For each image tile, precision ($P_{ocr}$) and recall ($R_{ocr}$) are computed with respect to the reference string $S_{Ref}$ and the resulting (recognized) string $S_{Res}$. The definitions of $P_{ocr}$ and $R_{ocr}$ are as follows:

$$P_{ocr} = \max(L_{Ref}, L_{Res}) - d_{lev}(S_{Ref}, S_{Res})$$

$$R_{ocr} = \max(L_{Ref}, L_{Res}) - d_{lev}(S_{Ref}, S_{Res})$$

where $L_{Ref}$ and $L_{Res}$ denote the length of the reference string and the length of the resulting string, respectively, $d_{lev}$ is the Levenshtein distance introduced in [25]. For multiple images, the overall precision, recall and f-score ($\bar{P}_{ocr}$, $\bar{R}_{ocr}$

![Fig. 8. Font size (horizontal axis, from 6 to 60) vs. number of recognized characters (vertical axis, from 0 to 36) for OCR engines. 36 combinations of characters (26 combinations with aA to zZ, 10 combinations with 00 to 99) are used for the evaluation so that some ambiguous recognitions, such as c and C, o and O, are avoided.](image)

![Fig. 9. Performance of OCR engines in document images with respect to the text orientation varying from 0 to 10 degrees.](image)
and $\bar{F}_{ocr}$) are computed as the mean values of corresponding measures from single image tiles.

The summarized generalized performance of OCR engines is listed in Table I. In comparison to GOCR and OCRAD, Tesseract and Cuneiform are able to provide superior recognition results. An overview of performance versus font size for the four considered OCR engines is presented in Fig. 8. As read in the graphics, stable performance can be obtained if the font size is not less than 24 which corresponds to a stroke width between 3 and 4 pixels. In Fig. 9, performance of OCR engines is plotted with respect to the varying text orientation. Obviously, there is a decline in performance\(^1\) while text objects are rotated away from the ideal orientation (0 degree).

E. Proposed Application

As indicated through evaluation of general cases, Tesseract and Cuneiform provide more solid performance in contrast to GOCR and OCRAD. Additionally, in consideration of the better technical transparency and the greater community, Tesseract is selected for further text recognition.

According to the conducted analysis with respect to varying text size and orientation, modifications of binarization results have to be made to ensure practical recognition performance. First of all, after determining the stroke width in original images, these images are resized by applying the bilinear interpolation to reach the desired stroke width of characters. The bilinear interpolation is preferred to other interpolation methods employing polynomials of higher degree for avoiding possible overshoot. New binarization results are generated for the resized images and used for final text recognition. Furthermore, based on blob analysis and line fitting, similar approaches as those suggested in [26] are employed for line finding and base line estimation. By means of such processing, vertical and horizontal text lines in the image can be separated and further processed individually. All estimated text lines are realigned to better match an ideal orientation to avoid a decline in OCR performance.

V. RESULTS & DISCUSSION

A. Binarization

In Table II, performance of several state-of-the-art binarization methods (Niblack [9], Sauvola [10], Wolf [11], Feng [12], Patvardhan [15], Gatos [16], Zhang [14], Otsu [20])

\[^1\]some OCR engines have the integrated function for orientation estimation which is designed for document images. However, in our tests, it was unable to give any reasonable results for text objects in PCB images. Thus, this function was permanently disabled in our evaluation.

and the proposed method (“−” and “+” for without and with post-processing, respectively) is listed. Using the brute-force search, appropriate parameters are determined for all concerned state-of-the-art approaches. Obviously, the proposed method is superior to all others. Moreover, only 6 out of 860 image tiles in the data set are with erroneously estimated background. Gatos’ method slightly outperforms the rest (except the proposed) since it uses a sophisticated background estimation.

To make the state-of-the-art approaches stable against variations of background color, the proposed background estimation is also applied to their final binarization results and “*” is used to indicate this additional operation in Table III. In comparison to Table II, a significant improvement is obtained for most approaches and the modified Otsu’s method provides the best performance. On the other hand, no significant change is observed for Gatos’ method.

B. Text Recognition

For evaluating the text recognition, the $f$-score ($F_{ocr}$) defined in Eq. 10 is used. To illustrate the effectiveness of binarization methods, the text recognition results from native Tesseract OCR engine without any front-end processing is also presented and used as the reference for comparison. All text lines that were found in binarized images are realigned to obtain the expected orientation (0 degree to horizontal lines) before sent to the OCR engine. The cancellation of small disturbances is also carried out on all binarization results.

Referring to Fig. 8, a stroke width of more than 3 pixels should be used for stable OCR performance. In PCB images, even higher resolution is required for preserving details of character structures in case of difficult imaging conditions (chromatic aberrations, shadows, specularity, etc.). Therefore, if insufficient stroke width is detected, original images are resized to reach a minimal stroke width of 5 pixels.

Recognition results from original and modified methods are presented in Table IV and Table V, respectively. Again, our proposed method outperforms all others. Improvement of performance is obtained for most state-of-the-art methods after applying the background estimation.

C. Parameter Tuning

It should be mentioned that the proposed method is rather insensitive to parameter variations. Two major parameters need to be set for achieving reasonable performance: $T_{contrast}$ for contrast improvement and $K$ for local thresholding. $T_{contrast}$ was varied between 0.1 and 0.5, and $K$ between 0.05 and 0.5.
A significant decline in quality of binarization and text recognition results was hardly observed.

VI. CONCLUSION

To achieve a comprehensive analysis of PCBs for a more efficient and environment-friendly recycling, text recognition in PCB images for retrieval of relevant information is presented in this paper. Two key aspects of a recognition procedure are mainly considered: 1. binarization of text contents; 2. recognition of detected text objects using OCR engines. As improvement of state-of-the-art local thresholding approaches, an algorithm for automatic selection of window size for local analysis is developed. Using a histogram analysis of distance values of determined character key points, window size is set to adaptively match the font size of text objects. Furthermore, to deal with varying background color in PCB images, a novel, non-parametric and effective criterion is proposed for background estimation in binary images. After applying the proposed background estimation, significant improvement is observed for most binarization methods. Benefiting from adaptive window size, our method is able to provide the best performance among local thresholding approaches. To avoid possible mismatch between binarization and OCR engine-based text recognition for better performance, technical limitations of commonly used open source OCR engines are investigated. As indicated through quantitative evaluation, font size and text orientation are the two factors with great impact on performance. Finally, binarization results are modified correspondingly to meet the determined requirements and superior results are obtained. The proposed methods and the presented evaluation results should also lead to improved performance in general cases of text recognition.

REFERENCES