Text Recognition for Information Retrieval in Images of Printed Circuit Boards

Wei Li[1], Stefan Neullens[1], Matthias Breier[1], Marcel Bosling[2], Thomas Pretz[2] and Dorit Merhof[1]
[1]Institute of Imaging and Computer Vision, [2]Department 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 asses 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].

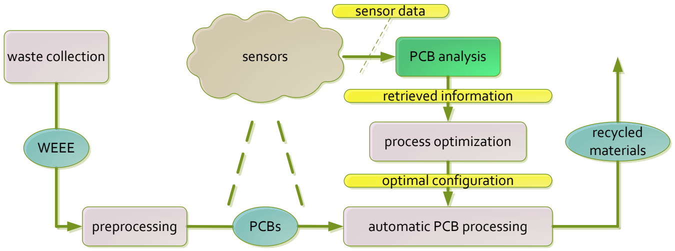


Fig. 1. Automated PCB recycling system.

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

978-1-4799-4032-5/14/$31.00 ©2014 IEEE

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.

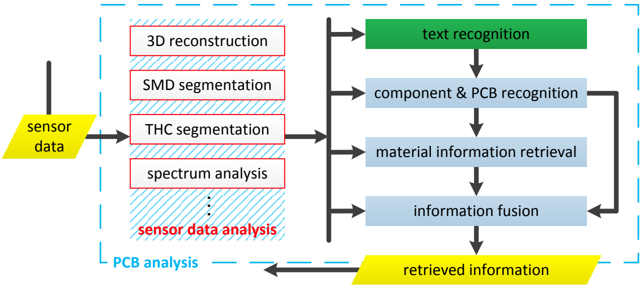


Fig. 2. PCB analysis for information retrieval.

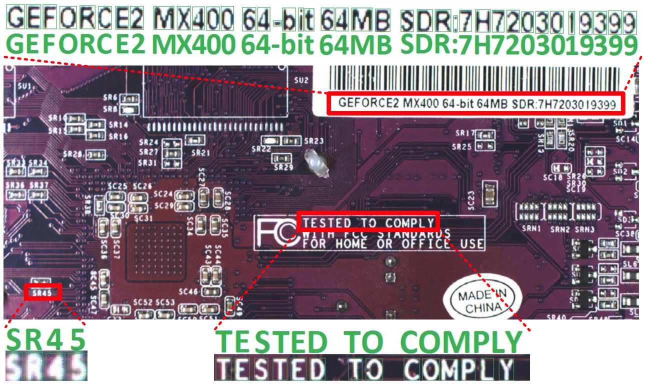

GEFORCE2 MX400 64-bit 64MB SDR:7H7203019399
GEFORCE2 MX400 64-bit 64MB SDR:7H7203019399
SR45
SR45
TESTED TO COMPLY
TESTED TO COMPLY

Fig. 3. Demo PCB image with exemplarily marked (in red rectangles) and recognized (in green) texts.

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

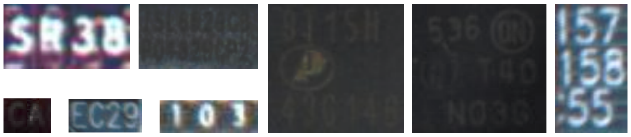


Fig. 4. Image tiles illustrating the great variations of text contents and background on PCBs.

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 $\bar{\delta}$ of each image is calculated as

$$\bar{\delta} = \frac{1}{M} \cdot \sum_{g_i \in \boldsymbol{G}} |g_i - \bar{g}| \tag{1}$$

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

$$g_{i,new} = \begin{cases} \exp(g_i), & \text{if } \bar{g} > 0.5, \\ \exp(1) - \exp(1 - g_i), & \text{otherwise.} \end{cases} \tag{2}$$

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 $\boldsymbol{p}_i$ in $\boldsymbol{G}$, the local threshold $T_{Wolf,i}$ is determined as

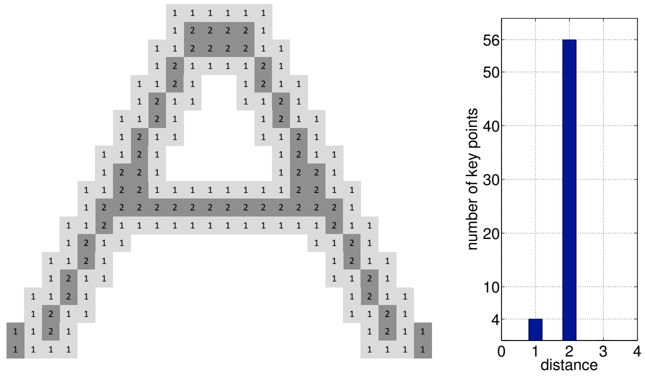
$$T_{Wolf,i} = (1 - K) \cdot m_i + K \cdot M + K \cdot \frac{r_i}{R} \cdot (m_i - M) \tag{3}$$

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 $\boldsymbol{p}_i$ respectively, M is the minimal value of m and R is the maximal value of r with $M \leq m_i$ 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 $\boldsymbol{C}_T$ and $\boldsymbol{C}_B$ denote boundaries of detected text and background areas, respectively. The distance between a pixel $\boldsymbol{p}_i$ and a boundary $\boldsymbol{C}$ is defined as $d(\boldsymbol{p}_i, \boldsymbol{C})$. The key points $\{\boldsymbol{p}_i^k\}$ of characters are those pixels of text objects with

$$\underset{\boldsymbol{p}_i^k}{\forall} \quad \underset{\boldsymbol{p}_j \in \mathcal{N}(\boldsymbol{p}_i^k)}{\forall} d(\boldsymbol{p}_i^k, \boldsymbol{C}_B) \geq d(\boldsymbol{p}_j, \boldsymbol{C}_B) \tag{4}$$

where $\mathcal{N}(\boldsymbol{p}_i^k)$ is the neighborhood of the pixel $\boldsymbol{p}_i^k$. 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 the neighborhood in the local 5×5 window. In principle, there are many metrics available for calculating distance values. The "chessboard" distance is selected in this work for its appropriateness and computational efficiency. Then, the distance value $d(\boldsymbol{p}_i, \boldsymbol{C})$ is defined as



(a) (b)

Fig. 5. Found key points of a capital letter "A" using the 8-connected neighborhood and the distance value histogram of key points. (a) All pixels of the letter "A" are illustrated as gray (light and dark) squares. Numbers denote the "chessboard" distance values of text pixels to background. Found key points are marked in dark gray. (b) Histogram of distance values of all key points. Obviously, the stroke width of this letter is $(2 \cdot d_{SW} - 1) = 3$ pixels where $d_{SW} = 2$ is the distance value of the peak in the histogram.

$$d(\boldsymbol{p}_i, \boldsymbol{C}) = \underset{\boldsymbol{p}_j \in \boldsymbol{C}}{\arg\min} \ max(|x_i - x_j|, |y_i - y_j|) \tag{5}$$

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 as gray (light and dark) squares. Numbers in squares denote the "chessboard" distance values of the corresponding text pixels to background. Key points are those text pixels with local maximum distance values marked in dark gray. Through a histogram analysis, determination of SW is straightforward. 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 $\boldsymbol{\Omega}_T$ and $\boldsymbol{\Omega}_B$ denote detected text and background areas, respectively. The observations can be mathematically described as

$$\sum_{\boldsymbol{p}_i \in \boldsymbol{\Omega}_T} d(\boldsymbol{p}_i, \boldsymbol{C_B}) < \sum_{\boldsymbol{p}_i \in \boldsymbol{\Omega}_B} d(\boldsymbol{p}_i, \boldsymbol{C_T}). \tag{6}$$

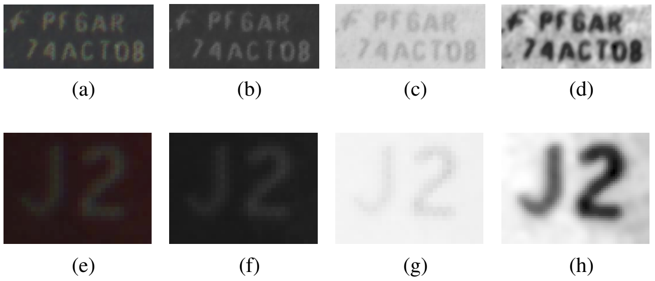


Fig. 6. Projection of color values for improved contrast. (a) and (e): original RGB images. (b) and (f): gray value images. (c) and (g): complementary gray value images. (d) and (h): images after color value projection with improved contrast.

In most cases, the image boundaries are also part of background. To include this fact into background estimation, the searched pixel $\boldsymbol{p}_j$ in $\boldsymbol{C_T}$ satisfying Eq. 5 needs to be restricted to images. In other words, the regions outside of images are assumed to be part of background.

The desired inequality in Eq. 6 is checked for each obtained binary image. If the condition is not fulfilled, a false background estimation is detected. In this case, the complementary image $\boldsymbol{G}^C$ is used for binarization, instead of the original image $\boldsymbol{G}$, with

$$\boldsymbol{G}^C = 1 - \boldsymbol{G}. \tag{7}$$

The proposed method for determining background is more intuitive than the method introduced in [16]. Moreover, since no single parameter is required, more reliable results are expected.

D. Color Value Projection

Due to erosion, aging, pollution, imperfect printing and non-ideal illumination/imaging conditions, images may possess inadequate text-to-background contrast. For this reason, a color value projection for improved contrast is applied. The pixel color values in the original RGB image can be divided into two classes according to the obtained thresholding results. Let $\bar{\boldsymbol{p}}_T^{RGB}$ and $\bar{\boldsymbol{p}}_B^{RGB}$ denote the class centroids of text pixels and background pixels in RGB color space respectively. The direction vector $\boldsymbol{q}^{RGB}$ with

$$\boldsymbol{q}^{RGB} = \frac{\bar{\boldsymbol{p}}_T^{RGB} - \bar{\boldsymbol{p}}_B^{RGB}}{||\bar{\boldsymbol{p}}_T^{RGB} - \bar{\boldsymbol{p}}_B^{RGB}||} \tag{8}$$

is determined for the 1D projection of color values and therewith the maximum inter-class distance can be achieved after projecting RGB values onto the axis defined by $\boldsymbol{q}^{RGB}$. In Fig. 6, two image tiles with low text-to-background contrast are used for illustrating the described color value projection. Quasi-black-white images similar to scanned document files are obtained after this operation.

E. Post-processing

Frequently, there are still some small non-text objects present in binarized images. Most of them have sizes much smaller than characters. To omit such objects, connected 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 $\#(\{N(\boldsymbol{p}_i) \in \boldsymbol{\Omega}_B\})$ denote the number of pixels belonging to background in the 8-connected neighborhood of an edge pixel $\boldsymbol{p}_i$, and $\#(\{N(\boldsymbol{p}_i) \in \boldsymbol{\Omega}_T\})$ the number of pixels belonging to text, respectively. $\boldsymbol{p}_i$ is then removed if $\#(\{N(\boldsymbol{p}_i) \in \boldsymbol{\Omega}_B\}) > \#(\{N(\boldsymbol{p}_i) \in \boldsymbol{\Omega}_T\})$.

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

$$\begin{aligned} P_{pixel} &= \frac{\#(TP_{pixel})}{\#(TP_{pixel}) + \#(FP_{pixel})}, \\ R_{pixel} &= \frac{\#(TP_{pixel})}{\#(TP_{pixel}) + \#(FN_{pixel})}, \\ F_{pixel} &= 2 \cdot \frac{P_{pixel} \cdot R_{pixel}}{P_{pixel} + R_{pixel}} \end{aligned} \tag{9}$$

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

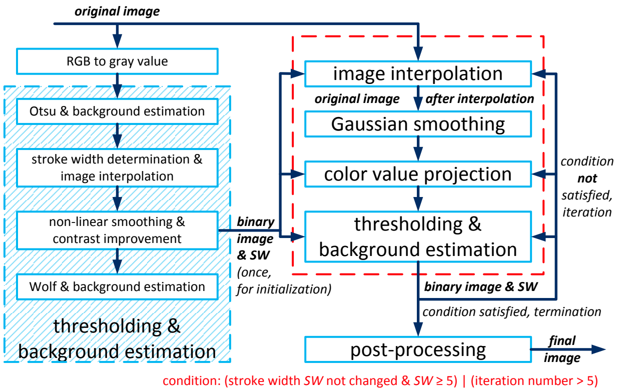

original image
RGB to gray value
Otsu & background estimation
stroke width determination & image interpolation
non-linear smoothing & contrast improvement
Wolf & background estimation
thresholding & background estimation
binary image & SW (once, for initialization)
image interpolation
original image
after interpolation
Gaussian smoothing
color value projection
thresholding & background estimation
condition not satisfied, iteration
binary image & SW
condition satisfied, termination
post-processing
final image
condition: (stroke width SW not changed & SW ≥ 5) | (iteration number > 5)

Fig. 7. Flow diagram of proposed binarization approach.

TABLE I
GENERAL PERFORMANCE OF OCR ENGINES ON DOCUMENT IMAGES.

OCR engines	precision (P_{ocr})	recall (R_{ocr})	f-score (F_{ocr})
Tesseract	0.934	0.944	0.939
Cuneiform	0.941	0.944	0.943
GOCR	0.711	0.831	0.766
OCRAD	0.851	0.879	0.865

text objects and the number of true text pixels in found background regions, respectively. For multiple images, the overall *precision*, *recall* and *f-score* ($\bar{P}_{pixel}$, $\bar{R}_{pixel}$ and $\bar{F}_{pixel}$) 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

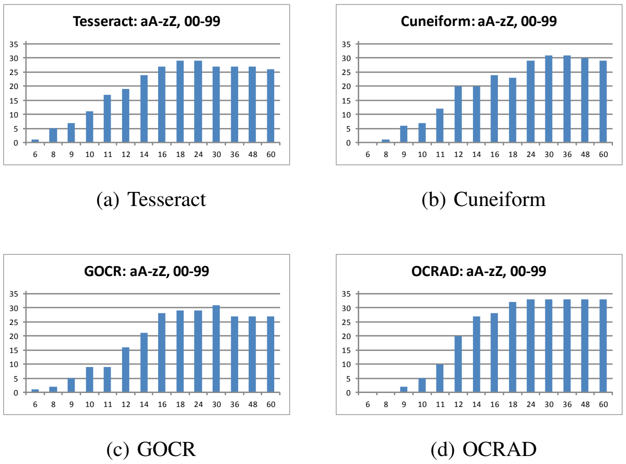
The considered OCR engines are first evaluated for general cases. A set of document images are generated to this end. In documents, texts appear in three commonly used fonts for PCB labeling: “Arial”, “OCR-A” and “OCR-B” with diverse font sizes and shapes (normal, **bold**, *italic*). All text contents 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.



(a) Tesseract (b) Cuneiform

(c) GOCR (d) OCRAD

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.

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

$$\begin{aligned} P_{ocr} &= \frac{max(L_{Ref}, L_{Res}) - d_{Lev}(S_{Ref}, S_{Res})}{L_{Res}}, \\ R_{ocr} &= \frac{max(L_{Ref}, L_{Res}) - d_{Lev}(S_{Ref}, S_{Res})}{L_{Ref}} \end{aligned} \tag{10}$$

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* ($\bar{P}_{ocr}$, $\bar{R}_{ocr}$

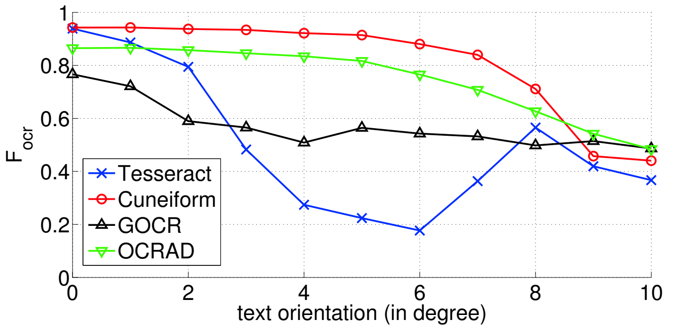


Fig. 9. Performance of OCR engines in document images with respect to the text orientation varying from 0 to 10 degrees.

TABLE II
BINARIZATION WITHOUT BACKGROUND ESTIMATION.

method	$\bar{F}_{pixel}$	method	$\bar{F}_{pixel}$	method	$\bar{F}_{pixel}$
Niblack	0.456	Wolf	0.443	Patvardhan	0.465
Sauvola	0.445	Gatos	0.509	Otsu	0.470
Feng	0.430	Zhang	0.459	proposed$^-$	**0.831**
				proposed$^+$	0.825

TABLE III
BINARIZATION WITH BACKGROUND ESTIMATION.

method	$\bar{F}_{pixel}$	method	$\bar{F}_{pixel}$	method	$\bar{F}_{pixel}$
Niblack*	0.708	Wolf*	0.804	Patvardhan*	0.853
Sauvola*	0.751	Gatos*	0.504	Otsu*	**0.863**
Feng*	0.774	Zhang*	0.654	proposed$^-$	0.831
				proposed$^+$	0.825

and $\bar{F}_{ocr}$) are computed as the mean values of corresponding measures from single image tiles.

The summarized general 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]) 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 (expect 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.

[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.

TABLE IV
TEXT RECOGNITION WITHOUT BACKGROUND ESTIMATION.

method	$\bar{F}_{ocr}$	method	$\bar{F}_{ocr}$	method	$\bar{F}_{ocr}$
Niblack	0.225	Wolf	0.248	Patvardhan	0.295
Sauvola	0.281	Gatos	0.119	proposed$^-$	0.552
Feng	0.229	Zhang	0.271	proposed$^+$	**0.589**
		Otsu	0.322	Tesseract native	0.278

TABLE V
TEXT RECOGNITION WITH BACKGROUND ESTIMATION.

method	$\bar{F}_{ocr}$	method	$\bar{F}_{ocr}$	method	$\bar{F}_{ocr}$
Niblack*	0.285	Wolf*	0.325	Patvardhan*	0.363
Sauvola*	0.394	Gatos*	0.119	proposed$^-$	0.552
Feng*	0.328	Zhang*	0.334	proposed$^+$	**0.589**
		Otsu*	0.386	Tesseract native	0.278

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

[1] "Weee directive 2012/19/eu," *Official Journal of the European Parliament*, July 2012.

[2] I. Dalrymple, N. Wright, R. Kellner, N. Bains, K. Geraghty, M. Goosey, and L. Lightfoot, "An integrated approach to electronic waste (weee) recycling," *Circuit World*, vol. 33 Issue 2, pp. 52–58, 2007.

[3] J. Sohaili, S. K. Muniyandi, and S. S. Mohamad, "A review on printed circuit boards waste recycling technologies and reuse of recovered nonmetallic materials," *Int. Journal of Scientific & Engineering Research*, vol. 3, no. 2, pp. 1–7, 2012.

[4] A. C. Marques, J.-M. C. Marrero, and C. de Fraga Malfatti, "A review of the recycling of non-metallic fractions of printed circuit boards," *SpringerPlus Engineering*, vol. 2, pp. 521–531, 2013.

[5] J. Li, P. Shrivastava, Z. G, and H. Zhang, "Printed circuit board recycling: A state-of-the-art survey," *IEEE Trans. on Electronics Packaging Manufacturing*, vol. 27, Issue: 1, pp. 33–42, 2004.

[6] W. Li, B. Esders, and M. Breier, "Smd segmentation for automated pcb recycling," in *Proc. of the 11th IEEE Int. Conference on Industrial Informatics*, Bochum, Germany, July 2013, pp. 65–70.

[7] D. Herchenbach, W. Li, and M. Breier, "Segmentation and classification of thcs on pcbas," in *Proc. of the 11th IEEE Int. Conference on Industrial Informatics*, Bochum, Germany, July 2013, pp. 59–64.

[8] T. Koch, M. Breier, and W. Li, "Heightmap generation for printed circuit boards (pcb) using laser triangulation for pre-processing optimization in industrial recycling applications," in *Proc. of the 11th IEEE Int. Conference on Industrial Informatics*, 2013, pp. 48–53.

[9] W. Niblack, *An introduction to digital image processing*. Englewood Cliffs, Prentice Hall, N.J, 1986.

[10] J. Sauvola and M. Pietikäinen, "Adaptive document image binarization," *PATTERN RECOGNITION*, vol. 33, pp. 225–236, 2000.

[11] C. Wolf and J.-M. Jolion, "Extraction and recognition of artificial text in multimedia documents," *Pattern Analysis & Applications*, vol. 6 Issue 4, pp. 309–326, February 2003.

[12] M.-L. Feng and Y. peng Tan, "Contrast adaptive binarization of low quality document images," *Ieice Electronic Express*, vol. 1, no. 16, pp. 501–506, 2004.

[13] K. Khurshid, I. Siddiqi, C. Faure, and N. Vincent, "Comparison of niblack inspired binarization methods for ancient documents," in *Document Recognition and Retrieval XVI*, ser. SPIE Proceedings, K. Berkner and L. Likforman-Sulem, Eds., vol. 7247. SPIE, 2009, pp. 1–10.

[14] C. Zhang and J. Yang, "Binarization of document images with complex background," in *Proc. of the 6th Int. Conference on Wireless Communications Networking and Mobile Computing*, Sep 2010, pp. 1–4.

[15] C. Patvardhan, A. Verma, and C. Lakshmi, "Document image denoising and binarization using curvelet transform for ocr applications," in *Proc. of Nirma University Int. Conference on Engineering*, Dec 2012, pp. 1–6.

[16] B. Gatos, I. Pratikakis, and S. J. Perantonis, "Text detection in indoor/outdoor scene images," in *1st Int. Workshop on Camera-based Document Analysis and Recognition*, Seoul, Korea, 2005, pp. 127–132.

[17] S. Tabatabaei and M. Bohlool, "A novel method for binarization of badly illuminated document images," in *Proc. of the 17th IEEE Int. Conference on Image Processing*, Sep 2010, pp. 3573–3576.

[18] W. Li and J. Klein, "Multichannel camera calibration," in *Proceedings SPIE 8660*, vol. 8660. SPIE, February 2013, pp. 8660–02.

[19] P. Perona and J. Malik, "Scale space and edge detection using anisotropic diffusion," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.

[20] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[21] "Tesseract-ocr," accessed: 03-09-2013; version: 3.02. [Online]. Available: http://code.google.com/p/tesseract-ocr

[22] "Cuneiform-linux," accessed: 03-09-2013; version: 1.1.0. [Online]. Available: https://launchpad.net/cuneiform-linux

[23] "GOCR," accessed: 03-09-2013; version: 0.50. [Online]. Available: http://jocr.sourceforge.net

[24] "OCRAD," accessed: 03-09-2013; version: 0.22. [Online]. Available: http://www.gnu.org/software/ocrad

[25] V. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[26] R. Smith, "An overview of the tesseract ocr engine," in *Proc. of the 9th Int. Conference on Document Analysis and Recognition*, 2007, pp. 629–633.