

Image-Based Competitive Printed Circuit Board Analysis

Simon Basilico

Department of Electrical Engineering
Stanford University
Stanford, CA
basilico@stanford.edu

Ford Rylander

Department of Electrical Engineering
Stanford University
Stanford, CA
fordr@stanford.edu

Abstract— This paper describes an image processing algorithm to detect and identify integrated circuits (ICs) on a printed circuit board (PCB). The identification algorithm leverages a maximum a priori probability (MAP) detector, morphological filtering, vertex detection, and SIFT feature extraction to segment ICs in the image, correct for rotations, and recognize SIFT descriptors to identify ICs. The processing algorithm is implemented and evaluated in MATLAB.

Keywords—image processing; printed circuit board; integrated circuit detector; MAP; SIFT descriptors

I. INTRODUCTION

Competitive analysis is a crucial task in the hardware electronics industry that drives both product pricing and functionality. Specifically, a list of the components on a printed circuit board (PCB), or a Bill of Materials (BoM), can be used to directly obtain a reliable cost estimate of the product. Compiling a BoM by means of manual analysis can be tedious, time consuming, and prone to human error. Instead, modern image processing techniques can be used to expediently produce a BoM.

This paper discusses an implementation of an image processing algorithm to detect and identify integrated circuits (ICs) on a PCB. First, an examination of relevant prior art is given. Then, a new architecture to detect and identify ICs is proposed, and each stage of the processing pipeline is described in detail. A discussion of algorithm selection and associated tradeoffs in each stage of the pipeline is also given. Finally, the described architecture's performance is evaluated across a range of test images.

II. PRIOR ART

Image processing techniques have been applied in the past to read labels off of IC packages on an assembly line to check for errors [1] and detect IC orientation within the package [2]. These applications operated on individual IC packages on an assembly line, not fully populated PCBs.

Prior art also exists on the detection and identification of ICs on populated circuit boards [3]. The authors implemented an Android-based mobile application to recognize IC part numbers. Several modules of their chosen pipeline architecture proved to be useful insights that motivated design decisions in the

implementation described herein, such as the use of a maximum a posteriori (MAP) detector as an image segmentation step. Using the author's published code to reproduce their results, we found that detection performance dropped when running the algorithm over a richer test set - the implementation herein seeks to improve upon the robustness of this prior art.

We seek to advance the science of detecting and identifying ICs on populated PCBs, particularly in regards to recognition accuracy and robustness to different PCBs and lighting.

III. ALGORITHM DESCRIPTION

The following section examines the steps taken by the proposed algorithm to identify ICs on a populated PCB using the image of a PCB shown in Figure 1 as an illustrative example.

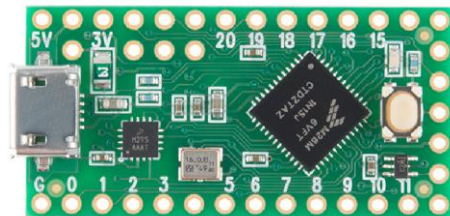


Figure 1. Original image to be processed.

A. IC Segmentation

The first objective in the processing pipeline is to segment regions in the image that contain pixels corresponding to ICs. There are several distinctive features that ICs have in common - clean, straight edges, and a black/brown coloration. These characteristics immediately lend themselves to several approaches that may be effective at segmenting IC regions.

The first approaches examined were edge detection techniques to identify the long, well defined edges of the ICs. Several difficulties immediately presented themselves with applying edge-based segmentation in a PCB environment. On a PCB, there are many well defined edge structures that are similar to those on an IC - the edge detector outputs strong responses from traces, board edges, and other non-IC devices mounted on

the board. We attempted to remedy this problem by computing the eccentricity of the closed edge regions, and rejecting regions whose eccentricity is above a chosen threshold (in hopes of rejecting long regions corresponding to traces). However, this approach was fundamentally limited by the inability to discern between short traces and legitimate IC edges, resulting in many false positives.

Due to the challenges associated with applying edge-detection to images of PCBs, we chose to use a MAP detector to segment ICs. The MAP detector exhibited considerably better performance on the PCB test images - since ICs on a populated PCB have a very distinctive black/brown color compared to the background, the detector excelled at identifying IC regions. A key factor limiting the performance of the MAP detector across a rich test set was the size and diversity of the training set across lighting variation, PCB color, and image quality. We overcame this difficulty by generating a small but diverse training set to that generalized well across a variety of test images. The MAP output of the original test image is shown in Figure 2.

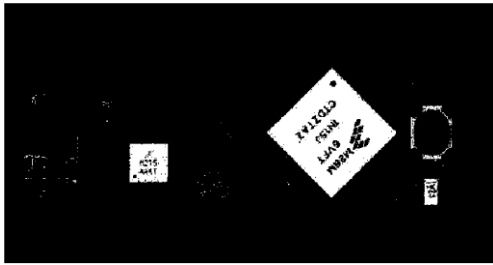


Figure 2. Output of the MAP detector.

B. Segmentation – Post Processing

While the MAP detector proved effective at roughly highlighting regions in the image containing ICs, the MAP detector output still contained considerable noise due to false positives (produced by other dark regions in the image) and holes in legitimate IC regions (due to the white letters in the interior of the IC inducing negative results). Additionally, in many test images we observed that individual legitimate IC regions became disconnected in the MAP output. Without additional processing, this would cause the later stages of the processing pipeline to detect the presence of multiple regions, when in fact there is only one IC. We will now outline the steps taken to remediate these concerns.

First, a dilation with a 2x2 square structuring element is performed to connect any erroneously disconnected IC regions in the output of the MAP detector. The size of the structuring element is critical in this step - a structuring element that is too large will result in the erroneous connecting of regions that are distinct IC regions, but an element that is too small will fail to connect regions that are legitimately contiguous.

The second processing post-processing step is a closing by a 8x8 square structuring element. This step eliminates small holes that may still exist in IC regions, often caused by the presence of the light lettering in the interior of the region.

Finally, we perform region counting and eliminate regions whose size are below a chosen threshold. This effectively eliminated noise in the MAP output resulting from false positives (dark regions on the PCB resembling the color of an IC). The choice of this threshold was chosen experimentally to maximize results across the test image set. The final post-processed segmentation result is shown in Figure 3.

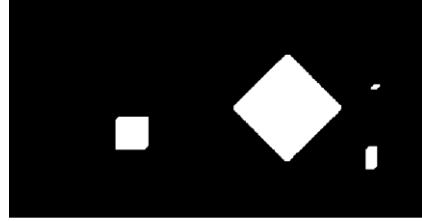


Figure 3. Segmentation result after post-processing

C. Polygon Detection

An algorithm is needed to identify the vertices of the rectangular regions comprising each detected IC to allow subsequent processing stages to operate on distinct IC regions. There are a number of heuristics that have been proposed in prior art to achieve this task [3]. These heuristics involve computing the convex hull of each IC region and eliminating supporting points on the basis of proximity and adjacency angles. There are a few issues applying this approach to the cleaned MAP output - given the variability of the MAP output in the presence of lighting gradients and shadows, only a single edge or corner of an IC may be identified by the MAP detector. Given that the detector has still identified a legitimate IC region, it is desirable to leverage the detected edge and corner geometries to reliably reconstruct the rectangular region that comprises the IC's perimeter - using only the computed convex hull points will cause only subsections of the IC region to be identified in many instances.

Eq. (1) presents a heuristic to find the best-fit rectangle vertices for the i th detected IC region over the set of region rotations θ_i .

$$\operatorname{argmin}_{\theta_i} \left(\left[\max_x(A_{\theta_i} V_i) - \min_x(A_{\theta_i} V_i) \right] * \left[\max_y(A_{\theta_i} V_i) - \min_y(A_{\theta_i} V_i) \right] \right) \quad (1)$$

$$s = \{[x \ y] \mid (x, y) \in R_i\}, \quad V_i = [s_1^T \ s_2^T \ \dots \ s_n^T]$$

$$A_{\theta_i} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{bmatrix}$$

The optimization problem in Eq. (1) can be summarized as follows: For each θ_i , rotate the detected IC region pixels by θ_i , and compute the smallest bounding box that fully contains the rotated region. We choose $\theta_{i,opt}$ that gives the smallest bounding box area. This gives the tightest fit rectangle over the set of region rotations θ_i . This heuristic is repeated for all detected regions. The output of this process is shown in Figure 4. This procedure worked well over our test set. As mentioned above, even when the MAP detector output only contained

single corners of an IC region, this heuristic allows us to accurately recover the original IC geometries.

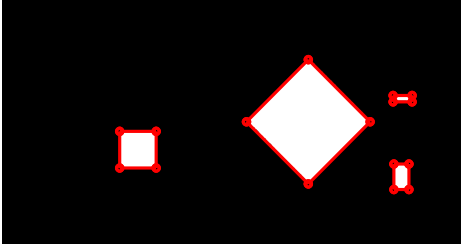


Figure 4. Best fit polygonal detection result.

D. IC Identification

The final step is to identify the part number of the detected ICs on the PCB. Initially, optical character recognition (OCR) was explored as a technique to parse the text printed on each segmented IC. To ensure peak character recognition performance, we took several pre-processing steps before invoking OCR. First, we rotate the i th region by $\theta_{i,opt}$ with respect to the vertical edge to correct for rotational misalignment. Then, a sharpening morphological filter is applied to enhance character edge sharpness, followed a binarization and region-counting that removes sufficiently small noise components. Each pre-processed IC region is then fed to the OCR software. The last step is to compute the Levenshtein distance between the text output of the OCR software and a database containing the names of each IC in the test set. We choose the word that minimizes the normalized Levenshtein distance (with respect to the length of the query). Ultimately, we found that OCR performance was inadequate to meet the desired project identification rate goals. The OCR was plagued by inadequate character definition due to low-resolution test images, noise in the binarized images, and company logos present in the inscription on each IC.

A SIFT descriptor matching approach was preferred for IC identification versus OCR. In this descriptor matching approach, SIFT descriptors are computed over a set of IC images that appear in the test set, and are stored into a vocabulary tree data structure. For each computed rectangular boundary containing an IC, we immediately compute the SIFT descriptors (no rotation required here due to rotational invariance of SIFT descriptors) and index into the vocabulary tree. We choose the IC corresponding to the entry in the vocabulary tree that minimizes the histogram distance:

$$\rho = \frac{\sum_{i=1}^n \min(Q_i, D_i)}{\sum_{i=1}^n D_i} \quad (2)$$

where D_i is the histogram of the database entry and Q_i is the query histogram.

This approach worked well for IC detection, and demonstrated resilience against the noise, low-resolution test images, and non-textual information that plagued the OCR software. Identification rates can be further boosted in the

future by building the vocabulary tree and testing with higher quality images. A depiction of this process is shown in Figure 5.

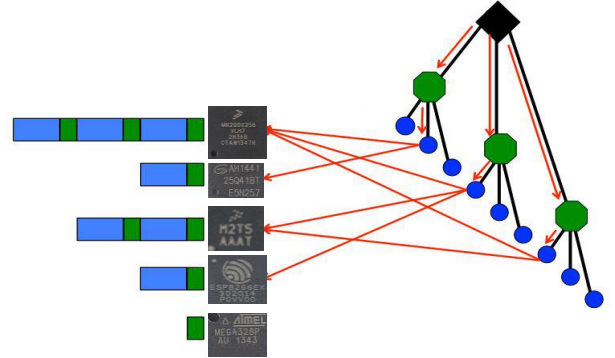


Figure 5. Vocabulary tree lookup based on SIFT descriptors.

IV. RESULTS

The results of the algorithm described in section III are presented here for four test images. The segmentation column refers to the results after polygon detection, but before attempting to identify the IC in the database. This result is visualized as rectangles drawn on the original image. Identification refers to correctly matching a segmented IC to its corresponding database entry. This result can be visualized as text displayed next to each segmented IC. For each image, three metrics are reported. *Number Missed* refers to the number of ICs that were not found during segmentation or a segmented region that could not find a match in the database during identification. *Number Incorrect* refers to a segmented region that is not an IC or a segmented region that found an incorrect match during identification. Finally the overall success rate (SR) is defined as:

$$SR_{Segmentation} = \frac{\#correctly\ segmented\ ICs}{\#total\ ICs}$$

$$SR_{Identification} = \frac{\#correctly\ identified\ ICs}{\#correctly\ segmented\ ICs}$$

for segmentation and identification respectively. Note ‘N/A’ is counted as a miss.

TABLE I. SUMMARY OF RESULTS

Image Number	Algorithm Results		
	Metric	Segmentation	Identification
1 (3 ICs)	<i>Number Missed</i>	0	1
	<i>Number Incorrect</i>	1	0
	Overall Success	0.75	0.67
2 (11 ICs)	<i>Number Missed</i>	1	1
	<i>Number Incorrect</i>	0	5
	Overall Success	0.92	0.27
3	<i>Number Missed</i>	0	1

Image Number	Algorithm Results		
	Metric	Segmentation	Identification
(6 ICs)	Number Incorrect	0	0
	Overall Success	1.0	0.83
	<hr/>		
4 (5 ICs)	Number Missed	1	2
	Number Incorrect	0	1
	Overall Success	0.8	0.25
<hr/>			
Total		0.89	0.46

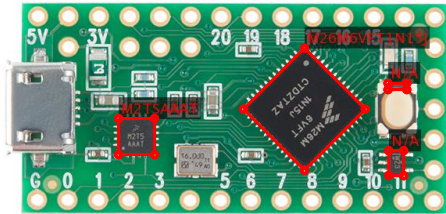


Figure 6. Test image 1 results

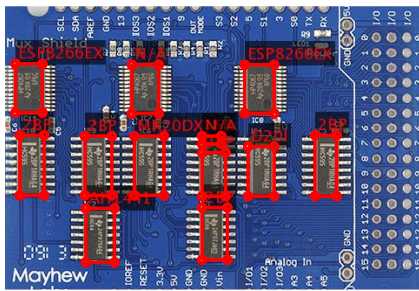


Figure 7. Test image 2 results

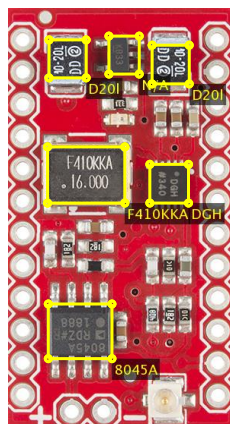


Figure 8. Test image 3 results

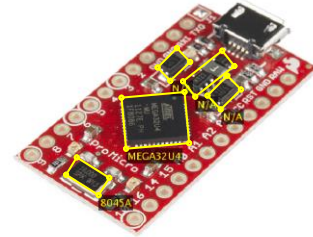


Figure 9. Test image 4 results

V. CONCLUSIONS

This paper has presented an image-based approach for detecting ICs on fully populated PCBs. An image-processing architecture was proposed that leveraged a MAP detector and morphological filtering techniques to segment IC regions, a polygon detection heuristic to reconstruct the vertices of ICs, and a SIFT descriptor matching technique to identify IC part numbers. Design tradeoffs and implementation details at each stage of the processing pipeline were examined on the basis of application performance. Finally, the paper presented the measured results of the proposed image-based IC detection approach. Future identification rate performance can be achieved by training the algorithm on both higher resolution images and a greater diversity of PCBs (across lighting and color variations). Similarly, using a higher performance camera to capture test images is expected to greatly improve IC identification rate performance, specifically during the SIFT descriptor matching step. We hope that future work can be undertaken to refine and improve this novel approach to the detection of ICs on populated PCBs.

REFERENCES

- [1] N. Amin and M.S. Khadem., "Interface development for cost effective automated IC orientation checking systems," In *Computer and information technology, 2007. iccit 2007. 10th international conference on*, pages 1–6, 2007.
- [2] P. Blaes and C. Young, "Mobile IC Package Recognition", Stanford Digital Image Processing Projects, Spring 2012.
- [3] R. Nagarajan, Sazali Yaacob, Paulraj Pandian, M. Karthigayan, Shamsudin, "A real time marking inspection scheme for semiconductor industries," *The International Journal of Advanced Manufacturing Technology*, 34(9-10):926–932, August 2006

APPENDIX

We participated equally in developing the algorithm architecture, the MatLab implementation, the poster presentation, and the final report.