MACHINE VISION IMPLEMENTATION IN RAPID PCB PROTOTYPING

IMPLEMENTASI *MACHINE VISION* PADA PEMBUATAN DUPLIKASI PCB

Yosafat Surya Murijanto, Rusman Rusyadi, Maralo Sinaga
Department of Mechatronics, Faculty of Engineering
Swiss German University, Bumi Serpong Damai
Banten 15321, Indonesia
yosafat. murijanto@student.sgu.ac.id; rusman.rusyadi@gmail.com; maralo.sinaga@sgu.ac.id

Recieved: August 24th, 2011; Revised: October 27th, 2011; Accepted: December 1st, 2011;
Published online: December 22th, 2011.

Abstrak

Pemrosesan *image* yang merupakan inti dari *machine vision* telah membuktikan dirinya menjadi bagian yang penting pada industri saat ini. Penerapan pemrosesan *image* telah membuka pintu yang baru dalam proses manufaktur yang lebih baik. Karya tulis ini memaparkan penerapan *machine vision* dalam proses desain modul yang mampu mengekstrak lubang bor dan mengarahkan jalur koordinat PCB dari kondisi yang belum tersusun menjadi jalur koordinat yang tersusun. Algoritma ini dimulai dengan proses *pre-capturing*, diikuti dengan segmentasi *image* dan pemfilteran, deteksi sisi dan kontur, ekstraksi koordinat dan diakhiri dengan pembuatan *G-code*. *Library* yang ada di OpenCV dan Qt IDE merupakan perangkat utama yang dipergunakan. Melalui beberapa percobaan yang dilakukan, disimpulkan bahwa algoritma ini mampu memberikan hasil yang cukup baik. Algoritma pengeboran dan ekstraksi jalur menghasilkan rata-rata 90% dan 82% dari jumlah lubang bor dan jalur yang ada pada PCB yang di-*scan* dalam total waktu pemrosesan kurang dari 3 detik. Hal ini dapat dicapai pada kondisi pencahayaan yang baik, permukaan PCB yang baik, dan *webcam* yang berkualitas baik.

Kata kunci: *PCB scanning*, ekstraksi koordinat, *G-code*, *machine-vision*.

Abstract

Image processing, the heart of machine vision, has proven itself to be an essential part of the industries today. Its application has opened new doorways, making more concepts in manufacturing processes viable. This paper presents an application of machine vision in designing a module with the ability to extract drills and route coordinates from an un-mounted or mounted printed circuit board (PCB). The algorithm comprises pre-capturing processes, image segmentation and filtering, edge and contour detection, coordinate extraction, and G-code creation. OpenCV libraries and Qt IDE are the main tools used. Throughout some testing and experiments, it is concluded that the algorithm is able to deliver acceptable results. The drilling and routing coordinate extraction algorithm can extract in average 90% and 82% of the whole drills and routes available on the scanned PCB in a total processing time of less than 3 seconds. This is achievable through proper lighting condition, good PCB surface condition and goodwebcam quality.

Keywords: PCB scanning, coordinate extraction, G-code, machine vision.

I. INTRODUCTION

This paper mainly focuses on the application of vision in manufacturing processes. The concept was born from the reverse engineering concept suggested by Koivunen [1], in which she stated that integrating reverse engineering and computer vision will benefit manufacturing processes in many ways. In this case, vision is used to generate design data of an existing printed circuit board (PCB). The data is then further processed to form a computer aided design (CAD) model and other manufacturing information, before finally a prototype of the scanned component is built in a manufacturing cell. G-code acts as the CAD model mentioned previously. The concept will greatly decrease design time and at the end could save much manufacturing cost.

II. METHODOLOGY

A. Creating Proper Lighting Condition

Martin [2] uttered that “the quality and appropriateness of lighting are critical aspects for creating a robust vision inspection”. Two lighting modules were constructed: back light to support drilling coordinate extraction (un-mounted PCB case) and front light to support routing coordinate extraction. Based on Figure 1 which is shows a comparison of common machine vision lighting, sources LED arrays was chosen as the lighting source of these lighting modules due to its

doi: 10.14203/j.mev.2011.v2.79-84

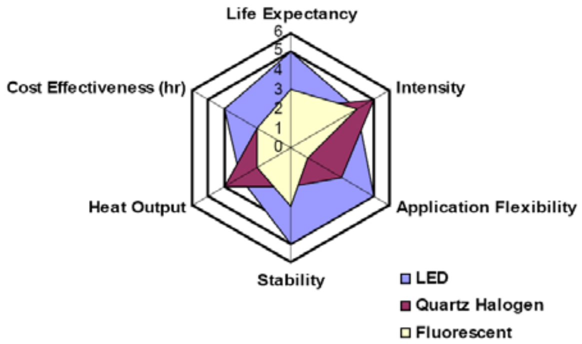


Figure 1. Comparison of common machine vision lighting sources [2].

superiority to other light sources: stability, flexibility, lifetime, and cost effectiveness (low power consumption).

Lighting techniques can be classified as bright field (on-axis lighting), partial bright field (directional lighting) and dark field (off-axis lighting). Each has its own application field and further categorized as direct and diffuse lighting. Figure 2 groups lighting techniques according to their application fields.

The application focuses on scanning PCBs, a surface with relatively uneven topology and mixed surface reflectivity, and thus bright and dark field lighting can be applied in the front light. However, bright field is chosen because of its better capability in generating contrast and enhancing topographic details [2]. When dealing with metals, however, bright field may create glares on the metal surface. This could result in incorrect contour extraction and at the end, false coordinate extraction. This problem can be reduced by applying a diffuser in our front light module [3].

Back lighting provides excellent image contrast as it produces silhouettes of the image against the bright background [2]. This is used to expose PCB edges and drill holes. The back light design can be observed in the figure 3.

B. Image Preparation

Image preparation will consist three operations. Those operations are color conversion, smoothing and edge detection described as follow.

1) *Color conversion*

Each frame grabbed by the webcam is stored as an red, green and blue (RGB) image. This is of course useful if it is used for visual inspection by humans, but not for image processing done by computers [4]. RGB image stores the value of each pixel in 3 channels and in many cases computers are only able to process operation in 1 channel at a time. Processing 3 channel images could take too much time and consume too much memory. Therefore RGB image has to be converted to grayscale image, which represents major features of the RGB image in a single channel.

2) *Smoothing*

Smoothing (or blurring) is done to remove unwanted camera artifact or noises [5]. The simplest burring technique is applied by replacing a pixel value with the average of its surrounding neighbor-pixels.

3) *Edge detection*

Edge detection is the base of contour extraction operation, which is the main algorithm of drilling and routing coordinate extraction. A method developed by J. Canny is applied. This technique takes the first and second derivatives of intensity of an image pixel, as shown in figure 4. An edge is defined if the second derivative result is zero and the first derivative result is above the defined threshold.

C. Routing Coordinate Extraction

The objective of developing this algorithm is the ability to extract lines from a complex image (PCB image). To do so, contour extraction and polygon approximation techniques are applied.

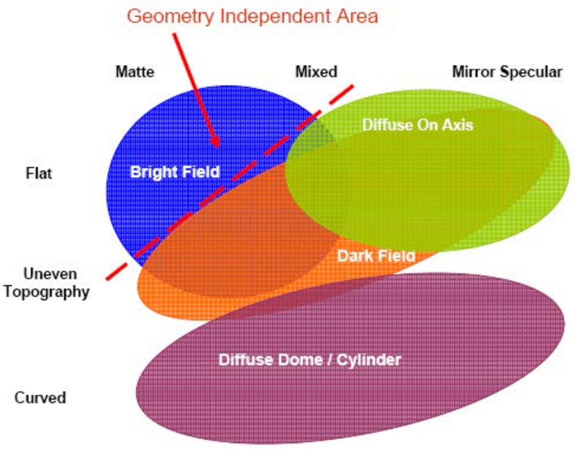


Figure 2. Lighting techniques application fields [2].

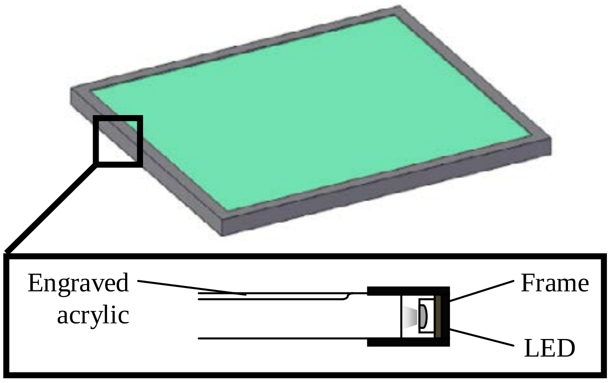


Figure 3. Back light design.

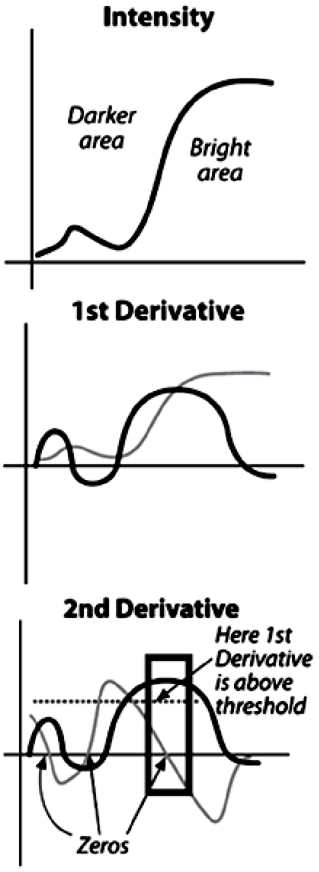


Figure 4. Canny edge detection concept [5].

A contour is a list of points that represent a curve in an image [5]. The contour extraction algorithm comprises component labeling and assembling. Component labeling differentiates holes from contours. Then these found contours are assembled according to the defined method. This module will assemble the contours as trees.

Polygons are the approximated from these contours. This is done to ensure that the extracted points form a close-loop and to filter the results from excessive points caused by image noises. First, a line is drawn from two extreme points of the contours. A farthest point of the contour, measured from that line, is then defined and lines are drawn to connect the newly-defined point with the existing points. This continues until the length of each line exceeds the defined precision parameter. Figure 5 illustrates the polygon approximation.

D. Drilling Coordinate Extraction

The challenge of developing the drilling coordinate extraction is to be able to detect small and imperfect circles. To meet this requirement, two main algorithms were applied: Hough [6] and thresholding-based circle detection. Hough circle is based on edge detection and first-order Sobel derivative. It provides more practical way of detecting the drill holes. The center coordinate of the circles is also directly returned. However ,the algorithm is very prone to noises.

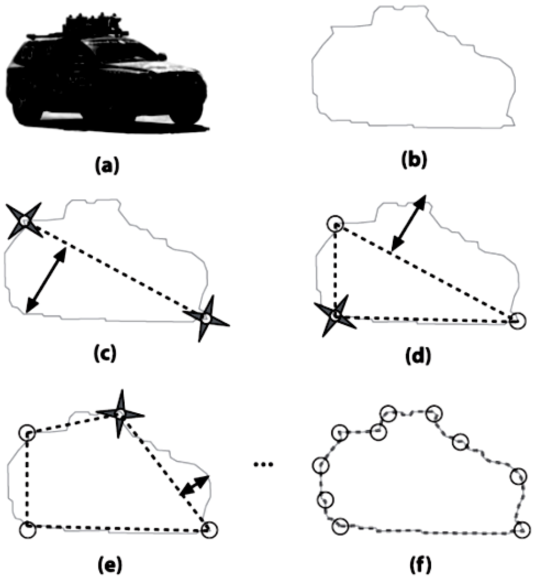


Figure 5. Polygon approximation [5].

The thresholding-based circle detection integrates tophat image morphology, basic thresholding, contour extraction and polygon approximation. Tophat algorithm is applied according to Equation (1). The input image is subtracted by the opened image. The opened image itself is done by first eroding and then dilating the image. Figure 6 illustrates the tophat image morphology. The result of the tophat image morphology is then thresholded and.dilated once again; this will reveal the drill holes. Finally, contour extraction and polygon approximation are applied and the drilling coordinates can be extracted. The advantage of this algorithm is the ability to preserve edges even on glare area or in case uneven reflection exists. However, the time needed to complete the operation highly depends on the number of circles to be detected.

$$\text{tophat(src)} = \text{src} - \text{open(src)} \quad (1)$$

E. Generating G-Code

The communication between systems should use standardized data formats [1]. For this reason, G-code file creation is chosen as the final step of the main program. After the coordinates are extracted, they are masked to form standardized G-code based on RS274D and DIN 66025 [7]. To be manufacturable, the G-code should include the drilling and routing coordinates and the drilling and engraving parameters, including spindle speed, feed rate and depth of cut. The G-code creation utilizes Qplain Text Edit class provided by Qt.

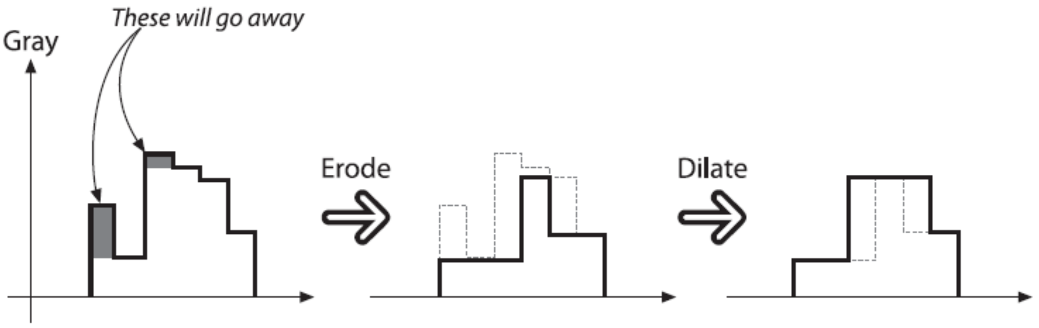


Figure 6. Tophat image morphology [5].

III. RESULTS AND DISCUSSION

A. Routing Coordinate Extraction

The routing coordinate extraction was tested using a simple “S curve” image and 4 PCB types.The result shows that it is able to detect in average more than 80% of all routes. As mentioned in the previous section, the algorithm highly depends on edge detection. However the edge detection failed in some cases. This is due to the uneven contrast and reflection caused by the front light module. If the input is an “ideal” image, the algorithm will perform perfectly, this can be seen as it was tested using a simple binary image. Evaluating the processing time, the algorithm managed to detect and process 1698 contours in 828ms time. The test conclusion can be observed in table 1 and the routing coordinate extraction result can be seen in figure 7.

B. Drilling Coordinate Extraction

The detections using hough circle and thresholding-based circle are discussed as follow.

1) Hough circle

Hough circle offers easy and practical way of detecting drill holes. However this algorithm delivers poor performance as it is applied in this case. The algorithm is based on Sobel operation, which is rather sensitive to noises. Setting the threshold values too high would eliminate most drill holes, but setting it too low would result in so much noises and false circle detection. This algorithm was only able to detect 13 out of 842 correct holes

2) Thresholding-based circle detection

This algorithm performs much better compared to the Hough circle. The holes in the glare area can also be really preserved due to the tophat algorithm. Though there are some problems around the glared and shadowed area, the result is still acceptable. The algorithm managed to detect 817 out of 842 holes on the matrix PCB. The test conclusion can be observed in Table 2 and the drilling coordinate extraction result can be seen in the Figure 8.

C. G-code Generation

After doing some observations and testing of the generated G-code, it can be seen that the algorithm delivers precise result. The test was done by manufacturing a simple PCB from a generated G-code. The result showed that every line can be executed correctly. The unit used in the G-code is mm as a result of applying G71 in the code. An example of generated G-code that shows the drilling operation can be observed in the Figure 9.

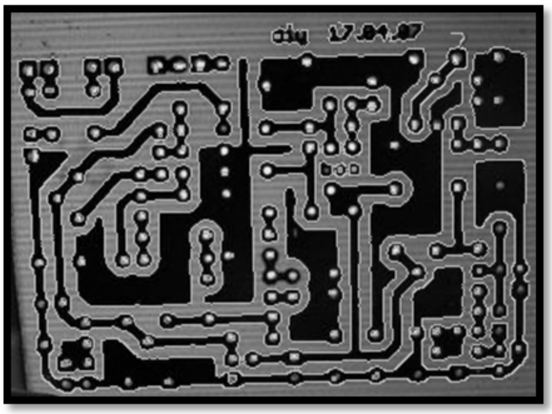


Figure 7.Routing coordinate extraction result.

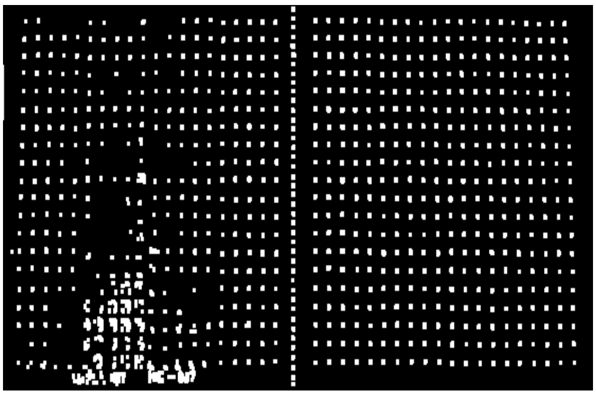


Figure 8. Drilling coordinate extraction result. (The rectangles show improper drill detection).

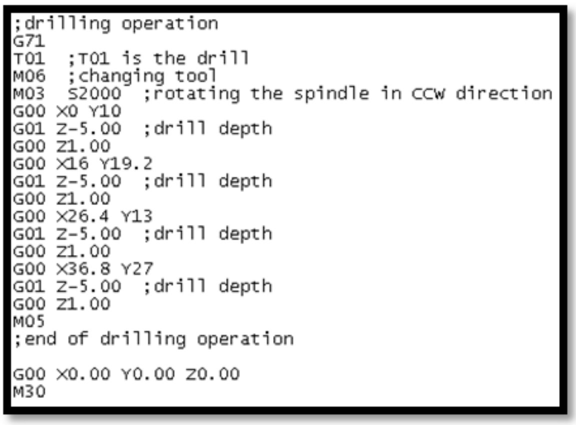
Table 1.
Routing coordinate extraction performance.

	SBI	SPCBA	SPCB B	DPCB	MPCB
Processed image size (pixels)	242 x 336	341 x 296	339 x 249	162 x 172	561 x 271
Detected raw contours	1	248	475	78	1698
Detected routes	1	23	34	47	227
Detected routes –approx.	100%	70%	95%	100%	45%
Processing time (ms)	78	265	438	156	828

Note: SBI: Simple Binary Image; SPCB-A: Single layer PCB A; SPCB-B: Single layer PCB B; DPCB: Double layer PCB; MPCB: Matrix PCB.

Table 2.
Overall drilling coordinate extraction performance.

	Hough circle	Basic thresholding	Tophat + thresholding
Detected circles	15	757	817
Correct holes	13	627	757
Error	98.46%	25.53%	10.10%
Processing time (ms)	438	16	140



```
;drilling operation
G71
T01  ;T01 is the drill
M06  ;changing tool
M03  S2000  ;rotating the spindle in CCW direction
G00 X0 Y10
G01 Z-5.00  ;drill depth
G00 Z1.00
G00 X16 Y19.2
G01 Z-5.00  ;drill depth
G00 Z1.00
G00 X26.4 Y13
G01 Z-5.00  ;drill depth
G00 Z1.00
G00 X36.8 Y27
G01 Z-5.00  ;drill depth
G00 Z1.00
M05
;end of drilling operation

G00 X0.00 Y0.00 Z0.00
M30
```

Figure 9. Generated G-code example.

IV. CONCLUSION AND RECOMMENDATION

The project managed to integrate vision sensor in reverse engineering process through the PCB scanning module. It is able to extract in average 80% of all routes and 90% of all drill holes of a PCB in less than 3s time. It is also able to output manufacturable G-code as the CAM data.

To perfect the module, the front light can be changed into dome or coaxial lighting, which will eliminate glares and uneven reflection. Computer graphic techniques such as path refining, line approximation and circle approximation can also be applied to output better engraving movement. Finally, a component library can be added to the database to improve the drilling coordinate extraction accuracy. This component library should comprise the basic components, especially standard integrated circuits (ICs).

REFERENCES

[1] Visa Koivunen and Ruzena Bajcsy, (1992, Aug.). *Rapid Prototyping Using Three-Dimensional Computer Vision. Scholarly commons.* [Online]. Available: http://repository.upenn.edu/cgi/viewcontent.cgi?article=1496&context=cis_reports&sei-redir=1#search=%22machine+vision+rapid+prototyping%22 [Accessed: November 4, 2011]

[2] Daryl Martin, *A Practical Guide to Machine Vision Lighting*. Rochester, VT: Advanced Illumination, October 2007.

[3] CVI Melles Griot, "Machine Vision Lighting Fundamentals," in *Fundamentals of Imaging and Machine Vision.* CVI Melles Griot 2009 Technical Guide, Volume 2, Issue 1. pp. 3. 2009.

[4] G. Bradski, *Open Source Computer Vision Library in Robotics*, New Jersey: Prentice Hall, 2004.

[5] Bradski, G. and Kaehler, A., "*Learning OpenCV*", 1st ed., CA: O'Reily Media Inc., 2008.

[6] Brovička, J., "COMS30121 – Image Processing and Computer Vision," *Circle Detection Using Hough Transform Documentation*, March 2003.

[7] Power Automation America, Inc., "*Standard List of CNC Codes.*" Machine Mate Inc. [Online]. Available: http://www.machinemate.com/StandardCodes.htm [Acessed: November 4, 2011]

[8] Y. Murijanto, *Machine Vision Implementation in Rapid PCB Prototyping.* Thesis Report, Swiss German University, BSDCity, Tangerang, Indonesia, July 2011

[9] D.A. Forsyth, & J. Ponce. *Computer Vision: A Modern Approach*. NJ: Prentice Hall, 2003.

[10] T.Q. Chen, J.X. Zhang, Y.N. Zhou, and Y.L. Murphey. *A Smart Machine Vision System for PCB Inspection*. University of Michigan, 2000.