Date of publication MAR-29, 2023, date of current version FEB-07, 2023.
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.22.1.2877

Chains Defects Detection in a Printed Circuit Board Image by the Plane Partition and Flood-filling of Traces

ROMAN MELNYK, TETYANA KOROTYEYEVA, YEVHENIYA LEVUS
Lviv Polytechnic National University, Ukraine,
(ramelnyk@polynet.lviv.ua, tetyana.o.korotyeyeva@lpnu.ua, yevheniia.v.levus@lpnu.ua, http://pz.lp.edu.ua)

Corresponding author: Tetyana Korotyeyeva (e-mail: tetyana.o.korotyeyeva@lpnu.ua).

ABSTRACT An approach to dividing the printed circuit board into parts to increase visibility of defects in a PCB image is considered. The approach is based on a piecewise linear approximation of a cumulative histogram. The last one is calculated for numbers of informative pixels placed in rows and columns of an image matrix. Informative pixels are those indicating contacts and C traces. The histogram demonstrates a distribution of informative pixels along the OX and OY axes. The beginning and ending points of linear lines are taken as coordinates of the divided parts of the PCB board. The flood-fill algorithm is used to color and separate the PCB chains components. The start pixels are taken from a set of informative pixels. To measure defect values, a mean intensity and gradient functions for the template and manufactured images are used. Distributed cumulative histograms are applied to PCB components to detect places and intensity of defects.

KEYWORDS printed circuit board; defect detection; chain; contact; pixel; trace; flood-fill; short; open; distributed cumulative histogram; piecewise linear approximation; cumulative histogram; distribution of pixel.

I. INTRODUCTION

COMPUTER vision algorithms for printed circuit board defect detection are widely described in literature. For example, in [1-4] many of them were selected and described. Some specific inspection techniques for fault diagnostics, services and artificial networks for image processing as well as defects classification were considered in works [5-8]. The Machine Vision PCB Inspection System used in manufacturing and creating bare PCB was presented in [9, 10]. Comparing a standard PCB image with a PCB image to be inspected, the approach uses a simple subtraction algorithm. In the works [13-14] PCB color images are grayed, wavelet denoising and histogram equalization techniques are applied. The defects are extracted through the differential processing between them and the image segmentation. In publications [15-18], to compare two PCB images a subtraction algorithm to detecting the defected regions was used. In works [19-22] different methods of PCB defect detection were presented, and in [22-25] description of databases of a PCB image for conducting experiments with methods of detecting defects in PCB images was given.

In this article detection of defects in a PCB image is performed due to multilevel segmentation applied to the whole PCB image to get separated parts that are images of smaller sizes. The flood-filling of different objects in a PCB image helps to find specific features of separated chains. The found pixels are input data for checking algorithms to find defects in traces and extra material on a PCB background.

II. PCB IMAGE PARTITION

The main part of defects in manufactured PCB samples relates to their chains realized by traces and contacts. Defects are of different nature. One group of them relates to extra conducting material or a lack of conducting material. The second group unites defects caused by incorrect shape of traces and defects of connectivity which are also caused by a lack or extra amount of conducting material. In PCB images all these defects are reflected by black and white pixels arising in comparison to a template sample.

An example of the PCB image with the two defects is shown in Fig. 1. Open and short defects are encircled with red color. In the first case a small area of a black trace is changed by white pixels. And vice versa white pixels of the space between two chains changed their color to black.

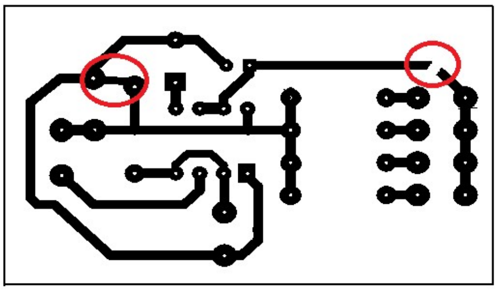


Figure 1. Example of PCB image with defects

As a rule, a few pixels reflecting defects is a very small amount compared to a number of white or black pixels of the whole image. Even having a template sample, they are superimposed on the inaccuracies of manufacturing PCBs. Also, they are small comparatively to the size of a row or column in the image matrix. An example of defects influencing in a column of the mean intensity function is demonstrated by graphs in Fig. 2. Graphs are calculated for the template (black) and defective (red) PCB images. They differ between themselves in two places. The difference is very small. For the row function the difference is even smaller.

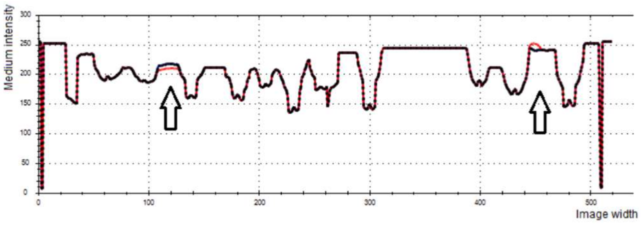


Figure 2. Mean intensity functions in columns of the etalon and defective PCB images. OX: width 0-500, OY: intensity 0-300

There is one way to increase visibility of pixels indicating white and black defects. Their relative weight must be increased that means the relative weight of surrounding pixels must be decreased. It can be reached by decreasing the area for measurement. It is illustrated by the graphs in Fig. 3 where the mean intensity functions are calculated for the upper half of the PCB image.

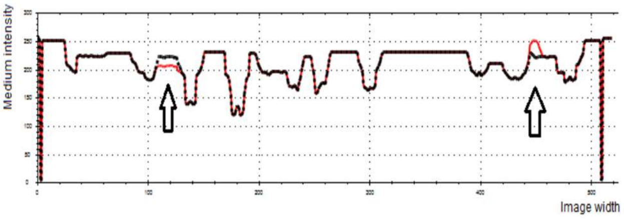


Figure 3. Mean intensity functions in columns of the etalon and defective PCB images with indicated difference, OX: width 0-500, OY: intensity 0-300

Thus, searching for defective pixels requires much smaller areas for comparative measurement, whether it is intensity, variance, or other features.

The first step of the algorithm is to solve the decomposition problem for the PCB image as a partition into smaller fragments of the etalon and manufactured PCB images. The simplest approach is a division of a plane into cells by a covering grid. The disadvantage of this approach is that the cutting lines divide too many traces and contacts.

To minimize the number of intersections of lines and PCB elements the partition procedure is considered. The procedure is based on the method of multilevel segmentation [11] of a special cumulative histogram. To illustrate the main idea of the ordinary and the cumulative special histograms they are calculated. But instead of the traditional probability/intensity axes they use probability/coordinate axes (OX and OY):

$$G_c(s)=\sum G(j), j=1,\ldots,s,\ s=1,\ldots,N, \tag{1}$$

where $G(j)$ is the overall number of pixels in PCB elements for the j-th coordinate, $G_c(s)$ is the cumulative number of pixels for the s-th coordinate, N is the number of intervals.

In Fig. 4 special ordinary and cumulative histograms are calculated for the PCB image. The special ordinary histogram is red. It is difficult in this histogram to find points dividing the image width in two, three or more fragments and satisfying a minimal number of intersected PCB elements such as white traces and contacts.

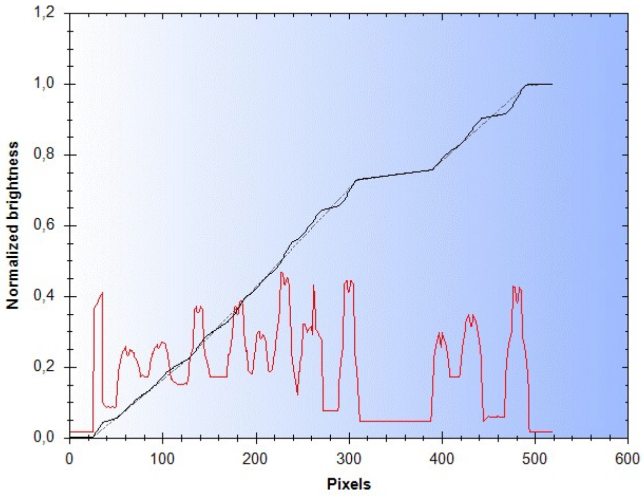


Figure 4. Special ordinary (red) and cumulative histograms (blue) and its approximation

The function $G(x)$ of the special cumulative histogram is approximated by the piecewise-linear function $g(x)$ determined on the interval $a_i \le x_i \le a_{i+1}$ with a set of sections having the end points (α_i, β_i) and $(\alpha_{i+1}, \beta_{i+1})$:

$$g(x) = \sum_{i=0}^{N-1} g_i(x), \tag{2}$$

where $g_i(x) = \begin{cases} m_i x + b_i & \text{for} \quad \alpha_i \le x \le \alpha_{i+1} \\ 0 & \text{in other case} \end{cases}$,

where m_i is the angular coefficient (slope). After approximation the coordinates of the breaking points are obtained. Found points are the beginning and ending coordinates of lines with a certain angle of inclination. The angle indicates how quickly the pixel number increases in this interval. The lines are shown in Fig. 5.

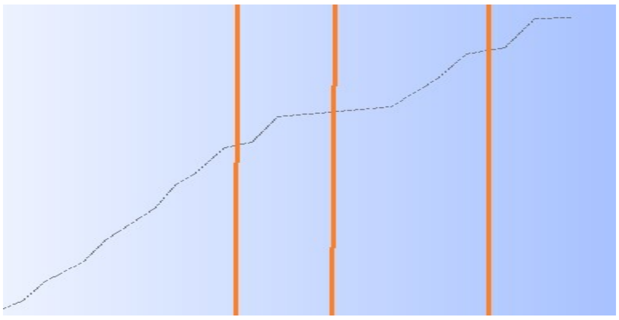


Figure 5. Division of interval lines of the piecewise linear function

The angles are calculated for all intervals. Then the intervals are sorted by corresponding values of angles. Resulting positions of intervals indicate a few parts for division of the image by one coordinate. The first position indicates two parts, the second position (together with the first one) indicates three parts and so on. The coordinate for division is taken as a middle of the interval.

A similar procedure is realized for the OY axis. The found coordinates of dividing lines are applied as for the etalon PCB image and for the control (manufactured) PCB image. Thus, the found parts are taken as input data for the next algorithm of defective chains detection.

For the etalon PCB image four intervals on the OX axis and two intervals on the OY axis create a 2x4 partition for analysis and comparison of eight separate parts of the PCB image Corresponding dividing lines are shown in Fig. 6.

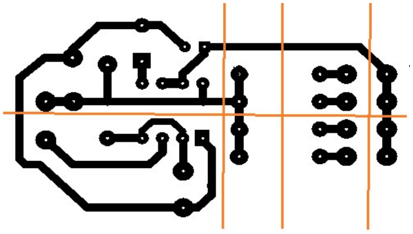


Figure 6. 2x4 partition of the PCB image

Three upper parts of the image are shown in Fig. 7.

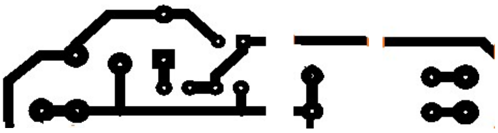


Figure 7. Three upper parts of the PCB image

III. FLOOD-FILL ALGORITHM FOR CHAINS SELECTION AND PLACEMENT CONTROL

Two binary parts of the etalon and the control PCB images are characterized by global and local features indicating absence or available defects. The most general feature is a cumulative histogram calculated without prevailed white color. In Fig. 8 two partial images are shown and their cumulative histograms have values 0.176 for the correct part and 0.170 for the image having a defect. So, they differ between themselves by 4 percent.

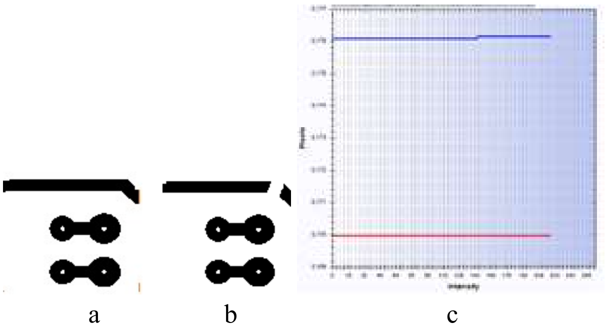


Figure 8. Parts of the etalon (a) and control (b) PCB images and their cumulative histograms (c), OX: intensity 0-255, OY: number of pixels 0.169-0.177

A difference of 4 percent is small enough and by other statistical features this value can be much smaller. That is why the next decomposition must be realized. Especially since the PCB image contains separate chains.

For one part of the image all black pixels are placed to the table with their attributes: coordinates, color, an ID number of a chain. One more smaller table is being formed to contain features of the chains. Every pixel from the first table is a candidate to be accepted as a center of wave propagation for the flood-fill algorithm.

The task is to find chains and to calculate for them all possible features. The number of chains is also a global feature for the partial image. The following procedure illustrates it for one chain. Taking one black pixel from the table as the start and the target points the flood-fill algorithm replaces colors for all neighboring black pixels. When the flood-fill algorithm ends its work and separates one chain its features are being calculated.

The simplest features are mathematical expectations of pixels coordinates:

$$\bar{x}(s) = 1/k_s \sum X(s), \bar{y}(s) = 1/k_s \sum Y(s),$$

$$x_i \in X(s), y_i \in Y(s), \qquad (3)$$

and the dispersion of pixel coordinates of the chain:

$$E^2(x,y,s) = (1/k_s)\cdot\sum_{x_i\in X(s),y_i\in Y(s)}(x_i - \bar{x}(s))^2 + (y_i - \bar{y}(s))^2, \qquad (4)$$

where k_s is the number of pixels in the chain; $\bar{y}(s), \bar{x}(s)$ are mathematical expectations of pixel coordinates; $X(s)$, $Y(s)$ are sets of pixel coordinates forming the chain; E^2 is the dispersion of pixel coordinates.

To calculate the parametric features connected with axis coordinates the image surface is divided by a grid. A step depends on the needed accuracy. In Fig. 9 dimension of the grid is 10x15. Thus, the numbers of the grid cells are from 1 to 150 and the numeration order is from left to right and from bottom to down.

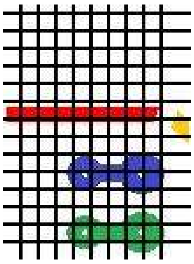


Figure 9. One part of the control PCB image covered by a grid

Mathematical expectations of pixel coordinates are placed in the corresponding cells. The cell number is a coordinate itself and contains a coordinate on the OY axis and a coordinate on the OX axis. Having two statistical coordinates of the pixels center (two mean values) coordinates of the grid cell are calculated. They are as numbers of a row and a column of the grid to which the center belongs:

$$i = (\overline{x}(s)) / L_x)_{cells} + 1, \quad (5)$$

$$j = (\overline{y}(s)) / L_y)_{cells} + 1, \quad (6)$$

where L_x, L_y are the numbers of cells in a row and a column of the grid, j_s*L_x+i_s or i_s*L_y+j_s is the real sequence number depending on the accepted order.

To compare all chains in the image, a part of these chains is remained, but flood filled with different colors. Examples of the etalon and control PCB parts with one defect are given in Fig. 10.

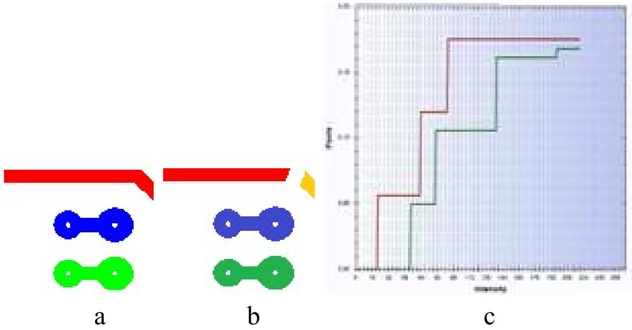


Figure 10. Parts of the etalon and control PCB images with marked chains (a,b) and cumulative histograms (c), OX: intensity 0-255, OY: number of pixels 0.0-0.20

Due to different start pixel the same chains are marked by different colors. One chain is broken and an additional color is used. Different numbers of colors used for flood filling are confirmed by cumulative histograms of the parts (Fig. 8c). The final difference between histograms is the same 4 percent but the second histogram demonstrates 4 colors instead of 3 colors in the first histogram.

The most valuable feature of the chain is a mean value of pixels coordinates that is a point at the geometric center. These centers are illustrated in Fig. 11. In this case colors do not matter. Illustration of changing centers and the new centers of defects is shown in Fig. 11 where a height corresponds to the coordinates and the OX axis corresponds to intensity.

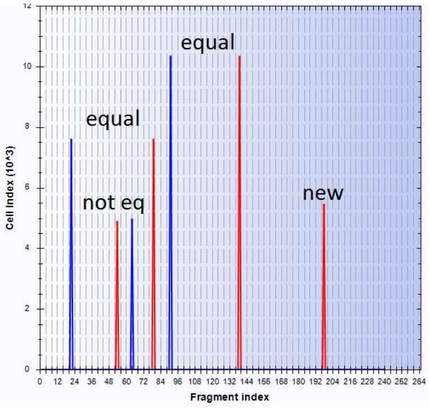


Figure 11. Coordinate of chains centers for the etalon and control images

IV. MEASUREMENT OF DEFECTS INTENSITY

Flood filled with different colors chains on the etalon and real PCB images are objects to compare. The comparison is made for all components or only for those that do not satisfy checking conditions. There are many approaches allowing us to measure and compare robust and suspicious components. Some of them are presented here.

A. INTENITY MEASUREMENT

To highlight defects in the chain image having extra or missing pixels, different functions are used. They are applied to chains of the template and manufactures PCB images to find defect positions and their intensity. The chains are flood-filled with different colors and are selected as input data to detect different types of defects.

Mean intensity function. To measure the chain inaccuracy a pixel intensity function in the W columns and H rows of the image matrix is used:

$$\bar{I}(i) = 1/W(\textstyle\sum_{j=1}^{W} I(i,j)), i = 1,2,\dots,H, \quad (7)$$

$$\bar{I}(j) = 1/H(\textstyle\sum_{i=1}^{H} I(i,j)\,, j = 1,2,\dots,W, \quad (8)$$

where $I(i,j)$ is pixel intensity in the i-th row and the j-th column ($1 \le i \le H$, $1 \le j \le W$).

In Fig. 12 a pair of the etalon and the real components are given. They demonstrate the correct and the defective chains.

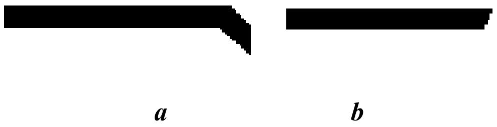


Figure 12. Etalon (a) and control (b) chains

The mean intensity functions in columns and rows of these two samples illustrate the different numbers of pixels in the two images in Fig. 13 and Fig. 14. The marked area indicates a difference between the two samples. Thus, some tolerance values could be assigned to take a decision about available defects.

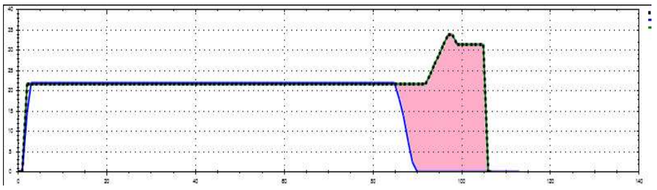


Figure 13. Mean intensity functions in columns of two PCB components and marked difference, OX: width 0-140, OY: intensity 0-40

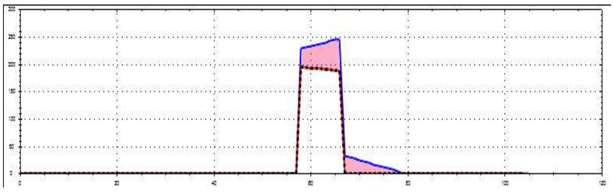


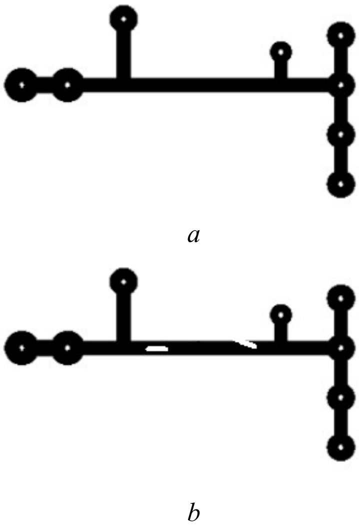
Figure 14. Mean intensity functions in rows and marked difference, OX: width 0-120, OY: intensity 0-300

To measure the difference between graphs of the real and etalon components, the distance D_{ch} between two graphs is calculated as the sum of the absolute (or quadrate) values of a difference between the functions for every intensity value:

$$D_{ch} = \sum_{s=1}^{W} \left| f_1(s) - f_2(s) \right| \qquad (9)$$

The values of the manufacture tolerance and calculated D_{ch} are input data for the program to take a decision: to reject or to accept the current PCB fragment as suitable.

Two more complex examples of the template chain and with defects are shown in Fig. 15.



a

b

Figure 15. Chains from PCB images: a – without defects, b – with defects

To increase visibility of defects in the intensity functions, two images are inverted. It is made to amplify a weight of defect pixels among pixels of a background. The mean intensity functions in columns for two inverted images from Fig. 15 are given in Fig. 16.

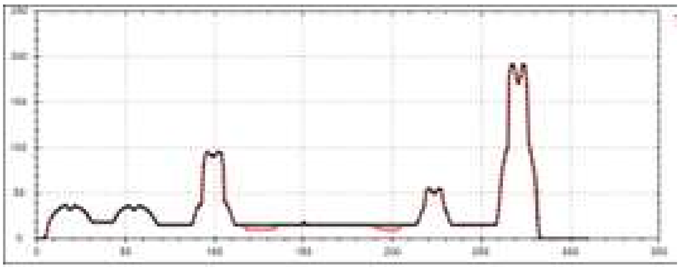


Figure 16. Mean intensity functions for template chain (black) and defective chain (red), OX: width 0-350, OY: intensity 0-250

The red graph of the defective chain has some deviation in places of defects in a body of the chain. To measure this deviation a difference of two functions is calculated to make conclusion about places and intensity of defects:

$$S = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{i_k} |I_i - \bar{I}_k|^2 < t, \qquad (10)$$

where t is a tolerance value which indicates what deviation is acceptable and what are the coordinates and intensity of defects. In Fig.16 absolute values of the template and defective images are 15.0 and 7.5 for corresponding coordinates. It is a relative deviation of near 50 %. For not inverted images these values are 244 and 239 with 2 % of a relative deviation. That is why black background gives wider range for taking a decision about possible inaccuracies.

Mean gradient functions. One more function to estimate defects is to measure mean gradient in rows or columns:

$$\bar{I}(i) = 1/W(\textstyle\sum_{j=1}^{W} I(i,j)), i = 1,2,\ldots,H, \qquad (11)$$

$$G(j) = 1/H(\textstyle\sum_{i=1}^{H} G(i,j)), j = 1,2,\ldots,W, \qquad (12)$$

where $G(i,j)$ is gradient value for two neightbouring pixels in the i-th row and the j-th column ($1 \leq i \leq H$, $1 \leq j \leq W$). For the two chains in Fig. 15 their mean gradient functions are given in Fig. 17.

The red graph of the defective chain has some deviation in places of defects in a body of the chain. In Fig. 17 absolute values of the template and defective image are 3.8 and 6.5 for corresponding coordinates. It is a relative deviation of near 70 %. For not inverted images these values are the same. So, in the case of gradients there is no difference what types of images are being processed.

Such a procedure is being applied for every chain which the connectivity algorithm marks as suspicious.

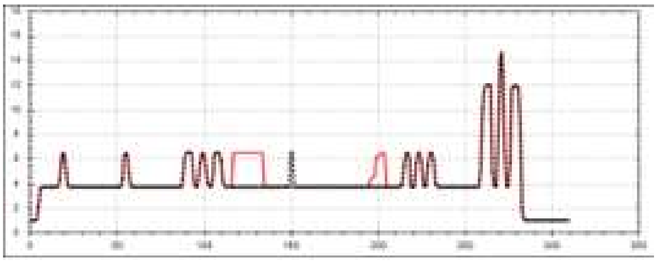


Figure 17. Mean gradient functions for template chain (black) and defective chain (red), OX: width 0-350, OY: number of changes 0-20

B. EVALUATION BY DISTRIBUTED CUMULATIVE HISTOGRAM

To visualize defects a distributed cumulative histogram (DCH) [11] is more productive. It is possible to notice changes in the difference between pixels on the etalon and real samples. Fig. 18 contains two DCH of the etalon and defective chains, and their difference overlaid with a chain.

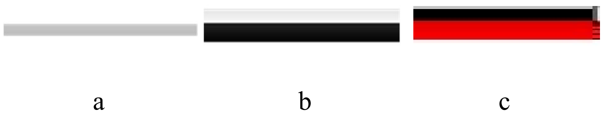


Figure 18. Distributed cumulative histograms (a,b) and their difference (c)

To increase the difference of the intensity function and visualize data reflecting defects, the following approach from the work [11] is used. In Fig. 19 two images of distributed cumulative histograms of two previous images are shown. A simple observation and comparison of images does not give an answer as to how much these images differ between themselves. Special procedures are developed to compare two images.

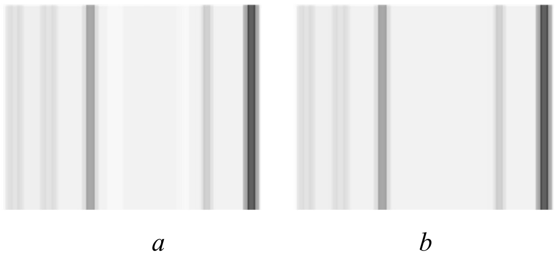


Figure 9. Distributed cumulative histograms of the etalon (a) and real PCB trace (b)

It is based on the following formulas:

$$\begin{aligned} &I_r(x,y)=I_d(x,y)\,, \\ &\forall\,|\,I_e(x,y)-I_d(x,y)\,|<Tol\,, \\ &I_r(x,y)=RGB(|\,I_e(x,y)-I_d(x,y)\,|,00)\,, \\ &\forall\,|\,I_e(x,y)-I_d(x,y)\,|\geq Tol\,, \end{aligned} \tag{13}$$

where $I_r(x,y)$ is the pixel intensity of the resulting image, $I_e(x,y)$ is the pixel intensity of the first image, $I_d(x,y)$ is the pixel intensity of the second image, Tol is the tolerance value for controlling the difference between the pixel intensity of the input and controlled samples.

Applying this formula to the DCH images from Fig. 19 the resulting image with marked places of different intensities is obtained and shown in Fig. 20a. A width of the resulting image is the same as a width of the DCH image and is the same as a width of the chain image. The red stripes indicate where in the defective image are some pixels which are absent in the template trace. Their OX coordinates correspond to the OX coordinates of defects. To illustrate defects and their positions, the DCH difference and the defective trace images with 50 percentage of opacity are added. The resulting image is shown in Fig. 20b.

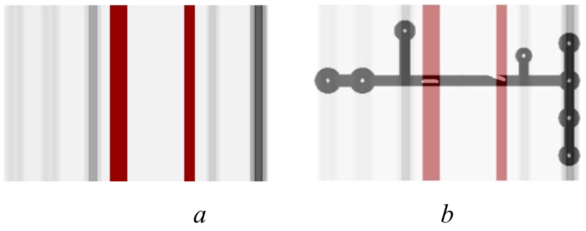


Figure 20. Difference of DCH images (a) and an overlay of the defective chain image with this difference (b)

So, the DCH image allows detecting small defects in the trace of the chain. Moreover, this approach allows us to detect much bigger defects connected with broken short traces. The procedure is the same: selection and separation of the template and the real chains, processing them to obtain DCH images, subtraction of two images and marking suspicious places of defects. The considered technique allows detecting defects as by their coordinates and by the intensity of incorrectness.

V. DEFECTS OF CONNECTIVITY

Two simple fragments of the PCB images are used to illustrate the approach to detecting defects in connectivity. The first image is with an open defect. And the second image has a short within two chains. They are given in Fig. 21.

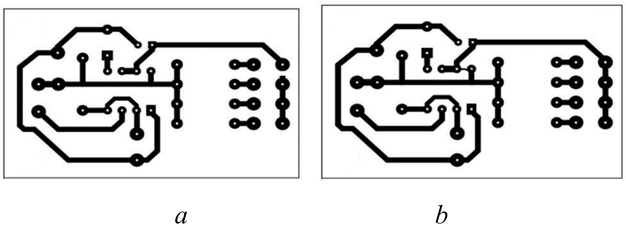


Figure 21. PCB images: a – with open, b –with short defects

In the template image all pixels of all chains are marked with different colors by the flood filling algorithm. As a result all black (white) pixels are united in the groups with assigned colors in Fig. 22a. Pixels of these groups create chains marked with different colors as it is shown in Fig. 22b.

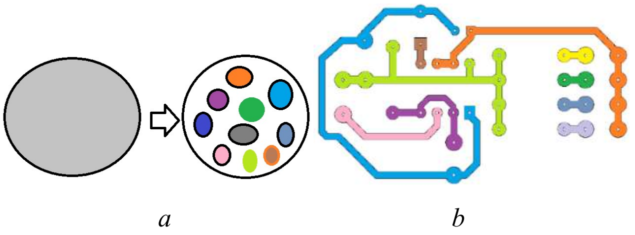


Figure 22. Division of all pixels into separate groups (a) and marked chains (b)

A few determined chains in the control image coincides with the correspondent number of chains in the template image for two cases when the template image has no defects of connectivity and when the number of broken traces equals to the number of open defects. In other cases, the determined number of chains in the control image indicates connectivity defects.

The procedure fulfills iterations according to the number of found chains. For one chain of the manufactured PCB the flood-fill algorithm takes an arbitrary pixel from the set of pixels $R_i\{r_{i1}, r_{i2},\dots,r_{im2}\}$ of a chain in the template image. This pixel is taken as a start point with its position and color.

When the chain in the control image is built the answer to the following question "is any open or short defect in the chain?" gives the following condition check:

- if the current chain does not contain a full set of own pixels from $R_i\{r_{i1},r_{i2},\dots,r_{im2}\}$ and the difference of pixels is more than a tolerance, it is the open defect;
- if a current chain contains all contacts of $R_i\{r_{i1},r_{i2},\dots,r_{im2}\}$ and no one from other sets, there is no open defect in the chain;
- if the current chain contains pixels from two and more sets of pixels and the difference between pixel numbers is more than a tolerance, there is a short defect between different chains.

Application of the flood-fill algorithm is illustrated in Fig. 23 by an example of the PCB image. One chain having a break (open) is marked with red and blue colors. A breaking line divides the chain into two parts. When checking chains these two fragments do not include all native pixels included in the etalon set. Here it is the first checking rule fulfilled.

The second case is illustrated in Fig. 24. Due to incorrect connection two chains touch one another and the flood-fill algorithm marks them with red color as one chain. When checking one chain this component includes all native pixels placed in two etalon sets. Here it is the third checking rule fulfilled.

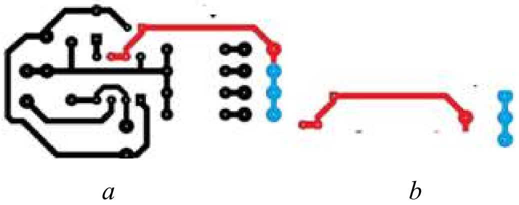


Figure 23. PCB image with one chain with an open defect (a) and with two of its components (b)

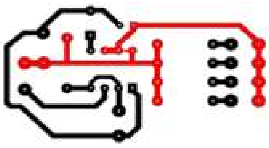


Figure 24. PCB image with filled two chains with short defect

In practical work when the algorithm terminates its work the marked defective chain is copied on the reserve image to be highlighted.

VI. CONCLUSION

The partition algorithm based on multilevel segmentation is used to divide the PCB board into smaller parts. The flood-fill algorithm is implemented and applied to the PCB image to separate all connection traces. To detect defects in the PCB image chains, the following approaches are used: mathematical expectation of pixel coordinates belonging to different chains are calculated and compared, mean intensity and gradient functions as chain features are compared, distributed cumulative histograms are used to detect PCB defects and visualize them. The resulting algorithm could indicate wire breaks, short circuits, hump, hole, and other types of defects.

The checking procedure detects defects of connectivity and extra components on the background. For visualization distributed cumulative histograms are calculated and subtracted.

References

[1] D. B. Anitha, M. Rao, "A survey on defect detection in bare PCB and assembled PCB using image processing techniques," *Proceedings of the International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2017, pp. 39-43. https://doi.org/10.1109/WiSPNET.2017.8299715.

[2] K. P. Anoop, N. S. Sarath, V. V. S. Kumar, "Review of PCB defect detection using image processing," *International Journal of Engineering and Innovative Technology (IJEIT)*, vol. 4, issue 11, pp. 188-192, 2015.

[3] S. McClure, "Extracting and classifying circuit board defects using image processing and deep learning," [Online]. Available at: https://towardsdatascience.com/ building-an-end-to-end-deep-learning-defect-classifier-application-for-printed-circuit-board-pcb-361b3a76232.

[4] Seven types of defects, [Online]. Available at: https://blog.matric.com/7-types-of-pcb-soldering-defects-to-watch-out-for.

[5] Visual Fault Analyzer (VFAT) speeds PCB fault Diagnostics and Repair, [Online]. Available at: http://www.intellitech.com/company/

[6] V. A. Adibhatla, H.-C. Chih, C.-C. Hsu, J. Cheng, M. F. Abbod, and J.-S. Shieh, "Defect detection in printed circuit boards using you-only-look-once convolutional neural networks," *Electronics*, vol. 9, 1547, 2020. https://doi.org/10.3390/electronics9091547.

[7] Y. Hanlin, W. Jun, "Automatic Detection Method of Circuit Boards Defect Based on Partition Enhanced Matching," *Information Technology Journal*, vol. 12, issue 11, pp. 2256-2260, 2013. https://doi.org/10.3923/itj.2013.2256.2260.

[8] J. Zhu, A. Wu, X. Liu, "Printed circuit board defect visual detection based on wavelet denoising," *IOP Conference Series: Materials Science and Engineering*, vol. 392, pp. 1-6, 2018. https://doi.org/10.1088/1757-899X/392/6/062055.

[9] F. B. Nadaf, V. S. Kolkure, "Detection of bare PCB defects by using morphology technique," *International Journal of Electronics and Communication Engineering*, vol. 9, no. 1, pp. 63-76, 2016.

[10] P. R. J. Nayaka, K. Anitha, B. D. Parameshachari, R. Banud, P. Rashmi "PCB fault detection using image processing," *IOP Conference Series: Materials Science and Engineering*, vol. 225, pp. 1-5, 2017. https://doi.org/10.1088/1757-899X/225/1/012244.

[11] R. Melnyk, R. Tushnytskyy, R. Kvit, "Cloudiness images multilevel segmentation by piecewise linear approximation of cumulative histogram," *International Journal of Computing*, vol. 19, issue 2, pp. 199–207, 2020. https://doi.org/10.47839/ijc.19.2.1762.

[12] R. Melnyk, Yu. Kalychak, "Detection of defects in printed circuit boards by flood-fill algorithm and distributed cumulative histogram," *Proceedings of the XIII International conference on Computer sciences and information technologies (CSIT-2018)*, 2018, pp. 155–159. https://doi.org/10.1109/STC-CSIT.2018.8526733.

[13] S. Ganesan, M. Pecht, "Open trace defects in FR4 printed circuit boards," *Circuit World*, vol. 32, issue 1, pp. 3-7, 2006. https://doi.org/10.1108/03056120610616490.

[14] Why Do We Use Copper to Make PCB Traces? [Online]. Available at: https://yic-assm.com/why-do-we-use-copper-to-make-pcb-traces/.

[15] A. P. Singh Chauhan, S. C. Bhardwaj, "Detection of bare PCB defects by image subtraction method using machine vision," *Proceedings of the World Congress on Engineering*, vol. II, WCE, 2011, pp. 6-8.

[16] Bond, "An Efficient and Versatile Flood Fill Algorithm for Raster Scan Displays," 2011. [Online]. Available at: www.crbond.com.

[17] K. Kamalpreet, K. Beant, "PCB defect detection and classification using image processing," *International Journal of Emerging Research in Management &Technology*, vol. 3, issue 8, pp. 42-46, 2014.

[18] M. H. Tatibana, R. de A. Lotufo, "Novel automatic PCB inspection technique based on connectivity," *Proceedings of the X Brazilian Symposium on Computer Graphics and Image Processing*, 1997, pp. 187-194.

[19] M. Moganti, F. Ercal, C. H. Dagli, S. Tzumekawa, "Automatic PCB inspection algorithms: A review," *Computer Vision and Image Understanding*, vol. 63, issue 2, pp. 287-313, 1996. https://doi.org/10.1006/cviu.1996.0020.

[20] K. P. Anoop, N. S. Sarath, V. V. Kumar, "A review of PCB defect detection using image processing," *International Journal of Engineering and Innovative Technology (IJEIT)*, vol. 4, issue 11, pp. 188-192, 2015.

[21] S. H. Putera, Z. Ibrahim, "Printed circuit board defect detection using mathematical morphology and MATLAB image processing tools," *Proceedings of the 2nd International Conference on Education*

Technology and Computer, 2010, vol. 5, pp. 359-363. https://doi.org/10.1109/WiSPNET.2017.8299858.

[22] F. Vikas Chaudhary, I. R. Dave, K. P. Upla, "Visual inspection of printed circuit board for defect detection and classification," *Proceedings of the International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2017, pp. 732-737.

[23] Guide to PCB issues. [Online]. Available at: https://www.mclpcb.com/guide-pcb-issues/.

[24] A PCB Dataset, [Online]. Available at: https://www.123rf.com/photo_40192468_computer-chip-showing-high-tech-and-circuit.html.

[25] PCB images, [Online]. Available at: https://www.eleccircuit.com/5-30-minuts-timer-alarm-circuit-using-ic555/.

***Roman Melnyk**, a Professor of Software Engineering Department, Institute of Computer Sciences and Information Technologies, Lviv Polytechnic National University. Research interests: fuzzy clustering, recognition of visual information*

***Tetyana Korotyeyeva**, an Associate Professor of Software Engineering Department, Institute of Computer Sciences and Information Technologies, Lviv Polytechnic National University. Research interests: neural networks, recommendation systems.*

***Yevheniya Levus**, an Associate Professor of Software Engineering Department, Institute of Computer Sciences and Information Technologies, Lviv Polytechnic National University. Research interests: computer graphics, software quality*

• • •