Defects Detection in Printed Circuit Boards by Separation and Comparison of Chains

[1]ROMAN MELNYK, [2]ROMAN KVIT
[1]Software Department,
Lviv Polytechnic National University,
12 Stepan Bandera St., Lviv, 79013,
UKRAINE

[2]Department of Mathematics,
Lviv Polytechnic National University,
12 Stepan Bandera St., Lviv, 79013,
UKRAINE

Abstract: - The known K-means clustering, flood-filling, and thinning algorithms are used to find coordinates of contacts in PCB images. Images of different types and colors are considered. The clustering algorithm is used to reduce the number of colors, to get uniform colors in the PCB image. The thinning algorithm is used to build skeletons and find pixels of contacts. The flood-fill algorithm is used to mark and separate chains, defect connectivity, and metal defects. Different subtraction operations are applied to original, transformed, and distributed cumulative histogram images.

Key-Words: - PCB, defects, chain, contact, flood-filling, thinning, clustering, subtraction, histogram, approximation

Received: April 24, 2022. Revised: April 18, 2023. Accepted: May 19, 2023. Published: June 26, 2023.

1 Introduction

There are many approaches for printed circuit board defect detection in the published articles. They are grouped into three classes. The first one unites works describing artificial networks for fault diagnostics. For example, the works, [1], [2], [3], use neural network tools. Works of the second class are based on subtraction algorithms as it is presented in the papers, [4], [5], [6], where defects are separated and classified. The article, [7], considers defects detection based on a connected table of a reference image. This table is often unknown, and the coordinates of contacts are to be determined by different methods.

Some works considering defects in PCB images and the method of their extraction are given in the surveys, [8], [9], [10]. In [11], different defect methods are divided into groups of image subtraction and feature extraction for the PCB image control. The algorithms in [12], [13], use parts of the board to increase the visibility of defects. Simple techniques to divide a plane are used. In the works, [14], [15], authors propose an algorithm for machine vision inspection systems to detect the defects short circuits, open circuits, and depressions defects in a PCB image. The work, [13], considers parallelization issues for sequential algorithms of thinning. Algorithms of thinning, clustering, and comparison are used in the work, [16], to find defective contacts and traces.

In this article three approaches are considered: different types of subtraction and reprocessing, thinning and flood filling to select and separate chains, measurement of traces to detect positions and intensity of PCB defects. Algorithms of K-means clustering and calculation of distributed cumulative histograms are applied. PCB samples for testing of algorithms are taken from [17], [18].

Many subtraction approaches give in result an image that is controlled by the operator. He can miss defective cases. The proposed approach is fully automatic based on the consequence of the developed algorithms.

Also, the subtraction operations are very sensitive to the deviation of the sizes of contacts and traces from the reference ones and give extra or lacking pixels that do not influence the normal work of the circuit.

2 Subtraction and Comparison Formulas

2.1 Logical Subtraction Operations

A simple way to isolate defects in a manufactured PCB image (denoted as B_m) is to subtract its binary form C from the corresponding binary form B_{2r} of the reference PCB image (denoted as B_r). But in this case, along with defects, the result contains manufacturing inaccuracies that often do not affect the functionality of the scheme.

The XOR operations with pixels are as follows:

$$0 \wedge 1 = 1, 1 \wedge 0 = 1, 0 \wedge 0 = 0, 1 \wedge 1 = 0,$$

where 0 denotes white and 1 denotes black or vice versa.

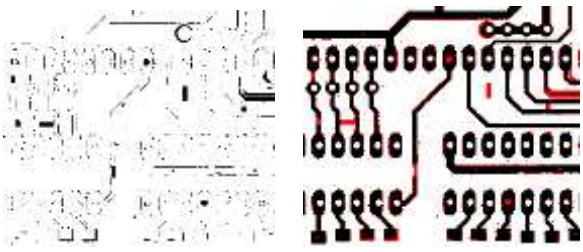
Applying XOR operations to the binary images of printed circuit boards considered below, the resulting image with marked defects is obtained, which is shown in Figure 1a. Manufacturing inaccuracies are mixed with various defects. Some defects are isolated from traces and contacts. It is difficult to remove defects from such an image. It is impossible to classify them on the received image.

The second approach involves attaching inaccuracies to objects on the board. For this, a modified OR logical operator is used. The modification is that the result of the two operations is marked with a red

$$0 \wedge 1 = 1 \text{ (red)}, \ 1 \wedge 0 = 1 \text{ (red)}, \ 0 \wedge 0 = 0, \ 1 \wedge 1 = 1,$$

where 0 denotes white and 1 denotes black (except for those mentioned) or vice versa.

Then, in the image in Figure 1b, [4], objects are black, and defects with manufacturing inaccuracies remain red.



a b

Fig. 1: Difference between two PCB images: defects and inaccuracies (a), objects, defects, and inaccuracies (b)

2.2 Mathematical Subtraction Comparison Formulas

To obtain a binary form segmenting and thresholding procedures are required. It is a more simple way to compare the input original images. Two images are compared according to the following comparison formulas

$$\begin{aligned} &I_r(x,y) = I_d(x,y) \\ &\forall \, | I_e(x,y) - I_d(x,y) | < Tol, \\ &I_r(x,y) = RGB(| I_e(x,y) - I_d(x,y) |, 00) \\ &\forall \, | I_e(x,y) - I_d(x,y) | \geq Tol, \end{aligned} \tag{1}$$

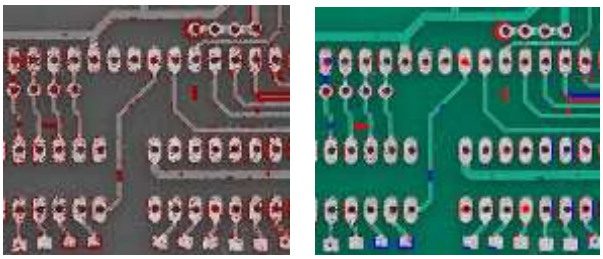
where $I_r(x,y)$ is the pixel intensity of the resulting image, $I_e(x,y)$ is the pixel intensity of the first image, $I_d(x,y)$ is the pixel intensity of the second image, Tol and is the tolerance value for controlling the difference between the pixel intensity of the reference and controlled samples.

The resulting image obtained by formula (1) is shown in Figure 2a. Pixels corresponding to positive or negative intensity differences are marked in red.

To account for the sign of the pixel intensity deviation, a refinement is used to specify the nature of the deviation

$$\begin{aligned} &I_r(x,y) = RGB(| I_e(x,y) - I_d(x,y) |, 00) \\ &\forall I_e(x,y) - I_d(x,y) \geq Tol, \\ &I_r(x,y) = RGB(0, | I_e(x,y) - I_d(x,y) |, 0) \\ &\forall I_e(x,y) - I_d(x,y) \geq Tol \ . \end{aligned} \tag{2}$$

The resulting image obtained by formula (2) is shown in Figure 2b. Pixels corresponding to positive intensity differences are marked in red, and negative intensity differences are marked in blue.



a b

Fig. 2: Difference between the two PCB images: red defects and inaccuracies (a), red and blue defects and inaccuracies (b)

All four approaches make it possible to isolate defects along with manufacturing inaccuracies. The latter distinguishes two groups of defects: excess

and missing bonding metal. The resulting image is planned to be controlled by the operator. It is difficult to interpret the results of the subtraction, which type of defects has occurred because it requires constant comparison with the input images. In this way, the operator can miss defective cases. Thus, a fully automated approach is preferable.

In addition, the subtraction operation is very sensitive to the deviation of pin and trace sizes from the reference image and sometimes gives extra or missing metal pixels that do not affect the normal operation of the circuit.

The mathematical comparison formulas are auxiliary and designed to visualize defects in individual circuits. It is difficult to interpret the results of the subtraction, which type of defects has occurred because it requires constant comparison with the input images.

In this sense, the following approach is more universal: for flood-filling, thinning, and other algorithms, the sizes of tracks and contacts are irrelevant. The main condition is the complete display of the physical board with an image.

3 Pre-processing Algorithms

3.1 Distributed Cumulative Histogram

To build the image of a distributed cumulative histogram two sets of ordinary histograms of the number N of columns and M rows are calculated

$$\begin{aligned} V_i(c) &= \{V_{ij}(c)\},\ j = 0,255, i = 1,N; \\ V_j(r) &= \{V_{ji}(r)\},\ i = 0,255,\ j = 1,M. \end{aligned} \tag{3}$$

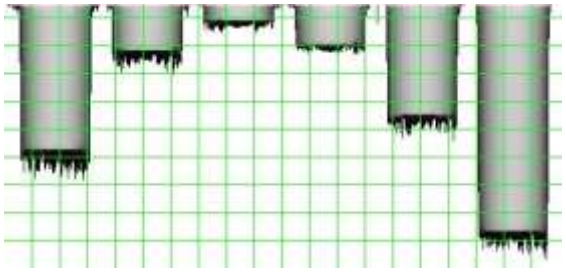
Then two distributed cumulative histograms assets of frequency sums

$$\begin{aligned} V_j(cc) &= \left\{ \sum_{l=0}^{i} V_{li}(r),\ i = 0,255 \right\},\ j = 1,N; \\ V_j(cr) &= \left\{ \sum_{l=0}^{i} V_{li}(r),\ i = 0,255 \right\},\ j = 1,M, \end{aligned} \tag{4}$$

where, $V_i(cr)$, $V_j(cc)$ are cumulative histograms in rows and columns $V_{ij}(c)$, $V_{li}(r)$ are intensity frequencies in a column and row, N, M are columns and rows numbers.

The last equations are the mathematical models of distributed cumulative histograms of an image. An example of such processing (3)-(4) is given in Figure 3, where the original image of "circles" is shown in Figure 3a, and its DCH image is shown in Figure 3b. The last image is covered by a grid designed to measure intensity.

a



b

Fig. 3: The "circles" image (a), a 2-D image of distributed cumulative histogram (b) covered by a grid

The advantages of the DCH image are the following: 1) the intensity of circles is automatically compared, 2) the intensity of every circle is automatically measured by horizontal lines of the green grid, and the intensity interval $0-255$ corresponds to the height of the grid, from down to the top on the OY axis, 3) the DCH image reflects a noise belonging to the surface of all circles and a noise of a white background.

3.2 K-means Clustering Algorithm

The K-means clustering algorithm approximates the PCB image by K segments of pixels with the same intensity.

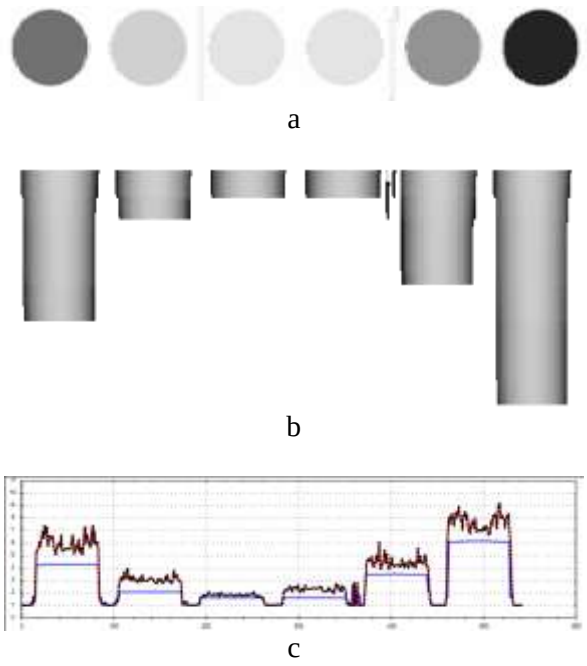
This algorithm assigns pixels into different clusters to minimize the sum of distances between the centroids and pixels within a cluster. The criterion function is as follows:

$$S = \sum_{i=1}^{m} \sum_{k=0}^{K} w_{ik} \mid I - \bar{I}_k \mid,\ S = \sum_{i=1}^{m} \sum_{k=0}^{K} w_{ik} [I - \bar{I}_k]^2, \tag{5}$$

where $w_{ik} = 1$ if a pixel of intensity I_{ir} belongs to the cluster k, otherwise, $w_{ik} = 0$, $\bar{I}_k$ is the intensity of the *i-th* centroid.

The DCH image serves as control information for processing and correction of the original image. An example of such correction by the clustering algorithm (5) is given in Figure 4. The original image is clustered by seven clusters (circles and

background). The resulting image of the clustered circles is given in Figure 4a. Its DCH image is shown in Figure 4b.



a b c

Fig. 4: The clustered "circles" image (a), its DCH image (b), and graphs of intensity (c)

4 Processing of Reference PCB Image

4.1 Clustering into a Binary Form

The first stage of the method includes the steps of selecting circuit chains and marking their coordinates. They include three algorithms: transforming to the binary form by the clustering algorithm, selection of chains by the flood-filling algorithm, and searching of specific points with the application of the thinning algorithm.

The application of the clustering algorithm is explained below. For illustration, the image histogram is calculated

$$h(i) = card\{(u,v) \mid I(u,v) = i\}.$$

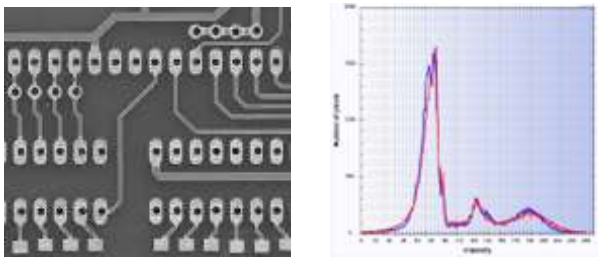
Also, the cumulative histogram is calculated:

$$H(i) = \sum_{j=1}^{i} h(j),$$

where $I(u,v)$ is the pixel intensity, and $h(i)$ is intensity frequency, $H(i)$ is the accumulating frequency for the given intensity.

The histogram of the input grey image from [19], in Figure 5a has three maximum values (Figure 5b). The first corresponds to the grey background and is relatively large. This representation of the PCB image is not suitable for the subtraction algorithm. Therefore, the image must be transformed. The approach from [19], applies three-interval segmentation to select wires, pins, and holes. After thresholding and redrawing, the selected segments are used for a subtraction algorithm to find defects.

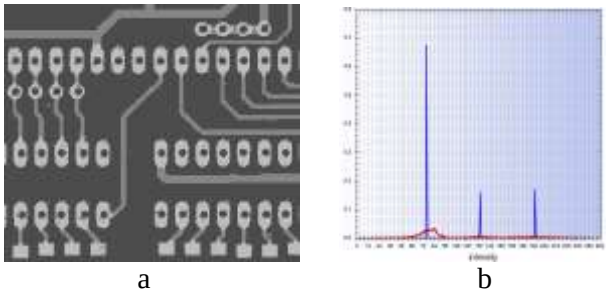
The weaknesses of this approach are the necessary algorithms for determining thresholds, the lack of defect types and coordinates, as well as the lack of any quality characteristics of defects (area, intensity, coordinates, etc.).



a b

Fig. 5: A PCB image (a) and its histogram (b)

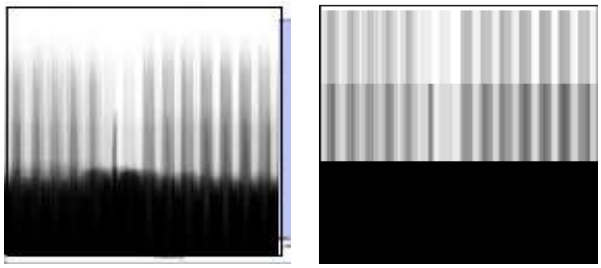
A clustering algorithm is applied to the image to obtain three segments ($K = 3$). The advantages of the approach are as follows: 1) no pixel is deleted, 2) pixels are grouped naturally near their centroids, 3) segment colors are uniform and their values do not affect the subsequent fill. The cluster image is shown in Figure 6a. Visually changes are not noticeable. Its usual histogram is shown in Figure 6b. For comparison, the very small graph is a red histogram of the original input image of the PCB.



a b

Fig. 6: A clustered PCB image (a) and its histogram (b) with original histogram

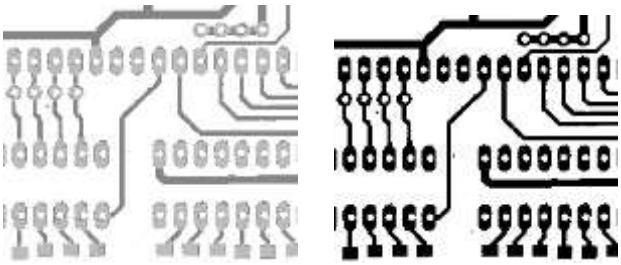
The DCH image distinctly illustrates how pixels are distributed in the input and clustered images. Figure 7b shows three groups of pixels corresponding to contacts, traces, and backgrounds.



a b

Fig.. 7: DCH images for input (a) and clustered images (b)

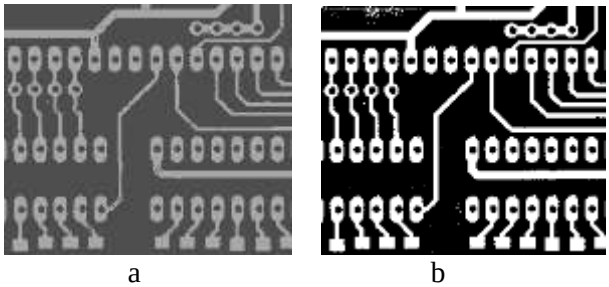
There are two ways of converting clustered segments into a binary form: 1) image segmentation by intensity levels from Figure 6b or Figure 7b, separating the required parts, or 2) sequentially filling the required part with white and all other parts with black. The transformation of the clustered image from Figure 4a is shown in Figure 8.



a b

Fig. 8: Two steps of flood-filling: a background with white (a) and all other elements with black (b)

The simplest case is when the clustering algorithm is applied to form two segments ($K = 2$). Two images are obtained depending on redrawing color. They are shown in Figure 9: two clusters in Figure 9a and this image after redrawing in Figure 9b.



a b

Fig. 9: Two clusters in the PCB image: grey (a), and after redrawing (b)

The binary PCB images in Figure 8b and Figure 9b are ready for further processing.

4.2 Separation of Connected Components

The reference image of the printed circuit board consists of the background and various components distributed on its surface. Components are isolated contacts, traces, and chains (contacts connected to traces). The actual PCB image may contain some isolated metal extra surfaces. The purpose of the developed algorithms is to separate these components and compare them instead of comparing whole images.

Hilditch's thinning algorithm, [8], works with binary (black and white) images. This tool is very important because it finds the pixel coordinates that are accepted as chain contacts. This is useful for locating, highlighting, and separating electrical circuits as connected components on a board. These are used as starting points for the filling algorithm to build and select chains in the reference and real PCB images for comparison.

Figure 10 shows that the skeleton image has three types of elements: chains with specific points placed (contacts and traces), circles without any specific points, and isolated specific points of noise or extra metal on the background. In general, the last elements should not be placed on the background because the image is made for the template PCB. If noticed, they must be deleted from the image in a manual or programmed way.

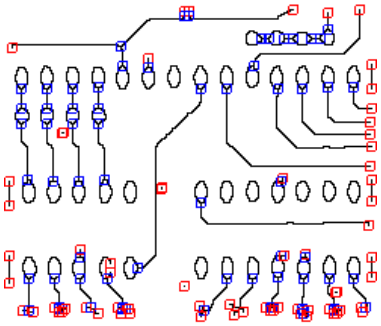


Fig. 10: Skeleton with endings and switches

Figure 10 shows that some circles in the skeleton have no specific points. They correspond to contacts without traces. It means that a simple way of thinning does not allow us to separate these components for comparison.

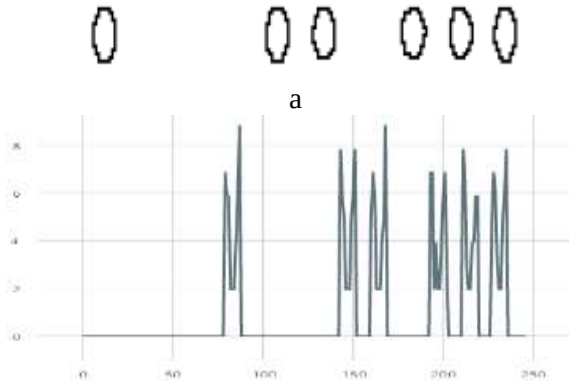
Therefore, another method is used. The circles separated from the skeleton are shown in Figure 10a.

To illustrate the basic idea, the custom histogram is calculated, but instead of the traditional probability/intensity axes, it uses probability/coordinates axes (OX or OY)

$$h(i) = card\{(i,v) \mid I(i,v) \in I(b) \div I(e)\},\ i = 0,1, \ldots, \text{width (height)}, \quad (7)$$

where $I(i,v)$ is the pixel intensity, $h(i)$ is the number of pixels, and $I(b)$, $I(e)$ are borders of the intensity interval of the counted pixels.

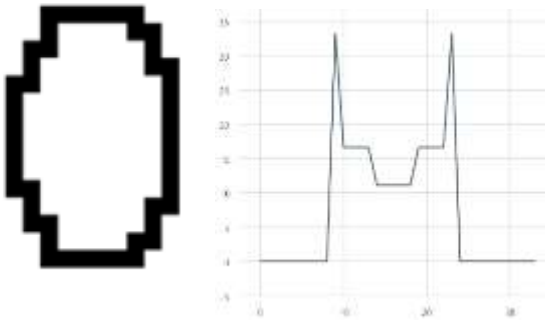
Histograms of black pixels in columns and rows of circles are calculated. Figure 11b shows the histogram for all circles by the axis OX. Figure 12 shows the increased image of one circle and its histogram by the axis OY.



a

b

Fig. 11: Circles image (a), a histogram by the OX axis (b)



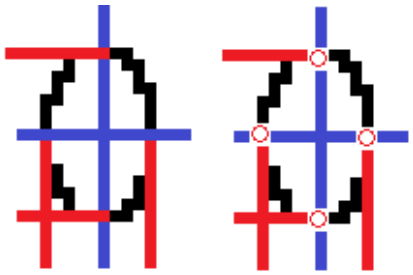
a b

Fig. 12: An image of one increased circle (a), and its histogram by the OY axis (b)

Two histograms have extrema values. For one circle every histogram has two maximums and one minimum, which are found in the graph with W points

$$h_{\max(\min)} = \max(\min) h(i), \quad 1 \le i \le W.$$

Three coordinates in the axis OY and three coordinates in the axis OX give coordinates of four points on the circle-skeleton. The process of their finding is shown in Figure 13: a - finding of lines, b - crossing lines. Blue lines correspond to minimum values in two histograms and red lines correspond to maximum values in two histograms. The intersection of two different lines gives the specific point on the circle-skeleton. They are four small circles in Figure 13b.



a b

Fig. 13: Circle-skeletons with lines from the axes OX and OY (a), and marked positions of specific points (b)

This procedure is applied for each component in which specific points were not found by the thinning algorithm. Why should it be done? The circle-skeleton of the reference image of the printed circuit board is placed in the middle of the component and the coordinates of their pixels with a high probability coincide with the corresponding pixels on the manufactured image of the printed circuit board. Thus, their respective components will be highlighted and separated. This procedure is required only when the board contains contacts without traces and their defects are unacceptable.

After finding the coordinates of endings and switches in the skeleton image the following step is to bind chains in the reference PCB image with these specific points. It is realized by applying the flood-filling algorithm. Arbitrary ending or switch pixels are taken as start points. The resulting PCB image for filled three chains is shown in Figure 14. All other chains are not flood-filled.

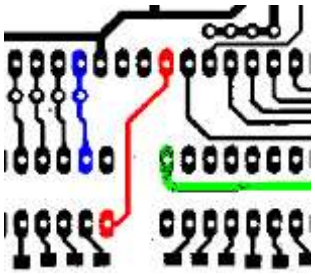


Fig. 14: The reference PCB image after flood-filling of three chains

An order in which chains are flood-filled gives us their ID numbers After flood-filling all endings and switches are with corresponding colors and ID numbers of chains to which they belong. Thus, a set

of pixels as specific points for the reference PCB image $E=\{E_1,E_2,...,E_n\}$ is formed, where E_i is a set of pixels belonging to the i *-th* chain. Every pixel is described by four parameters

$$p_{ij}=\{i,c_i,x_j,y_j\},$$

where i is the number of a chain, j is the number of a pixel in the set E_i, x_j,y_j are the coordinates of a pixel, c_i is the color of the i *-th* chain. Every set E_i contains two and more pixels. The set E_i also includes pixels found for isolated contacts through their skeletons.

4.3 Separation of Chains and Connectivity Defects Detection

All endings and switches are placed in the middle of contacts and traces. So, their positions are not very sensitive to the manufacturing process. And that is why they are accepted as the starting points to provide manipulations with objects in the manufactured PCB image.

As the first step a procedure to separate chains and determine open and short defects is considered.

1. Starting points are formed by logical projecting of the set E_e on the board of the manufactured PCB. Physical operations are not performed. Logical assignment operations are performed

$$D=:E,\{D_1,D_2,...,D_n\}=:\{E_1,E_2,...,E_n\},$$

where D is a set of specific points for the manufactured PCB image and D_i is a set of pixels for the i *-th* chain.

2. The filling algorithm is applied as many times as there are chains in the scheme. Taking elements from the set D_i as starting points the flood-fill algorithm fills chains with different colors selecting and separating them in the manufactured PCB image.

An example of the results of the filling algorithm is shown in Figure 15. Figure 15a shows two sets of starting points $D_1=\{B,C,D,H\}$ $D_2=\{A,E,F,G\}$ and two filled chains in the reference image. Figure 15b shows one correct chain and one chain with an open defect. Figure 15c shows two chains as one chain caused by a short defect. Two last examples refer to the manufactured PCB image.

3. Defects detection. To select the red chain with the open defect two iterations of flooding are used: for point B and point H. Two iterations indicate an available open defect. One color for two and more sets of starting points indicates an available short defect. Two blue chains with a short defect are flooded and selected from one starting point and one iteration.

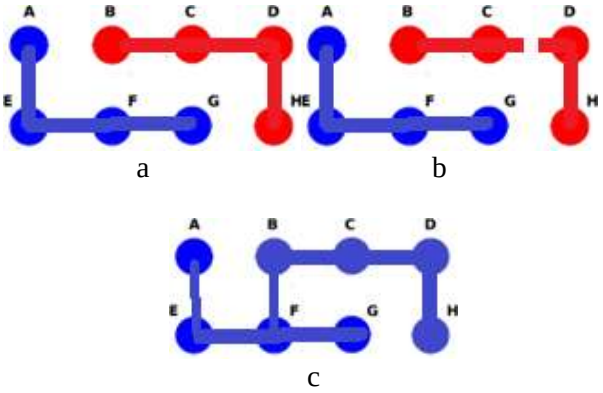


Fig. 15: Illustration of flood-filling and selection of chains: chains in the reference (a), open (b), short (c)

The resulting PCB image after three steps of flood-filling is shown in Figure 16. One step requires two iterations because the red chain has one open defect resulting in two components. In the other two steps, blue components are built from two chains having short defects between themselves.

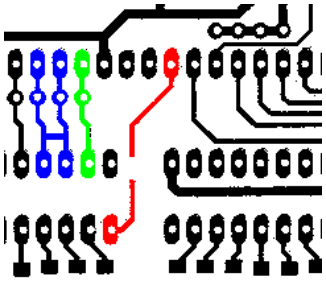


Fig. 16: The manufactured PCB image after three steps (four iterations) of flood-filling

When chains are selected, and separated they are grouped into three classes: correct, with an open defect, and with a short defect. The coordinates of the defects are not known. They are only visible for external inspection. To find their positions, two ways can be used: by application of the thinning algorithm to the manufactured PCB image or by comparison of two components from the reference and manufactured PCB images. These approaches are considered below.

4.4 Comparison of Mask Images

The binary presentation of the manufactured PCB image is taken for the thinning algorithm. In the

result, there are two sets of specific points: the first reference skeleton $E=\{E_1,E_2,...,E_n\}$ and the second skeleton $D=\{D_1,D_2,...,D_n\}$. Every element of these sets contains two and more pixels with their identification numbers and coordinates of detected specific points, for example, the first chain contains k pixels

$$E_1=\{p_{11},p_{12},...,p_{1k}\},\quad p_{ij}=\{i,c_i,x_j,y_j\}.$$

On the basis of these two sets, two so-called mask images are built: a mask C of the Etalon PCB and M_d the manufactured PCB. Every mask image has the dimension of the corresponding PCB image. Black circles or quadrates (in our case for switches blue and for endings red) are drawn on a white or transparent background. Centers of circles or quadrates coincide with the coordinates x_j,y_j of specific points. A radius of a circle or a quadrate reflects manufacturing precision. Its value is controlled by the user. Two examples of circles and quadrates on skeletons and not filled are shown in Figure 17.

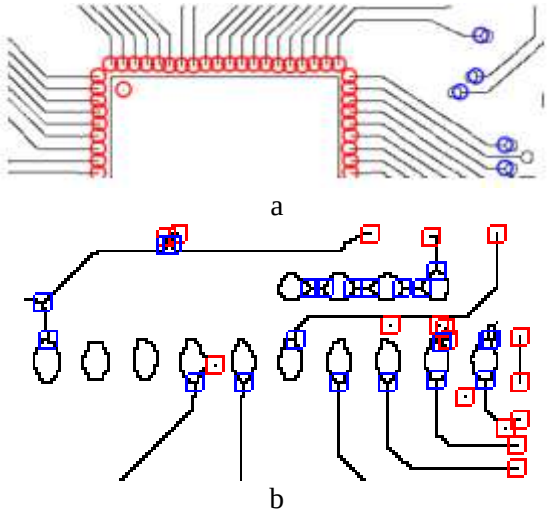


Fig. 17: Circles (a) and quadrates (b) marking specific points on skeletons

Then the circles and quadrates are separated from the skeletons and filled with the corresponding colors as it is shown in Figure 18. Thus, the mask image is built and prepared for the following manipulations.

Fig. 18: Flood-filled **c**ircles

Simple observation and comparison of the mask images do not answer how much they differ. The special comparison procedure (1) is used to compare two images.

The etalon mask is the output image. A pixel of this image changes its color if two corresponding pixels are with different intensities. A pixel becomes red or blue if an absolute value of intensity difference is greater than a tolerance value. Red for a positive difference and blue for a negative one. If the difference is within a tolerance value, the resulting pixel remains as it was before a comparison. For example, the separated chain with defects is shown in Figure19a. Its mask image with blue and red circles is shown in Figure 19b. The reference chain and its mask are not presented. Two mask images are accepted as operands in the comparison formula. The resulting image from the comparison operations is shown in Figure 19c.

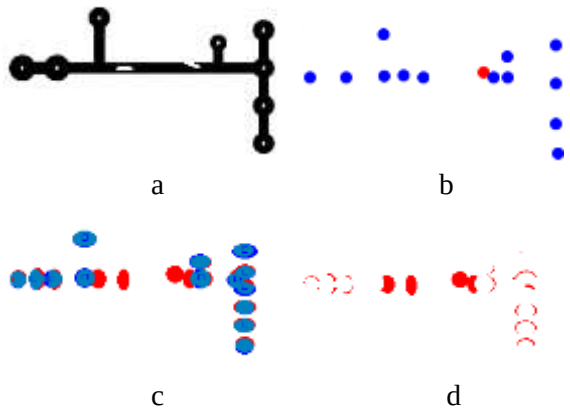


Fig. 19: Chain with defects (a), and its mask images: a full mask (b), difference (c), and with only defects (d)

Blue correct circles are remained for the illustration but are eliminated for the second illustration in Figure 19d. Red circles mark defects in the chain which are absent in the reference chain. Also, some red pixels mark inaccuracies in the positions of specific points. The reason is that the centers of circles on the etalon PCB and the real PCB masks do not coincide exactly. A distance between centers of corresponding circles on the etalon and real masks indicates a shift of the

concrete-specific points (corresponding contacts and traces)

$$D(b) = sqrt\left\{[x_b(r) - x_b(m)]^2 + [y_b(r) - y_b(m)]^2\right\} < Tol,$$

where b is the ID number of a specific point; r and m are indexes of the PCB images, Tol and is a tolerance value. Practically this parameter reflects several pixels by which the real PCB image is shifted from the etalon PCB image. It also indicates that the real PCB is defective and must be removed.

The mask images are imposed on the original manufactured PCB image to underline found and suspicious defects that are demonstrated in the example in Figure 20.

a b

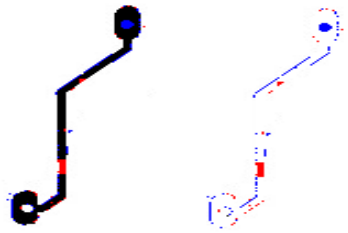
Fig. 20: Overlay of the defective chain with the full comparison image (a) and only marked defects (b)

Such visual checking is realized for every chain having a total number of specific points greater than the corresponding number in the reference chain. The following approach is developed to determine the intensity of defects and their coordinates.

5 Defects Detection by Measurement of their Sizes

The previous methods separate every chain from the reference and manufactured PCB image marking places of defects and their types: short or open. The following approach widens the opportunity for researchers to detect other types of defects and ways to measure their intensity.

For defects visualization, the comparison formula is used for every chain. For the input images as the defected chain and reference chain, the comparison image is with positive (red) and negative (blue) defects and shown in Figure 21a. If required, defects can be separated by segmentation or flood-filling of the black body of the chain (Figure 21b). Defects have their concrete coordinates connected with the considered chain.



a b

Fig. 21: Comparison image of two chains (a) and their segmented version (b)

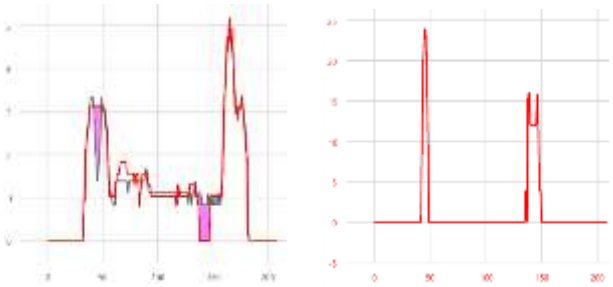
To decide on whether defects are critical for circuit functionality their intensity must be measured. After measurement, they are compared with the corresponding characteristics of the reference sample.

Surface histograms of pixels that form the previous chain or defects are calculated by the following formula:

$$h(i) = card\{(i,v) \mid I(i,v) \in I(black)\},\ i = 0,1,...,H,$$

where H is the height of the image, $I(i,v)$ is the pixel intensity, $h(i)$ is the number of pixels, and $I(black)$ is intensity of black pixels.

Their graphs are shown in Figure 22.

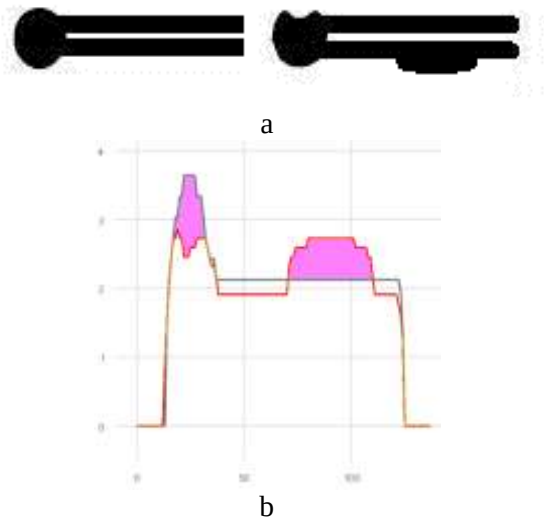


a b

Fig. 22: OY Histograms of two chains (a) and of defect pixels (b)

Figure 22a shows histograms OY of the defective (red) and reference (blue) chains. Two marked areas indicate a difference between the numbers of pixels in places of defects. Extra pixels correspond to higher values; a wire break corresponds to zero value. The second chart in Figure 22b reflects blue and red pixels of defects in the image received after comparison.

In this way, all inaccuracies in the components of PCB can be measured for deciding whether the circuit is ready for exploitation or not. Two other examples are shown in Figure 23. The graphs demonstrate extra and lack of pixels and thinner traces in the defective component.



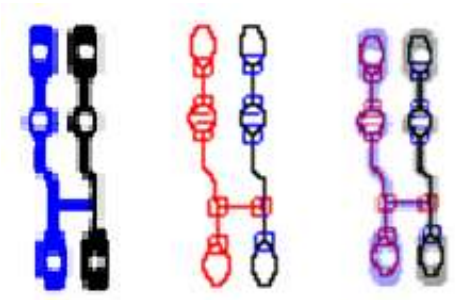
a b

Fig. 23: Two components (a) and their *OX* histograms (b) with marked defect regions

The marked area of the difference between the two graphs is measured automatically. The difference can be caused not only by defects but not equal wides of traces, shifts of contacts, and traces. So, the user or special program system must take a decision.

Defects of connectivity are detected by two previous approaches: during the separation of a chain having these types of defects and by comparison of mask images. Visualizing short defects also is useful in the approach of comparison of reference and defective chains.

Figure 24 shows three cases: the comparison image of two chains (with a short and a reference), their skeleton with marked specific points, and the overlay of two previous images. In the first image, the blue part consists of one chain and the shorting line.



a b c

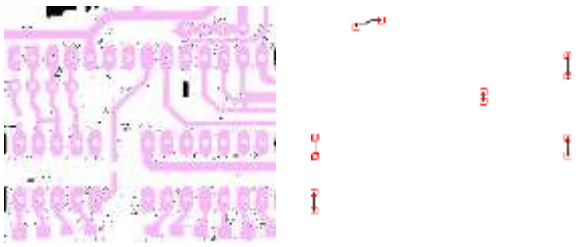
Fig. 24: Defective chains (a), skeleton (b), and overlay of them (c)

Measurement of defects intensity, visualization, and determination of their coordinates helps to divide printed circuit boards into working, defective, and subject to repair.

6 Separation of Defects of Extra Metal

The previous approach works with all chains defined in the control image. In the picture of the printed circuit board, separates each circuit on the board and specifies it as one of four possible types: correct and the same as in the control circuit; chains of an irregular shape or with a defective metal filling, but conduct signals; chains with a break defect; short circuits.

For further analysis of the image of the printed circuit board after separation, identification of possible defects, and classification, the circuit as an object of consideration is removed from the produced image. This operation is very simple: one of the contact pixels is taken as the starting point for the filling algorithm. The target color is white. After this operation, the manufactured PCB image will be fully white or with black inclusions of extra metal on the background. In Figure 25a all checked components instead of removed are marked with pink for better presentation. Black extra metal traces remained in the image. For measuring their intensity access to them is realized through their specific points in Figure 25b.



a b

Fig. 25: Extra metal and removed chains (a) and specific points (b)

Measurement of these residues is required by the user to decide the robustness of the PCB.

7 Conclusion

This approach allows us automatically to inspect all open and short defects visible to a camera and unnoticeable to the user. The K-means clustering algorithm, flood-filling, thinning, and distributed cumulative histograms are used to transform PCB images to a binary form with uniform distribution of colors, to find the coordinates of points identifying every chain. The algorithms select and separate every chain in the reference and real images, detect

defects of connectivity during the separation of chains, measure the intensity of defects for chains with extra and lack of metal filling, and visualize all defects for every chain or the whole image.

A visualization approach is based on different types of subtraction and comparison operations. Overlaying images are used to project-specific points on defective chains. It helps to indicate places the wire breaks, short circuits, and other types of defects.

A large number of algorithms are developed as participants in the process of the detection of defects. To build a robust program system of printed circuit board inspection by their images they must be united in four modules: preprocessing module, open and short defects detection module, the module for the detection of defective traces, and the module to decide the robustness of the circuit.

References:

[1] Jungsuk Kim, Jungbeom Ko, Hojong Choi and Hyunchul Kim, Printed Circuit Board Defect Detection Using Deep Learning via A Skip-Connected Convolutional Autoencoder, Sensors, No.21(15), 2021, 4968.

[2] S. McClure. Extracting and Classifying Circuit Board Defects using Image Processing and Deep Learning, towardsdatascience.com, Feb. 4, 2020 [Online]. Available: https://towardsdatascience.com/building-an-end-to-end-deep-learning-defect-classifier-application-for-printed-circuit-board-pcb-6361b3a76232.

[3] 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, No.9, 2020, pp. 1547.

[4] Vikas Chaudhary, Ishan R. Dave and Kishor P. Upla, S. V., Visual Inspection of Printed Circuit Board for Defect Detection and Classification. International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 2017, pp. 732-737.

[5] Y. Hanlin and W. Jun, Automatic Detection Method of Circuit Boards Defect Based on Partition Enhanced Matching, Information Technology Journal, Vol.12, No.11, 2013, pp. 2256-2260.

[6] Zhu, A. Wu and X. Liu, Printed circuit board defect visual detection based on wavelet denoising, IOP Conference Series: Materials Science and Engineering, Vol. 392, 2018, pp. 062055.

[7] M. H.Tatibana, R. and de A. Lotufo. Novel Automatic PCB Inspection Technique Based on Connectivity, Proceedings X Brazilian Symposium on Computer Graphics and Image Processing, 1997, pp. 187-194.

[8] M. Moganti, F. Ercal, C. H. Dagli, and S. Tzumekawa, Automatic PCB inspection algorithms: A review, Computer Vision and Image Understanding, Vol.63, No2, 1996, pp. 287-313.

[9] D.B. Anitha, and M. Rao, A survey on Defect Detection in Bare PCB and Assembled PCB using Image Processing Techniques, International Conference on Wireless Communications, Signal Processing and Networking, 2017, pp. 39-43.

[10] K. P. Anoop, N.S. Sarath and V. V. Sasi Kumar, Review of PCB Defect Detection Using Image Processing, International Journal of Engineering and Innovative technology (IJEIT) Vol.4, Is.11, 2015, ISSN: 2277-3754.

[11] J. Nayaka, K. Anitha, B.D. Parameshachari, R. Banud and P. Rashmi, PCB Fault Detection Using Image Processing, IOP Conference Series: Materials Science and Engineering, Vol.225, 2017, pp. 1-5.

[12] Y. Hanlin, W. Jun, Automatic Detection Method of Circuit Boards Defect Based on Partition Enhanced Matching, Information Technology Journal, Vol.12(11), 2013, pp. 2256-2260.

[13] F. B. Nadaf and V. S. Kolkure, Detection of Bare PCB Defects by using Morphology Technique, Morphology Technique International Journal of Electronics and Communication Engineering. Vol.9, No.1, 2016, pp. 63-76.

[14] S. Guan, F. Guo, A New Image Enhancement Algorithm for for PCB Defect Detection, International Conference Intelligence Science and Information Engineering (ISIE), 2011, pp. 454-456.

[15] J. P. R. Nayak, K. Anitha, B. D. Parameshachari, R. Banu, and P. Rashmi, PCB Fault Detection Using Image Processing, IOP Conference Series: Materials Science and Engineering, Vol. 225, 2017, pp. 012244.R. Melnyk, D. Hatsosh, and Y. Levus, Contacts detection in PCB image by thinning, clustering, and flood-filling, IEEE 16th International conference CSIT 2021, 2021, pp. 370-374.

[16] W. Huang and P. Wei, A PCB dataset for defects detection and classification. arXiv preprint arXiv:1901.08204, 2019.
[17] S. Miles, "Computer Chip Showing High Tech And Circuit," 123RF.com. [Online]. Available: https://www.123rf.com/photo_40192468_computer-chip-showing-high-tech-and-circuit.html.
[18] "Circuit Board Drawing," Circuit Board Drawing at Painting Valley.com, Explore a collection of Circuit Board Drawing. Available [Online]: https://paintingvalley.com/circuit-board-drawing.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed to the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare.