

TITLE OF THE INVENTION

METHOD AND SYSTEM FOR DETECTING DEFECTS
ON A PRINTED CIRCUIT BOARD

5

FIELD OF THE INVENTION

The present invention relates to inspection systems and
to methods for detecting defects on electronic circuits such as printed
10 circuit boards. More specifically, the present invention is concerned with
such systems and methods based on edge detection.

BACKGROUND OF THE INVENTION

15 Electronic circuits may take many forms, such as PCB
(Printed Circuit Boards), lead frames, and hybrid circuits. These circuits
usually include a plurality of components, for example conductors, holes,
pads, dielectrics, photopolymer resist film, etc. These components may
be assembled on layers that are mounted on top of each other. An
20 example of such multi-layer assembly is the well-known conductors-holes-
glaze circuit structure. In the following description we will refer to printed
circuit boards (PCB) only as an example. However, it is to be understood
that the present invention is not limited to that embodiment of electronic
circuits.

25

In the past, inspection of PCBs has been performed visually
by a person who would have looked at a circuit, using a magnifier, trying

to find differences with an acceptable circuit model. Drawbacks of visual inspections are numerous: subjectivity, slowness, difficulty to collect quantitative information on defects, etc.

5 Automated methods of inspection have recently emerged following the increase of computer processing speed. Most of those automated methods consist in taking a digital image of the PCB to be inspected and analyzing this digital image to determine the presence of defects. Examples of such automated methods are those based on the
10 detection of edges.

An edge is a segment of a component's contour. It can be visually distinguished on a digital image as variations of hue from one component to the next, assuming that transitions between components (or
15 regions) have optically different characteristics. Edge based detection methods consist in characterizing and analyzing edges on a PCB image and by comparing such edges to known values and criteria.

Figure 1 of the appended drawings illustrates examples of
20 defects that can be found on a PCB or on other types of electronic circuits. The defects may be, for example, a bad overlap 10 between two layers of the PCB, a misalignment of two layers 12, a gap too small 14 between two adjacent tracks, a bridge 16 between two components, a track too narrow 18 or a broken track 19.

25

An example of an edge-based method for detecting defects on an electronic circuit is described in United States Patent No. 4,570,180,

untitled "Method For Automatic Optical Inspection", issued on February 11, 1986 and naming Baier *et al.* as inventors. This patent relates to a method and an apparatus for automatic optical inspection of a substantially two-dimensional pattern using digital image processing techniques. The method includes a first step in which grey level digital images are scanned for edges or lines and marking those edges in image storage. Then all non-marked regions in the image storage are scanned and tested for permissible grey levels.

10 Since anomalies are tested only by comparing non-marked regions to permissible grey levels, a drawback of Baier's method is that the quality of the detection is too dependent on the quality of the digital images. Indeed, the digitized image of an object may present variation in feature sizes and locations depending on the digitizing algorithms, magnification differences, etc.

20 Another problem of Baier's method is that, approximately the same computing power is required to test all pixels on the digital picture that do not correspond to edges, leading to waste of both computing resources and time.

25 Still another drawback of Baier's method is that it can be difficult to characterize a detected defect. Such characterization can be useful to find emerging error patterns in the electronic circuit's manufacturing process.

In United States Patent No. 5,452,368 issued on September 19, 1995 and entitled "Method of Detecting Defects In Semiconductor Package Leads", LeBeau teaches a method for detecting defects in objects by comparing a first grey level image of a first object to a second
5 grey level image of a second object. More precisely, the edge features of the first image are skeletonized and compared to the dilated edge features of the second image and vice versa. Contrary to Baier's method, LeBeau's does not test non-edge-related regions on the digital image.

10 However, a drawback of LeBeau's method is that defects are searched by comparing two images of objects that can be inflected by very similar defects positioned at about the same place. LeBeau's method does not allow for detection of such defects. This is a major drawback since a faulty manufacturing process can cause such repeated defects.

15 Accordingly, it is desirable to have a method and a system that allows for the detection of defects on an electronic circuit that does not rely on the quality and resolution of the digitized image of the electronic circuit and that does not interpret grey level variations as
20 defects.

It is also desirable to have a method and a system to detect defects that make use of design data.

25 It is further desirable to have such methods and systems that provide two levels of inspection, one to detect anomalies and another

to inspect these anomalies for defects, helping maximize the processing speed.

SUMMARY OF THE INVENTION

5

More specifically, in accordance with the present invention, there is provided a method for detecting surface defects on a Printed Circuit Board (PCB) having at least one layer including components, the method comprising:

- 10 providing a digital image of the PCB;
 identifying edges on the PCB image;
 providing a corresponding computer model for each component on at least one layer of the PCB;
 detecting anomalies on the PCB image by comparing said
15 identified edges to the computer model; and
 for each detected anomalies, determining if the detected anomaly corresponds to a surface defect.

 According to another aspect of the present invention, there
20 is provided a system for detecting surface defects on a PCB; the system comprising:

- a computer including a model of the PCB and being configured to identify edges on a PCB images, detecting anomalies on the PCB image by comparing the identified edges to the computer model, and
25 for each detected anomaly, determining if the detected anomaly corresponds to a surface defect;

an illumination assembly connected to the computer to provide illumination on the PCB;
a frame grabber connected to the computer;
a camera connected to the frame grabber to take an image
5 of the PCB; and
a positioning system connected to the frame grabber.

It is to be noted that the term "PCB" should be construed herein as any electronic circuit that may include visually distinguishable
10 surface defects.

Other objects, advantages and features of the present invention will become more apparent upon reading of the following non restrictive description of preferred embodiments thereof, given by way of
15 example only with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

In the appended drawings:
20

Figure 1, is schematic view illustrating examples of defects on a printed circuit board (PCB);

Figure 2 is a block diagram of a system for detecting surface
25 defects on a PCB according to an embodiment of the present invention;

Figure 3 is a flow chart of a method for detecting surface defects on a PCB according to an embodiment of the present invention;

5 Figure 4 is a flow chart of the contour-identifying step of Figure 3;

10 Figure 5 is a schematic view of both images of the PCB and of a computer model therefrom, illustrating the selection of reference points for their alignment;

Figure 6a is a schematic view of superimposed polygons from a three layer model of a PCB;

15 Figure 6b is a schematic view of the polygons of Figure 6a after subtraction of the following layers;

Figure 6c is a schematic view of the intersections of the polygons of Figure 6a;

20 Figure 7a is a schematic view of the polygons of Figure 6a, illustrating a narrow region;

Figure 7b is a schematic view of the polygons of P1 and P3 from Figure 7a, after dilatation;

25 Figure 7c is a schematic view of the results of the superimposition of the polygons of Figure 7a;

Figure 8 is a schematic view illustrating the detection of missing contours on a region of a PCB;

5 Figure 9 is a schematic view illustrating the definition of two active half-planes in the step of detecting a defect among anomalies of Figure 3;

10 Figure 10 is a schematic view illustrating the measurement of the distance between two segments when the segments are not parallel;

 Figure 11 is a schematic view illustrating the measurement of a cumulative interval between two parallel segments;

15 Figure 12 is a schematic view illustrating the determination of an Anomaly Analysis Region (AAR);

20 Figures 13a, 13b and 13c are a flow chart of a method to determine if an anomaly violates the allowed width or spacing;

 Figure 14 is a schematic view illustrating defects in a narrow space region;

25 Figure 15 is a decision tree to detect defects in a narrow space region;

Figure 16 is a schematic view of a polygon from a vectorial model, illustrating the erosion process; and

Figure 17 is a schematic view of the polygon of Figure 16
5 after the erosion process.

DESCRIPTION OF THE PREFERRED EMBODIMENT

Generally stated, a method for detecting surface defects on
10 a PCB according to embodiments of the present invention includes 1) identifying the edges on a digital image of the PCB, 2) detecting anomalies by comparing the identified edges to a computer model of the PCB and 3) determining if the detected anomalies correspond to a surface defect by a characterization of each anomaly.

15

A method according to the present invention therefore presents two levels of inspection: a first level that allows to rapidly identify anomalies and which requires relatively little computing time, and a second level, more refined, that compares identified anomalies to design
20 specifications so as to characterize detected defects.

To implement the method, a digital image of the inspected PCB and a computer model of a similar PCB must be provided.

25 Referring to Figure 2 of the appended drawings, a system 20 for detecting surface defects on a PCB, according to an embodiment of the present invention, will be described.

Generally stated, the system 20 allows to gather an image of a PCB under inspection and to detect possible defects using a method according to the present invention, as will be described hereinbelow in more detail.

The system 20 comprises a computer 22, a frame grabber 24 and an illumination assembly 26, both connected to the computer 22, a camera 28, connected to the frame grabber 24 and a positioning system 30, connected to the frame grabber 24 via an axis controller 32 and a servo controller 34.

The computer 22 is advantageously in the form of a personal computer that is so configured as to store both the image taken by the camera 28 of the PCB to be inspected and a computer model of a similar PCB. The computer 22 is also advantageously programmed so as to control the illumination assembly 26 and to do the computations required to analyze the image and, using a method according to the present invention, assess whether the PCB under inspection includes surface defects.

The illumination assembly 26 allows the regulation of the light intensity on the PCB under inspection.

The frame grabber 24 is in the form of a card that interfaces the camera 28 and the computer 22. The frame grabber 24 can advantageously operate in either one of two modes. According to a first

mode, the frame grabber 24 captures an image line by line, and sends each line to the computer 22. In a second mode, the frame grabber 24 captures a full image of the PCB at once before sending the image to the computer 22.

5

An optional conventional image-processing card 36 advantageously interfaces the frame grabber 24 and the computer 22 and allows faster processing of the image taken by the camera 28. When the card 36 is not included, the computer 22 is configured to process the
10 image.

The camera 28 allows digital images of PCBs to be taken. The camera 28 is advantageously in the form of a CCD (Charge-Coupled Device) camera that may operate in linear or in matrix mode. The
15 resolution of the camera 28 may vary according to the nature of the PCB to be inspected and the precision required.

The use of a linear CCD camera requires that the positioning system 30 scan the PCB in the direction perpendicular to its
20 pixels. However the resolution obtained in the perpendicular direction may be practically infinite. Its line resolution is about 8000 pixels.

Matrix CCD cameras have a resolution of about 4000 x 4000 pixels. The use of such cameras is advantageous in sparse regions
25 of a PCB. In this case, the positioning system 30 may be configured so as to move from region to region.

The positioning system 30 includes axles and servomotors (not shown) that allow the movement of supports on which the PCB lies.

5 The axis controller 32 allows synchronizing the motion of the positioning system 30 with the requirements of the inspection system 20.

The servo controller 34 controls the positioning system 30 and the retroactive loop that regulates the amplitude of the power current required to energize the electric motors (not shown) of the positioning
10 system 20.

Since frame grabbers, servo controllers, positioning systems and CCD cameras are believed to be well known in the art, they will not be described herein in further detail.
15

Of course, the system 20 may have other configurations without departing from the spirit and nature of the present invention.

The system 20 may obviously be part of a production line.
20

Turning now to Figure 3, a method 200 for detecting surface defects on a PCB or on another electronic circuit, according to an embodiment of the present invention, will now be described in more detail.

25 In step 202, a digital image of the PCB is provided to the computer 22. The image is advantageously in the form of a bitmap having grey levels. The number of levels of grey may vary according to the

precision required. According to a most preferred embodiment of the present invention, the system 20 produces and analyses digital images having 256 grey levels. Alternatively, the initial image may have other formats and a conversion step may then be added to provide a bitmap
5 image.

In step 204, contours are identified on the PCB image. Step 204 produces an Actual Contour Chart (ACC) which is a representation of contours on the PCB image. As will be explained hereinbelow, the
10 positions of the contours are computed at a greater resolution than the resolution of the PCB image as taken by the system 20. Step 204 is summarized in Figure 4.

In sub-step 204a (Figure 4), the luminosity of the PCB
15 image is corrected. More specifically, the luminosity (pixel intensity) of the image is advantageously normalized using the following equation:

$$lum = 256 \times (lum - LUM_MIN) / (LUM_MAX - LUM_MIN) \quad (1)$$

where lum is the luminosity at each pixel of the image;
20 LUM_MIN is the average luminosity in the darkest region of the image; and
 LUM_MAX is the average luminosity in the brightness region of the image.

The above equation allows for distribution of the pixel intensity between *LUM_MIN* and *LUM_MAX* so as to cover all the 256 grey levels.

5 Obviously, the above equation may be corrected according to the number of grey levels of the image. Other normalization equations may also be used.

10 Although sub-step 204a is optional, it has been found to be advantageous to correct the luminosity to increase the efficiency of the thresholding sub-step (204d).

15 It is believed to be within the reach of a person skilled in the art to conceive an algorithm to determine the darkest and brightness region.

In the optional sub-step 204b, the PCB image is filtered so as to reduce noise caused by the numerization of the image. The following gaussian filter is advantageously applied on a 3 by 3 pixel window (eight-connected window):

$$\begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix}$$

25 The gaussian filter may be applied on a pixel window having another dimension. Other types of filter may also be used.

In sub-step 204c, the numerical gradient is computed on the eight-connected window using a conventional Prewitt method. Alternatively, another method may also be used. The intensity of the gradient is saved in a new image. Since the Prewitt method is believed
5 to be well known in the art, it will not be described herein in more detail.

The gradient image is then compared to a predetermined threshold to identify pixels in regions of the image where the luminosity changes rapidly (sub-step 204d). These regions usually correspond to
10 pixel tracks overlapping the contours in the image.

In sub-step 204e, a first approximation of the edges is computed by skeletization of the pixel tracks identified in sub-step 204d. Indeed, every edge on the image corresponds to a chain of connected
15 pixels, each having at least one neighbour in an 8-connected window. The skeletization process allows rendering of the pixel chain to a track having one pixel wide. The results are advantageously saved in an image file (ACC_{image}) having the same dimension as the initial image but having only binary value, i.e. 1 for pixels belonging to an edge and 0 for the other
20 pixels.

In the sub-step 204f, every chain of pixels of the ACC_{image} is listed as co-ordinates of pixels that form the ACC. These co-ordinates are only first estimates of the real co-ordinates, as will be explained
25 hereinbelow.

More precisely, the chain is followed in counter-clockwise direction and the co-ordinates of every consecutive pixel are saved in a file: the ACC. Therefore the ACC contains a series of vector, each representing the co-ordinates of one of the contours of the image.

5

Each contour is then analyzed pixel by pixel so as to estimate the position of the passage by zero on the second derivative, which has been found to be a reliable indication of the co-ordinates of an inflection point of the luminosity. The corresponding pixels are supposed
10 to be part of contours since an inflection point in the luminosity profile is an indication of a transition between two materials.

More precisely, the process consists first in finding an estimate of the direction perpendicular to the contour at each pixel. This
15 is achieved first by doing a line fit at the current pixel position, including on the same contour the 2 pixels before and the 2 pixels after. The line fit indicates the angle of the contour tangent at the pixel position. The perpendicular direction is obtained by adding 90 degrees. Then the luminosity profile which lies along that direction and which crosses the
20 current pixel is extracted in the initial PCB image, as originally taken by the system 20. The second derivative is computed on the luminosity profile. The passage by zero of the second derivative corresponds to the inflection point of the luminosity profile. The co-ordinate of this point is saved in the ACC, replacing the first estimate of the co-ordinate.

25

Of course, other numerical method may also be used to find the estimate of the direction perpendicular to the contour at each pixel.

5 It is believed to be within the reach of a person skilled in the art to conceive other methods of extracting contours from the digital image of a PCB under inspection.

As will be explained in more detail in step 208, anomalies
10 are detected on the PCB under inspection by comparing the contour positions saved in the ACC and in the ACC_{image} with the contour positions as designed. Indeed, a contour or part of a contour is suspect on the ACC if its position is different from its designed position.

15 In step 206 of the method 200 (Figure 3) a computer model of the PCB under inspection is then provided. The computer model advantageously includes a vectorial model, in the form of polygons, of every layer with contours of components. In addition to being an adequate representation of contours, polygons have been found to be an
20 efficient model for comparison with the ACC, as will be described hereinbelow.

A conventional Gerber model of the PCB is advantageously converted in a vectorial model where every component on each layer is
25 represented as a polygon. Indeed, even an element including circle arcs may be represented as an adequate vectorial model if arc edges are represented by short vertexes. Alternatively, of if a Gerber model of the

PCB is not available, other multi-layer models of the PCB may also be used to extract the relevant information.

Before comparing the vectorial model of the PCB with the
5 ACC and the ACC_{image} , transformations may be performed on the vectorial model so as to align the model with the image. Indeed, during fabrication of the PCB, the layers are conventionally applied at different moments by different machines and the alignment of the layers may slightly differ. It is then preferable to have a multi-layer model of the PCB and to realign the
10 layers of the model according to the PCB under inspection thus yielding a better model of the PCB.

For every layer, predetermined points are selected as a reference for the alignment process. As can be better seen from Figure 5,
15 showing the model of a layer 32 and a PCB image 34, it has been found advantageous to select intersection 40 between horizontal 36 and vertical 38 segments as reference points. Obviously, regions of a layer not covered by elements from another layer provide more reliable reference points. It is to be noted that the PCB image 34 includes
20 components, such as components 42 that are not present on the single layer model 32 since the PCB image 34 includes all the layers.

Once the list of reference points is established, the following transformation can be applied on every point of a layer to align it
25 according to the reference points on the PCB image:

$$u = \frac{a_u \times x + b_u \times y + c_u}{dd \times x + ee \times y + 1} \quad (2)$$

$$v = \frac{a_v \times x + b_v \times y + c_v}{dd \times x + ee \times y + 1} \quad (3)$$

- 5 where (u,v) are the co-ordinates in the PCB image,
 (x,y) are the co-ordinates in the model of the layer,
 and a_u, b_u, c_u, dd and ee are coefficients that allow to
 compensate for rotation, translation, scale, and perspective between the
 PCB image and the model. These coefficients may be different for each
 10 layer.

The reference points are advantageously selected before the inspection of PCBs to minimize delays in computation.

- 15 Once all the layers of the model are aligned, they are all superimposed.

- Since all layers of the model include a plurality of polygons, the superimposition of 2 layers consists in superimposing each polygon of
 20 the first layer with all the polygons from the second layer. If the model includes, for example, a third layer, the polygons from the third layer are superimposed with the new polygons resulting from the superimposition of the first two layers.

This superimposition process is illustrated in Figures 6a to 6c.

Figure 6a shows a three-layer model including three polygons P1, P2 and P3, one on each layer. The superimposition consists in superimposing the polygon P3 (Pv3) on the results of the superimposition of the polygon P1 and P2.

The superimposition of P2 over P1 yield:

$$Pv1=P1-P2 \quad (4)$$

$$Pv2=P2. \quad (5)$$

After the superimposition of P2 over P1, P2 is intact while P1 must subtract the part of P2 that overlaps. The superimposition of P3 on the superimposition of P2 and P1 yields (see Figure 6b):

$$Pv1=P1-P2-P3 \quad (6)$$

$$Pv2=P2-P3 \quad (7)$$

$$Pv3=P3. \quad (8)$$

In some cases, the subtraction of 2 polygons produces independent polygons. Pv1 and Pv1 can be seen as lists of polygons. When there is more then one polygon by layer, P1, P2 and P3 can also be seen as lists of polygons.

The superimposition process allows pointing out overlapping regions. A polygon on an overlapping region must be updated after the superimposition of a new layer.

5 As can be better seen on Figure 6c, O21 is the resulting polygon resulting from the intersection of P1 and P2 after the superimposition with the P3.

10 Again, since the intersection of two polygons or the difference between two polygons may yield more than one polygon, O21 must be interpreted here as a list of polygons. This list may include no polygon if no polygon overlaps between layers 1 and 2.

15 The superimposition process described above also allows keeping track of narrow regions, as will be described hereinbelow. An example of a narrow region is region T31 in Figure 7a. Region T31 is extracted as follows.

20 As can be seen from Figure 7b, the polygons P1 and P3 are dilated to yield $D(Pv1)$ and $D(P3)$. The intersection therebetween is then found to yield the polygon $D31$ that obviously includes the region that is to be extracted. The polygon $O31$ is dilated to yield $D(O31)$ that is subtracted from $D31$ to yield the narrow region T31 (see Figure 7c).

25 The operation illustrated in Figures 7a to 7c is summarized by the following equation:

$$T31 = (D(P3) \wedge D(Pv1)) - D(O31) \quad (9).$$

Here again, T31 can be seen as a list of polygons for the following two reasons: the intersection of two polygons may yield more than one polygon and D(031) may also yield more than one polygon.

5

To summarize, the superimposition of the layers produces the following three results:

$$Pv_{k+1} = (Pv_k - P_{k+1}) \cup P_{k+1} \quad (10),$$

the list of visible polygons;

10

$$O_{k+1} = (O_k - P_{k+1}) \cup (P_{k+1} \cap Pv_k) \quad (11),$$

the list of polygons from overlapping regions; and

$$T_{k+1} = (T_k - P_{k+1}) \cup ((D(P_{k+1}) \cap D(Pv_k)) - D(P_{k+1} \cap Pv_k)) \quad (12),$$

the list of polygons from the narrow regions,

with,

15

$$T_1 = D(P_1) \cap D(P_1) \quad (13); \text{ and}$$

O_1 is nul (empty).

The process is iterative, i.e. each new layer is superimposed to Pv_k , O_k and T_k obtained by the superimposition of the previous layers.

20

The k indicia indicates the number of layers that has been superimposed.

The operations performed on the lists of polygons are union, intersection, subtraction and dilatation. The union ($\cup$) consists in grouping in a single list two lists of polygons. The intersection ($\cap$) between two lists consists in the intersections of every polygon from a first list with every

25

polygon from a second list. Similarly, when two lists are subtrated, every polygon of the first list are subtracted to all polygons in the second list.

The list of visible polygons Pv_k describes all contours that should appear on the PCB image. However, the list Pv_k , as described hereinabove, is not in a format appropriate to be rapidly compared with the ACC. Indeed, every point in the ACC should correspond with at least one polygon segment from Pv_k to ensure rapid comparison.

Transforming the list Pv_k so as to replace every polygon's vertex with the segment that it forms with the next vertex yields the Characteristic Contour Chart (CCC). The segments of the CCC are thus divided into a list of polygons, including the description of all their segments instead of the entire vertexes, as in the Pv_k list. More specifically, every segment in the CCC is characterized by the equation of a line, the co-ordinates of the end points of the line, and a distance threshold allowed from the segment. Moreover, the CCC includes, for all polygons, a characterization of the rectangle that closely encloses the polygon. This rectangle is so dilated as to ensure that every point in the ACC that is not enclosed in the rectangle does not correspond to a polygon's segment.

Detection of Anomalies

Once the CCC is created, step 208 of the method 200 (see Figure 3) consists in detecting anomalies on the PCB by comparing the contours in the ACC with the computer model that is now in the form

of the CCC. More specifically, the anomalies are advantageously indexed and characterized in the following two charts:

ECC, the Excess Contour Chart, and
MCC, the Missing Contour Chart.

5

The first step in the detection of contours in excess is to find a segment in the CCC that corresponds to each point in the ACC. If a point is not sufficiently near any segment of the CCC, it is considered in excess and is saved in the ECC. The computed distance is compared to
10 a pre-determined threshold. The format of the ECC is the same as the ACC; it is a list of contours, with each contour characterized by a list of connected points.

More precisely, for each point in the ACC:

15

- the CCC is scanned until one of its listed rectangles encloses the point;
- the corresponding polygon is scanned until a segment to which the point may correspond the point is found;

20

- if no segment corresponds to the point, another polygon is scanned;
- if a segment is detected, the process continues with the next point, but beginning advantageously with the same segment. Indeed, since the points forming a
25 polygon are consecutive, the probability that the next point corresponds to the same segment is good;

- a point not corresponding to any segment is indexed in the ECC. If the point is part of a consecutive series of points, it is recorded in the ECC as being part of this series.

5

For each contour of the ACC, the points are advantageously scanned in the same direction in the ACC and in the CCC.

Of course, other methods may be used to detect contours in
10 excess in the ACC.

The detection of missing contours will now be described in more detail.

15 The detection of missing contours also implies the detection of contours that deviate too much from the positions of contours in the CCC. For high-density electronic circuits, the allowed deviation zone of a contour is relatively narrow. But since CCC is in a vectorial format, its resolution is not limited by the resolution of the image. Similarly, the ACC
20 has a resolution 7 to 8 times the resolution of the image since the ACC has been compiled after the detection of contours with a sub-pixel resolution. This is advantageous since it allows detecting misplaced contours even when the maximal allowed deviation is narrow.

25 It is to be noted that the detection of missing contours is relatively not binding in terms of precision. The objective of the detection of the missing contour process is only to verify if a contour on the CCC is

absent on the image of the PCB. Indeed, the contours that are present on the PCB image, but misplaced relative to the computer model, will be detected by the detection of the exceeding contour process and indexed on the ECC. Therefore, the detection of the missing contours can be
 5 achieved at the resolution of the image and can thus be done by comparing images.

To obtain the MCC, the contours on the CCC are first drawn on a binary image, the CCC_{image} , having the same resolution than the
 10 ACC_{image} . The comparison between the CCC_{image} and the ACC_{image} , yielding the MCC, is summarized by the following equation:

$$MCC = D(ACC_{image}) \otimes CCC_{image} \cap CCC_{image} \quad (15)$$

where $D(ACC_{image})$ is the ACC_{image} after dilatation. The process of computing MCC is also summarized in Figure 8.

15

The MCC and the ECC comprise region descriptions corresponding to anomalies in the PCB image. These regions are inspected in step 210 of the method 200 (see Figure 3) to detect defects.

20 Inspection of Anomalies

Since an anomaly can generate more than one contour in the ECC, but each contour of the ECC corresponds to only one anomaly, each contour of the ECC is advantageously considered as an independent
 25 anomaly and inspected separately. The inspection process ends when all contours in the ECC have been inspected. The contours in the ECC

which are found beyond predetermined manufacturing standards are advantageously indexed and characterized in a file.

5 This file may take many forms without departing from the spirit and nature of the present invention and includes, for example, defect position and dimension, as well as a description of the PCB inflected with such a defect.

10 The majority of defects that can be found on a PCB correspond to wrong width of elements printed on the PCB or incorrect spacing between these elements. Other anomalies, such as contaminants, grooves, and feathering can be problematic when the width of an element or the distance between two elements diverge too much from design specifications.

15 Defects among anomalies are thus detected when either one of the following two is beyond manufacturing standards: the width of an element having an anomaly on it, and the distance between two neighbouring elements, if the anomaly is not on an element. These two conditions are verified independently as will be explained next.

25 The verifications of these two conditions are very similar. Generally stated, in both cases, all the segments of the CCC that can be found in the neighbourhood of the anomaly are stored in a computer file in the form of a list that will be referred to herein as the Neighbour Segment List (NSL). For every segment near the anomaly, every other segment of the NSL that are opposed to the first segment are found. The

distance between the first segment and all of these opposed segments is verified to assess if the distance is within design or manufacturing standards. Those standards are obviously predetermined and pre-stored in a computer file so as to be accessible by the computer 32.

5

The verification of the distance between two segments and the verification of the width therebetween only differs by the definition of what are considered the two opposed segments, as will be explained hereinbelow.

10

As can be seen in Figure 9, every segment of a contour, such as segment 44, can be interpreted as a frontier between 2 half planes 46 and 48. The plane 46, on the side of the segment 44 opposite the contour 50, is on the "exterior" side of the segment 44. The plane 48, on the same side of the segment 44 as the contour 50, is called the "material" side of the segment 44. The material side is set as "active" when the distance between segment 44 and the "opposed" segment 52 is verified.

20

The exterior side is active when the width of the track 54 between segment 44 and 56 is verified. Two segments are paired for analysis if they face each other and if they are in the same active plane. In the example of Figure 9, segments 52, 58 and 60 are alternatively paired with segment 44.

25

The detection process used thresholds, advantageously expressed in percentages, thus providing a universal method that does

not require that new thresholds for the width and distances be programmed for every new type of circuit. Of course the percentages are expressed according to the design values. For example, the threshold is set to 50%, when the distance between two segments is problematic if it is less than two times what has been designed.

Obviously, thresholds based on absolute values can also be used to verify the width between segments.

Different approaches may be chosen to determine the spacing between two segments, which will depend on the orientation of the segments and whether there are one or more anomalies between the two segments.

If an anomaly is detected between two parallel segments, the spacing between the two segments is simply defined as the distance between the two segments.

If the two segments 62 and 64 are not parallel (see Figure 10) the spacing is arbitrarily defined as the angle between them. In such case the distance of a point to the reference segment 62 is measured as follow: the line 66 intersecting the intersection point 68 between the two segments and the point is measured. The distance between the point and the reference segment is defined as the angle 72 between the line 66 and the reference segment 62.

In both cases, the distance between each point of the anomaly and the reference segment (the opposed segment) is also computed. The maximum and minimum distance recorded define an interval that includes the anomaly. In the presence of an anomaly, the effective spacing between two segments is the spacing between the two segments minus the interval. The normalized spacing is the ratio between the effective spacing and the spacing between the two segments.

Figure 11 illustrates an example of computation of spacing when a plurality of anomalies 74, 76 and 78 are detected between two segments 80 and 82.

In this case, intervals are computed for each anomaly. If the interval of more than one anomaly overlaps (see, for example, anomalies 74 and 76), these intervals are regrouped so as to form a single cumulative interval (a-c). The cumulative (a-c) and non-cumulative (d-e) intervals are added to yield the effective spacing. Again, the normalized spacing is the ratio between the effective spacing and the actual spacing between the two segments.

20

The verification of the spacing and of the widths are only performed in regions of the PCB image near an anomaly. The following description explains how these regions are determined. These regions will be referred to herein as Anomaly Analysis Regions (AAR).

25

Turning now to Figure 12, the AAR is first determined as the rectangle 84 that encloses the anomaly 86. Well known algorithms can

be used to determine such rectangle, knowing the co-ordinates of points forming the contour of the anomaly 86.

The rectangle 84 is then dilated by a Dilatation Factor (DF)
5 defined by:

$$DF = \frac{1 + \%res}{1 - \%res} \quad (14),$$

where *%res* is a predetermined threshold on the normalized spacing or on the normalized width to yield the effective AAR 88.

10 Elements that are out of the region enclosed in the rectangle 88 are advantageously not considered for further analysis since it is presumed that they cannot cause a defect in combination with the anomaly 86. However, all segments 89-91 and 89-91' which are paired as described hereinabove, and which are in the AAR 88, are inspected for
15 defects.

The AAR is advantageously determined for each anomaly when an anomaly is detected in step 208 (see Figure 3) and is saved in the ECC.

20 If the AAR of a second anomaly is overlapping the AAR of a first anomaly, then it is assumed that the combination of these two anomalies can cause a defect. When an anomaly is inspected for defects, verification is thus done as if its AAR does not overlap the AAR of other
25 anomalies. If it does, a new rectangle is created that encloses all the anomalies involved. The new rectangle is dilated to yield a new AAR and

this last process is repeated until the new AAR does not overlap any other AAR. Again, in this AAR, all pairs of segments are verified for defects.

5 Since an AAR may include more than one anomaly or may
be relatively important in size, a test may be performed to verify if each
anomaly is located in the inter-space between two segments. The
following two rules are advantageously implemented to perform the test.

10 The first condition implies that the rectangle defining the
AAR should overlap the two active half planes of the two paired segments.

15 The second condition implies that the two segments should
be sufficiently close. More specifically, the following criterion is used: the
sum of the distances of the anomaly between the first and second
segment should be less than half the maximum distance therebetween.

20 The maximum distance between the two segments is saved and will be
referred to herein as *dsegmax*. Although such criterion has been found
to yield good results, other criterion can also be used to determine if two
segments are sufficiently close to be inspected.

20

When the first and second conditions are satisfied, the
anomaly is considered in the inter-space between the two segments and
is considered when the pair of opposed segments are inspected for
defects.

25

Turning to Figures 13a, 13b and 13c, a method 300 to determine if an anomaly causes a defect by violating the allowed width or spacing will now be described.

5 As will be discussed hereinbelow, two additional inspections will additionally be performed in the case of regions of narrow space and overlapping regions.

 The anomalies are advantageously divided into two groups:
10 anomalies that are on an element which can cause a width defect, and
 anomalies that are not on an element but which can cause spacing defect.

 Anomalies that are on an element are relatively easily selected. Indeed, every contour on the ECC corresponds to a series of
15 consecutive points. A contour of the ECC overlapping a contour of the CCC would then be broken in two. In relation thereto, if any point of a contour of the ECC is on an element, all the points of the contour are considered on this element. If only one point of an ECC contour is not on an element, then the contour is not considered on the element. Verifying
20 *only one point of every contour on the ECC therefore allows discriminating* anomalies in two groups that are referred to herein as ECC_{spacing} and ECC_{width} . The method 300 is used independently on the two groups.

 Since the differences between the application of the
25 method 300 on each of the two groups are very similar, and for concision purposes, the method 300 will be described by referring to the overall ECC.

The first step 302 is to take the next non-inspected anomaly listed in the ECC_{width} . Obviously, the first time the step 302 is performed, the next anomaly is the first one listed. This anomaly will now be referred to as the Current Anomaly (CA).

Step 304 consists in finding all anomalies in the ECC having its corresponding AAR overlapping the AAR of the CA.

In step 306, these anomalies are marked as inspected in the ECC lists and placed in a Current Anomalies List (CAL). The AAR is re-evaluated (step 306) as discussed hereinabove and steps 304 and 306 are repeated until no non-inspected anomalies have its AAR overlapping the new AAR.

In step 308, all the contours in the CCC enclosed in the AAR are listed in a new NSL.

In step 310, it is verified that all the segments in the NSL are marked as analyzed. If they are not, the next non-analyzed segment in the NSL becomes the Current Segment (CS), the CS is marked as analyzed in the NSL (step 311), and the process continues to step 316.

If all the segments in the NSL were marked as analyzed then, in step 312, verification is made that all anomalies in the ECC have been marked as inspected. If not, step 302 is repeated with the next

anomaly. If so, the inspection of anomalies ends and the overall process is stopped (step 314).

Step 316 consists in verifying if the next non-analyzed
5 segment of the NSL is opposed to the CS using the criteria described hereinabove. If such is the case, then the process continues with step 320. If this is not the case, and the end of the NSL has been reached (318), then step 310 is repeated. If not, step 316 is repeated.

10 In step 320, *dsegmax* is computed, as explained hereinabove, considering the CS and the segment opposed to the CS.

In step 322, verification is made as to whether the CAL
includes another non-inspected anomaly. If such is the case, the next
15 non-inspected anomaly in the CAL is set as the CA (step 324) and the process continues with step 326. If not, the method proceeds with step 330.

In step 326, if the ARR of the CA does not overlap the active
20 half planes of both the CS and the segment opposed thereto, then step 322 is repeated. If so, then the method proceeds with step 328.

In step 328, the minimal distance to the CS (MD1) and to
the segment opposed (MD2), and the maximal and minimal spacing (MAS
25 and MIS) are computed for every point of the CA.

Among all detected anomalies, the ones that are in the inter-space of the two segments are determined (step 330). The test is as follows: if

$$MD1 + MD2 > DSEGMAX * \%RES$$

5

then the anomaly is not considered in the inter-space of the two segments. The anomalies that are not in the inter-space are removed from the CAL (step 332).

10 If the CAL is empty, then the process returns to step 310 (step 334).

In step 336, the regions occupied by each anomaly are saved in a list of All Occupied Regions (AOR) that is advantageously in
15 the form of intervals. The AOR list includes MAS and MIS for each anomaly. If two regions overlap, then their intervals are added.

In step 338, the normalized spacing is computed as discussed hereinabove. Indeed, the effective spacing between the CS
20 and the segment opposed to it is computed. It is equal to the spacing between the two segments minus the sum of the intervals in the AOR. The normalized spacing is equal to the ratio of the effective spacing to the actual spacing between the two segments.

25 A defect is detected in step 340 if the nomalized spacing is inferior to the value $\%RES$.

In step 342, the detected defect is characterized and saved in a defect list. More specifically, the anomalies and the two segments (CS and the segment opposed to CS) are saved. The process then returns to step 310.

5

Inspection of Regions of Narrow Space

The polygons T_{k+1} from the narrow regions are also advantageously inspected for the following two reasons:

10

- their dimension on the multi-layer model may be too small; and
- even if an inter-space between two elements on the model is correct, the corresponding region on the image may be incorrect.

15

Generally stated, two opposite segments in the list of polygons T_{k+1} yields an inter-space having a width. Some of those inter-spaces correspond to a defect if the corresponding width is not within a predetermined threshold.

20

More specifically, a dimension on the multi-layer model that is too small can be indicative of an important alignment gap between the layers. This is detected by testing the inter-spaces with predetermined criteria. For example, the spacing between two different materials may be verified to ensure that it is not inferior to a minimum

25

spacing value as set, for example, by the manufacturers of the PCB. If it is less, then an alignment problem is detected between two layers.

It is possible that the detection of anomalies according to
5 the method 300 may fail to detect a wrong spacing between two elements of a PCB.

Indeed, as was discussed hereinabove, the detection of anomalies is done within a predetermined allowance between the contours
10 in the model and the contours as detected in the PCB image. However, *the minimal dimension of a defect allowed in an inter-space depends on the dimension of the inter-space.* For example, if the allowance on the contour position is 25% of the width on the model, and the minimal width of the region is 50% of the model's width, then the anomalies will not be
15 detected by the algorithm 300.

Turning now to Figure 14, a first anomaly 90 can cause a deviation of a contour 91 within the tolerance margin and be at 25% of the nominal width 92 and so is a second anomaly 94 near the opposite
20 contour 93. None of the two anomalies are detected by the method 300 since they are both within the tolerance margins 92 and 96. However, the combination of the two anomalies contributes to narrowing the inter-space by 50% of its nominal width. This however corresponds to a defect.

25 The following criteria, illustrated in Figure 15, are set according to the method of the present invention to detect a defect in a region of narrow space (inter-space).

In step 400, if the computed width of the inter-space is inferior to the threshold, as specified for example by the manufacturer, an alignment error of the layers is detected.

5

If not, and the width is inferior to four times the tolerance margin on the contour position (step 402), then a defect may be present and the width of the inter-space is measured on the PCB image.

10

If the measured width is inferior to 50% of the width on the model (step 404), then the region between the elements (inter-space) is incorrect and a defect is detected.

As it is illustrated in Figures 16 and 17, the measurement of the inter-space 98 on the vectorial model is advantageously done by gradually narrowing the polygon 100 (see arrows 101) until segments of the polygon cross (see 102 in Figure 17). This process is called erosion.

The rate of erosion is an indication of the width of the inter-space 98. For example, the erosion rate is advantageously set high until a crossing occurs 102. Then, the rate is lowered to dilate the polygon 100. Then the rate is lowered again to restart the erosion until another crossing occurs. The process is repeated until the erosion rate is equal or inferior to the required resolution. The rate is expressed in fractional number of pixels since the resolution is advantageously in the order of the sub-pixel as it has been discussed hereinabove.

The measurement of the inter-space on the image is advantageously done by the extraction of the contours that mark off the inter-space. It is to be noted that these contours do not necessarily correspond to the same element or do not necessarily produce closed
5 contours. The contours are then closed to form the vertex of a polygon. The method discussed hereinabove for the erosion of the polygon in the vectorial model is used to measure the width of the contour.

Other methods can also be used to measure the inter-space
10 both on the vectorial model and on the image, without departing from the spirit of the present invention.

Of course, other thresholds or criteria may also be set to detect defects in narrow regions.

15

Inspection of Overlapping Regions

The overlapping regions obtained by equation (11) are inspected for the same two reasons stated in the prior section.

20

However, in the case of the overlapping regions, another problem may arise: one or more contours of a layer may be hidden on the PCB image by an overlapping layer. In this case, two possibilities exist.

25

A portion of a segment may be visible, allowing the extrapolation of the contour on the superior layer. Otherwise, only visible contours are detected.

Since, the detection of defects in overlapping regions is generally based on the same criteria described in Figure 15, it will not be described herein in more detail.

5

Defects that are caused by narrow regions or by overlapping regions are advantageously characterized and saved in a file for further analysis. This file may be the same as the one used to store and characterize the other detected defects.

10

Although the present invention has been described hereinabove by way of preferred embodiments thereof, it can be modified, without departing from the spirit and nature of the subject invention as defined in the appended claims.

WHAT IS CLAIMED IS:

1. A method for detecting surface defects on a Printed
Circuit Board (PCB) having at least one layer including components, said
5 method comprising:

providing a digital image of the PCB;

identifying edges on said PCB image;

providing a corresponding computer model for each
component on said at least one layer of the PCB;

10 detecting anomalies on the PCB image by comparing said
identified edges to said computer model; and

for each of said detected anomalies, determining if said
detected anomaly corresponds to a surface defect.

15 2. A method for detecting surface defects on a PCB as
recited in claim 1, wherein said computer model of each component on
said at least one layer including segments.

3. A method for detecting surface defects on a PCB as
20 recited in claim 1, wherein said edges identifying step includes:

computing the numerical gradient on said PCB image to
yield a gradient image;

thresholding said gradient image to yield a thresholded
image;

25 skeletting said thresholded image; and
finding edges on the skeletted image.

4. A method for detecting surface defects on a PCB as recited in claim 3, wherein the edges finding step yields at least one chain of pixels and includes:

5 saving each of said at least one of said chain of pixels of the skelleted image as a vector of pixel co-ordinates;

using said pixel co-ordinates to estimate the second derivative of said at least one chain of pixels; and

10 estimating the co-ordinates of the passage by zero of said second derivatives;
whereby, the passage by zero correspond to said edges.

5. A method for detecting surface defects on a PCB as recited in claim 3, further comprising correcting the luminosity of the PCB image.

15

6. A method for detecting surface defects on a PCB as recited in claim 5, further comprising filtering the corrected image;

20 7. A method for detecting surface defects on a PCB as recited in claim 1, wherein said provided computer model includes a vectorial model for each of said layer.

8. A method for detecting surface defects on a PCB as recited in claim 7, wherein said vectorial model includes a polygon
25 representation of every components on each layer.

9. A method for detecting surface defects on a PCB as recited in claim 2, wherein each of said at least one layer of said computer model being aligned with said PCB image before the anomalies detecting step.

5

10.A method for detecting surface defects on a PCB as recited in claim 9, wherein at least one reference point is selected on the PCB image for each of said at least one layer of the computer model; said reference point is used to align said corresponding layer with said PCB image.

10

11. A method for detecting surface defects on a PCB as recited in claim 9, wherein transformations are applied on every segment of each layer of said computer model to align said each of layer with said PCB image so as to compensate for at least one of rotation, translation, scaling and bad perspective of said each layer.

15

12.A method for detecting surface defects on a PCB as recited in claim 9, wherein said aligned layers are superimposed.

20

13. A method for detecting surface defects on a PCB as recited in claim 12, wherein said provided computer model is a vectorial model including a polygon representation of every components on each layer; said polygon including at least one segment; said layers of said computer model are superimposed by superimposing each polygon from each layer with each polygon from the other layer.

25

14. A method for detecting surface defects on a PCB as recited in claim 13, wherein said superimposition is done using the following iterative formula for each layer:

$$Pv_{k+1} = (Pv_k - P_{k+1}) \cup P_{K+1}$$

5 where

P_k is a list of vectorial representations of segments in the current layer;

Pv_k is a list of vectorial representations of segments produced by the superimposition of the previous layers;

10 Pv_{k+1} is a list of vectorial representations of segments produced by the superimposition a layer with the superimposition of the previous layers; and

$$Pv_1 = P_1.$$

15 15. A method for detecting surface defects on a PCB as recited in claim 1, wherein said computer model includes contours of each component.

20 16. A method for detecting surface defects on a PCB as recited in claim 15, wherein said computer model further includes, for each contour, a model of a rectangle enclosing said contour.

25 17. A method for detecting surface defects on a PCB as recited in claim 16, wherein said anomalies detecting step includes: a) for each pixel of the PCB image, verifying if a contour among said contours of each component has its corresponding rectangle enclosing said pixel; if not, considering said pixel part of an anomaly; if so, b) verifying if said

pixel is part of said contour having its corresponding rectangle enclosing said pixel; if not, considering said pixel part of an anomaly.

18. A method for detecting surface defects on a PCB as
5 recited in claim 1, wherein said anomalies detecting step includes using said computer model to create an image of said computer model and comparing said PCB image with said image of said computer model.

19. A method for detecting surface defects on a PCB as
10 recited in claim 2, wherein the determination of defects includes *comparing said detected anomalies to design specifications.*

20. A method for detecting surface defects on a PCB as
recited in claim 19, wherein the step of detecting anomalies on the PCB
15 image further including detecting edges in the PCB image that are in excess relative to said segments in said computer model; for each of said detected anomalies, determining defects by a) determining a first segment in said computer model that is positioned within a predetermined distance of one of said detected anomalies; b) for each segment opposed to said
20 opposed segment, determining if the distance between said first segment and said segment opposed to said first segment is within a predetermined threshold.

21. A method for detecting defects on a PCB as recited in
25 claim 20, wherein said distance between said first segment and said segment opposed to said first segment is computed as the effective

spacing therebetween when more than one anomalies is located between said first segment and said opposed segment.

22. A method for detecting defects on a PCB as recited in
5 claim 21, wherein said anomalies include overlapping and non-overlapping anomalies; said effective spacing is computed as the sum of intervals between all non-overlapping anomalies and regrouped overlapping anomalies.

10 23. A method for detecting defects on a PCB as recited in claim 20, wherein said predetermined distance is defined by a rectangle that encloses said detected anomaly.

24. A method for detecting defects on a PCB as recited in
15 claim 23, wherein said enclosing rectangles being dilated.

25. A method for detecting defects on a PCB as recited in claim 23, further including verifying if said enclosing rectangle encloses another anomaly and, if so, redefining a rectangle that encloses said
20 another anomaly.

26. A method for detecting defects on a PCB as recited in claim 25, wherein said rectangle redefining step is iterative.

25 27. A method for detecting defects on a PCB as recited in claim 20, wherein said threshold is in the form of a ratio.

28. A method for detecting surface defects on a PCB as recited in claim 13, wherein said anomalies detecting step includes determining, for each layer k of said computer model, a list of polygons T_k , located in narrow regions of said layer k , using the following iterative
 5 formula:

$$T_{k+1} = (T_k - P_{k+1}) \cup ((D(P_{k+1}) \cap D(Pv_k)) - D(P_{k+1} \cap Pv_k))$$

where

P_{k+1} is a list of vectorial representations of polygons in the previous layer;

10 Pv_k is a list of vectorial representations of polygons produced by the superimposition of the previous layers;

Pv_{k+1} is a list of vectorial representations of polygons produced by the superimposition a layer with the superimposition of the current layer;

15 D is a function that causes the dilatation of all polygons in a list of polygons;

and

$$T_1 = D(P_1) \cap D(P_1);$$

two opposite segments in said list of polygons yielding an
 20 inter-space having a width; each of said inter-space corresponding to a defect is said corresponding width is not within a predetermined threshold.

29. A method for detecting surface defects on a PCB as recited in claim 28, wherein said width of said inter-space is determined
 25 by erosion of each of said polygons.

30. A method for detecting surface defects on a PCB as recited in claim 13, wherein said anomalies detecting step includes determining, for each of said at least one layer k , a list of polygons O_k in overlapping regions on the PCB's computer model using the following
 5 iterative formula:

$$O_{k+1} = (O_k - P_{k+1}) \cup (P_{k+1} \cap Pv_k)$$

where:

P_{k+1} is a list of vectorial representations of polygons in the previous layer;
 10 Pv_k is a list of vectorial representations of polygons produced by the superimposition of the previous layers;
 and
 O_1 is nul;

two opposite segments in said list of polygons yielding an inter-space
 15 having a width; each of said inter-space corresponding to a defect is said corresponding width is not within a predetermined threshold.

31. A method for detecting surface defects on a PCB as recited in claim 30, wherein said list of polygons includes visible and
 20 hidden contours; said hidden contours being extrapolated using said visible contours.

32. A method for detecting surface defects on a PCB as recited in claim 1, further comprising indexing said detected anomalies in
 25 a computer file.

33.A method for detecting surface defects on a PCB as recited in claim 1, wherein said image is in the form of a bitmap.

34.A method for detecting surface defects on a PCB as
5 recited in claim 33, wherein said bitmap includes grey levels.

35.A system for detecting surface defects on a PCB; said system comprising:

a computer including a model of the PCB and being
10 configured to identify edges on a PCB images, detecting anomalies on the PCB image by comparing said identified edges to said computer model, and for each detected anomaly, determining if said detected anomaly corresponds to a surface defect;

an illumination assembly connected to said computer to
15 provide illumination on the PCB;

a frame grabber connected to said computer;

a camera connected to said frame grabber to take an image of the PCB; and

a positioning system connected to said frame grabber.

20

36.A system as recited in claim 35, wherein said computer is further configured so as to control the illumination assembly.

37.A system as recited in claim 35, wherein said positioning
25 system is connected to said frame grabbed via an axis controller and a servo controller.

38.A system as recited in claim 35, further comprising an image processing card for interfacing said frame grabber and said computer.

5 39.A system as recited in claim 35, wherein said camera is a Charge-Coupled Device (CCD).

40.A system as recited in claim 39, wherein said CCD is configured to operate in linear mode.

10

41.A system as recited in claim 39, wherein said CCD is configured to operate in matrix mode.

42.A system as recited in claim 35, wherein said positioning
15 system includes at least one servomotor.