

INSPECTION SYSTEM USING DYNAMICALLY OBTAINED VALUES AND RELATED TECHNIQUES

FIELD OF THE INVENTION

5 This invention relates generally to inspection systems and more particularly to image processing systems and techniques for use with inspection systems including inspection systems used to inspect printed circuit boards.

BACKGROUND OF THE INVENTION

As is known in the art, an inspection system refers to a system used to inspect any
10 real world process, device or object. An Automated optical Inspection System (AOI) performs inspections largely without human intervention. AOIs may take a variety of shapes and configurations depending upon the particular application in which they are used. Typically, however, such systems include one or more sensors which are mounted within a fixture (sometimes referred to as an inspection head). The inspection head is
15 adapted for controlled movement relative to the object being inspected. Each of the one or more sensors captures an image of the object (or part of the object) being inspected and provides the captured image to an image processing system. The most typical type of sensors are cameras that are sensitive to the visible lights spectrum. Others, for instance, are sensitive to X-Rays. The image processing system compares the captured image of the
20 actual object being inspected to a software model of objects of that type. Based upon the results of the comparison, the inspection system provides an indication of how well the captured image matched the model. Thus, the inspection system uses models in the inspection process.

As is also known, a software model or more simply, a model, is a representation
25 of a real world process, device or concept which has been “realized” or “represented” in a software. The model thus provides a representation of selected or entire aspects of a structure, behavior or operation or other characteristic of a real world process, concept or system. The real world process, device or concept is referred to as an object class.

In order to generate a model, one must first identify an object class and then select
30 attributes of the object class to be encoded in the model. The object class thus typically

includes a group of objects or instances of objects which share one or more characteristics or attributes.

It is generally desirable to select for inclusion in the models those attributes which can concisely summarize the object class and which allow an inspection system using a model which includes these attributes to identify particular objects as “true positives” and to distinguish other objects which are not part of the class as “true negatives.” Since there are many attributes from which to choose for inclusion in the model and since some attributes result in the model, when applied, having a high rate of success in identifying “true positives” and distinguishing “true negatives”, it is difficult to determine which attributes to include in the model. The problem becomes more difficult when we cannot determine which attributes allow an inspection system to distinguish true positives from true negatives. In addition, if we have a hypothesis about what attributes to include, we may not know how to measure or represent them.

An object which is labeled as “true positive” is an object which properly belongs to a particular object class with which the object is being compared. For example, if the object class is integrated circuit package types and the object is an integrated circuit, then the integrated circuit would be considered a true positive with respect to the integrated circuit object class.

An object which is a “true negative,” on the other hand, is an object which does not properly belong to a particular object class with which the object is being compared. For example, assume the object class is integrated circuit package types and the object is a lumped element resistor. In this case, the lumped element resistor would be considered a true negative with respect to the integrated circuit object class because a lumped element resistor does not belong to the same object class as objects having integrated circuit package types.

In order to match a model to an object to determine if it is part of the object class, a matching method is used. The matching method extracts the chosen attributes from the object being inspected and compares the measured attributes of that particular object to the attributes of the object class as stored in the model. One important aspect of the matching method is that it correctly calculate or determine the value of the attributes from

the object being inspected. These calculated or selected attributes are then compared to the model attributes. One example of an attribute used to model components on a printed circuit board are the part boundary edges with the printed circuit board and any internal edges of the component. Given an image that may contain a part, large image gradients or discontinuities are considered as potential "edge candidates" that are the result of the placement of the component on the board.

One problem with this approach, however, is that many matching methods are not able to correctly determine, or determine in a realistic amount of time, which data from the image which contains the object being inspected should be included in an attribute measurement. This dilemma is often called the correspondence problem. When a precise correspondence is necessary between the measured data from the image of object being inspected and the model attributes but the data is ambiguous with respect to a particular attribute, match methods tend to yield poor results. In the worst case, if there are n attribute candidate measurements in the image and m attribute measurements in the model of the part, there are m^n possible combinations. Often all must be evaluated to choose the best correspondence. Even then a true match may not exist if one of the critical attribute measurements in the image was not measured properly.

As is also known in the art, conventional printed circuit board (PCB) inspection techniques typically use only a single type of model having a single attribute. Also, conventional inspection systems use a single matching method. Most model matching schemes compute instances of attributes in the image and compare them to all instances attributes in the model. As described above, the number of correspondences that must be evaluated is exponential. Many techniques try to refine this set by ruling out combinations that are unlikely or that violate some heuristically generated rules.

Different types of models are also known. One type of model referred to as an image model is generated from an image of an instance of an object being inspected. In practice, the model is often derived or built from an image of a sample or a typical one of the objects to be inspected. The sample or typical object may be that of an entire circuit component or a portion of a circuit component or from a portion of a PCB to be inspected. The image model typically includes only a single attribute, for example,

luminance. The luminance distribution is arranged in a fixed spatial configuration. A matching method is used to translate the image of the object being inspected (e.g. the component or the portion of the circuit being inspected) into a set of attributes like those included in the model. For example, if luminance attributes are included in the image model, then the matching method generates a set of luminance attributes from the object being inspected. The single image model is then used to perform an inspection process. One problem with the image model technique, however, is that if the appearance of true positives changes over particular instances of the object(s) to be inspected, the image models tend to be a poor representation of the actual data. That is, the image of the circuit component or PCB from which the single image model is provided may not be an accurate representation of a typical circuit component or PCB being inspected during the inspection process. It also may not be a good representation of a typical circuit component which may have several acceptable appearances. Consequently, the image model will not accurately match images of the circuit components or PCBs being inspected and thus the inspection system using the image model will not yield accurate test results.

Another type of model referred to as an edge model is often provided from an idealized edge representation of the component or a circuit portion of a circuit to be inspected. A matching method is used to translate the image of the object being inspected (e.g. the component or the portion of the circuit being inspected) into a set of edge attributes. One problem with this approach, however, is that a new image to be inspected may include many edges. In such a case, it may be unclear which set of edges to use to match the new data from an object being inspected to the set of edges or lines in the model thus making it difficult to measure the corresponding features in the new image and in the model. It is also possible that due to poor lighting conditions, camera noise, low contrast between the object and the background, or numerous other conditions, the image processing system was not able to discern a true component edge. When the matching method has not accurately translated the image circuit components into the desired attributes, the inspection system will not yield accurate test results.

When the models do not yield accurate test results, the inspection system provides a significant number of “false positives” and a significant number of “false negatives”. In the printed circuit board inspection context, a “false positive” means that the inspection system indicates that a circuit component is present on a PCB when the circuit component
5 actually is not present. Similarly, a “false negative” means that the system indicates that a circuit component is not present on a PCB when the circuit component actually is present.

Automated optical inspection of PCBs is relatively difficult for a variety of reasons. For example, circuit components having a dark color can be disposed on PCBs having a dark color. Thus, in this case a camera does not detect any significant contrast
10 between the circuit component and the PCB due to a dark part (i.e. the circuit component) being disposed on a dark background (i.e. the PCB).

Also, PCB’s can include “false edges” which are due to silk screening processes used on the PCB, as well as false negatives and positives which are due to the high amount of variability in component and printed circuit board appearance. Such variations
15 also make it difficult for inspection systems to consistently recognize parts on the PCB.

It is undesirable to have false negatives because it is time consuming for a human to look at and dismiss these failure conditions. It is undesirable to have and false positives since it is time consuming and expensive to later determine that circuit components are not there or are faulty. Thus, inspection systems utilizing the single model and matching
20 method approach typically result in increased PCB manufacturing costs and reduce the rate at which the PCBs can be manufactured.

It would, therefore, be desirable to provide an inspection system and technique which results in relatively few false positives and false negatives. It would also be desirable to provide a technique which increases the capacity / rate at which PCBs can be
25 inspected and manufactured and which processes images of printed circuit components relatively rapidly. It would be further desirable to provide a system which produces results which are both reproducible and repeatable.

SUMMARY OF THE INVENTION

In view of the above problems and limitations of existing inspection systems,
30 including circuit board inspection systems, and in accordance with the present invention,

it has been recognized that combining the need for accurate inspection test results with the usefulness and desirability of performing rapid image analysis can be achieved by dynamically obtaining one or more characteristics of the circuit board during an inspection process and using the one or more characteristics in a model used to inspect the circuit board. Appropriate types of characteristic include but are not limited to color, texture and luminance of the object (e.g. the circuit board) to be inspected. This, in turn, leads to the problem of how to select and utilize the appropriate number and types of characteristics to use in the inspection process.

It has, in accordance with the present invention, also been recognized that circuit boards have a relatively small number of colors associated with them. Thus, in a printed circuit board inspection system the characteristic of color can be used during an inspection process to reduce the number of false positives and false negatives while at the same time increasing the speed with which the circuit components or PCBs are inspected.

It has been recognized that an unpopulated circuit board has a relatively small number of distinct colors distributed about the surface of the unpopulated circuit board. In present technology, unpopulated circuit boards are fabricated with a variety of materials and a variety of processes. For example, unpopulated circuit boards can be fabricated from paper composites, fiberglass, and poly-tetrafluoroethylene (PTFE). A dominant color of the unpopulated circuit board is often associated with a solder mask layer that is deposited on both outer surfaces of the unpopulated circuit board during its manufacture. The solder mask layer is provided in a variety of colors, including, but not limited to, blue and green. The solder mask layer is applied to most of the surface of the unpopulated circuit board, including all areas of the unpopulated circuit board that do not receive solder paste. The solder mask is not applied to areas of the unpopulated circuit board that do receive the solder paste, the solder paste areas provided to attach electrical components. The solder mask is also not applied to unpopulated circuit board areas that must otherwise be exposed, for example connector pads. The areas that receive solder will generally be either silver or gray both before and after solder paste is applied, and the areas that are otherwise exposed are generally silver, copper, or gold color. In addition to the above colors, a silkscreen is often applied to the surface of the unpopulated circuit

board on the surface of the solder mask, the silkscreen having alpha-numeric reference designators and sometimes also body outlines corresponding to electrical components. The silkscreen is generally white, but can be a variety of colors.

5 Regardless of the material of the circuit board, the color of the solder mask, the color of the areas that receive the solder paste, the color of the otherwise exposed areas, and the color of the silkscreen, it has been recognized that there are a relatively number of colors associated with the surface of a particular unpopulated circuit board.

10 Circuit components disposed on the circuit board can have many colors. For example, a resistor can be black, green, or brown. In many instances, the circuit component has a color that is very distinct from any color associated with the unpopulated circuit board. In other cases, the electrical component has a color that is close to a color associated with the unpopulated circuit board.

15 Therefore, it would be desirable to provide a system that can distinguish the colors associated with the unpopulated circuit board from the colors associated with circuit components disposed on the circuit board such that a determination can be made as to whether a component is disposed at a particular region of interest (ROI) on the circuit board.

20 Thus, in accordance with the present technique, a method for generating a characteristic palette for a circuit board for use in a circuit board inspection system includes identifying a value range of a first characteristic on the circuit board, establishing a plurality of characteristic categories for the circuit board, and selecting a first plurality of locations on the circuit board with each of the plurality of locations having a characteristic value which is representative of at least one of the characteristic categories with the first plurality of locations corresponding to first palette regions for the circuit
25 board.

With this particular arrangement a palette region which can be used in an inspection process is provided. The palette regions correspond to physical locations on the printed circuit board at which a value representative of the characteristic of interest can be measured on the circuit board. Once the palette regions are identified, the palette
30 regions can be used in a circuit board inspection process. In particular, for each circuit

board being inspected, a characteristic (e.g. color) of the circuit board can be dynamically measured at each of the palette regions. Once values at each palette region have been measured, measurements of the characteristic value at one or more regions of interest (ROIs) on a circuit board can be dynamically obtained. The dynamically obtained values at the ROIs can then be compared with the palette region values dynamically obtained for that circuit board. Based upon the comparison of the dynamically obtained ROI values and the dynamically obtained palette region values, each ROI on the PCB can be placed into one of the established categories.

In accordance with a further aspect of the present invention, an inspection system includes a color palette and one or more inspection models. With this particular arrangement, an inspection system which utilizes a color palette during an inspection process (e.g. inspection of an object such as a printed circuit board) and provides relatively few false positive and false negative results is provided. The particular number of colors to use in the color palette used in the inspection process depends, at least in part, upon how many colors are on the printed circuit board.

In accordance with a yet further aspect of the present invention, a process to inspect an object includes dynamically measuring values at one or more palette regions on the object, dynamically measuring values at one or more grid regions in a region of interest on the object, comparing each of the grid region values with the palette region values and based on the result of the comparison, categorizing each of the grid regions. With this particular arrangement, an inspection process which uses dynamically obtained values and which results in an accurate inspection process is provided. The categorized grid region values resultant from the dynamic measurements can then be compared with expected categories for that region of interest on the circuit board.

In accordance with a further aspect of the present invention, a process to inspect a circuit board includes representing the circuit board with a relatively small number of color categories and using the color categories to determine whether a component is present on the circuit board. With this particular arrangement, a process for inspecting circuit boards which utilizes a relatively small number of colors is provided. It should be appreciated that the process can be used to inspect circuit boards and other types of

objects.

While the present invention is described herein in the context of a circuit board inspection system, it should be appreciated that more generally the invention relates to dynamically measuring a particular characteristic of a piece being inspected and using the dynamically obtained values in an inspection process. In a preferred embodiment, the characteristics are measured at specific locations on the piece and these specific locations are referred to as "characteristics palette regions" or more simply "palette regions." The particular characteristic being measured can be any one characteristic of the piece including but not limited to color, texture and luminance of course, in some embodiments, it might be desirable to utilize a combination of characteristics, e.g. color and texture, or color and luminance or texture and luminance. The decision to select one particular characteristic depends upon the particular application and the particular characteristic which is important in that particular application. For example, in one application directed toward inspection of printed circuit boards, color may be a characteristic of importance. In this case, the characteristic palette may be referred to as a color palette (i.e. the characteristic of interest being color) and the regions of interest may be referred to as color palette regions. In another example, texture may be a characteristic of importance in a printed circuit board inspection system and in this case the characteristic palette may be referred to as a texture palette (i.e. the characteristic of interest being texture) and the regions of interest may be referred to as texture palette regions. In short, any characteristic (or combination of characteristics) of the piece being inspected can be used to form the characteristic palette and the palette regions are selected accordingly.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing features of this invention, as well as the invention itself, may be more fully understood from the following description of the drawings in which:

Fig. 1 is a block diagram of an inspection system;

Fig. 2 illustrates the steps for inspecting a particular printed circuit board type;

Fig. 3 illustrates the steps of automatically obtaining a snapshot of a component rotated to be at the default orientation;

Fig. 3A describes a learning process, given a set of example images (bare, paste, place, and a part snapshot);

Fig. 4 illustrates a learning process to pick the best set of example images to be used in Fig. 3A;

5 Fig. 5 illustrates the steps of an inspection process for a component; and

Figs. 6 and 6A illustrates a specific implementation of the Figure 5 to inspect a component.

Figs. 7 and 7A show an image model, a structural model and a geometry model in nominal orientations as trained on captured image regions.

10 Fig. 8 shows the image structural and geometry models of Figs. 7 and 7A inflated from an expected angle.

Figs. 9-9C show image, structural, and geometry models applied to three different cases of inspection.

15 Figs. 10-10D, are plots of the structural model score on instances of paste and placed images for a package type CC0805.

Figs. 11 and 11A, are plots of structural model score versus instance is shown for a series of placed parts and paste parts of the type RC1206

Fig. 12 shows an image model matched to a component identified as an RC0805.

Fig. 13 shows image model scores for paste and place images of the type RC1206.

20 Figs. 14 – 14A show a technique for learning a model or set of models which provides good classification of images for a part type.

Fig. 14B shows a histogram of number of occurrences of scores having two curves fit through the data points.

25 Fig. 15 is a flow chart showing the steps of an exemplary optical process for printed circuit board inspection using dynamically obtained values;

Fig. 16 is an electronic image showing a placed circuit board, a circuit board upon which some electrical components have been placed;

Fig. 16A is a pictorial representation of a portion of the placed circuit board of FIG. 16;

30 Fig. 17 is a flow chart showing an exemplary learn paste board process and an

exemplary learn paste board process, as well as an exemplary negative model process;

Fig. 18 is a pictorial diagram showing portion of a paste circuit board to which a grid area having grid regions has been applied;

Fig. 18A is a pictorial diagram showing a portion of a placed circuit board to which a grid area having grid regions has been applied;

Fig. 19 is an electronic image showing a portion of a circuit board having regions to which solder paste is incompletely applied;

Fig. 19A is another electronic image, corresponding to the electronic image of Fig. 19, to which the regions to which solder paste is incompletely applied are electronically filled in to indicate solder paste;

Fig. 20 is yet another electronic image showing a portion of a circuit board having regions to which solder paste is incompletely applied;

Fig. 20A is yet another electronic image, corresponding to the electronic image of Fig. 20, to which the regions to which solder paste is incompletely applied are electronically filled in to indicate solder paste;

Fig. 21 is another pictorial diagram showing a portion of a placed circuit board to which a grid area having grid regions has been applied;

Fig. 21A is yet another pictorial diagram showing a portion of a placed circuit board to which a grid area having grid regions has been applied; and

Fig. 22 is a flow diagram showing the steps to manufacture a printed circuit board.

DETAILED DESCRIPTION OF THE INVENTION

Before describing the processing and apparatus of the present invention, it should be appreciated that, in an effort to promote clarity, reference is sometimes made herein to inspection of certain objects or inspections performed in a particular field of use. Such references and accompanying examples are only intended to facilitate an appreciation of the invention and should not be taken as limiting use of the concepts described herein to use with only systems of the type described herein. Rather, as mentioned above, the present invention finds application in a wide variety of different fields and generally finds application to the problem of object recognition or detection. The present invention can be used to recognize objects such as faces or to detect object such as wafer defects. Other

areas include image database indexing, medical image analysis, and surveillance and monitoring applications.

Also in the description hereinbelow, reference is sometimes made to a system having a particular imaging system or imaging system components or a particular lighting system or lighting system components or lights which operate at particular frequency profiles or temperatures. Those of ordinary skill in the art will appreciate, of course, that the concepts described herein apply equally well to inspection systems having any type of imaging or lighting systems (including lighting systems operating over a wide range of frequency profiles) or components provided that the systems or components have the desired operational or functional characteristics. Reference is also sometimes made herein to a lighting system having lights disposed in a particular topology. Those of ordinary skill in the art will appreciate that the principles of the present invention can be implemented using a variety of light topologies and that those presented herein are only examples and should not be construed as limiting.

Reference is also sometimes made herein to inspection of certain objects such as printed circuit boards and circuit components disposed on printed circuit boards. As described herein, a circuit component or more simply a component, refers to a part such as an integrated circuit which is mounted or otherwise coupled to a printed circuit board (PCB). The object may also be a printed circuit board defect. The PCB may be of any type.

Those of ordinary skill in the art will appreciate that the principles of the present invention can find use in a variety of applications including, but not limited to, inspection of printed circuit boards and components as well as inspection of any other types of objects. For example, the present invention finds applications where one object is disposed over and possibly in contact with or in close proximity with another object or where one object is embedded in another object, or where it is desirable to identify a foreground object from a background in an image. , Likewise, the techniques described herein may be used for any type of printed circuit board or circuit board component without regard to its function.

Accordingly, those of ordinary skill in the art will appreciate that the description

and processing described herein as taking place on "printed circuit boards" and "components" could equally be taking place on a person's face or a fingerprint or a company logo or any other image. Likewise, the processors described hereinbelow may include any type of integrated circuit, hardwired or programmed to perform a particular task or function.

Referring now to Fig. 1, a processing system 10 for performing inspections of printed circuit boards (PCBs) includes a database component 12 having stored therein a package library 14 which contains detailed information concerning certain objects. For example, in the case of a PCB inspection system, the package library 14 includes detailed information concerning the shape and size of an integrated circuit does not include any information related to how the parts would be disposed on a PCB. The database 12 also includes an inspection plan library 16 which is coupled to an inspection plan generator 18 which generates an inspection plan for a particular PCB and stores the results of the inspection plan in the inspection plan library 16.

An image processing system 20 coupled to the database 12 includes an image capture system 22, an image processor 24 and an image interface unit 25. The image capture system 22 may be provided, for example, as one or more cameras or sensors which capture an image of an object to be inspected. In a preferred embodiment, the cameras correspond to color cameras. After the image is captured, there is subsequent processing on the data, such as Bayer color recovery, white balance, contrast enhancement, etc.... The image interface unit 25 may be provided as a graphical user interface (GUI) for example, through which a user can interface with the image processing system 20.

The images captured by the image capture system 22 are delivered to a runtime system 26. The runtime system 26 determines from the inspection plan 16 which parts to inspect in one camera field of view for a particular board type. The runtime system 26 also determines what parts need to be inspected over several camera fields of view (e.g. if the part crosses a camera frame boundary or a part is too big for one camera frame).

To inspect one particular part, the runtime system 26 invokes an inspector module 28. The inspector module 28 includes an occlusion detector 30, a theta estimator 32, a

learn and debug system 34 an image model processor 36, a structural model processor 38
a geometric model processor 40 and an orientation mark detector 42. The runtime system
26 can invoke the inspector module 28 in an “inspect mode”, a “learn mode”, or a “debug
mode.” In both learn and debug modes, the system 10 will learn and save attributes about
5 the appearance of parts and update or add to the corresponding image, structural and
geometric models. The runtime system 26 can take input from a user via a user interface
module. For instance, during a debug process, the user of the system can be asked
questions via this user interface.

When the runtime system invokes the inspector module 28 in the inspect mode, an
10 inspection process is initiated. The inspection process will be described in detail in
conjunction with Fig. 5 below. Suffice it here to say the inspection system 10 utilizes a
plurality of modules during the inspection process. In one embodiment, the inspection
module 28 utilizes the image model 36, the structural model 38 and the geometric model
40 in a particular order to perform an inspection. Thus, the inspection system 10 utilizes
15 an image model, a structural model and a geometric model to inspect objects.

The three model types are combined in a way that uses the strengths of each type
of model. The image model 36 is first applied to a larger region of interest on the PCB to
determine if there exists an object in this larger region of interest that looks extremely
similar to the picture stored in the image model of an object being inspected to determine
20 if the part being inspected “looks like” the image model. Such a use should be
distinguished, for example, from conventional uses of image models in which the image
model is used to determine whether an object being inspected is present in an image. Use
of the image model in accordance with the present invention provides a relatively rapid
technique for identifying objects which look very similar. The attributes included in the
25 image model can correspond to color, luminance, etc.... It should be appreciated that an
image model is usually a fixed pattern of binary, luminance, color pixels. Usually these
pixels have a one to one correspondence to an imaged view of an object. However, an
image model can also be a fixed pattern of processed image features, such as gradients or
texture features. An image model may also exist at many different resolutions. A
30 disadvantage of an image model is that many features on a bare or pasted board may look

very similar to the features on an object.

After the image model 36 is applied to the image of the object being inspected, a structural model 38 is applied.. Specifically, the structural model 38 is applied to make the decision of whether the object is truly present in the region of interest. If the image
5 model indicates that it thinks the part is present at a particular location, the structural model checks to see if the indicated part has all of the structural features that should be present on the part. The structural model may be used to provide a closer approximation of the location of the object. If the image model indicates that it does not think a part similar to its internal image is present in the ROI. The structural model looks over the
10 whole region for a part that looks different from the image model, but has the right structural components.

The output of the image and structural matching steps is an indication that either 1) the part is absent or 2) the part is present at rough location $\langle x,y \rangle$ with an estimate of rotation at r degrees. If the part is present, a geometric model is applied to determine
15 precisely the location of the part or object being inspected.

Finally, a geometric model 40 is applied to determine precisely the location of the part or object being inspected. The geometric model searches for all edges of the object substantially simultaneously with the constraint that the edges match the "top level" model description. The assumption is made with the geometric model that the part or
20 object is already known to be in a particular place and the geometric model 40 determines the exact details of the part or object placement. The geometric model utilizes strong gradients in luminance, color, etc... to precisely locate the part or object being inspected.

It should be appreciated that the geometric model can use features other than strong gradients. For example, it can analyze the image for regions containing inflection points, other geometric feature, and even image features, such as a distinct precisely positioned
25 mark. Thus, use of the multiple models in the inspection system 10 results in the system 10 having increased speed, fewer false fails, and greater measurement resolution than prior art systems.

Figs. 2-6A, 15 and 17 are a series of flow diagrams showing the processing
30 performed by a processing apparatus which may, for example, be provided as part of the

inspection system 10 (Fig. 1) to inspect printed circuit boards (PCBs). Alternatively, the processing steps may be implemented by an image processing system which simply matches images in a process other than PCB inspection. The rectangular elements (typified by element 54 in Fig. 2), are herein denoted "processing blocks," and represent
5 computer software instructions or groups of instructions. The diamond shaped elements (typified by element 50 in Fig. 2), are herein denoted "decision blocks," and represent computer software instructions, or groups of instructions which affect the execution of the computer software instructions represented by the processing blocks.

Alternatively, the processing and decision blocks represent steps performed by
10 functionally equivalent circuits such as a digital signal processor (DSP) circuit or an application specific integrated circuit (ASIC). The flow diagrams do not depict the syntax of any particular programming language. Rather, the flow diagrams illustrate the functional information one of ordinary skill in the art requires to fabricate circuits or to generate computer software to perform the processing required of the particular
15 apparatus. It should be noted that many routine program elements, such as initialization of loops and variables and the use of temporary variables are not shown. It will be appreciated by those of ordinary skill in the art that unless otherwise indicated herein, the particular sequence of steps described is illustrative only and can be varied without departing from the spirit of the invention.

20 Some of the steps in the flow diagrams of Figs. 2-6A below are described as non-board specific meaning that no information from any specific printed circuit board is needed in order to carry out the step. Other steps in the flow diagram are described as board specific meaning that at least some information about one or more specific printed circuit boards (or specific types of printed circuit boards) is needed in order to carry out
25 the step. In particular, if a particular element or step performed during a process of inspecting a printed circuit board is said to be non-board specific, this indicates that these elements apply irrespective of what board is being built. Examples of non-specific board information include the size of the parts, the snapshot of a particular part, the default structural model and geometry model. Board specific information is used mostly in the
30 training step where the models learn the difference between an instance of a part on paste

in a region of interest and a region of interest with pasted or bare pads. Any step that requires contextual information about the part on the board is board specific.

Turning now to Fig. 2, the PCB inspection process is shown. In the inspection process, processing begins in step 50 where it is determined whether a package library is populated. If the package library is populated then processing flows to step 52 where an inspection plan is generated for a specific PCB. If the package library is not populated then processing flows first to step 54 where the package library is populated and then to step 52 where an inspection plan is generated for a specific PCB.

The package library is annotated to include a visual class type. In one particular PCB inspection application, eleven different visual class types are defined. It should be appreciated that some applications may require more than eleven visual classes and that other applications may require less than eleven visual classes. The particular number of visual classes used in any particular application will be selected in accordance with the needs of that application. Parts classified as a particular visual class have a common structure. For instance, all discretes with metal endcaps have two metal endcaps on either side of a colored body. Thus, a capacitor of size 60 by 30 mils (referred to as package type CC0603) and a resistor of size 120 by 60 mils (referred to as package type RC1206) have the same visual class even though they are of different sizes and perform different functions.

Libraries of parts typically describe the part as a specific package type with known dimensions and also a functional value. A part number is often given to a package type with a particular function value. In these libraries, it is not noted how the parts look or how they should be grouped via visual characteristics. An image processing system on the other hand e.g. image processing system 20 (Fig. 1), expects a grouping of parts based on their structure or appearance. Thus, it is useful to annotate any part libraries with a definition of visual classes.

It should be noted that if a library is already populated (i.e. if data on the objects to be inspected is already stored in the data base) or can be read from a computer database or other storage device, the population step may not be needed in which case processing begins by annotating the library. In practice, it is infrequent that the library will have no

parts stored therein and a check is made to determine if there are entries in the library for every part on the board.

The package library includes information such as: (1) part dimensions; (2) part name; (3) assigned part number; (4) vendor; (5) body size; (6) does the part have leads?; 5 (7) size of leads; (7A) type of leads (8) lead pitch; (9) does part have an orientation mark?; (10) where on the part should orientation mark occur. In one implementation the class types are one of eleven visual class types: (1) discretes with metal endcaps; (2) gull wing integrated circuits (ICs); (3) J-leaded ICs; (4) tantalum capacitors; (5) flat leaded (6) CAP; (7) discrete arrays; (8) MELF ; (9) SOTs; (10) diodes; and (11) ball grid arrays 10 (BGAs).

It should be appreciated that other visual class types can be added as need. Also, in some applications it may be desirable to include class types which are not visual class types of parts. For instance, it may be desirable to generate a set of visual classes around specific defects such as tombstoning or billboarding (part has one side touching the board 15 and the other side is lifted up in the air), damaged part, and smeared paste. The package library is preferably provided as a database which can be accessed by various parts of the inspection system as needed. The part library population/ annotation step is a non-board specific step.

It may also be desirable in some applications to put snapshots in the library. It is 20 presently believed that the snapshot is board independent. However, a few tests have shown that a part on a light board will look brighter than a part on a dark board.

As shown in step 52, after populating the package library, the inspection plan is generated. The inspection plan is generated by taking in board specific information. Board specific information can be in the form of computer aided design (CAD) data, pick 25 & place data, and/or PCB layout data (e.g. Gerber data). The inspection plan describes the location of each component on the PCB, describes the part type and what orientation it should have. The inspection plan also describes where the fiducials and vias are on the board. The inspection plan also describes the models to use to inspect each part. The inspection plan is initialized with default models for each component type.

30 Once the basic inspection plan is generated, processing proceeds to step 56 which

corresponds to a board-specific learning step in which the information in the basic inspection plan is augmented. This board-specific learning process relates the known part information in the plan, such as geometric information, to its observed visual characteristics. This learning process is more fully described in conjunction with Figures 3, 3A, 4, and 14 below. Steps 50-56 above thus correspond to a set-up and learn procedure which takes place before an inspection step is performed.

After generation of the inspection plan and completion of the board-specific learning process, processing proceeds to step 58 in which an inspection step is performed. The inspection step is a board specific step and is described in detail in conjunction with Figs. 5, 6 and 6A below. In step 58, the inspection is performed on a test data set. Generally, we should only expect to get three training boards when we first setup or train. The first is bare, the second is a pasted board and the third is a placed board. For each type of part on the board, we want to train the models to be able to detect when the part is present and when it is absent. Usually there is more than one instance of the part on the board. We can break up the instances into two groups, the learn and the test group. Each group should have labeled positive (part present) and negative (part absent) examples. The system will train the models on the learn group and verify they are working properly on the test set. (It is possible for the test and learn set to have elements in common.)

After completing the inspection of the test data set in step 58, processing proceeds to step 60 in which decision is made as to whether a debug mode should be initiated. The decision is substantially based upon the results of the inspection of the test data set 58. In particular, if the results of step 58 indicate that the models yield good results for all more components entire PCB, then decision is made in step 60 to proceed directly to step 62 where a determination is made as to whether any more boards should be inspected. If, on the other hand, the results of step 58 indicate that the models do not yield good results for one or more components or for an entire PCB, then decision is made in step 60 to proceed to step 64 where another learn step is implemented. This learn step will take the new false positives and false negative, along with other training data, and revise the models. In addition to retraining on the false positives and negatives, the user or the system may change the snapshot in the image model, change the dimensions of the part to better fit

the data, add more models of a particular type, change the sequence of calling the models, and change the decision function for determining part presence or absence.

It should be noted that the user can set the system to debug mode at any time. It should also be noted that debug mode is mainly used to correct problems with any of the models in the inspection plan. If decision is made to enter a debug process then processing flows to the debug learning process of steps 64, 66. If a new model for a component has been learned in background mode, it can be substituted for one in the inspection plan. Otherwise, the learning steps for a particular model are repeated on a different set of images. Processing then flows to an update inspection plan step in which the model for a problem component is replaced with a new “debugged” model.

The decision as to whether the debug process steps should be followed can take place in either of two ways. In one approach, the inspection system itself identifies a problem inspecting a particular part (or stated differently, the inspection system itself identifies a problem with a particular model - e.g. the model yields poor results in inspecting a specific part). In this case, the inspection system itself as part of the debug learning process can ask the user a series of questions such as: (1) the part in question has a very different appearance than the other parts of this kind. Is this a normal variation?; and (2) the system is having difficulty inspecting this part. Is it damaged?

The debug and subsequent learning process results in a revised specific model or set of models for that part which the system identified as a problem part. After the new model or set of models is generated, the inspection plan for the problem component is updated.

In a second approach, the system does not recognize that it has a problem correctly inspecting a part (i.e. the system does not realize that a model it is using is resulting in incorrect inspection results). For example the system does not recognize that it is providing false positives (i.e. labeling a bad part as a good part, or labeling an empty location as having a part) and false negative (i.e. labeling a good part as a bad part, or incorrectly labeling the part absent). In this case, external intervention is required (e.g. a user or something else external to the system must recognize that the system has a problem identifying a part). Once the problem is identified, the system is notified and

instructed to change the model to reduce the number of false positives and false negatives. In response to the manual instruction and in order to reduce the number of false positives and false negatives, the system executes the debug mode process, learn, and inspection plan update. During the debug and subsequent learn process, the user, for
5 example, can provide the images to the system which resulted in the false positives and false negative results.

It should be noted that the debug process can be implemented at any time (e.g. after inspection of a single component or board or after inspection of one-hundred components or boards).

10 Processing then proceeds to a decision step where decision is made as to whether more boards remain to be processed.

Upon completion of the learn step 64, the inspection plan is updated for the problem component or PCB. That is, the new model generated in the learn step 64 is associated with the part or PCB which resulted in the poor inspection results. Thus, the
15 next time that particular part is inspected, the new model or models are used in the inspection process. It should be noted that all inspection plans are board specific.

Processing then again proceeds to step 62 where decision is made to inspect more PCBs. If more PCBs remain for processing, then processing proceeds to step 68 where the next board is inspected.

20 After the next board is inspected, processing proceeds to step 70 where an optional background model build process step takes place. Thus, during regular inspection, a background model learning step can be performed.

The steps for performing a background model learn are the same as the steps for the learn. Background model learning can process data over many boards and thus
25 includes an amount of data which is relatively large compared with the amount of data used in the initial learn process of step 56.

As mentioned above, if a decision is made not to debug, then processing directly proceeds to the decision step where decision is made as to whether more boards remain to be processed.

Thus, if in step 62 decision is made that no more boards remain to be processed, then processing ends. If on the other hand, more boards remain to be processed, then a loop is entered in which the next board to be inspected is identified and inspected and the optional steps of background model builds and debugging are repeated. It should be appreciated that in some embodiments it may be preferable to not always perform background model learning or debugging. It should be noted that the part plan can be saved and used later if the board associated with the plan is built again.

Figs. 3, 3A, and 4 describe elements of a learning process. The models may need to be trained on board specific images. For one part type, the models need to see examples of: (1) a tightly cropped image of the part without any surround rotated to the correct orientation, (2) the part, on paste, with its surround (known as the “place” image), (3) examples of pasted pads and the surround without the part (known as the “paste” image), (4) examples of bare pads and the surround without the part (known as the “bare” image). Therefore, in one particular embodiment, the minimum number of example images that are required to train the models for a part is four. In some applications, however, the minimum number of examples can be less than four. For example, in some applications, (e.g. some printed circuit board inspection applications, the bare board has been found to be unimportant. Preferably images (2)-(4) above are from the same reference designator (i.e. from the same part at the same location on a PCB) and image (1) is from a different reference designator (i.e. from the same part at a different location on the PCB than examples (2) –(4)). Image (1) may also be captured independently by a non-board specific technique (e.g. it can be imaged alone without a board).

In learning how to inspect a part, there exist several variabilities that occur. These include part color/luminance variability, part size changes, non-uniform illumination, less or more oxidized metal, different types of paste, and a variety of bare board features. The bare board features include different colors on the same board, a variety of thicknesses and finishes, vias, traces, and silkscreen. By constraining the bare, paste and place images to come from the same reference designator, we at least can eliminate the board variabilities during the learn. It is important that the snapshot come from a different reference designator or even a different board because of how the image model performs

a match. The image model checks to see if there is a part exactly like the saved snapshot in the region of interest. If the snapshot came from the same reference designator as the place image, the snapshot would match the part perfectly (with a difference of 0). Parts actually vary quite a bit in appearance from instance to instance. We would like to try to
5 quantify that variability in the learn process. A snapshot matched to itself does not tell the system anything about the amount of part appearance variations.

It is desirable to have the system be able to automatically crop the snapshots from an example when needed. The alternative is to have the user import an image of the board containing the part or an image of the part on a background, rubber band the part, copy
10 and paste the part to a new image, rotate the cropped image to the appropriate orientation and save it. This user intervention gets quite cumbersome when there are many part types.

In Fig. 3, a process for automatically obtaining a snapshot of a component rotated to be at the default orientation are shown and thus the method described in Fig. 3 describes how to automatically obtain a cropped image of a part.

15 Referring now to Fig. 3, processing begins with step 72 in which a “default snapshot” of an object to be inspected is obtained. The best way to get a real snapshot of a part is to use the inspection process previously defined to localize the desired part in a region of interest. If the part is well localized, the image of the part can be easily rotated to the default orientation and cropped. Currently, both the image model and the structural
20 model require a snapshot in order to inspect. In order to break the circular nature of the problem, we can bootstrap the system with a synthetic snapshot that has the same geometry of the desired component and some of the key visual characteristics of the part. Also, a snapshot of a similar looking part may also be used. For instance, if we need a snapshot of a CC0805 of part number 123-000-000, we may use a previously captured
25 snapshot which shares the same package type, CC0805, but is of part number 123-000-001. These default snapshots may be stored in database 12 or in the inspection plan itself.

Regardless of how the default snapshot is provided, processing the proceeds to step 74 where the snapshot is used to build a default image model and a default structural model. The image and structural models generated in step 74 are referred to as “default”
30 models since the models do not yet any board specific information.

Processing then proceeds to step 76 where the geometry model is generated or built purely geometric information alone (e.g. the geometric information available in the package library). One can thus build a geometry model directly from the information about the part in the package library.

5 Next, as shown in step 78, a region of interest (ROI) on the PCB to be inspected is identified. The ROI should contain the object under consideration (e.g. the part such as the circuit component to be inspected). In processing step 80, the default image and default structural models are applied to the ROI to decide if the part is present within the ROI. Processing then proceeds to step 82 where a determination is made as to whether
10 the part is present in the ROI. Thus, the image and structural models are used to determine whether the part is present in the ROI.

 If in decision block 82, decision is made that the part is not present, then processing proceeds back to step 78 where a new ROI is input for application of the image and structural models in step 80. This loop is repeated until the image and
15 structural models indicate that the part is present in some ROI or until all of the ROIs have been explored and the part is deemed not present.

 If in decision block 82, decision is made that the part is present, then processing proceeds to step 84. In step 84, the structural model provides the roughly localized part position on the PCB in the ROI to the geometry model. The output from the structural
20 model includes the center of the part in the ROI $\langle dx, dy \rangle$ (delta position from expected location) and a rough estimate of how the part is rotated. The structural model may go a step further convert $\langle dx, dy \rangle$ and theta to the positions of each edge of the part that is considered geometry model.

 The geometric model localizes (i.e. finds) the boundaries and any rotation angle
25 (θ) of the part to an accuracy of a sub-pixel order of magnitude (e.g. with a few tenths of a pixel).

 As shown in step 86, once the exact boundaries of the part and its rotation are known, it can be rotated to the default orientation. The part is rotated by an amount equal to the minus of the rotation angle ($-\theta$) to obtain a “nominal angle” which is defined in the
30 part library (i.e. the part library holds a definition of what zero degrees rotation is and this

can be related to what is found on a particular printed circuit board.) For instance, if the system find the part to be rotated by 1 degrees counter clockwise and the default orientation of the part is at 90 degrees clockwise, the system can rotate the part image by 91 degrees clockwise to transform the image into the correct orientation.

5 Once the snapshot has been found and cropped as shown in step 88, a reference designator may be chosen at random to provide the placed, paste, and bare examples.

Figure 3A discusses one way to train the models on the snapshot and board specific examples. This corresponds to step 56 in Figure 2.

Referring now to Fig. 3A describes the learn process for the image in the
10 structural model. Processing begins in step 90 in which the structural and image models are imported and applied to cropped, paste, placed and bare example ROIs. It should be noted that in the process flow of Fig. 3A, the images are randomly selected. In some applications, however, it may be desirable to select the best set of example images to be used in the learn process. Such a technique will be described below in conjunction with
15 Fig. 4.

Processing then proceeds to step 92 in which the default qualitative and quantitative properties or attributes of the structural model are replaced with values learned by applying the structural model to the images. In one embodiment, four images – i.e. the cropped, paste, placed and bare images are used. It is recognized, however, that
20 the invention need not be limited to the four images since this is a very generic learn flow. The attributes of the structural model may be modified at several levels. At a top level, the structural model may just record how well or poorly it performed on the paste and place images. The performance can be stored as a set of scores with at least one score for the paste image and one score for the place image.

25 The structural model may be modified at a lower level. We may instruct the structural model to change its quantitative and qualitative relationships to best fit the placed image and to best distinguish itself from the paste image.

Processing then proceeds to step 94 in which the default attributes of the image model are replaced with values learned by applying the image model to the four images
30 (the cropped, paste, placed and bare images).

Similar to the structural model, the image model may be modified at several different levels. First, the snapshot is associated with the model. Second, The image model scores on the paste and place images may be stored. In addition, other attributes within the image model may be modified to best fit the placed image and to best distinguish itself from the placed image. For instance, it may learn that in the paste image at locations (x1, y1) and (x2,y2) the image model provides a good match. This means that there are features on the bare board that look very much like the part. We can generate a score function such that the image model when applied to a new region gives a low confidence match score if it finds the best fit at (x1, y1) or (x2,y2).

Processing then proceeds to step 95 in which the default attributes of the geometry model are replaced with values learned by applying the geometry model to the four images. For instance, the geometry model can measure the true dimensions of the part and its subparts from the placed image and the snapshot. The dimensions stored in the default geometry model are the mean of the expected dimensions for that part type across all vendors. The geometry model may also learn the strength of the gradients at the part and board boundaries and also between subpart boundaries. The geometry model may also learn the best color channel, if the image is provided in color, to compute the gradients.

Processing then ends.

It is appreciated that choosing a random set of bare, paste, place images to train on (as described in Fig. 3A) may not be the optimal procedure. Ideally, the system should look at all reference designators across the different types of boards (e.g. bare, paste and place) to choose the example images that provide the best performance.

Thus, referring now to Fig. 4 the steps which could be performed if it were desired to select the best cropped, paste, placed and bare images to use in the learn process described above in conjunction with Fig. 3A are shown.

Processing begins in step 96 by importing: (1) the board types (bare, paste, place), (2) the inspection plan. The example place boards are commonly referred to as "bronze boards" because there may have been errors or omissions in the placement of the parts on the boards.

In processing step 98 default models associated with each part type are generated

and in step 100, for a particular part type, the default models are trained on one more reference designators and applied to every other reference designator that is associated with the part type. It should be noted that each location (or board position) on a printed circuit board at which a component is to be placed is assigned a unique reference
5 designator. Thus, in the inspection plan each reference designator specifies a particular, unique location on the printed circuit board as well as a particular part which is to be located that board position.

In step 102, for a particular reference designator, the default image and structural models of a particular part type are used to first check if the part is present in the "bronze" board and absent in the paste and place boards at that reference designator
10

Processing then proceeds to step 104 in which once the set of assertions from step 102 are verified, the models are trained on the three images derived from the example reference designator (Note that it is not require to train on all three image. In some embodiments the bare board is not found to be useful). As shown in step 106, the learned
15 models can then be used to inspect the rest of reference designators associated with the part type both on the "bronze" place board and on the paste and bare boards.

As shown in step 108, the models trained on a particular reference designator can be rated in terms of how well they separate the true positive examples, "the part is present", from the true negative examples, " the part is absent". For each reference
20 designator used as a training example, one can rate how effective it is at providing this separation. As shown in step 110, the example set of images which provides the best separation should be chosen as the training example to be used in full production board inspection.

As shown in step 112, it may be found that one set of training images is not
25 sufficient for the models of one part type to separate the true positives and the true negatives. For instance, for one part type, there may be both brown and white instances on the board. Rather than having to choose between a brown or a white example, we can train two sets of models, one on a brown part and one on a white part. A simple clustering algorithm on the outputs of the image and structural models scores can be used
30 to determine if more than one reference designator is required. The clustering algorithm

will determine how many reference designator examples are required.

In step 114, for each cluster a reference designator set of images is chosen. Default image and structural models are generated for each cluster. A learn is then performed for each set of <models, reference designator>.

5 Processing then proceeds to step 116 where the learned models are then saved back into the inspection plan. Processing then ends.

It should be understood that Fig. 4, describes a scenario in which only a single trio of images (one example of a bare, paste and placed image) are used for training. The other N-1 images are used for testing to determine or gauge how well the models work. It
10 should be appreciated, however, that in some applications more than one reference designator could be used for training in step 104 and therefore it would be desirable to select the best set or sets of reference designators which give the models the best discrimination ability over the test set (i.e. whatever is left).

For example, if two reference designators are available, one can be used to
15 provide a snapshot and one can be used to train the models in the paste and placed images. As another example, given three reference designators a first reference designator could be used for the snapshot and the remaining two references design would be used for the training. This can be extended to the case where N reference designators are available and M (where M is less than N) are used for training leaving N-M reference
20 designators available for the testing. Lastly, it should be noted that this process can be used to choose the best snapshot as well as the best training set of ROIs

It should be appreciated that in the description provided above, no distinction is made between selecting the snapshot from selecting the paste, place, and bare training images. In the worst case, every reference designator should be examined as a candidate
25 for the snapshot. In practice, however, this is too time consuming.

Once the models for each part type are generated and are trained on a set of example images, the system is ready to inspect boards. Figure 5 illustrates the steps of an inspection process which utilizes multiple models.

Referring now to Fig. 5, processing begins in step 118 by first obtaining an image
30 of a region of interest on the board that should contain a specific part type. Processing

then proceeds to step 120 models associated with that part type are obtained.

Next, as shown in step 122, the image model is applied to the captured image. With any model it is desirable that we apply the model to all possible center locations at all possible rotations to determine the part center and rotation. This is known as an
5 exhaustive search approach and ensures that if there is a part in the that looks similar to the snapshot in the image model, the image model will find it. For speed issues the image model can sample the set of centers and possible rotations when searching the ROI. The greater the sampling the quicker the operation. However, as sample size increases, the probability of the model finding the part decreases. To try to maximize both speed and
10 effectiveness, we apply a coarse to fine strategy; first sampling coarsely and then sampling finely around the regions that are the best candidates for the real part center and rotation. This strategy can be used by any model, not just the image model.

Processing then proceeds to decision block 124, a decision is made as to whether the image model indicates that a part is present.

15 If decision is made that the image model believes a part is present, then processing proceeds to step 126 where the structural model is used to search around a relatively small area where the image model found the part. If decision is made that it is not known if the part is present, then processing proceeds to step 128 where the structural model is used to perform a full search of the ROI. When the structural model searches the whole
20 ROI, it may employ the coarse to fine searching method

Next, in step 130, based upon the results of the structural model, a final decision is made as to whether a part is present. (Note in deciding whether the part is present, the structural model may use the image model score as part of its decision function) If the part is deemed not present, then processing ends.

25 If, on the other hand the part is deemed present, then processing continues to step 132 in which the geometry model is used to locate the part precisely and also to provide more detailed rotation information about the part. Processing then ends.

Figures 6 and 6A illustrate in more detail the steps of an inspection process for a particular component at a particular reference designator. Steps other than the application
30 of the three models are identified. Decision procedures to determine if the image and

structural model indicate the part is present are also described.

In step 134, for a part type, the models are loaded into the inspection plan. Next as shown in step 136, the system acquires a picture of the part and its surround for a particular reference designator. This image is labeled the "region of interest" or ROI.

- 5 The ROI may be cropped from a larger picture of the board that has already been acquired or pieced together from several camera frames. Determining the size of ROI is dependent upon the part type, its size, the position tolerances inputted by the user, and the size of any the surround regions in the models.

- 10 A first optional step 138 is to inspect the ROI for features that appear on the bare and pasted board, but should be occluded when the part is present. These features include vias, traces, and pads (pasted or bare) that are commonly hidden under large parts such as integrated circuits or parts with hidden connections such as ball grid arrays. One implementation of this step is to look for circular, rectangular or linear features of an expected size. Another implementation is to compare current ROI to the same ROI on
15 the learned paste board. If decision is made in step 140 that the occlusion features are present with high confidence, then processing ends since this means that the part is absent.

- If decision is made in step 140 that the occlusion features are not present, it is assumed that something (e.g. the correct part or something else), is in the image. The
20 system then performs a second optional step 142 to process the ROI to look for dominant angular features. One method to do this is to compute the gradient direction and magnitude at each point in the image. Based on the angular features, the system computes the dominant angle of the objects in the image. If the part is not present, the background of the board traces and pads will produce dominant angles of 0, 90, 180 or
25 360 degrees. This is because these features are usually aligned or orthogonal to the camera frame. If the part is present, its features will contribute to the set of dominant angles. The best dominant angle, theta, other than 0, 90, 180, or 360 is recorded for later use. This gives the system an estimate of how the part is rotated, but not where the part is located.

- 30 Next as shown in step 144, the image model is applied to the ROI image. The

image model essentially looks to see if the ROI contains a pattern that looks very similar to the cropped snapshot and very different from the bare or paste images. It uses the learned data to compute a probability or a confidence that the correct part is present. The image model searches over the entire ROI. The image model should match the snapshot at multiple rotations at multiple locations in the image in case the part itself is translated from the expected position and is rotated beyond the expected rotation. To increase speed, the image model may not check for the part at every location in the ROI. Part size and type of part currently determines how coarsely the image model samples the image space. In addition, to increase speed, the image model only checks for two rotations, the expected rotation and theta. After a coarse search, the image model may do a fine search around the best candidate for the part center. The image model outputs the best location, $\langle x1, y1 \rangle$ and angle hypothesis, theta2, for the part in the ROI. It also outputs a probability, phi, that the part is present.

Next, in step 146, a decision is made as to whether the image model indicates that a part is present. This is based upon a first threshold value, here denoted phi. It should be appreciated that although a relatively simple threshold value is here used, in some applications it may be advantageous or desirable to utilize a relatively complex function to provide the threshold. For instance, a more complex function could take the computed probability phi, the actual match score between the image snapshot and the candidate location for the part, the candidate location, and the probability that the image is not paste to determine whether the part is really present.

If the image model is confident that the part is present and looks exactly like what it has learned, then processing flows to step 148 and the structural model is used to verify that the part is present. It tries to increase the localization resolution by searching around a relatively small area where the image model found the part. In this case, it only checks for the part, in small step sizes such as 1 or 2 pixels, around location $\langle x1, y1 \rangle$ and angle theta2.

If the image model is not confident that a part is present, then processing flows to step 150 in which the structural model does a full search of the ROI. Again for speed, the structural model does not check for the part at every location and rotation in the ROI.

Part size currently determines how the coarsely structural model samples the ROI. In addition, for further speed increase, the structural model only checks for two possible part rotations, the expected rotation and theta.

5 The structural model ultimately decides if the part is present. If, in either case, it determines the part is in the ROI, it returns the best location, $\langle x2, y2 \rangle$ and angle, theta3 as shown in step 152. This information is sent to the geometry model, which then localizes the part to subpixel accuracy and refines the theta estimate as shown in step 156. The geometry model returns the label "present", the final center position, $\langle x3, y3 \rangle$, and angle, theta4 as shown in step 158.

10 If the structural model determines that the part is not present, the system stops and returns the label "absent".

The flow described in Figs. 6 and 6A is optimized for speed. The optional precursor steps of comparing bare board features and calculating a theta estimation can be skipped. In the most thorough mode, the image and structural models can search every
15 possible position for the part in the ROI. They can also perform the search looking for the part at a variety of angles. In addition they may do a search for both position and rotation in a coarse to fine manner.

It is possible via the learn and debug that an alternate image or structural model has been specified. If this is the case, at the appropriate time, both would be applied. If
20 either matched, the process would continue as specified in Figure 4.

It should be noted that the decision functions in Figures 6 and 6A can be base on other factor or expressions more complicated than "is the probability $> \phi$ ". This is just an example. Also, a mixed combination of model scores may be used in the decision function, such as in the structural model decision function.

25 It should also be noted that although Figures 6 and 6A only show outputs of "present" and "absent". There is an intermediate scenario, where something is in the image, however, it is not the correct part. In this case, at the structural model stage, using the second decision function, if the probability is medium that the part is present, the system could branch to another analysis procedure which performs a diagnosis and comes
30 out with a result of "part damaged", "wrong part", "smudged paste", "part tombstoned"...

Finally, this work is aimed at printed circuit board inspection, but the strategy is applicable across image processing domains. One could use the same strategy for face recognition, person recognition, and image database search for instance.

Referring now to Figs. 7 and 7A, an image model 160, a structural model 162, and
5 a geometry model 164 shown in nominal orientations are trained on captured image regions 166, 168. The image regions 166, 168 correspond to portions of a PCB. The image region 166 is a so-called “paste” image meaning that the PCB has paste (e.g. solder paste) disposed in areas in which a circuit component will be mounted. Thus, no circuit component is shown in image 166. The image region 168 is a so-called “placed” image
10 meaning that a circuit component 170 should be found in a particular location within the region 168.

Each of the three types of models are provided for a predetermined visual class. Eleven class types used in one printed circuit board inspection embodiment are listed above in conjunction with Fig. 2. As noted above, fewer or greater than eleven class types
15 can be used and the particular number of class types as well as the class types themselves will be selected in accordance with the needs of a particular application.

To determine the number of visual classes, we considered all the typical component types for surface mount printed circuit boards. We essentially clustered the components based common visual characteristics of the parts. Our goal was to generate
20 the smallest number of visual classes that spanned all the types of typical parts. In the case of PCB components, we discovered the component lead type dictated the visual appearance of the components. For instance, components with endcaps are generally resistors and capacitors, all which have two metal endcaps and a colored body. Components with gull wing leads are usually integrated circuits that usually have a black
25 rectangular body with metal leads that protrude from the body, bend down in the z axis, and then straighten out. In each case, even though the size of the whole package and its subparts can change, the overall configuration of the component stays the same. This is why the structural model is well-suited to represent these classes.

This technique of clustering visual stimuli into visual classes is not unique to the
30 printed circuit board inspection application. It is applied to most computer vision

applications in order to generate a set of models that cover the class of visual stimuli for that application. One example is in the area of face detection. Faces under different illumination conditions, with different expressions, of people with different genders and ages can be clustered into groups based on visual similarity. For instance, in prior art
5 work related to image processing of faces, it was found that frontal faces could be clustered into six visual classes. Although the visual classes had no name or semantic meaning to humans, the clusters greatly aided the problem of face detection. If a new image, when processed, fell into one of the visual clusters, it was identified as a face. If it fell outside the clusters, it was identified as a non-face. A measure of confidence in the
10 diagnosis, was how close the new image was to a face cluster.

Visual class clustering is common in other applications such as medical imaging (e.g. characterizing and clustering the appearance of tumors through a sensor), military applications (e.g. classifying patterns for detecting objects from synthetic aperture radar), and even applications such as traffic flow monitoring (e.g. classifying different patterns of
15 traffic flow).

In this particular example, the models 160-164 each belong to the visual class type called DISCRETES. Each of the eleven visual class types includes nine elements to the overall matching method (1) occlusion features; (2) theta estimator regions; (3) image model properties; (4) structural model properties; (5) geometry model properties; (6)
20 orientation mark types; (7) alternate models; (8) learn parameters (9) the decision function

The occlusion attribute identifies occlusion features expected to be present in the ROI. The theta estimator attribute identifies the region(s) over which the theta estimator should be used. The structural model attribute describes the regions, region properties and relations from which the model is comprised. The geometry model attribute
25 describes the composition of high gradient or edge attributes. The orientation mark identifies the type of mark expected to be present on the part body.

The alternate models attribute identifies alternate models which can be used for each part in the particular class type. For instance, an alternate model might include an image model of the part when it is placed upside-down. The learn properties include
30 information concerning which characteristics should be learned and saved for each model

in each class type. Finally the last property describes the decision function for whether the part is present or absent. This decision function may take as input the outputs of one or more model types.

An example of the visual class information for the visual class of discretes used in a printed circuit board application is shown in Table 1 below.

TABLE 1 Visual Class: DISCRETES

Occlusion	Theta Estimator	Image Model	Struct. Model	Geom. Model	Orientation Mark	Alt. Models	Learn Notes	Notes
pasted pads	Use over the whole ROI or the Error bounding Rectangle	Use Full Image with potential emphasis on endcaps	3 Body Regions 2 Paste Regions 2 Background Regions	Multiline Order of operations: 1)theta 2)Dx Dy 3)Theta revision Fit external borders of endcaps first Use body borders to verify result Change dimensions from original specs. To do a best fit of part.	In most cases no orientation mark In a few cases orientation mark is a stripe	Could have models for flipped parts	Geom. Model will tell true overall size and endcaps. Save all structural All model params. Incl: - determine ratio between endcap and paste. - Body and background	Structural Model is the decision maker on presence / absence

The image model 160 corresponds to a cropped image which can be provided, for example, in accordance with the techniques described above. The structural model 162 here corresponds to a structural model for a predetermined component in the visual class DISCRETES. Thus the structural model includes a first portion 162a which represents a main body of the discrete part, proximate end portions 162b, 162c which represent "end caps," (i.e. the leads of a discrete component) side portions 162d, 162e which represent board background portions and distal end portions 162f, 162g which represent pad paste regions. The geometric model 164 includes endcap regions 164a, 164b and body region 164c.

In one embodiment, the system trains at two different rotations. Thus, FIG. 7A shows a second orientation of image structural and geometric models 160', 162', 164'. The models 160', 162', 164' are substantially the same as the models 160, 162, 164 but are simply rotated at an angle which is different than the angle of the models 160, 162, 164.

In Fig. 7, the models 160-164 and regions 166, 168 are shown at rotations corresponding to 0° and 180°. As indicated in the figure, the structural model portions are rotated to match circuit components having that general orientation. In Fig. 7A, the image, structural, and geometry models as well as the regions 160' – 170' are shown at nominal $\pm 90^\circ$. Fig. 7A again shows a paste and placed portions of a PCB. The models are intended to match and identify in a real-time process portions of the PCB being inspected within the regions 166, 166' 168 168'. The above description views these as instances of ROIs of true positives and true negatives at the two/four different orientations.

Referring now to Fig. 8, the image, structural, and geometry models 160-164 are inflated from the expected angle of 0 degrees. The term inflation of a model corresponds to the process of generating different but possible variations of the original model. For instance, in a region of interest the part may actually be rotated beyond its designated rotation. Also, the part in reality may be a slightly different size from the default sizes in the database. We can anticipate these variations by creating different instances of the model. Practically, we cannot generate, store, and match a model at every possible rotation or size. In the inflation step we can sample the different possible variations to generate a few models that span each variation space.

In this particular example, two new models are generated which are $\pm 10^\circ$ about the nominal angle. The sampling of each space, such as the space of possible rotations, may change according to the type of part, part size, and the known part tolerances.

Referring now to Figs.9-9C in which the image, structural, and geometry
5 models 160-164 are applied to three different cases of inspection shown in Figs. 9A-9C respectively. Note that the image model 160 contains a snapshot that is black with white writing and gray endcaps. In the first case, shown in Fig.9A, there is a good match between the image model 160 and a subregion of image 170 generated or captured by the inspection equipment. The part in image 170 is also black with white
10 writing and gray endcaps. After the image model 160 matches a subregion the captured image 170, the center of the subregion (x,y) , corresponding to the center of the part, is saved in the image processing system. (Note that we can refer to a subregion in many different ways, such as referring to its top left hand corner). The current state of the image processing system is that it has a rough hypothesis that the
15 correct part is located around position (x,y) in image 170. The structural model 162 (Fig. 9) is now used to verify that the component 172 is at the noted location (marked by a circle) and to refine the location.

In one embodiment, the structural model 162 verifies the existence of the component at the mark location by placing its collection of regions in a rigid spatial
20 configuration around a location (x, y). If the region and region relation properties are satisfied., the structural model indicates the part is present. In applications with time constraints, it may be desirable to utilize the this embodiment.

In another embodiment, the structural model may actually deform its component parts 162a – 162g and spatial positions to align with regions of the
25 component. If the structural model is able to align its regions 162a-162g without deforming any of the regions 162a-162g beyond acceptable limits, then an indication is provided that the part is present. If, on the other hand, the structural model is not able to align its regions 162a-162g without deforming any of the regions 162a-162g beyond acceptable limits, then an indication is provided that the part is not present.

30 It should be appreciated that the structural model is needed to perform a search around a relatively small area which includes the values (x,y). Such an approach is required for two reasons: (1) the image model was run in coarse mode, meaning it checked every N'th pixel in the image; and (2) even if the image model was run in

fine mode, the image model is extremely sensitive to high contrast regions such as the writing on a part. If the writing is not exactly in the middle of the part, the image model will match not to the part center, but to the center of the writing.

Once the structural model verifies the component is present in the ROI around
5 location (x,y) and refines its center position, the geometry model 166 is applied. As described above in conjunction with Figs. 1-6, the geometry model is, in essence, a sophisticated edge finder. The geometry model is used to calculate the fine dx, dy and theta values for the component 172. The dx, dy theta values represent the distance and angle by which the component 172 being inspected in the image deviates from
10 expected or ideal positional values for that component as computed via the inspection plan. It should be noted that the dx, dy and theta values are all computed with respect to the center of the component 172. It should be appreciated that the center of the component is calculated differently for different types of components. For circuit components classified as discretes, this is center of the circuit component body and
15 the endcaps. For circuit components classified as IC's only the leads matter in determining dx, dy and theta (the body is uncontrolled).

In the example shown in Fig. 9B, the image model 160 is matched to an image
180 of a component 182 captured by an inspection system during an inspection process. As described above, in this particular example, the image model 160 is black
20 with gray endcaps and the image of the component 182 is green with very bright (saturated endcaps). Thus, the image model 160 is not well matched to the image of the component 182. In this instance, the structural model 162 is used to search a region of interest (ROI) 184 to locate the circuit component 182. If the circuit component 182 is found, then the geometry model is utilized to calculate the fine dx,
25 dy and theta values for that particular circuit component. It should again be noted that the dx, dy and theta values are all computed with respect to the center of the component 182.

In the third example shown in Fig. 9C, the image model is once again not well
matched to the image captured by the image inspection equipment. In this case, the
30 image model is not well matched because the circuit component is missing and the pasted pads and circuit board background in image 186 are significantly different in color and luminance from the image model. This indicates to the system that either the component is missing or a component which looks different than the image model

is present. Thus, as shown in Fig. 9C, the structural model 162 is again used to search the whole region of interest 186 to find the component. In this particular example, the structural model 162 does not find the circuit component within the region of interest 188 and thus the inspection system determines that the part is absent from the location on the printed circuit board at which it should be found.

It should be noted that in some cases the image model may match well to an image where the component is absent. In this case the pasted pads and background may look very much like the image model snapshot. The same processing as described with Fig 9B would occur. The structural model would do a search around the most likely center position of the part, provided by the image model. It is unlikely that the image would match the stringent specifications in the structural model. The structural model, thus, would declare the part absent.

Referring now to Figs. 10-10D, a plot of the structural model scores on instances of paste and placed images for a parts of a particular package type, CC0805. The plot also indicates whether the instance was identified as having the component present or the component absent.. In the plot, two groupings 190, 192 of identification points are shown. Grouping 190 indicates the part was absent and grouping 192 indicates the part was present. In this particular example, the score versus instances indicates that the models were able to accurately distinguish placed parts from paste images. All points in group 190 are true negatives and all points in group 192 are true positives. For instance, in figure 10A and 10B image 196 and image 198 were analyzed by the system correctly labeled as having the part absent because they had scores in group 190. In Figs. 10C and 10D, images 200 and 202 were analyzed by the system and were correctly labeled as having the part present because they had scores in the group 192. As shown in figure 10, there are 197 paste instances and 788 place instances that were analyzed. The wide separation 194 between the paste and placed instances 190, 192 respectively indicates that the system is able to confidently distinguish between these two groups.

Referring now to Figs. 11 and 11A, a plot of structural model score versus instance is shown for a series of placed parts and paste parts of package type RC1206. As can be seen from Fig. 11, one placed parts grouping 204 is highly separated from a paste parts grouping 206. Another placed parts grouping 208, however, is in a region which does not have good separation from the paste parts grouping 206 and the placed

parts grouping 204. The components which resulted in the grouping 208 are those components which the structural model could not confidently identify as placed images.

The results which occur in the region 208 between the lower and upper regions 204, 206 correspond to components which the structural model could not identify as either a paste or a placed part. In this particular example, the package type is a so-called RC1206; a resistor of size 120 by 60 mils. Thus, the particular structural model being used for package type RC1206 are not accurately identifying the set of parts in group 208.

The images denoted in group 208 were analyzed to look for any commonality. The resulting analysis found that all images from group 208 were from the particular reference designator R29 on the different instances of the printed circuit board.

Since the structural model being used for package type RC1206 are not accurately identifying a particular part at reference designator R29, a new structural model is generated for this particular part at this particular location on the board. Once the new model for package type RC1206 and reference designator R29 is used, as can be seen in Fig. 11A, good separation is achieved between the placed and paste parts as indicated by the position of grouping 210. Group 210 shows the scores generated by the new structural model for the images of placed parts at reference designator R29. Thus, Fig. 11A shows that the identification scores for both the R29 model (group 210) and the non-R29 model (group 206 for paste images and group 204 for placed images) result in good separation between the paste and placed parts.

Figs. 11 and 11A illustrate that by examining clusters of example images, it is possible to identify components which require specialized or specially trained models for detection and recognition. For example, steps 100-108 described above in conjunction with Fig. 4 illustrate a technique of generating a separation plot, which is a measure of how well the models can discriminate between positive and negative examples, for a set of paste and placed images. Figure 4 discusses how to choose the best set of models that gives a good separation between paste and placed images.

Note that at the end of the processing in Fig 11A, we have added a new structural model for part type RC1206. This means that two structural models are associated with this part type. It is possible to use the same process to determine if we what to create a new image or geometry model.

Referring now to Fig. 12, an image model match to a component identified as

an RC0805 is shown. Again, in this particular example, there is good separation between the paste and placed part groupings 214, 216 thus indicating that the image model can correctly distinguish between a paste and a placed part. This analysis shows that for this part type, the matching method will most likely follow the flow as described in figure 9A where the structural model is used more for verification than
5 part detection and location identification.

Figure 13 shows the image model scores for the paste and place images of the type RC1206. The paste scores are shown as group 218 and the place scores are denoted by dashes in Figure 13. In contrast to the results achieved in accordance with the techniques of the present invention and with reference to Fig. 13, it should be
10 appreciated that when prior art techniques are used (i.e. using only an image model to identify and distinguish placed and paste regions of a printed circuit board), little or no separation occurs between the paste and placed images as shown in Fig. 13. Note that the process described in the current invention is able to compensate for the limitations of the image model by using the structural model. The structural model does provide a
15 good separation between paste and place images as already shown in figure 11A.

This analysis shows that for this part type, the matching method will most likely follow the flow as described in figure 9B and 9C where the structural model is used for part detection and location identification.

Referring now to Figs. 14 – 14B, a technique for learning a model which provides good classification of images for a part type is shown. The process begins with step 230 in which a model for a part type is selected from a set of model types. The model can be an image model, a structural model or a geometry model. While each of these models may be learned independently, we currently learn an image and a
20 structural model together.

As show in step 232, the model is applied to all “placed images” of the same part type. A “placed image” refers to an image in which the object being inspected is in one of a range of expected locations in the image. Processing next proceeds to step 234 where a “placed image score” is computed between the selected model and each
25 “placed image” in the region of interest (ROI). Each “placed image score” is a value which represents or indicates the goodness of the match between the model and a particular placed image.

Processing then proceeds to steps 236 and 238, in which the same model is

applied to all paste images of the same part type and a “paste image score” is computed between the selected model and each “paste image” in the region of interest (ROI).

Once the placed image scores and paste image scores are computed they are saved for use in later processing. Next, in step 242 a check for “outlier scores” (or more simply “outliers”) is performed. The term “outliers” refer to those placed and paste image scores which appear to be well outside the typical range of values for the placed image scores and the paste image scores. If an outlier is identified, then the reason for the outlier should be determined. That is, if the outlier score occurred due to an anomalous situation or characteristics which is not expected to be repeated, then the outlier point should not be included in a computation of a distribution of the scores. If, on the other hand, the outlier score occurred due to a situation or characteristics which is expected to be repeated, then the outlier point should be included in a computation of a distribution of the scores. For instance, a discrete component with paste on the endcaps will provide a score that is an outlier. We would want to eliminate this score from the set of good placed scores because it is an actual defect. Also, if a placed image is inaccurately labeled, meaning the expected object is not in the image, we would want to remove the score associated with this image from the set of place scores. On the other hand, if a discrete component generally has a black body and there occurs a valid instance of the part with a white body (or a drastically different appearance), we would want to include this valid instance in the distribution of good placed scores

Referring briefly to Fig. 14B for example, two histograms of number of occurrences of scores is shown. Each element on the X axis denotes a range of scores. The Y axis denotes the number of appearances of that score or range of scores.. Curve 254 corresponds to a gaussian fit to histogram of the placed image scores and curve 256 corresponds to a gaussian fit to the histogram the paste image scores. Point 258 represents an outlier on the placed image score. That is, point 258 corresponds to a point which was not included in the computation used to produce curve 254. Similarly, point 260 represents an outlier on the paste image score and thus point 258 corresponds to a point which was not included in the computation used to produce curve 256.

Referring again to Figs. 14 and 14A, processing proceeds to step 244 in which

a separation function is computed. The manner in which the separation function is computed depends upon a variety of factors including but not limited to the type of model which was selected in step 230, the characteristics of the placed and paste images and the particular type of application in which the model is being used. For, example in an application such as a printed circuit board inspection process where the model corresponds to an image model, the separation function may be generated from scores of a correlation function. The scores may be generated from any model matching method. For instance, the model may be of a face with a complex matching function. The resulting scores may be input into the same process to generate a separation function as described in step 244.

The separation function in this case tries to fit a gaussian curve to the positive examples and another gaussian curve to the negative examples. It is possible that one gaussian may not be sufficient to achieve sufficient curve fitting. Several gaussians may thus be required to approximate the data in each class. For instance, the paste examples may produce a bi-modal distribution if the PCB has two very distinct background colors. There are several clustering algorithms, such as K-means, well known to those of ordinary skill in the art that are suitable for this purpose. Another way to compute a separation function is to find the best curve that fits the data. This assumes that we do not know or we are not imposing the distribution function.

Given a new data point, one can lookup the value of the curve at that point. Based on the calculated value and the nature of the distribution, one can compute the probability or likelihood that the data point belongs to that class. In some cases the point may fall in the intersection of two or more distributions. In this case, we can compute how likely the point belongs to each distribution. In the simplest case, we would label the point as belonging to the distribution with the highest likelihood. We can however, report this diagnostic with a low level of confidence.

After the separation function is computed, processing flows to decision block 246 in which decision is made as to whether the separation function is acceptable. Again, the manner in which this decision is made and the factors considered in making the decision depends upon a variety of things. For example, in an application such as a printed circuit board inspection process where the model corresponds to an image model and the separation function includes correlation functions, the decision as to whether the separation function is acceptable may be made by computing the

difference between minimum and maximum correlation values and comparing the difference value to a predetermined threshold value.

In other applications, however, the separation function could be how well the relative relations of the regions and the region properties match the structural model or
5 how well a gradient pattern matches an edge model.

It should be appreciated that the process is essentially the same for any application in which a score is generated for positive and negative examples. It is possible, however, to have multiple score outputs. In this case, a gaussian of a higher dimension is required to approximate the data.

10 If in decision block 246, it is determined that the separation function is acceptable, then processing proceeds to step 248 where the model is stored for further use. If, on the other hand, it is determined in decision block 246 that the separation function is not acceptable, then processing proceeds to step 250 where the model is identified as one that will not be used (or the model can simply be marked as a model
15 that will be disposed of). This "bad" model can be used to benchmark future models.

Processing then proceeds to step 252 where decision is made as to whether any other models of the same type to evaluate exist. If there are no more such models, then processing ends. If, on the other hand, it is determined in decision block 252 that other models of the same type to evaluate do exist, then processing flows to step 253
20 in which the next model type is selected and then processing flows back to step 232. At step 232, the processing steps described above are again repeated.

The question whether there are any other models may be interpreted several ways. In the most limited scope, the question is asking whether there are any other models of the same type. For instance, if the model type is an image model, then the
25 question is are there any other correlation images to be tried against all the data. If the question is asking whether there are any other models for that part or part type, the next model type could be a structural or geometry model.

The learning method or decision function generator as described in figures 14-14B is a method that uses all true positives and true negatives that are available. This
30 decision function may be refined over time as new examples are seen. The refinement of the model may occur, for instance, in the background learning step of module 70 in Figure 2.

This learning method is different from that described in figures 3A and 4. This

method uses all data available to generate the models. Figures 3A and 4 are focused on getting the most representative data to train the models.

It should be noted that in Figs. 1 and 14A only two types of diagnosis are assumed to be available: part there, part not there. We may, however, have other
5 classes that we would like to represent such as part damaged, wrong part, paste smudged. We can either compute a distribution of scores for each of these labeled images or generate a more complex function to classify a new image, if the new measured image falls in between the true placed and the true placed.

Before describing the techniques and systems of the present invention, some
10 introductory concepts and terminology are explained. The term “unpopulated,” is used hereafter to describe a circuit board to which no circuit components have been applied. The unpopulated circuit board, as used herein, can either include no solder paste, or can include solder paste. Solder paste will be understood by one of ordinary skill in the art to be a liquid solder that is applied to only certain pads or regions of a
15 printed circuit board. The solder paste is often applied using a silkscreen process. The pasted regions correspond to locations at which circuit components will be first placed and then soldered at subsequent manufacturing steps. While surface mount circuit boards to which solder paste is applied will be described in the following discussions, it should be appreciated that the invention can be applied equally well to
20 through hole circuit boards that undergo a different soldering process.

Conversely, the terms “populated” and “partially populated” are used hereafter to describe a circuit board to which circuit components have been applied.

The dynamic color identification process makes use of a recognition that an unpopulated circuit board has a finite number of distinct colors distributed about the
25 surface of the unpopulated circuit board. In many cases, the finite number of distinct colors is a relatively small number of distinct colors. Using technology, unpopulated circuit boards are fabricated with a variety of materials and a variety of processes. For example, unpopulated circuit boards can be fabricated from paper composites, fiberglass, and poly-tetrafluoroethylene (PTFE). A dominant color of the unpopulated
30 circuit board is often associated with a solder mask layer that is deposited on both outer surfaces of the unpopulated circuit board during its manufacture. The solder mask layer is provided in a variety of colors, including, but not limited to, blue and green. The solder mask layer is applied to most of the surface of the unpopulated

circuit board, including all areas of the unpopulated circuit board that do not receive solder paste. The solder mask is not applied to areas of the unpopulated circuit board that do receive the solder paste, the solder paste areas provided to attach circuit components. The solder mask is also not applied to unpopulated circuit board areas
5 that must otherwise be exposed, for example connector pads. In addition to the above colors, a silkscreen is often applied to the surface of the unpopulated circuit board on the surface of the solder mask, the silkscreen having alpha-numeric reference designators and sometimes also body outlines corresponding to circuit components. The silkscreen is generally white, but can be a variety of colors.

10 The colors on a printed circuit board can be classified into a finite number of color categories. Thus, regardless of the material of the circuit board, the color of the solder mask, the color of the areas that receive the solder paste, the color of the otherwise exposed areas, and the color of the silkscreen, the colors associated with the surface of a particular unpopulated circuit board can be grouped into a relatively small
15 number of color classes. Using a fiberglass unpopulated circuit board having a green solder mask and a white silkscreen as an example, the unpopulated circuit board is generally green over the surface of the circuit board where the solder mask has been applied. Some otherwise exposed areas can be copper colored, corresponding to copper conductors or layer, silver, corresponding to solder plated conductors or layer,
20 or gold, corresponding to connector pads or the like. Of the areas to which solder mask has been applied, there are generally two shades of green. A first shade of green, corresponding to a light board color, is associated with a region under which a copper or solder plated conductor or layer is either at or near the surface of the unpopulated circuit board, visible below the solder mask. A second shade of green,
25 corresponding to a dark board color, is associated with a region under which there is no copper or solder plated conductor at or near the surface of the unpopulated circuit board, visible below the solder mask. The variety of unpopulated circuit board colors correspond to "color categories."

It will be recognized that a particular color category corresponding to a
30 particular individual circuit board can vary over the surface of the circuit board. For example, in some processes, the solder mask is applied in a silkscreen process. It should be noted that this solder mask silk screen process is different than the silkscreen process described above in conjunction with providing lines on a printed

circuit board for reference designators or circuit components. As the silkscreen solder mask is applied, there can be variations in the thickness of the solder mask, resulting in color variations. However, these color variations are generally small.

5 The exemplary fiberglass unpopulated circuit board, having a green solder mask and a white silkscreen can have different colors from circuit board to circuit board, and in particular, from one production lot to another production lot. For example, the solder mask can be a different shade of green from one unpopulated circuit board to another unpopulated circuit board within a production lot of the same unpopulated circuit boards. While the unpopulated circuit board material, here
10 fiberglass, is often specified by the designer of the unpopulated circuit board, the solder mask is often not specified. Thus, in another example, the solder mask can change from one unpopulated circuit board production lot to another unpopulated circuit board production lot. For example, the solder mask can be green in one production lot, and unless otherwise specified, the solder mask for the same
15 unpopulated circuit board can be blue in another production lot. Regardless, the number of colors associated with any one unpopulated circuit board remains a relatively small number, regardless of the color of the materials used to construct the circuit board.

The present technology of unpopulated circuit board manufacture known to
20 one of ordinary skill in the art, includes a variety of process steps and a variety of materials. It is expected that future advancements in the art of unpopulated circuit board manufacture will result in circuit boards also having a relatively small number of colors. The dynamic color identification process is not limited to current unpopulated circuit board manufacturing methods or materials.

25 Referring now to FIG. 15, a technique for processing a printed circuit board includes a group of steps 302 in which certain regions of the circuit board, referred to as palette regions, are selected and characteristics of the selected regions are measured, sampled or otherwise obtained. It should be appreciated that in the description provided herein below, the circuit board characteristic of interest is color
30 and to promote clarity in the description, the processing described below will sometimes make specific reference to palette regions as "color palette regions" and the processing of the circuit board characteristic will be explained in the context of a color characteristic. It should be understood, however, that other circuit board

characteristics, including but not limited to texture and luminence characteristics of the circuit board can also be used instead of or in conjunction with color.

It will be appreciated from the discussion below, that the colors of a circuit board in the color palette regions can be later used by a variety of circuit board inspection models including but not limited to those models described above in conjunction with FIGS. 1-14. Some of the models may, for example, determine if circuit components are disposed at each location where a component should be disposed on the circuit board and in some cases, to determine if the circuit components are properly disposed on the printed circuit board. The circuit board being processed can be either a paste, placed, unpopulated, or reflowed circuit board.

At step 308, color palette regions on the circuit board are identified. Each of the palette regions is preferably selected such that each palette region corresponds to one of the plurality of different colors on the circuit board. One particular example of the color palette regions is shown in association with FIG. 16A. Suffice it here to say, however, that each of the color palette regions can correspond to a relatively small area of the circuit board. Preferably, each of the color palette regions is selected such that each region is a different color. In this way, each of the plurality of color palette regions has a color which corresponds to a different one of the plurality of color categories described above. In a preferred embodiment, the color palette regions are selected to be in locations that will not receive circuit components at later assembly steps. One way to ensure this is to select the color palette regions on a populated printed circuit board.

In one particular embodiment related to inspection of circuit boards, five color palette regions are selected. Each of the five color palette regions can correspond to one of the following color categories: (1) a bare pad color; (2) a dark board color; (3) a light board color; (4) a silkscreen color; and (5) a paste color. From the discussion above, it will be recognized that the bare pad color corresponds to areas of the unpopulated circuit board that are otherwise exposed and not covered with solder mask. It will also be understood that the dark board corresponds to regions under which there is no copper or solder plated conductor at or near the surface of the unpopulated circuit board, visible below the solder mask. It will also be understood that the light board color corresponds to regions under which a copper or solder plated conductor or layer is either at or near the surface of the unpopulated circuit board,

visible below the solder mask. It will still further be understood that the silkscreen color corresponds to the color of the silkscreen, and the paste color corresponds to the color of the solder paste.

While five color categories corresponding to five color palette regions are
5 described above, it will be recognized that fewer or more than five color categories, each associated with a color palette region, can be used with this invention. Also, while the color palette regions corresponding to the bare pad color, the dark board color, the light board color, the silkscreen color, and the paste color are described, the color palette regions can correspond to any color categories that distinguish the variety
10 of colors associated with the unpopulated circuit board.

It should be appreciated that the number of color categories and the number of palette regions to use in an particular application depends upon a variety of factors including but not limited to the number of different colors in the piece being inspected, the variation (e.g. shade between colors) and the significance of the
15 different colors and shades of colors.

At step 310, which can be provided as an optional step, the color palette regions are scanned by an image processing system which may, for example, be similar to the imaging processing system 20 of FIG. 1. The image processing system optically scans the circuit board, thereby providing one or more pixel values
20 associated with each of the color palette regions. From the pixel values, the image processing system generates an initial palette value associated with each of the color palette regions. Each respective initial palette value can correspond to a color vector having conventional red, green, blue (RGB) values. Other color coordinate systems well known to those of ordinary skill in the art, may of course, also be used. Step 310
25 thus provides the palette value associated with each of the color palette regions.

It will be appreciated that the color palette region has a color palette region size that can include one or more pixels associated with the image processing system. It will be further appreciated that each of the one or more pixels is associated with a pixel value corresponding to a color vector having convention red, green, blue (RGB)
30 values. In one exemplary embodiment, each respective initial palette value is an average of the pixel values over the associated color palette region.

The color palette regions having initial characteristic palette values are each assigned a color category corresponding, for example, to the light board color, the

dark board color, the silkscreen, the bare pad color, and the paste color described above. All of the selected color palette regions must be associated with a color category before processing continues. The initial palette values, each associated with a corresponding color category, should comprise a set such that one could recreate an accurate image of the circuit board using only this set of colors.

At step 312, a “paste” circuit board, corresponding to an unpopulated circuit board having solder paste disposed thereon, is scanned by the image processing system. The palette regions (in this example color palette regions), having been selected at step 308 are used to dynamically generate another group of palette values, referred to herein as paste palette values (i.e. palette values as obtained from a paste circuit board). It will be recognized that the paste palette values need not be the same values as the initial palette values generated at step 310. However, the location and size of the color palette regions identified at step 308 remains unchanged. The paste palette values are each associated with the color categories assigned at step 310.

At step 314, a paste circuit board, is “learned.” As applied to a paste circuit board, the term “learned,” is used to describe a process by which a circuit board is examined by an image processing system to establish the circuit board colors at particular regions corresponding to regions of interest (ROIs), wherein the ROIs correspond to circuit board locations at which circuit components will subsequently be placed. The process of steps 312 and 314, collectively denoted as steps 304, will be further discussed in association with FIG. 17. Let it suffice here to say that at step 314 the ROIs are examined to determine the color at the ROIs, wherein the ROIs have no circuit components. Also in step 314 “paste grid values” are associated with each ROI.

At step 316, a “placed” circuit board is scanned by the image processing system. A placed circuit board (also referred to as a populated circuit board) is one on which circuit components are placed in their respective positions on the circuit board. The circuit components can be held in place by the solder paste previously applied to the solder pads upon which the circuit component is mounted, or additionally held in place by epoxy applied to the circuit board at an assembly step. The placed circuit board has not yet reached a manufacturing step in which the solder paste is heated, thereby solder bonding the circuit components to the circuit board. While the paste and the placed circuit board are described below, it will, however, be understood that

the techniques described herein can be applied to a reflowed circuit board after the soldering.

The color palette regions, having been selected at step 308 are again used to
5 dynamically generate yet another group of palette values, referred to herein as placed
palette values. It will be recognized that the placed palette values need not be the
same values as the initial palette values generated at step 310, nor the same as the
paste palette values generated at step 312. However, as described above, the location
and size of the color palette regions identified at step 308 remains unchanged. The
10 placed palette values are each associated with the color categories assigned at step
310.

At step 318, the placed circuit board is learned. As applied to the placed
board, the term “learned,” is used to describe a process by which a populated circuit
board is examined to establish the circuit board colors at the ROI locations established
15 above at step 314. At step 318, the ROIs correspond to circuit board locations at
which components have been placed. Thus, in step 318, the ROIs are examined to
determine the color values at the ROIs, wherein the ROIs have circuit components.
Also in step 318, “placed grid values” are associated with each ROI.

At step 320, a circuit board to be inspected, herein referred to as an inspection
20 circuit board is scanned by the image processing system. An inspection circuit board
can correspond to a paste, a placed, unpopulated or reflowed circuit board. The color
palette regions, having been selected at step 308 are yet again used to dynamically
generate yet another group of palette values, referred to herein as inspection palette
values.” It will be recognized that the inspection palette values need not be the same
25 values as the initial palette values generated at step 310, nor the same as the paste
palette values generated at step 312, nor the same as the placed palette values
generated at step 316. However, as described above, the location and size of the color
palette regions identified at step 308 remains unchanged. The inspection palette
values are each associated with the color categories established at step 310.

30 At step 322, the palette values generated are used in an inspection process and
in particular, can be used in an inspection process which utilizes one or more of the
image models, the structural models, and the geometric models described above in
conjunction with FIGS. 1-14. The palette values can also be utilized with a “negative

model” FIG. which uses the palette values explicitly to determine if components are absent and which is described below in conjunction with FIG. 17A.

Decision block 324 implements a loop in which steps 320 and 322 are repeated until all circuit boards are inspected.

5 Referring now to FIGS. 16 and 16A in which like elements are provided having like reference designations, a conventional placed circuit board 320 includes a bare pad region 322 having a bare pad color, a dark board region 324 having a dark board color, a light board region 326 having a light board color, a paste region 328 having a paste color, and a circuit component 330, having a circuit component color.
10 In FIG. 16A, a silkscreen 334 has been included for illustrative purposes.

Color palette regions 350a-350e corresponds to selected regions of the circuit board 320. In one embodiment, the color palette regions are selected manually by a user viewing a printed circuit board and identifying locations on the printed circuit board which are representative of each of the colors on the printed circuit board. The
15 color palette regions 350a-350e are thus selected to provide an indication of all colors associated with the circuit board 320.

In this particular example, the color palette region 350a corresponds to the bare pad region 322 having the bare pad color, the color palette region 350b corresponds to the dark board region 324 having the dark board color, the color palette
20 region 350c corresponds to the light board region 326 having the light board color, and the color palette region 350d corresponds to the silkscreen 334 having the silkscreen color. The circuit component 334 is associated with an unspecified “other” color.

It should be appreciated that although the color palette regions 350a-350e are
25 here all shown as having a rectangular shape, the color palette regions 350a-350e can be provided having a variety of sizes and a variety of shapes. It should also be appreciated that although the color palette regions are here shown in particular locations of the printed circuit board, the color palette regions can be located at any position on the printed circuit board as long as each color on the printed circuit board
30 is represented by one of the color palette regions. It should also be appreciated that although six color palette regions are shown in this example, the number of color palette regions used in any particular application is selected based on the number of

colors which must be represented. Thus, fewer or more than six color regions may be used.

In particular, the number of color palette regions 350a-350N having corresponding color categories is selected in accordance with a variety of factors, including, but not limited to, the number of distinct colors associated with the unpopulated circuit board, the subsequent processing load, and variation of a particular color category across a particular circuit board, such variation described above as being generally small. While in this particular embodiment, five color palette regions 550a-350e corresponding to five color categories are shown, in other embodiments, more than five or fewer than five color palette regions can be associated with respective color categories. It should be recognized that a color category "other" can corresponds to any color that is not associated with the unpopulated circuit board.

The shape of the color palette regions 350a-350N is selected in accordance with a variety of factors, including, but not limited to, the mechanical characteristics of the image processing system. The shape of the color palette region should be selected to ensure that the region encloses a single board color type.

The size of the color palette regions 350a-50N is also selected in accordance with a variety of factors, including, but not limited to the size of the circuit board feature to which a color palette region is applied. The size of the color palette region 350a-50N must be sufficiently small so as to encompass only the particular circuit board feature for which a color is desired. It is also desirable that the size of the color palette region 350a-50N be sufficiently large so that some averaging of color occurs over the pixels associated with the color palette region 350a-50N. The size of the palette regions can be selected having a variety of sizes in order to ensure that the selected color palette region encloses a single board color type and that the color palette region is large enough to get an average color over a sample of one or more pixels.

The locations of the color palette regions are selected in accordance with the position of circuit board features that can characterize each of the color categories, which in combination can describe a full range of colors associated with the unpopulated circuit board.

The color palette regions 350a-350e are processed in the way described in steps 304-308 of FIG. 15. In particular, at step 306 of FIG. 15, a color palette value is measured for each of the respective color palette regions 350a-350e. As described above, the color palette value in each color palette region can be expressed as an RGB color vector.

Thus, the color palette regions 350a-350e correspond to locations on the printed circuit board at which colors represented by the color categories, described above, are expected to be present on an unpopulated circuit board, or on unpopulated portions of the populated circuit board 320. As described above, it should be appreciated that there are only a relatively small number of color categories and thus a relatively small number of color palette regions 350a-350e associated with the populated circuit board 320. Also, it should be appreciated that, as described above, the specific colors of the color categories are not important for the purposes of this invention.

Referring now to FIG. 17, a process for learning the color layout of the unpopulated/pasted board at each ROI using dynamically generated palette characteristics (e.g. palette colors) is shown. This learned color layout is then used for inspection as described in Figure 17A. The process begins in processing block 406 in which measurements are dynamically made at palette regions on the printed circuit board. It should be appreciated that the term “dynamic” is used herein to describe techniques in which each circuit board is individually characterized by measuring/generating the characteristic palette values associated with the color palette regions, and wherein for a given type of printed circuit board, the color palette regions are constant in size, position and quantity from board to board.

It should be appreciated that the size, shape, and locations of the color palette regions can be selected as described above in conjunction with Fig. 15. In a preferred embodiment, an image processing system dynamically measures the circuit board characteristic (e.g. color) of an unpopulated printed circuit board in each of the palette regions. It should be understood, however, that while in this embodiment measurements are made on an unpopulated printed circuit board, similar measurements could also be made on a bare circuit board or even on a populated circuit board. In one embodiment, an image of the printed circuit board is obtained

(e.g. via an image processing system such as imaging processing system 20 described above in conjunction with FIG. 1) and the measurements are taken from the image.

In processing block 410, characteristic palette values are generated. In one embodiment in which color is the characteristic of interest, the palette values are
5 generated by representing the pixel data as a color vector. For example, dynamically obtained color palette values can be represented as conventional red, green, blue (RGB) values. Thus the palette values can be generated by converting or transforming the pixel data into color space values (e.g. values in an RGB color space).

Once the palette values are determined, a semantic label (e.g. "bare pad color,"
10 "dark board color," "light board color," "silkscreen," "paste") is assigned to each palette region. If the characteristic of interest were color and the measurements were made on the circuit board of Fig. 16A, for example, region 350a (Fig. 16A) would be assigned the semantic label "bare pad color," region 350b (Fig. 16A) would be assigned the semantic label "dark board color," region 350c (Fig. 16A) would be
15 assigned the semantic label "light board color," region 350d (Fig. 16A) would be assigned the semantic label "silkscreen" and region 350e (Fig. 16A) would be assigned the semantic label "paste."

In process block 412 an image of a region of interest (ROI) on the printed circuit board is obtained. A region of interest corresponds to an image of a part and
20 the area surrounding the part on the printed circuit board. Alternatively, an ROI can correspond to an image of a location on a printed circuit board where a part to be inspected is expected to be placed and the surrounding area.

In process block 414, grid areas are identified. The grid areas correspond to particular locations or regions within a region of interest (ROI). Each grid area is
25 provided having one or more grid regions as will be discussed below in conjunction with Fig. 18.

Once the locations of the grid areas are identified, the image processing system measures the desired characteristic (e.g. color) of each of the grid regions in the grid area as shown in process block 416. Then, in process block 418 the grid region values
30 are generated. In general, the grid region values are generated by comparing the grid region values obtained in block 416 to the palette values obtained in block 410. In the case of color, the semantic label is assigned by determining which color in the palette is closest to the grid region color value. In one embodiment in which the desired

characteristic is color, the grid region values can be generated by representing the pixel data as a color vector. For example, the dynamically obtained grid region values can each be represented via conventional red, green, blue (RGB) values.

Alternatively, the dynamically obtained grid region values can be represented as a color distribution which can then be analyzed using standard techniques. The RGB values can then be used in the comparison. Alternatively still, any technique well known to those of ordinary skill in the art can also be used to represent and compare the grid region values. Thus, in the case where color is the characteristic of interest, the grid region values can be generated by converting or transforming the pixel data in the grid regions into color space values (e.g. values in an RGB color space).

Once the grid region values are determined, a semantic label (e.g. "bare pad color," "dark board color," "light board color," "silkscreen," "paste") is assigned to each palette region. If the measurements were made on the ROI 450 of Fig. 18, for example, region 452a is assigned the semantic label "light board color," region 452b is assigned the semantic label "silkscreen," regions 452c – 452e are assigned the semantic label "dark board color," and regions 452f – 452g are assigned the semantic label "silkscreen."

It will be appreciated that there can be one or more grid areas on the printed circuit board and each grid area can have one or more grid regions. It will also be appreciated that each grid region has a grid region size that can include one or more pixels associated with the image processing system. It will be further appreciated that each of the one or more pixels is associated with a pixel value having a color vector which may be represented using any conventional technique (e.g. RGB values). In one exemplary embodiment, the grid region value is computed as an average of the pixel values over the grid region. It should be appreciated, however, that any conventional technique can also be used to assign a value to a grid region.

Decision block 420 implements a loop in which blocks 412 – 418 are repeated for all of the ROIs on the printed circuit board being learned.

As mentioned above, each circuit board of a particular circuit board design can have colors that are slightly or greatly different. For example, if two circuit boards have the same circuit board design and one circuit board is fabricated with a solder mask having a first color and the other circuit board is fabricated with a solder mask having a second color, the two circuit boards will have different colors. Both circuit

boards, however, can be processed by the process of Fig. 17 since the process dynamically adjusts to the first and the second solder mask colors, and provides results that do not vary significantly between any particular circuit boards even though the circuit boards are of different colors.

5 Fig. 17A describes the process for inspecting a populated printed circuit board. It should be understood, however, that a similar process applies to inspection of a bare circuit board or even to inspection of an unpopulated circuit board. It should also be appreciated that the inspection process of Fig. 17A, is repeated for each individual printed circuit board being inspected.

10 Referring now to Fig. 17A, a process for inspecting a populated printed circuit board begins in processing blocks 422 and 424 in which an image of the printed circuit board is obtained and measurements are dynamically made at palette regions on the printed circuit board to be inspected. It should be appreciated that the size, shape, and locations of the palette regions are selected as described above in
15 conjunction with Fig. 15. Also, the palette regions are learned in accordance with the process described above in conjunction with Fig. 17. In a preferred embodiment, an image processing system dynamically measures the color (or other desired characteristic) of a populated printed circuit board in each of the palette regions.

 In processing block 426, characteristic palette values (e.g. the pixel values at
20 the palette region) for this particular printed circuit board are generated. In the case where the printed board characteristic being used is color, the palette values can be generated by representing the pixel data as a color vector. For example, the dynamically obtained characteristic palette value can be represented having conventional red, green, blue (RGB) values. Thus, the palette values are generated by
25 converting or transforming the pixel data into color space values (e.g. values in an RGB color space).

 Once the palette values are determined, a semantic label is assigned to each palette region. Again, in the case where color is the characteristic being used, the semantic labels can correspond to "bare pad color," "dark board color," "light board
30 color," "silkscreen" and "paste." If the measurements were made on Fig. 16A, for example, region 350a is assigned the semantic label "bare pad color," region 350b is assigned the semantic label "dark board color," region 350c is assigned the semantic

label "light board color," region 350d is assigned the semantic label "silkscreen" and region 350e is assigned the semantic label "paste."

As mentioned above, each circuit board of a particular circuit board design can have characteristics that are slightly or greatly different. For example, if two circuit
5 boards have the same circuit board design and one circuit board is fabricated with a solder mask having a first color and the other circuit board is fabricated with a solder mask having a second color, the two circuit boards will have different colors. Both circuit boards, however, can be inspected by the process of Fig. 17A since the process dynamically adjusts to the first and the second solder mask colors, and provides
10 results that do not vary significantly between any particular circuit boards even though the circuit boards are of different colors.

Next in process block 428 an image of a region of interest (ROI) on the printed circuit board is obtained. The regions of interest correspond to an image of a part and the area surrounding the part on the printed circuit board. Alternatively, the region of
15 interest can correspond to an image of a location on a printed circuit board where a part to be inspected is expected to be placed and the area of the printed circuit board surrounding that location.

Then in process block 430, grid areas are identified. The grid areas correspond to particular locations or regions within a region of interest (ROI). Each grid area is
20 provided having one or more grid regions as will be discussed below in conjunction with Fig. 18.

Once the grid area locations are identified, the image processing system measures color of each of the grid regions in the grid area as shown in process block 432. Then, in process block 432 the grid region values are generated. The grid region
25 values can be generated by representing the pixel data as a color vector. For example, the dynamically obtained grid region values can each be represented via conventional red, green, blue (RGB) values. Alternatively, the dynamically obtained grid region values can be represented as a color distribution which can then be analyzed using standard techniques. Alternatively still, any technique well known to those of ordinary
30 skill in the art can also be used. Thus, the grid region values are generated by converting or transforming the pixel data in the grid regions into color space values (e.g. values in an RGB color space).

Once the grid region values are determined, a semantic label (e.g. “bare pad color,” “dark board color,” “light board color,” “silkscreen,” “paste”) is assigned to each grid region as shown in step 434. If the measurements were made on the ROI 450 of Fig. 18, for example, then region 452a would be assigned the semantic label
5 “light board color,” region 452b would be assigned the semantic label “silkscreen,” regions 452c – 452e would be assigned the semantic label “dark board color,” and regions 452f – 452g would be assigned the semantic label “silkscreen.”

It will be appreciated that there can be (and typically are) one or more grid areas, each having one or more grid regions. It will also be appreciated that the grid
10 region has a grid region size that can include one or more pixels associated with the image processing system. It will be further appreciated that each of the one or more pixels is associated with a pixel value having a color vector which may be represented using any conventional technique (e.g. RGB values). In one exemplary embodiment, the grid region value is computed an average of the pixel values over the grid region.
15 It should be appreciated, however, that any conventional technique can also be used to assign a value to a grid region.

It should be recognized that, at step 434 a grid region, having an unknown grid region value determined at step 432, can be assigned a semantic value, or color category, that is not associated with an unpopulated circuit board. For example, the
20 semantic value “other” can be used to indicate a grid region value that is not recognized to be among the characteristic palette values associated with the unpopulated circuit board.

Before describing steps 436-440, it should be noted that these processing steps relate to an inspection process which utilizes a negative model. It should be
25 appreciated that the negative model does not attempt to classify the circuit component. Rather the negative model merely classifies the grid area as having a characteristic associated with a paste circuit board, or having obstructed color characteristics.

At step 436, the measured grid region values are compared to previously stored grid region values obtained from an unpopulated circuit board. In the case
30 where the inspection process of Fig. 17A is performed on a populated circuit board, comparing the grid region values from the populated printed circuit board to the grid region values from the unpopulated printed circuit board can indicate a missing circuit component on the populated printed circuit board.

If in decision block 436 decision is made that the measured grid region values correspond to the previously stored grid region values obtained from the unpopulated circuit board, then processing proceeds to processing block 440 where it is indicated that a part is absent or an unpopulated circuit board is present.

5 If in decision block 436 decision is made that the measured grid region values do not correspond to the previously stored grid region values obtained from the unpopulated circuit board, then processing proceeds to processing block 438 where it is indicated that something is obscuring the unpopulated circuit board. Thus, if the grid region values do not match, an indication that there is something in the ROI
10 which causes the pasted grid region values to be obscured is provided. The obscuring object is most likely the component. The negative model is making the decision that it "IS NOT" the paste ROI.

 It should be appreciated that in one embodiment, at step 436, the grid region values generated at step 432 for the circuit board being inspected are compared with
15 the grid region values generated at step 416 of FIG. 17 for a paste circuit board. In a preferred embodiment, at step 436, the semantic palette label provided at step 434 for the inspection circuit board are compared with the semantic labels provided for the paste board at step 418 of FIG. 17. In a third embodiment, at step 436, the grid region values generated at step 432 for the inspection circuit board are compared with the
20 characteristic palette values generated at step 426 for the inspection circuit board. In a fourth embodiment, at step 436, the semantic palette labels provided at step 434 for the inspection circuit board are compared with the semantic palette labels generated at step 410 for the paste circuit board.

 It should be recognized that the semantic palette labels, or color categories,
25 assigned to the color palette regions of the paste circuit board are the same semantic labels assigned to the color palette regions of the inspection circuit board, although characteristic palette values associated with respective color palette regions can be different for the paste and for the inspection boards. In this way, for example, a semantic value, corresponding to a color category "dark circuit board" can be blue on
30 one circuit board and green on another circuit board, corresponding to two different characteristic palette values. The absolute color does not effect the negative model.

 Decision block 442 implements a loop in which blocks 428 – 422 are repeated for all of the ROIs on the printed circuit board being inspected. Once there are no

more regions of interest to consider, then processing for that printed circuit board ends and processing for another printed circuit board can begin again at processing block 422.

Referring now to FIG. 18, an ROI of a paste circuit board 450 includes an
5 exemplary grid area 452 provided having seven grid regions 452a-452g. While seven grid regions are shown, it will be appreciated that grid area 452 can include any number of grid regions (one or more) and that the grid regions can have any size, shape or relative position. The paste grid region number, size, shape and position are selected in accordance with a variety of factors, including, but not limited to, the size
10 of an electrical component within the ROI corresponding to the grid area, the tolerance limit of acceptable placement position of the electrical component, and the number of pixels within a grid region that can be averaged to provide the paste grid region values.

The circuit board 450 also includes two solder pads 454a, 454b and two trace
15 regions 455a, 455b. The circuit board 450 also includes a board region 456 having a color previously categorized by a second dynamic palette value as a light board color, a board region 458 having a color previously categorized by a third dynamic palette value as a dark board color, and silkscreens 460a, 460b having a color previously categorized by a fourth dynamic palette value as having a silkscreen color.

20 The color categories in the grid region 452a corresponds to the light board color, grid region 452b corresponds to the silkscreen color, grid region 452c corresponds to the dark board color, grid region 452d corresponds to the dark board color, grid region 452e corresponds to the dark board color, grid region 452f corresponds to the silkscreen color, and grid region 452g corresponds to the silkscreen
25 color. . As described above, the color categories are selected by the image processing system from among the color categories associated with the color palette regions identified at step 308 of FIG. 15. The particular color categories associated with the paste grid regions 452 are generated at step 414 of FIG. 17.

It will be appreciated that one or more of the grid region 452a-452g can
30 overlap circuit board features having different color characteristics. For example, the grid region 452f overlaps the light board region 456, the dark board region 458, and the silkscreen 460b. However, a processing algorithm associated with the image processing system can provide a determination as to the most likely color category to

which the grid region 452f belongs. The processing algorithm computes the distribution of colors in the grid region. The algorithm can also compute color statistics like mean and variance in the grid region. These color properties are computed and stored for the “paste”/“unpopulated” circuit board and are compared to
5 the “placed” circuit boards during inspection to determine if the component is present. The comparison of color properties can be done using a variety of standard color metrics derived from the color distributions. Here, the grid region 452f is associated with the silkscreen color category.

Referring now to FIG. 18a, an ROI of a placed circuit board 550 includes an
10 exemplary grid area 552, provided having seven grid regions 552a-552g. The circuit board 550 also includes two solder pads 554a, 554b and associated traces 555a, 555b all having a color previously categorized. In this case, the solder pads 554a, 554b and traces 555a, 555b are categorized as a paste color. The circuit board 550 also includes a board region 556 having a color previously categorized as a light board color, a
15 board region 458 having a color previously categorized as a dark board color, and silkscreens 560a, 560b having a color previously categorized as a silkscreen color. Here, ROI 550 also includes the circuit component 570. Thus color categories in the grid region 552a corresponds to the light board color, grid region 552b corresponds to the silkscreen color, grid region 552c corresponds to the dark board color, grid region
20 552d corresponds to the “other” color, grid region 552e corresponds to the “other” color, grid region 552f corresponds to the silkscreen color, and grid region 552g corresponds to the silkscreen color.

As described above, the color categories are selected by the image processing system from among the color categories associated with the color palette regions
25 identified at step 308 of FIG. 15. Here, the grid positions 552d, 552e correspond to no circuit board color from among the dynamic characteristic palette values associated with the color palette regions and measured at step 420 of FIG. 17. Thus, the grid regions 552d, 552e are associated with color category “other.” The particular color categories associated with the grid regions 552 are generated at step 414 of FIG. 17.

30 It should be appreciated that as mentioned above, the inspection process can use a so-called “negative model.” The “negative model” models the bare board in between the component pads, the component pads, and also the surrounding regions of the board using a fixed grid as described above in Figures 18 and 18A. The

negative model uses the fact that (1) each grid region of the board can be described by a small set of colors (corresponding to pad, paste, mask on copper, mask on substrate, and silkscreen), (2) the structure of the colors at specific location of a portion of the PCB remains the same even though the absolute colors of the set may change across
5 board and (3) component appearance can be distinct from the color and structure of the bare circuit board even if the components vary in appearance across boards. This model is useful in determining presence of a component when the component does not have distinct leads on the pads and when the body of the component can be distinguished from the unpopulated board. In general, the determination of the
10 absence of a circuit component is essentially performed by determining if the colors associated with the grid regions in the grid area are colors that match the color categories of an unpopulated circuit board.

Referring now to Figs. 19 and 19A, an optical image 600 of a portion of a paste circuit board includes an image portion 602 in which no paste is present.
15 Ideally, holder paste is properly applied to all portions of the circuit board and the existence of image portion 602 thus reveals that solder paste has been applied to only a portion of the corresponding solder pad. It has been recognized that it would be desirable to modify the image to electronically fill the un-pasted image portion 602 prior to performing the processes described above in conjunction with FIGS. 15-18.
20 The un-pasted image portion 602 could otherwise be categorized as the wrong color category if processed. Thus, in FIG. 19A, the un-pasted image portion 602 has been electronically filled to provide a uniform solder pad image 606.

Referring now to FIGS. 20 and 20A, an of an ROI image 610 of a portion of a paste circuit board includes an un-pasted image portion 612 showing that solder paste
25 has not been applied to any of the corresponding solder pad. For this invention, it would be desirable to electronically fill the un-pasted solder pad image 612 prior to performing the processes described above in association with FIGS. 15-18. The un-pasted solder pad image 612 could otherwise be categorized as the wrong color category if processed. Thus, in FIG. 19A, the un-pasted solder pad image 612 has
30 been electronically filled to provide a solder pad image 616 having solder paste.

Referring now to FIG. 21, an ROI a placed circuit board 650 includes an exemplary grid area 652 provided having twenty one grid regions 652aa-652gc. The grid regions are denoted 652xy, where x corresponds to a row, and y corresponds to a

column within the grid area 652. For example, grid region 652aa corresponds to a grid region at the first column and the first row. The circuit board 650 also includes two solder pads 654a, 654b and two trace regions 655a, 655b. The color of the trace regions has been previously measured and stored as a first dynamic palette value. The trace regions are typically categorized as light board color (mask over copper). The circuit board 650 also includes a board region 656 having a color previously categorized as a light board color, a board region 658 having a color previously categorized as a dark board color, and regions 660a, 660b having a color previously categorized as having a silkscreen color. The colors in the grid regions 652aa-652gc thus correspond to respective ones of the color categories described in Table 1 below.

Table 1

Grid Region	Color Category	Grid Region	Color Category	Grid Region	Color Category
652aa	light board color	652ab	light board color	652ac	light board color
652ba	silkscreen	652bb	silkscreen	652bc	silkscreen
652ca	paste color	652cb	dark board color	652cc	paste color
652da	other	652db	other	652dc	other
652ea	paste color	652eb	other	652ec	paste color
652fa	silkscreen	652fb	silkscreen	652fc	silkscreen
652ga	silkscreen	652gb	silkscreen	652gc	silkscreen

As described above, the color categories are selected by the image processing system from among the color categories associated with the color palette regions identified at step 302 of FIG. 15. The particular color categories associated with the placed grid regions 652 are measured/generated at step 414 of FIG. 17.

It should be appreciated that the circuit component 670 is properly placed so as to be substantially symmetrically oriented with respect to the pads 654a, 654b. Thus, when heat is applied to the circuit board to melt the solder paste disposed on the pads 654a, 654b, the circuit component has a high likelihood of being properly soldered to the pads 654a, 654b.

It will be recognized that since in this particular example the circuit component 670 is symmetrical placed, the color categories in each of the columns a, b, c, of the grid area 652 are symmetrically oriented about the center column (i.e. column b of grid area 652).

While twenty one grid regions 652aa-652gc are shown, it will be appreciated that more than or fewer than twenty one grid regions can be used. The size and placement of the grid regions are selected in accordance with a variety of factors, including but not limited to the size of the circuit component and the placement tolerance of the circuit component.

Referring now to FIG. 21A, an ROI of a placed circuit board 750 includes an exemplary grid area 752 provided having twenty one grid regions 752aa-752gc. The grid regions are denoted 752xy, where x corresponds to a row, and y corresponds to a column within the grid area 752. For example, grid region 752aa corresponds to a grid region at the first column and the first row. The circuit board 750 also includes two solder pads 754a, 754b and two trace regions 755a, 755b. The color of the trace regions has been previously measured and stored as a first dynamic palette value. The circuit board 750 also includes a board region 756 having a color previously categorized as a light board color, a board region 758 having a color previously categorized as a dark board color, and regions 760a, 760b having a color previously categorized as having a silkscreen color. The colors in the grid regions 752aa-752gc thus correspond to respective ones of the color categories as shown in Table 2 below.

Table 2

Grid Region	Color Category	Grid Region	Color Category	Grid Region	Color Category
752aa	light board color	752ab	light board color	752ac	light board color
752ba	silkscreen	752bb	silkscreen	752bc	silkscreen
752ca	paste color	752cb	other	752cc	paste color
752da	paste color	752db	other	752dc	paste color
752ea	other	752eb	other	752ec	paste color
752fa	silkscreen	752fb	silkscreen	752fc	silkscreen
752ga	silkscreen	752gb	silkscreen	752gc	silkscreen

20

As described above, the color categories are selected by the image processing system from among the color categories associated with the color palette regions identified at step 302 of FIG. 15. The particular color categories associated with the placed grid regions 752 are measured/generated at step 414 of FIG. 17.

It should be appreciated that the circuit component 770 is improperly placed so as to be not symmetrically oriented with respect to the pads 754a, 754b. Thus, when heat is applied to the circuit board to melt the solder paste disposed on the pads 754a,

754b, the circuit component has a low likelihood of being properly soldered to the pads 754a, 754b.

While twenty one grid regions 752aa-752gc are shown, it will be appreciated that more than twenty one or fewer than twenty one grid regions can be used. The size and placement of the grid regions are selected in accordance with a variety of factors, including the size of the circuit component, for example the circuit component 770, and the placement tolerance of the circuit component.

As described above, the image processing system can determine whether a component is not at an ROI corresponding with the grid area 752. In addition, having the grid area 752, with a sufficient number and spacing of the grid regions 752aa-752gc, the optical inspection system can determine that a component, for example the circuit component 770, is improperly placed upon the pads 754a, 754b.

It will be recognized that, in accordance with the asymmetrical placement of the circuit component 670, the color categories in each of the columns a, b, c are not symmetrically oriented about the center column (i.e. column b). From this information alone, the optical system can determine that the circuit component is skewed upon the pads 754a, 754b.

In addition to examination of the color categories, the optical system can also use the color categories, that were measured and saved for another placed circuit board of the same design, to more accurately determine an improperly placed circuit component.

Referring now to FIG. 22, in block 780 an unpopulated printed circuit board having electrical circuit lines etched, deposited or otherwise provided thereon is provided to a solder paste station as shown in block 782. The solder paste application station may be provided for example as a screen printer or any other device well known to those of ordinary skill in the art to apply solder paste to a printed circuit board. In some embodiments, the solder paste may be applied by hand. Regardless of the particular manner or technique used to apply solder paste to the printed circuit board, the solder paste is applied to predetermined regions of the printed circuit board. The solder paste should be applied in a predetermined amount within a given range. Processing then flows to block 784 in which a solder paste inspection system inspects the solder paste applied at the predetermined regions of the printed circuit board. Figs. To determine can be made as to whether the solder paste applied in block 782 was properly applied in each of the

appropriate regions of the printed circuit board. If decision is made that the solder paste was not properly applied in one or more of the examined regions, then the printed circuit board is returned to block 782 where the solder paste is reapplied in each of the regions in which it had not been properly applied in the first instance. Thus, blocks 782 and 784
5 are repeated until the paste inspection system determines that the solder paste has been properly applied in each appropriate region.

Processing then flows to block 786 in which the printed circuit board with the solder paste properly applied thereon is provided to a component placement station. The component placement station can include a so called pick and place machine or
10 alternatively, the placement station may involve manual placement of circuit components on the printed circuit board. The decision to use automated or manual component placement techniques is made in accordance with a variety of factors including but not limited to the complexity of the circuit component, the sensitivity of the circuit component to manual or machine handling, technical limitations of automated systems
15 to handle circuit components of particular sizes and shapes and the cost effectiveness of using automated versus manual systems.

Once the circuit component is placed on the printed circuit board, processing moves through block 788 in which a placement inspection station performs an inspection of the placed circuit component. Figs. In response to the instructions from
20 placement inspection station at block 788, processing can return to processing block 782 or processing block 786 depending upon the results of the placement inspection station in block 788.

Once determination is made in block 788 that the circuit component is properly placed and no other defects are detected, processing flows to block 790 in
25 which a solder reflow station reflows the solder thus coupling the circuit component to the printed circuit board. Solder reflow station may be provided as an automated station or as a manual station. After solder reflow in block 790, processing flows to block 800 where a placement and solder joint inspection station inspects each circuit component and solder joint of interest. If no defects are detected, then processing
30 flows to block 802 where a populated printed circuit board is provided.

It should be noted that in the above description, the inspection processes described herein above including those which use the palette regions can be used at any or all of the above-mentioned inspection processes 784, 788, 780.

Having described preferred embodiments of the invention, it will now become apparent to one of ordinary skill in the art that other embodiments incorporating their concepts may also be used. It is felt therefore that these embodiments should not be
5 limited to disclosed embodiments but rather should be limited only by the spirit and scope of the appended claims.

All publications and references cited herein are expressly incorporated herein by reference in their entirety.

What is claimed is:

CLAIMS

- 1 1. A method for generating a characteristic palette for a circuit board for use in a
2 circuit board inspection system, the method comprising:
3 identifying a value range of a first characteristic on the circuit board;
4 establishing a plurality of characteristic categories for the circuit board; and
5 selecting a first plurality of locations on the circuit board with each of the plurality
6 of locations having a characteristic value which is representative of at least one of the
7 characteristic categories with the first plurality of locations corresponding to first palette
8 regions for the circuit board.
- 1 2. The method of Claim 1 further comprising inspecting a first printed circuit board
2 by dynamically measuring, at each of the palette regions, a value of the first characteristic
3 of the first printed circuit board.
- 1 3. The method of Claim 2 further comprising dynamically measuring a value of the
2 first characteristic at one or more second locations of the first circuit board with the one
3 or more second locations being different than the plurality of first palette regions.
- 1 4. The method of Claim 3 wherein each of the one or more second locations
2 corresponds to a region of interest on the circuit board.
- 1 5. The method of Claim 3 further comprising comparing the values of the first
2 characteristic dynamically measured at each of the one or more second locations of the
3 first circuit board to the dynamically measured characteristic values at each of the first
4 palette regions on the first circuit board.
- 1 6. The method of Claim 5 further comprising identifying a category for each of the
2 values of the first characteristic at each of the one or more second locations based on the
3 comparison.

- 1 7. The method of Claim 6 comparing the category for each of the values of the first
2 characteristic at each of the one or more second locations to an expected category for each
3 of the values of the first characteristic at each of the one or more second locations.
- 1 8. The method of Claim 2 further comprising inspecting a second printed circuit
2 board by dynamically measuring a value of the first characteristic at each of the first
3 palette regions on the second printed circuit board.
- 1 9. The method of Claim 8 further comprising dynamically measuring a value of the
2 first characteristic at one or more second locations of the second printed circuit board
3 with the one or more second locations being different than the first palette regions.
- 1 10. The method of Claim 3 further comprising comparing the values of the first
2 characteristic dynamically measured at each of the one or more second locations of the
3 second printed circuit board to the dynamically measured characteristic values at each of
4 the first palette regions on the second printed circuit board.
- 1 11. The method of Claim 10 further comprising identifying a category for each of the
2 values of the first characteristic at each of the one or more second locations of the second
3 printed circuit board based on the comparison..
- 1 12. The method of Claim 11 further comprising comparing the category for each of
2 the values of the first characteristic at each of the one or more second locations to an
3 expected category for each of the values of the first characteristic at each of the one or
4 more second locations.
- 1 13. The method of Claim 1 wherein establishing a finite number of characteristic
2 categories for the first printed circuit board includes establishing a finite number of
3 characteristic categories for the printed circuit board based on the value range of the first
4 characteristic identified on the printed circuit board.

1 14. The method of Claim 1 wherein the first characteristic corresponds to color.

1 15. The method of Claim 1 wherein the first characteristic corresponds to texture.

1 16. The method of Claim 1 wherein the first characteristic corresponds to luminance.

1 17. The method of Claim 1 wherein the first characteristic is a first one of a plurality
2 of different characteristics and wherein

3 identifying a value range of a first characteristic on the printed circuit board
4 includes identifying a value range for each of the plurality of characteristics on the printed
5 circuit board;

6 establishing a plurality of characteristic categories for the printed circuit board
7 includes establishing a plurality of characteristic categories for each of the characteristics
8 on the printed circuit board; and

9 selecting a first plurality of locations corresponds to selecting first palette regions
10 includes selecting a plurality of palette regions on the printed circuit board with each of
11 the plurality of palette regions having a characteristic value which is representative of at
12 least one of the plurality of characteristic categories

1