THROUGH HOLE SOLDERING SYSTEM WITH AUTOMATIC OPTICAL INSPECTION

Colamba Liyanage Sameera Chathuranga Fonseka

(138016V)

Degree of Master of Philosophy

Department of Electronic and Telecommunication Engineering

University of Moratuwa

Sri Lanka

May 2019

THROUGH HOLE SOLDERING SYSTEM WITH AUTOMATIC OPTICAL INSPECTION

Colamba Liyanage Sameera Chathuranga Fonseka

(138016V)

Thesis submitted in partial fulfilment of the requirements for the degree Master of Philosophy

Department of Electronic and Telecommunication Engineering

University of Moratuwa

Sri Lanka

May 2019

Declaration

I declare that this is my own work and this thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date:

The above candidate has carried out research for the Masters/MPhil/PhD thesis/

Dissertation under my supervision.

Signature of the supervisor: Date

Abstract

Machine vision has been widely deployed in many applications in various sectors like industrial, medical, manufacturing, agricultural, surveillance etc. Such applications consist of numerous algorithms to fulfil the ultimate requirement of the end product. The role of automatic vision for classifying the quality of solder joints in the Electronics Manufacturing Services (EMS) industry has become significant because the increasing cost of labour, skill dependency, attitudes, time variance of human operators have narrowed down the quality of their service. Several major companies develop stand-alone Automatic Optical Inspection (AOI) systems with proprietary algorithms that contains multiple cameras operating inside a specific lighting environment for the inspection of Surface Mount Devices (SMD) and Through-Hole Technology (THT) components. However, the accuracy and maturity level for the automatic inspection of the quality of solder joints have yet to reach its ultimate goal due to the complex appearance of solder joint surface. Even though, many AOI systems are available, no vision system has been developed to integrate with a soldering robotic system to provide quality classification of solder joints in real time without prior teaching of individual joints and enclosed chambers with specific lighting to operate.

In this research, a THT soldering system with an AOI and Computer Aided Design (CAD) tool has been developed to classify the quality of THT solder joints without prior teaching and specific lighting requirements. The design consists of three main stages:

- AOI system for quality classification of THT solder joints (Major Research)
- CAD tool for extracting physical parameters of each component (Minor Research)
- THT Soldering system to perform automatic soldering (Minor Research)

The AOI system mainly operates in three stages. The first stage involves with precise alignment of the Printed Circuit Board (PCB) to the origin of the THT soldering system using feature extraction and template matching techniques. This approach provide a reliable and robust PCB verification and alignment capability compared to Hough transform based alignment method proposed by Nang Seng Siri Mar. Furthermore, this methodology rendered a better outcome even in distinguishing between vias, pads and test-points in addition to conventional fiducial markings. The identification of the solder pad and the verification of the component availability is performed prior to soldering in the second stage. The automatic identification of solder pad regardless of its plated colour and surrounding solder mask colour made a significant improvement over the methodology proposed by T. Y. Ong, Z. Samad and M. M. Ratnam, based on prior teaching of individual pads using artificial neural networks. In addition to that, the implemented methodology provides online positioning accuracy calibration which is not available in any commercial soldering robotic systems. The component availability is assured by precisely segmenting the component lead top from the identified solder pad. The developed algorithms could render a better outcome even for component leads which possess a minimum colour dissimilarity with their surrounding drill-hole region. No commercial soldering robotic system is capable of verifying component availability prior to soldering. Further, the segmented lead is used to establish a relationship between prior and post soldering stages of the solder pad in order to provide a robust solder quality classification capability. The classification of the soldering quality for short circuit (solder bridging), voids inside the drill-hole, voids on solder pad and excess solder, is carried out after soldering during the third stage. The implemented algorithms could render an improved recognition rate even with applied flux, illuminated pad regions, uneven illumination distribution and shadows on the solder joint surface. Such a widely distributed quality inspection capability is not covered in any of the reported studies. Neither commercial AOI systems nor soldering robotic systems perform real time inspection of soldering quality just after the soldering of THT components. Moreover, the precise localization of defective areas inside the solder joint, enables the robotic system to perform automatic reworking on defective solder joints adhering to IPC regulations with minimum user interaction. Such automatic reworking capability is not available with any commercial soldering robotic system in the market today.

The CAD system extracts the geometrical information of components and their pads such as the component location, its orientation, size of the solder pad and drill hole, height of nearby components, the width of the connected PCB track to the solder pad from the respective CAD file and visualize the 2D view of the PCB to the user in a Graphical User Interface (GUI). This information enhances the intelligence and the situational awareness of both robotic system and AOI.

The THT soldering system is a four-axis robotic platform that performs soldering on selected solder pads through the CAD system GUI. Its operation is mainly controlled by the vision system and the information acquired from the CAD system. The implemented THT soldering system together with integrated AOI and CAD tool provides a new concept in the EMS industry by replacing the manual inspection of THT solder joints with automatic inspection and providing automatic rework capability on defective solder joints within a single platform.

The performance of the complete system was evaluated under different illumination levels, flux residues, different types of component leads, colour combinations of solder pads and solder mask colours, wide variety of solder pad neighbourhoods and a range of solder pad sizes. Each stage of the AOI was able to provide a significant improvement over the reported studies and commercial systems. The automatic identification of solder pad and the verification of component availability could provide nearly a 98% of recognition rate for both cases. However, the existence of highly illuminated pad regions and overheated solder joints surfaces degrades the performance of the classification of not soldered regions by 2% and excess solder detection by 5% respectively (subjected to a sample size of 200 solder joints). Even though, a slight performance reduction is there due to such extreme conditions, the proposed approach provides an automated solution for soldering and quality assurance within a single platform while solving several problems in the reported studies and commercial systems effectively.

Keywords:

Automatic Optical Inspection, Solder Quality Classification, Localization of Component Lead inside a THT Solder Joint for Solder Defects Classification, Fiducial Verification

To my parents, wife and sister

Acknowledgement

Even though, it is a single name that is attributed as the author of this work, there are many people without whom it could never been completed. Firstly, I am immensely grateful to my family members who encouraged me throughout the period of this immense work.

I would be obliged to my supervisor, Prof. J.A.K.S Jayasinghe for providing me the correct guidance to achieve all the milestones in this research. I am honoured to say that this work could not be successfully completed, without the insights and guidance from him.

I would like to take this opportunity to provide my gratitude to National Science Foundation (NSF) for providing me the initial funding for the implementation of the robotic system architecture. My special thanks goes to Mr. J.G. Shanthasiri, Principle Scientific Officer-NSF for his kind support throughout the period of this research.

Finally, I am very thankful to Variosystems (Pvt) Ltd for providing me the funding and creating the opportunity to fabricate and implement the entire robotic system that helped to test the implemented vision system in a real time environment. My special thanks go to Mr. Chandana Dissanayake – Managing Director (Variosystems (Pvt) Ltd) for believing in this project and providing immense support to make this project a success.

Table of Contents

List of Figures

Page

List of Tables

List of Abbreviations

AOI	Automatic Optical Inspection
BRIEF	Binary Robust Independent Elementary Features
CAD	Computer Aided Design
CCOEFF	Correlation Coefficient
CIE	Commission Internationale de I'Elcairage
CV	Connection Validation
DCT	Discrete Cosine Transform
DMA	Direct Memory Access
DoG	Difference of Gaussian
DWT	Discrete Wavelet Transform
EDA	Electronic Design Automation
EMS	Electronics Manufacturing Services
FAST	Feature Accelerated Segment Test
FCM	Fuzzy C-Means
FLANN	Fast Library for Approximate Nearest Neighbour
FOV	Field of View
GPU	Graphical Processing Unit
GUI	Graphical User Interface
ICA	Independent Component Analysis
IPC	Institute for Interconnecting and Packaging Electronics Circuits
LoG	Laplacian of Gaussian
MCU	Module Control Unit
MST	Minimum Spanning Tree
NPI	New Product Industrialization
OCP	Over Current Protection
PC	Principle Curvatures

PCA	Principle Component Analysis
PCB	Printed Circuit Board
PUS	PCB Under Solder
ROI	Region of Interest
SD	Standard Deviation
SIFT	Surface Invariant Feature Transform
SMD	Surface Mount Devices
SMT	Surface Mount Technology
SPI	Serial Peripheral Interface
SQDIFF	Squared Difference
SURF	Speed-Up Robust Features
THT	Through Hole Technology

1. Introduction

Electronics, which is a part of physics and electrical engineering, has evolved for many years in the field of engineering. This is an area of study that demonstrates how the circuits made up of various components which control the flow of electricity, can be effectively used to solve real world problems. The deployment of such electronics circuits is abundant in control systems used for industrial automation [1], consumer electronics [2], medical equipments [3], surveillance equipments [4] etc.

The application of electronic circuits plays a vital role in majority of the activities carried out in a wide range of industrial sectors. Therefore, the increasing requirement of wide variety of electronic products has escalated in a higher demand in their manufacturing process. Consequently, the role of Electronics Manufacturing Services (EMS) industry has become increasingly significant. Furthermore, this advancement in electronics industry and increasing competitiveness have reduced the time to market of a particular product. Hence, each activity related with manufacturing process has to be accelerated to cater this growing demand. However, the human deployment to accomplish this, is not a viable option due to its skill dependency, time variance, increasing cost of labour and less availability of man power. Consequently, EMS companies have deployed an extensive use of sophisticated control systems to perform tasks that are time critical and yield very precise operation like Surface Mount Device (SMD) assembling, quality inspection for soldering defects, testing for electrical defects, Through Hole Technology (THT) component soldering, Printed Circuit Board (PCB) panel separation etc. Some of these applications make use of computer vision while some do not.

Automatic Optical Inspection (AOI) systems for quality inspection of assembled PCBs and soldering systems such as soldering robots [5], wave soldering machines [6] and selective soldering machines [7] for automatic soldering, are two of the major categories of automated platforms that are extensively used in EMS companies to ramp up their production while preserving the consistency of their quality of service by means of fast, time invariant, skill independent and precise operation comparing to human deployment. However, they still require considerable amount of user interaction for a better outcome.

In this research, a THT soldering system with integrated AOI and a Computer Aided Design (CAD) tool [8-9] was developed to provide soldering and quality assurance of THT solder joints in real time basis within a single platform. Sections 1.1 – 1.3 provide an overview of the existing automatic optical inspection systems, overview of the existing soldering robotic systems and how the implemented system in this research fills the existing knowledge gap by combining two different steps in an electronics manufacturing process, THT soldering and THT solder quality inspection, respectively.

1.1 Overview of Automatic Optical Inspection Systems

AOI is one of the prominent techniques used in electronics manufacturing industry to provide non-contact testing methodology for assembled PCBs. An AOI system provides fast and accurate inspection of assembled components and their respective solder joints in a PCB to ensure that the product leaving the production line is at high quality without any manufacturing faults. However, the growing complexity of modern circuits compared to the boards were in the recent past, urges more sophisticated methodologies to inspect their quality.

The debut of surface mount technology and the ensuing reductions of the sizes of SMDs, emphasize that PCBs are getting significantly compact in size and density. Even relatively average size PCB accommodates thousands of solder joints. This results in increasing its vulnerability to the existence of soldering defects. Such a growing complexity of PCBs, further implies that manual inspection is not a viable option for the foreseeable future. Even when it was an accepted approach, it was realized that it is not particularly effective because of its skill dependency, time variance and the increasing cost. Since the marketplace is requiring large volume of products nowadays, high quality products have to be brought to market very quickly through a reliable and precise manufacturing process that ensures the product quality remains high. AOI systems provide significant benefits to an EMS company by keeping the production costs as lower as possible through detecting the manufacturing faults early in the assembling process.

Several leading manufacturers like Nordson, Mek, Juki, Siemens have developed AOI systems with various capabilities to inspect SMD components. Some AOI systems from Mek, Nordson are even capable of providing inspection capability over THT solder joints as well. However, each of these systems requires manual teaching of every solder joint, specific lighting environments within fully enclosed chambers, multiple camera views together with proprietary algorithms to provide better results.

1.2 Overview of Soldering Robotic Systems

In modern EMS industry, automated production process is preferred due to increasing cost of labour, less availability of man power, skill dependency and time variance of manual operation. Soldering robotic systems are one of the versatile machines, which have been developed to replace the requirement of a human operator to perform soldering of THT components. These systems are primarily used to perform soldering on components with lower temperature endurance that limits the soldering through wave soldering or selective soldering machines. The Soldering robotic systems are available either of following forms in present market.

- Soldering arm is stationary and the PCB is positioned by precise X-Y positioning mechanism
- PCB is stationary and the soldering arm is positioned by precise X-Y positioning mechanism
- PCB is stationary and a robotic arm performs soldering (6 and 4 axis soldering robotic arms)

Each of these systems from reputed vendors like Japan Unix, Weller, Metcal, Unitechnologies, Janome, Thermaltronics are equipped with precise solder wire feeding mechanisms, position calibration mechanisms, precise soldering mechanisms, stable tip cleaning mechanisms and soldering iron heat controlling methodologies. Some soldering robotic systems like MSH150 MTA from Unitechnologies and TMT-R9800S from Thermaltronics have been equipped with a camera placed on the soldering head. However, it is only for visualization purpose for the user to inspect the soldering process.

Metcal, one of the most prominent manufacturers of soldering and de-soldering equipments, won 2018 New Product Industrialization (NPI) award for robotic soldering, as a result of their patented connection validation (CV) technology. However, this system performs a closed loop between the soldering system and the solder joint via the inspection of solder formation through soldering tip rather than inspecting the entire solder joint optically. Some soldering robotic systems replaces the operation of solder iron with micro-flame [10], laser [11], induction. Even though these systems include high tech electro-mechanical operations, and cutting-edge technologies like laser applications, they are still lacking of verifying the quality of their operation and performing rework on defective solder joints in real time basis.

1.3 Recent Works

Even though the commercial AOI systems contain proprietary algorithms, limited number of research works have also been reported that describe the implementation of optical inspection systems for solder joint defects classification as briefed in this section.

Nang Seng Siri Mar [12], presented an implementation of an automatic vision system for solder defect classification (both SMD and THT joints). His system uses Gabor filter, Discrete Wavelet Transform (DWT) and Discrete Cosine Transform (DCT) for the identification of soldering defects. This vision system:

- relies on gray scale image processing;
- follows edge detection based image alignment; and,
- targeted for already soldered SMD and THT solder joints and hence not suitable for a soldering system with AOI.

T. Y. Ong, Z. Samad, M. M. Ratnam, presented an implementation of an automatic vision system for solder defect classification of SMD joints using multi angle images and artificial neural networks [13]. This vision system also:

- relies on gray scale image processing; and,
- requires prior training using manually captured solder joints.

Kuk Won Ko, Hyung Suck Cho presented an implementation of an automatic vision system for solder defect classification of SMD solder joints using neural network and fuzzy rule-based classification method [14]. This technique includes the application of three neural network modules to classify solder joints by colour patterns obtained from a three-tiered circular colour illumination system. This vision system is not suitable for a soldering robot system, since the presence of three-tiered colour circular illumination system around camera makes it difficult to fix a soldering iron and a wire feeder. Furthermore, this system also requires extensive teaching of colour patterns of each solder joint.

The vision system described in [12], has been developed to inspect the quality of both THT and SMD solder joints. The lead position of a THT component plays a major role in determining the shape of the solder joint. However, this vision system does not consider the effect of the actual lead position towards the solder joint shape. Furthermore, the existence of dark regions on lead tops, leads to a similarity between the lead top and not soldered areas inside drill-hole. These two deficiencies may result in a significant increase of false detection rate. Further, this system processes the entire area of the THT solder joint and intended for final quality analysis (after cleaning flux residues) whereas a soldering robot needs an AOI which can classify the quality of fresh solder joint that contains a significant amount of flux around the solder joint and burnt flux residues on the solder joint surface. The larger the area of processing, the more the false detection rate due to presence of flux residues.

The vision systems described in [13] and [14] have been primarily designed for SMD joints. Since, the entire joint is processed for quality classification, both of them are not suitable to use for a THT soldering robot AOI due to the presence of flux residues. The application of gray scale images in the vision systems described in [12] and [13], leads to a significant loss of meaningful data. Hence the impact of the colour of solder pad (gold and tin plated) and PCB surface (green, blue, red, black) cannot be effectively used to reduce the false detection rate. Since each vision system described in [12], [13] and [14] process solder joints, they cannot provide a complete vision capability starting from the verification of actual pad to be soldered up to the verification of component availability prior to soldering. Furthermore, none of these vision systems do not compare the solder coverage percentage based on the class to which the PCB belongs as defined in the standards provided by the Institute for Interconnecting and Packaging Electronics Circuits (IPC) [15-16].

As described earlier in this section, there are many stand-alone AOI systems with proprietary algorithms. However, such systems are not suitable for soldering robot AOI as they:

- require individual teaching of each component and their solder joint;
- require specific lighting environments with fully enclosed chambers; and,
- have been designed to inspect the quality of completed PCBs.

The commercially available soldering robotic systems:

- do not contain in built AOI system;
- do not provide automatic rework on defective solder joints;
- require initial learning of position and soldering direction for a particular solder pad;
- are not capable of estimating the preheat time for a solder pad automatically;
- are not capable of providing the automatic assurance on the availability of the component to be soldered; and,

- require considerable level of user skill on machine operation and soldering programme development for a given PCB.

As a result of the absence of above capabilities in current research works and commercially available soldering robotic and AOI systems, a THT soldering system with an integrated AOI platform was developed that address to above problems effectively.

1.4 Original Contribution of the Research in Knowledge Gap and Industry

In this research, a THT soldering system with an integrated vision solution that requires a single camera view acquired from orthogonal direction to the placed PCB, was implemented to provide a complete solution to problems described in Section 1.3. This vision system has following three stages:

Fiducial Alignment Stage: *Aligns the PCB to its origin automatically and computes relative distances to each object on the PCB*

Solder Pad and Component Availability Verification Stage: *Identifies the solder pad automatically and the presence of the component before soldering*

Quality Classification Stage: *Performs automatic quality classification of solder joints*

This vision system operates with the assistance of a CAD system developed during this research which extracts all the physical information like location of the components and their respective solder pads, pad diameter, drill-hole diameter, height of the neighbouring components (if there are any) and the width of the connected PCB track to each solder pad. This information increases the intelligence and situational awareness of the entire system significantly.

In Section 1.3, several drawbacks have been discussed related to the reported studies and commercial AOI and soldering robotic systems. However, each of those issues could be effectively solved during the implementation of this system as described below.

- Replaces gray scale image processing described in [12-13] with colour image processing algorithms: *The application of colour image processing algorithms preserves meaningful data on the image to implement a robust and precise AOI system that requires no prior teaching of THT solder pads/solder joints.*

- Replaces edge detection based image alignment described in [12] using feature matching and image localization: *The alignment of a PCB to the soldering system origin is carried out by verifying the existence (using feature matching) and localizing the position (using image localization) of previously learnt model images and computing relative distances to each object on the PCB based on these positions.*

- Considers the impact of colour of solder pad and PCB surface (this was not analysed in [12-14]): *The most suitable colour models for different solder pad and PCB surface colour combinations were found that lead to a very precise segmentation of solder pad or joint from its immediate PCB background.*

- Replaces manual calibration required in commercial soldering robotic systems through on line calibration of positioning accuracy: *The identification of the pad to be soldered from its neighbours lead to perform online positioning offset adjustment with respect to the centre of camera FOV.*

- Considers the effect of the component lead (not took the advantage of THT component lead in [12-14]): *The precise segmentation of the component lead from bare solder pad is performed and localized its position after the soldering process to predict the shape of the solder joint, to minimize its likelihood of being detected as a defect and to assure the availability of the component prior to soldering.*

- Considers the effect of flux residues and shadows (not analysed in [12-14]): *The immediate surroundings of the segmented not soldered regions are processed for* *the availability of solder to minimize the effect of flux residues and shadows on the solder joint of being detected as defects of less solder.*

- Consider the effect of illuminated regions on the solder pad (not analysed in [12-14], since they process completed solder joints): *Mitigate the impact of highly illuminated regions on the solder pad on the accuracy of not soldered region classification through early retrieval of such regions prior to soldering process.*

- Consider the different geometrical regions (pad and drill-hole), whereas vision systems described in [12-14] process the whole solder joint/pad area: *A solder joint is fragmented into two regions based on their geometrical appearance, drill-hole (by early identification of this area from bare solder pad) and solder pad (by acquiring the diameter of solder pad from the CAD tool) and process them separately for quality classification.*

- Performs a comparison between the solder coverage percentage against the IPC standards (not analysed in [12-14]): *The less solder applied regions are precisely localized and compare the solder coverage percentage against the regulations provided by IPC. The outcome of this comparison is used to define its eligibility of being considered as a good joint based on the class to which the PCB belongs.*

- Provides automatic rework on defective solder joints (none of the commercial soldering robotic systems possess this capability): *If the identified less solder regions are not within the IPC regulations, automatic rework is performed on them by feeding the required amount of solder based on the size of defective regions.*

- Does not require learning of soldering direction for each solder pad (Commercial systems require the definition of this parameter during the learning phase): *The requirement of manual teaching of the direction in which the soldering iron touches a pad was eliminated with the retrieved information from CAD tool related to the height of nearby objects and their offsets from the point of interest.*

- Provides automatic estimation of the preheat time for a solder pad (the user has to define this parameter in commercial soldering robotic systems): *The width of the connected PCB track to a particular solder pad, extracted from the respective CAD file of a PCB, is used to define the preheat time automatically*.

- Requires less user skill level on machine operation and soldering programme development for a given PCB: *The required user skill level for the system operation and soldering programme development could be successfully minimized with the implementation of an interactive GUI that holds all the required information related to the PCB, retrieved from the respective CAD file*.

Consequently, a robust THT soldering system could be successfully developed with an integrated AOI and CAD system to provide automated soldering and quality inspection process, where none of the reported research work or commercially available systems is capable of performing. Sections 1.5 – 1.6 brief the system architecture of AOI and THT soldering system respectively.

1.5 Automatic Optical Inspection System Architecture

The AOI system described in this research takes control over every major action performed by the THT soldering system as illustrated in Figure 1.1.

- A : automatic PCB alignment to the origin of the robotic system XY positioning plane
- B : solder pad segmentation and real time positioning accuracy verification
- C : component lead top segmentation
- D : automatic classification of soldering quality
- E : verification of solder coverage to define automatic rework.

Implementation of each of these stages (A-E) will be discussed in Chapters 2-6 respectively. Section 2.2 briefs the architecture of the developed THT soldering system prototype in this study.

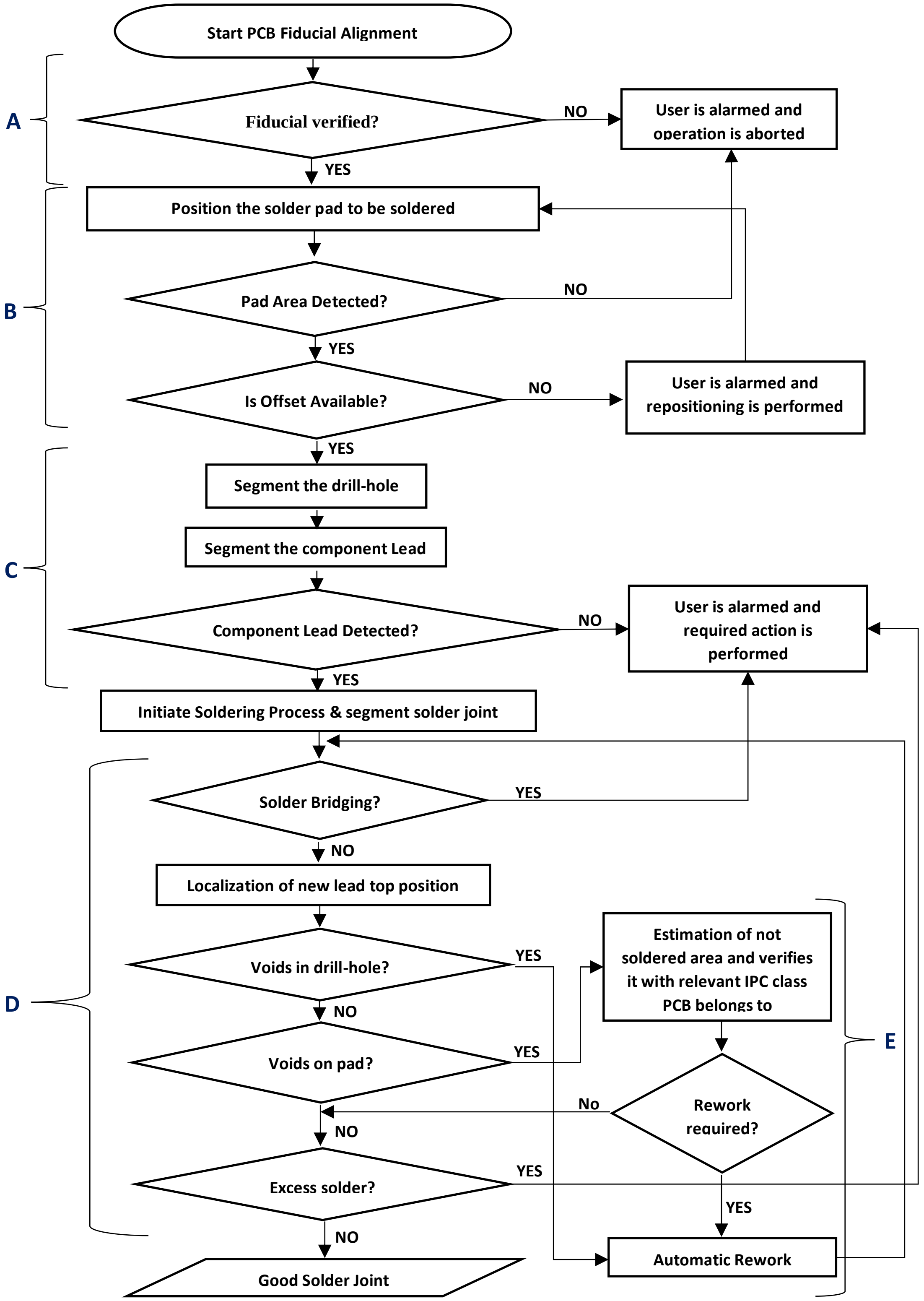

Start PCB Fiducial Alignment
A
Fiducial verified?
NO
User is alarmed and operation is aborted
YES
B
Position the solder pad to be soldered
Pad Area Detected?
NO
YES
Is Offset Available?
NO
User is alarmed and repositioning is performed
YES
C
Segment the drill-hole
Segment the component Lead
Component Lead Detected?
NO
User is alarmed and required action is performed
YES
Initiate Soldering Process & segment solder joint
D
Solder Bridging?
YES
NO
Localization of new lead top position
Voids in drill-hole?
YES
Estimation of not soldered area and verifies it with relevant IPC class PCB belongs to
E
NO
Voids on pad?
YES
Rework required?
No
NO
Excess solder?
YES
YES
NO
Automatic Rework
Good Solder Joint

Figure 1.1: AOI system architecture

1.6 THT Soldering System Architecture

Many soldering robotic systems have been developed from leading manufactures like Japan Unix, Metcal, Janome, Unitechnologies, Thermaltronics etc. Each of these systems requires user interaction and expertise knowledge up to a considerable extent for a better outcome, since none of them are equipped with real time inspection capability on their soldering process. The integration of a CAD tool together with AOI into the developed robotic system enables to provide a very precise and high-quality soldering process compared to the existing systems.

Figure 1.2 illustrates the operational block diagram of the developed THT soldering system. Each electronics circuit used in the robot system (except AC servo drivers used for *XY* positioning) was designed and developed from the scratch in my research in order to provide a flexible and modularized system architecture. This approach was beneficial in financial terms and further provides supplier independent system at a very low cost compared to commercially available soldering robotic systems. Further, the system has been designed in a way that the system mother board diagnoses the cause of failures of each sub module connected to it and alarms the user accordingly with the complete error report. Chapter 7 provides a detailed overview on the implementation of the THT soldering system.

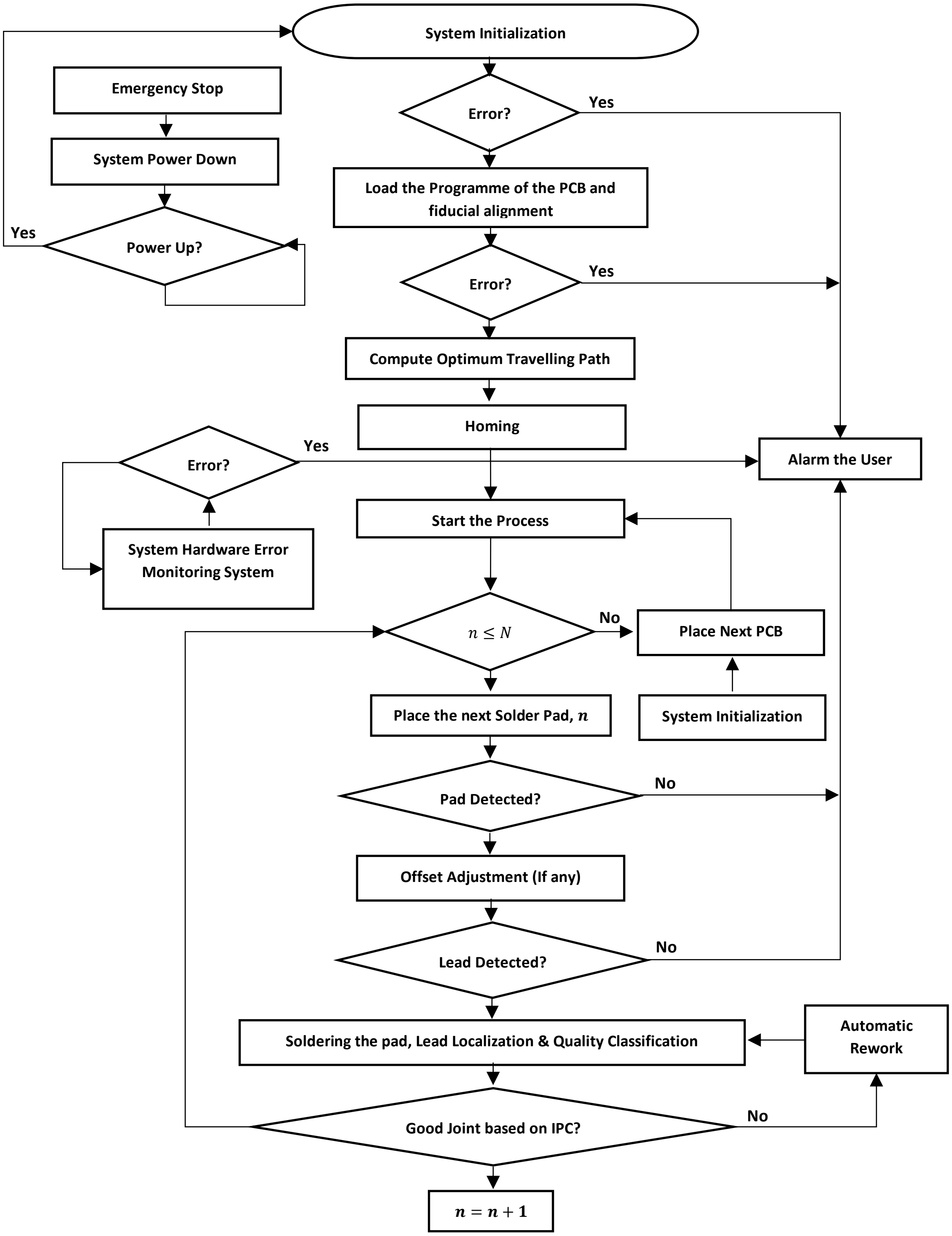


Figure 1.2: Robotic system architecture

2. Implementation of Automatic PCB Fiducial Alignment Process

In [12], the PCB alignment is carried out by detecting the edges of the PCB and computing the angle difference with the horizontal and vertical lines of the image box. However, this methodology is not robust and reliable for an industrial application. Therefore, a robust PCB alignment procedure based on the user defined master points available on the PCB was implemented in this research. Sections 2.1-2.3 describe how this procedure is implemented in this study.

2.1 PCB Fiducials and Importance of Localizing Them

PCB fiducial points are the user defined master points on a given PCB that can be used as reference points to calculate relative distances to each object on the PCB from a point that lies inside/outside of the PCB (In this application this point is the zero position/system origin of the PCB positioning system of the robotic platform). However, it is a necessity that the defined fiducials points must be included in the respective CAD file of the given PCB. The application of these master points is extensively used in SMD pick and place machines [17], AOI machines for SMD devices, Flying Probe – In Circuit Testers [18], PCB panel separation machines [19] in the field of EMS industry. These master points are sometimes defined by the PCB layout designer himself as illustrated in Figure 2.1.

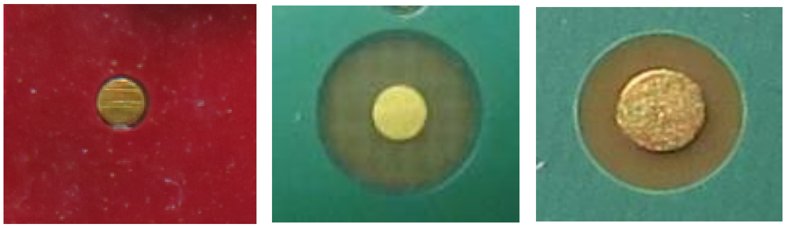


Figure 2.1: Fiducial points defined by the PCB layout designer

It is required to have at least of 3 fiducial points populated on a PCB to render a very precise outcome as described below.

1st Fiducial Point:

The first fiducial point enables the machine to recognize the respective location of the PCB is in its XY space.

2nd Fiducial Point:

The second fiducial point helps the machine to recognize the orientation of the PCB and the bend of PCB surface due to tightened screws. If the PCB is rotated even 0.1% of a degree this could completely ruin the assembly if it weren't for fiducials. The machine is able to measure the angle that the board is rotated down to the nearest 0.01% of a degree and compensate all of the placements accordingly.

3rd Fiducial Point:

Consequently, the third fiducial enables the machine to mitigate the slight deformation defects like any shrink or stretch of the PCB. In most cases, PCB's do vary by very small amounts over a long distance. Therefore, larger PCB's experience a much greater amount of stretch, shrink or both. This is effect is significant for double-sided Surface Mount Technology (SMT) assemblies. After the first side is reflowed, the board may have stretched, shrunk, bowed or flexed due to the higher temperature profile maintained inside the oven. Such an effect can be minimized with the application of a third fiducial point.

These fiducials can be fairly identified and localized by the vision system of such a machine using template matching algorithms [20-23]. This process is also known as fiducial verification. Once the fiducial points are identified by the respective vision system, it calculates relative distances to all the points on a PCB with respect to the calculated distances from the system origin to the identified fiducials on the PCB. However, when it comes to define fiducials on the secondary side of the PCB that holds the THT pads, dedicated fiducials points are not always available. Section 2.2 describes how to counteract such situations efficiently.

2.2 The Distinctive Behaviour of the Proposed Fiducial Verification Process in Automatic Optical Inspection System

The proposed vision system primarily focuses on quality assessment of THT solder joints which are placed on the bottom side (secondary side) of the PCB. In some cases, the fiducial points are not available on the bottom side of the PCB when,

- SMD components are placed only on the top side of the PCB
- The available space on the PCB is limited
- Mistake from the PCB layout designer

In such a case, it is required to define fiducial points from the available distinct objects on the PCB like mounting holes, vias, test-points and test pads. Figure 2.2, illustrates few examples of such objects available on the PCB.

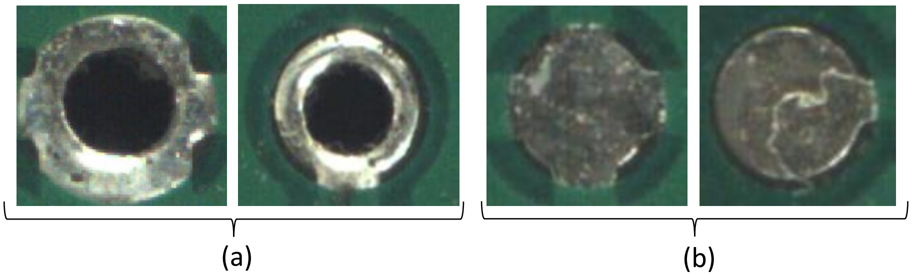


Figure 2.2: Distinct objects on the PCB in addition to fiducial markings provided by layout designer. (a). Vias available on a PCB. (b). Test Pads available on the PCB

Even though the positioned solder pads can be accurately identified by the vision system as described in Chapter 3, it still cannot differentiate between solder pads on different PCB types. Since the proposed AOI system has been designed to produce a fully automated THT soldering system, the vision system must be capable of precisely identifying whether the PCB under solder (PUS) is the correct one that should be soldered or not. In Figure 2.2, it can be seen that these objects on the PCB are more vulnerable to uneven illumination, environmental noise, lack of distinguishable features. Therefore, a more robust inspection mechanism is required to distinguish these points from their counterparts.

2.3 Algorithms for Fiducial Verification

Image localization can be mainly carried out by two methods.

- Template matching
- Feature extraction

In template matching method, the occurrence of a model image inside the input image (I) is computed by convolution between model image (I_M) and input image (I_L). Various algorithms have been proposed for template matching like computation of correlation coefficient (TM_CCOEFF), normalized version of correlation coefficient (TM_CCOEFF_NORMED), concept of correlation among I_M and I_L (TM_CCORR), concept of normalized version of correlation among I_M and I_L (TM_CCORR_NORMED), squared difference (TM_SQDIFF) error and normalized squared difference (TM_SQDIFF_NORMED) error [24-25]. The resulted image from above algorithms contain a global optima which corresponds to the best matching point as illustrated in Figure 2.4. Therefore, the application of template matching algorithms does not provide precise outcome, when the existence of the model image is not exactly known. In Figure 2.3 (a), it can be seen that fiducial verification process cannot be done effectively using above template matching algorithms.

In feature extraction methods, features are identified from a model image and their subsequent matching in the input image are found. There is no clear definition for ‘feature’, but there are some properties that can be evaluated at a given point, to decide whether that particular point/area is a good feature or not [26].

A feature should be,

- perceptually meaningful
- analytically special (Ex. minimum or maximum point in a given data set)
- easily distinguishable in different images
- invariant to scale, orientation and illumination.
- insensitive to noise

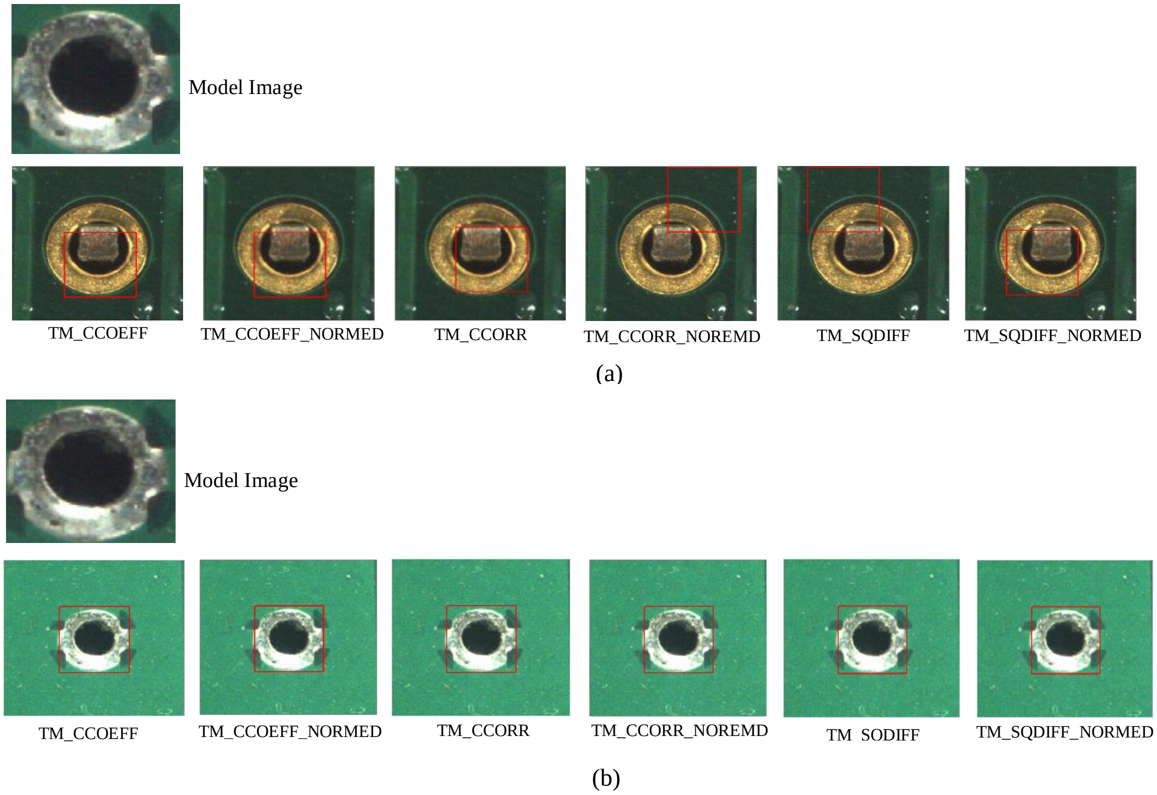


Figure 2.3: Best matching points detected by various template matching algorithms. (a) Matching of model image with an input image from a different PCB of same type. (b) Matching of model image with an input image from same PCB type

In order to compare images based on their features, many research works have been carried out. They mainly rely on extracting features / key points in a provided model image and find their corresponding matching inside a given input image at different scales and orientations. Various algorithms have been proposed for the extraction of features of a given image, like kadir and brady algorithm [27], Scale Invariant Feature Transform (SIFT) [25, 28-29], Speed-Up Robust Features (SURF) [30-31], Features from Accelerated Segment Test (FAST) [32]. In this research, SIFT, SURF and FAST algorithms have been taken into consideration because of their proven robustness, accuracy and fast detection of key points in an image (Sections 2.3.1-2.3.3). Figure 2.4 illustrates feature matched images using above stated algorithms.

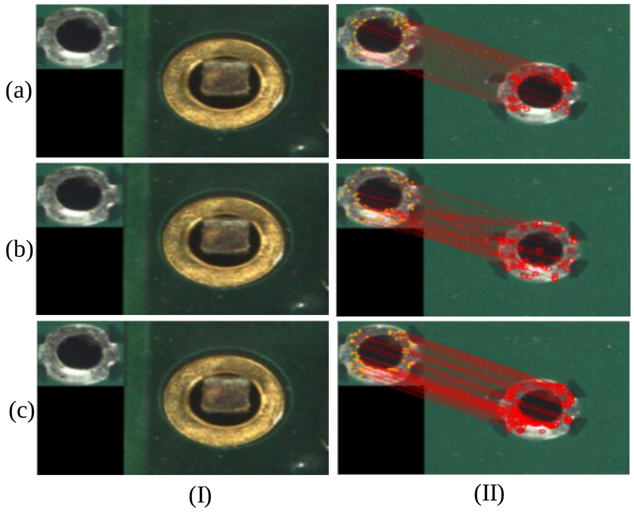


Figure 2.4: Feature matched images using feature extraction algorithms. Column (I) illustrates feature matching between model image and incorrect PCB. (II). Illustrates feature matching between model image and correct PCB. (a). Results obtained using SIFT over correct and incorrect PCB types. (b). Results obtained using SURF over correct and incorrect PCB types. (c). Results obtained using FAST over correct and incorrect PCB types

2.3.1 Scale Invariant Feature Transform

SIFT, Scale Invariant Feature Transform, is a method of extracting distinctive invariant features from a model image that can be effectively used to perform accurate matching between different views of that model image occurred inside a given input image (I) [29]. The detected features are proven to be invariant to scale, rotation and provide robust matching across a substantial range of affine distortion, noise, and illumination variation [29]. This algorithm can be described as a combination of following sub stages, where each stage adds a weighted outcome to the next stage.

- Constructing a scale-space:
  Implements a scale invariant platform by computing the difference of blurred images using a gaussian kernel to blur the original image and a set of scaled down images [29].

- Key point localization and selection of descriptive key points:

  Localization of key points extracted from generated Difference of Gaussian (DoG) images from scale space and selection of the detected keypoints based on their stability and repeatability of detection [29].

- Orientation assignment:

  Assignment of single or multiple orientations to each key point locations based on local image gradients to provide a rotational invariance [29].

- Generation of SIFT features:

  Descriptor to each located key point is generated that makes them invariant to local shape distortion and illumination variation [29].

<u>Constructing a scale-space:</u>

A natural property of real-world objects is that they only exist as meaningful substances over certain ranges of scale. A simple example is the concept of a branch of a tree, which makes only sense in a range of few centimetres to few meters. It is pointless to discuss this example at nanometre or kilometre level. At those levels, it is more convenient to discuss about the molecules in a tree leave and the forest where the tree grows respectively. The construction of scale space attempts to replicate this concept to digital images. The first step of key point detection is to identify locations and scales that can be repeatedly assigned under different views of the same object. In this process, the points must be localized in a way that they do not change their way of existence in the original image at different scales. These stable features can be estimated with the use of a continuous function of scale, referred as scale space [29].

SIFT uses gaussian function [35] to generate its scale space structure by progressively blurred out images resulted from the initial image and a set of scaled down images by a factor of 2 [29]. The generation of a gaussian blurred image is illustrated in Eq. (1).

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{1}$$

Where,

$L(x, y, \sigma)$: Resulted image from the convolution between $I(x, y)$ and gaussian kernel, $G(x, y, \sigma)$.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

Here x, y are the spatial coordinates in 2-D space and σ is the standard deviation of the gaussian kernel. Figure 2.5 illustrates the first two layers that contain progressively blurred out gaussian images in SIFT scale space structure.

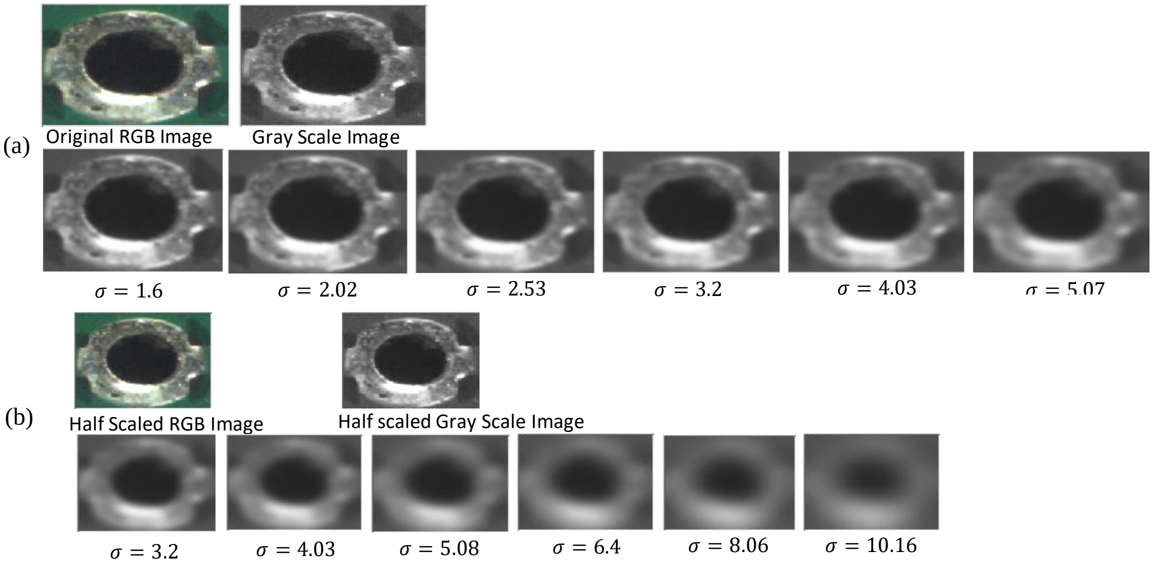


Figure 2.5: Few samples of scale space. (a). Resulted images after convolving the original image with gaussian kernel at five different σ values where $\sigma = 1.6, 2.02, \dots, 5.07$. (b). Resulted images after convolving the scaled down image by factor of 2 with gaussian kernel at five different σ values where $\sigma = 3.2, 4.03, \dots, 10.16$

There are two important parameters, number of blur levels and number of octaves, of this scale space structure that must be determined before going deeper of the algorithm. An octave is a set of blurred out images of same size. Then number of octaves defines how many down scaled levels are used to generate scale space structure. The author of SIFT suggests that the algorithm must generate $s + 3$ images in the stack of blurred images [29] for each octave in order to make sure that final feature (keypoint) detection covers a complete octave. Here 's', can be represented as illustrated in Eq. (2).

$$s = \log 2 / \log k \qquad (2)$$

Where, k is the multiplication factor between, two successive blur levels. s is an empirical factor determined by the author of SIFT, based on the analysis that has been carried out by plotting the percent of keypoints that are repeatedly detected at the same location and scale in a transformed image against the number of scales sampled per octave (number of scales used for extrema detection) [29]. The author of SIFT suggests to use $s = 3$, since maximum level of repeatability could be achieved when sampling 3 scales per octave. This results in 6 blurred out images per octave. Figure 2.6 illustrates how the scale space structure is generated in SIFT [29].

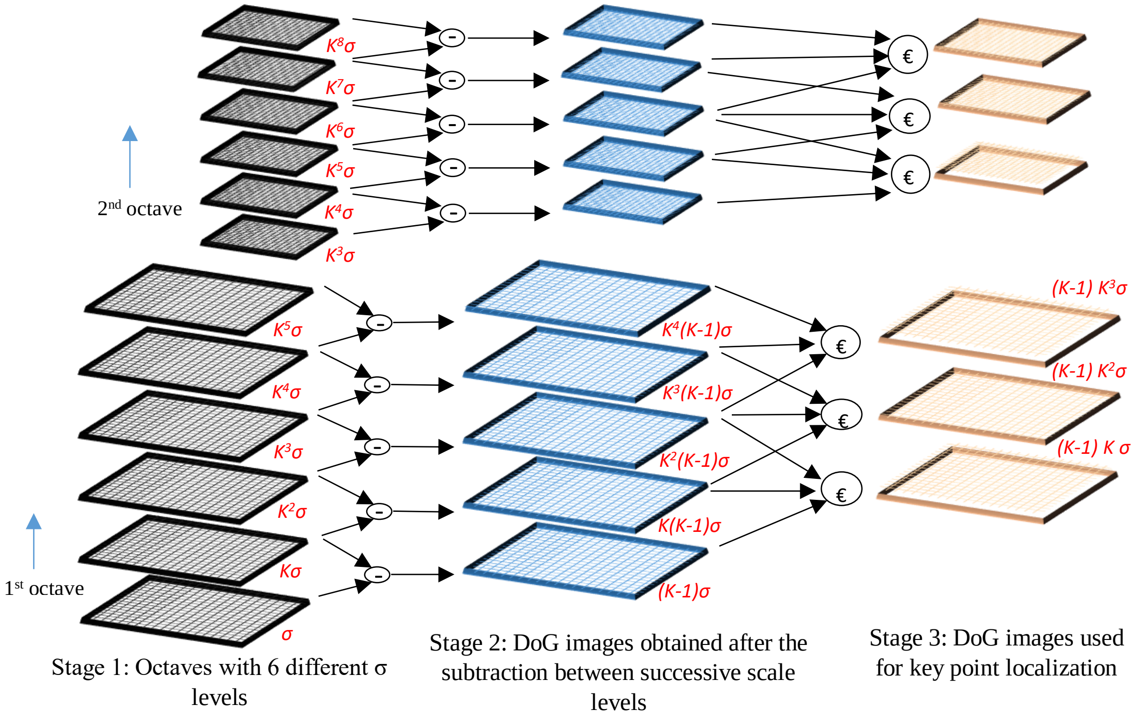


Figure 2.6: Generation of scale space structure in SIFT. Each black square illustrates a gaussian blurred image with $k^n\sigma$, where n=0,1,2... Each blue square illustrates computed DoG image by subtracting two adjacent gaussian blurred images $k^{n+1}\sigma$ and $k^n\sigma$). Each brown square illustrates a resulted image after neighbour pixel comparison method

In SIFT, DoG images are computed to detect stable feature locations in scale-space (refer Figure 2.6 Stage 2). The next step is to understand how the stable keypoints are detected by the algorithm.

Key point localization and selection of descriptive key points:

In order to estimate, whether a particular point is a keypoint or not, it must be differentiable from its neighbours. This is accomplished in SIFT in a way that, a particular pixel in a DoG image is compared with reference to its eight neighbours and the corresponding nine neighbours possessed by the above and below DoG images to determine whether that particular point is a local extrema or not [29]. Particular sample point will be decided as a local extrema, only if it is the maximum or minimum compared to its 26 neighbouring points [29] (Figure 2.6 Stage 3). Once the local extremum point is found, a distinctive relationship to nearby data for location, scale and ratio of principle curvatures [29] is computed. This relationship is used by the algorithm to eliminate key points with unstable properties. SIFT uses following procedure to estimate the stability of the detected keypoints.

Step 1: *Finds the sub pixel accuracy of detected keypoints using Taylor expansion [29].*

Step 2: *Performs contrast comparison of detected keypoints against a predetermined threshold value [29]. If the magnitude of the function computed from Taylor expansion at the keypoint location is below the defined threshold, it is discarded [29].*

Step 3: *The keypoints that lie on the edges of the image are discarded by computing Principle Curvatures (PC) of resulted DoG image [29]. This elimination is vital because of the increased vulnerability to noise due to strong edge response of DoG images [28, 29].*

At the end of this refinement process of detected keypoints, it is said the remaining keypoints are invariant to different scales of the image [29]. The next step is to compute the orientation of remaining set of keypoints, in order to make them rotational invariant.

Orientation Assignment:

A good feature extraction algorithm must be capable enough to detect same features inside the rotated versions of the same image which the keypoints were originally detected. Therefore, it is required to make a detected keypoint invariant to rotation. In SIFT, rotational invariance is achieved by assigning orientation to each key point based on local image properties. The estimation of the orientation of a given keypoint (θ_{ref}) is performed by calculating the gradient magnitudes and directions around a particular key point within a defined neighbourhood inside the respective gaussian blurred image ($L(x, y, k^n\sigma)$). The selection of $L(x, y, k^n\sigma)$ is clearly explained in [29]. The orientation assignment process is performed by calculating the gradient magnitude, $m(x, y)$ and the orientation, $\theta(x, y)$ based on the pixel differences over each pixel that falls within a defined circular neighborhood of the keypoint. The computation of $m(x, y)$ and $\theta(x, y)$ is illustrated in Eq. (3) and Eq. (4) respectively [29].

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1)^2} \qquad (3)$$

$$\theta(x, y) = \tan^{-1}((L(x, y+1) - L(x, y-1))/\big(L(x+1, y) - L(x-1, y)\big) \qquad (4)$$

The radius of the circular neighbourhood depends on the scale of the keypoint and it is estimated as $1.5 \times scale$ [29], where $scale$ can be denoted as $k^n\sigma$ where $n = 1,2, \ldots .p$. Once this process is completed, an orientation histogram is calculated that has 36 bins, covering 360^0 range of orientation, where the starting value of a bin can be expressed in general as $\theta_b = 2\pi(b-1)/36$, where $b = 1,2 \ldots 36$. A detailed explanation of this process is provided in [29].

Generation of SIFT features:

Algorithm has generated keypoints, which are invariant to scale and rotation up to this stage of processing. In real images, there will be lots of illumination variation and 3-D object transformations. Therefore, there must be a way of making a particular keypoint independent to these changes. At the final step of SIFT algorithm, a unique identifier is generated based on the local image region, that will make a scale and rotational invariant keypoint to be distinctive over above stated variations. In order to create SIFT descriptor for a keypoint (x_k, y_k), a square neighborhood with a width of $12k^n\sigma$ is centered at the keypoint, in the direction of θ_{ref} inside its respective gaussian smoothed image. Then this region is further divided into 4×4 sub regions, which has a width of $3k^p\sigma$ in each dimension. Then local gradients are computed inside each sub region. Once this is utilized, an 8-bin orientation histogram is computed for each sub region. In this case, the particular key point will have a descriptor that contains 128 (4 × 4 × 8) feature vectors. A detailed explanation on how this process is accomplished is provided in [29]. The detected key points on several model images by SIFT algorithm are illustrated in Figure 2.7. In Figure 2.7, it can be clearly seen that over 95% of the detected key points lie over the distinguishable object locations inside the input image.

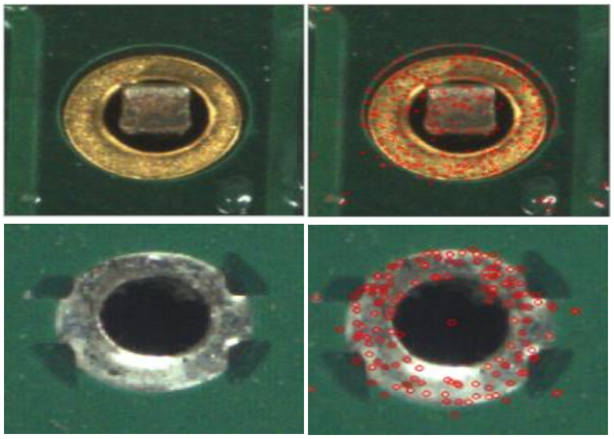

Original Image Key Point Detected

Figure 2.7: Detected key points on different objects using SIFT feature detector at number of scales per sample = 3, contrast threshold = 0.03, edge threshold = 10 and starting scale (σ) =1.6

Section 2.3.2 presents a detailed overview on the implementation of SURF algorithm.

2.3.2 Speed-Up Robust Features

SURF, Speed-Up Robust Features, is a fast and robust algorithm developed by H. Bay, for the detection of local invariant features and finds their subsequent matching in a provided scene [30-31]. SURF has been developed with the purpose of providing a fast computation on a given image, which makes it ideal for real time applications like tracking and object recognition. Similar to SIFT, SURF also relies on scale space representation for the detection of features that are scale and rotational invariant. This algorithm has been proved to render better results with higher speed and accuracy compared to SIFT. The implementation of SURF is be discussed in three steps.

- Generation of integral images and box filters:
  This contains the generation of integral images and box filters to approximate gaussian second order partial derivatives.
- Constructing a scale-space:
  Implements a scale invariant platform with the use of box filters together with integral images.
- Key point Localization:
  This contains the localization of stable key points with the use of hessian matrix in box filter space and computation the orientation of each detected features.
- Generation of SURF feature description:
  This contains the generation of a unique identifier for each detected key point.

Generation of integral images and box filters:

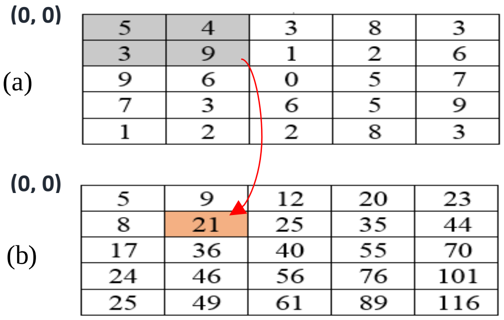
In SIFT, scale-space is generated from the convolution between a given input image and the Gaussian kernel at different scale levels. It is not efficient in real time applications due to the computational complexity of Gaussian kernel [31]. As a way of speeding up this process, SURF algorithm uses integral images together with rectangular shape uniform kernels, referred to as “box-filters”. Their combination approximates the role of gaussian kernels in an efficient way [31]. It is better to know what is actually an integral image and how will they be useful to reduce calculation complexities in generating scale space. Eq. (5) illustrates how the pixel value of an integral image (U) at a point (x, y), is computed from the corresponding I.

$$U(x, y) = \sum_{0 \le i \le x} \sum_{0 \le j \le y} I(i, j) \tag{5}$$

Where,

$U(x, y)$: Value of integral image at (x, y)
$I(i, j)$: Value of input image *I* at (i, j)

According to Eq. (5), it is quite clear that value of an integral image at point (x, y) means the summation of pixels above that point in both *x* and *y* directions inside I. Figure 2.8 illustrates how an integral image is formed from an input image of size (5×5).



5	4	3	8	3
3	9	1	2	6
9	6	0	5	7
7	3	6	5	9
1	2	2	8	3

5	9	12	20	23
8	21	25	35	44
17	36	40	55	70
24	46	56	76	101
25	49	61	89	116

Figure 2.8: Computation of integral image based on the pixel intensity values of a given *I*. (a). *I* with a highlighted region where the summation of pixel intensities is calculated to compute the corresponding pixel value of *U*. (b). *U* derived from the summation of intensity values in *I*

Red arrow points to the value of integral image at point, *X*= (*1*, *1*). This process utilizes only three additions and four memory lookups in the integral image to compute the sum of pixel intensities contained by any upright rectangular area formed inside I. The next step is to identify what are box filters and how they enhance the efficiency of generating scale space with the use of integral images. SURF also relies on generating of scale space composed with octaves and numbers of blur levels per octave as followed in SIFT.

In SIFT, Lowe approximated Laplacian of Gaussian (LoG) with DoG for generating scale space. The authors of SURF further improve this concept and approximate LoG with box filters [31]. Box Filtering is kind of image filtering, where an average value of surrounding pixels that lie in a given rectangular region, is computed. Integral images lessens the complexity of getting the average value of pixel intensities contained within any rectangular area. The main reason that made the authors of SURF to use box filters as scale space kernel is to have fast computation for their keypoint detector.

The authors of SURF rely on implementing their feature detector based on the hessian matrix because of its performance and precise operation [31]. Eq. (6) illustrates how the hessian matrix is described inside a given image at point, $X = (x, y)$.

$$H(X,\sigma) = \begin{bmatrix} L_{xx}(X,\sigma) & L_{xy}(X,\sigma) \\ L_{xy}(X,\sigma) & L_{yy}(X,\sigma) \end{bmatrix} \tag{6}$$

$H(X,\sigma)$: Hessian matrix at X with scale σ

$L_{xx}(X,\sigma)$: The result of the convolution between I and the gaussian second order derivative $\frac{\sigma^2}{\sigma x^2} g(\sigma)$ at point X.

Similarly gaussian second order derivatives for $L_{yy}(X,\sigma)$ and $L_{xy}(X,\sigma)$ can be denoted as $\frac{\sigma^2}{\sigma y^2} g(\sigma)$ and $\frac{\sigma^2}{\sigma x \sigma y} g(\sigma)$ respectively. Gaussians are optimal for scale space analysis, but in practice they have to be discretized and cropped due to the discrete nature of digital images [31]. This makes the authors of SURF to approximate these gaussian second order derivatives with second order finite difference operators as illustrated in Eq. (7) [30-31]. The finite difference is the discrete analog of a derivative.

$$H_{approx.}(X,\mu) = \begin{bmatrix} D_{xx}^{\mu} X & D_{xy}^{\mu} X \\ D_{xy}^{\mu} X & D_{yy}^{\mu} X \end{bmatrix} \tag{7}$$

Here D_{xx}, D_{yy} and D_{xy} are second order finite difference operators computed in *x*, *y* and *x*-*y* directions respectively. μ is known as the scale variable that has been used to parameterize the scale of gaussian second order derivatives, σ. The authors of SURF suggest that the relationship between μ and σ is defined by $\sigma = 0.4\mu$ [30], where the value of scale variable μ depends on the current octave level (o) and the number of scale levels (i) inside o. This relationship is illustrated in Eq. (8).

$$\mu = 2^o i + 1 \tag{8}$$

A detailed mathematical description on how the conjunction between the box filters and second order finite difference operators is made, has been presented in [30]. It further describes how the application of integral images reduces the computational complexity of the convolution between the second order difference operators and I. Next step is to understand how the scale space is generated in SURF using these approximated second order finite difference operators (box filters).

Constructing a scale-space:

In SIFT, the scale space is generated with the use of down scaling the original image at each octave while smoothing that down scaled image at different σ levels. However, the application of box filters together with integral images, eliminates the requirement of iteratively applying the same filter to the output of previously filtered layer in SURF [31]. SURF applies box filters of any size at exactly the same speed directly on the original image and even in parallel [31]. Therefore, the scale space (also known as box space) is generated in SURF, by up scaling the filter size rather than iteratively down scaling the image size [31]. This procedure outperforms SIFT in terms of decreasing computational complexity in generating scale space, while preserving high frequency components that can be lost in zoomed-in variants of the original image. According to the definition of μ in Eq. (8), it can be seen that the initial size of the box filter at first octave must be equal to 9×9. As in SIFT, SURF also creates different octaves where each octave represents a series of filter response maps obtained by convolving same input image with a filter of increasing size. For two successive levels, it is required to increase the size of μ by a minimum of 2 pixels (one pixel on every side) in order to keep the size uneven ensuring the presence of the central pixel. This results in the relationship denoted by Eq. (8). Figure 2.9 illustrates how a box filter of a previous blur level is used to generate box filter of next blur level.

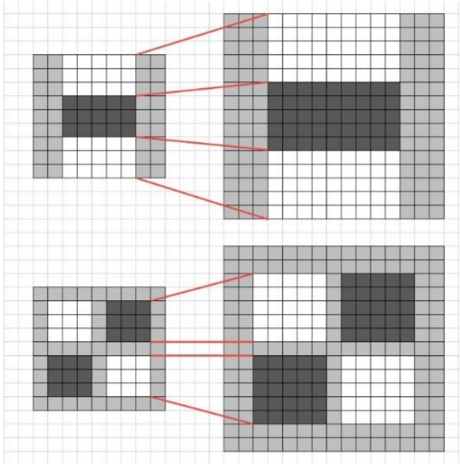


Figure 2.9: Generating of *(i+1)*th level of box filter using *i*th level of box filter inside *o*th octave level in both *y* and *xy* directions respectively

In SIFT, it has been stated that the starting scale of next octave layer must have a scale of twice of the size of initial scale of previous octave. A similar procedure is followed in SURF for generating new octave layer as illustrated in Eq. (9).

$$W_o = 3(\mu_{n-1} + 2) \tag{9}$$

Where W_o is the size of starting filter of o^{th} octave level and μ_{n-1} is the value of scale variable in previous octave ($n = 2, 3, \dots$). This will result in generating of a box space structure in SURF as illustrated in Figure 2.10.

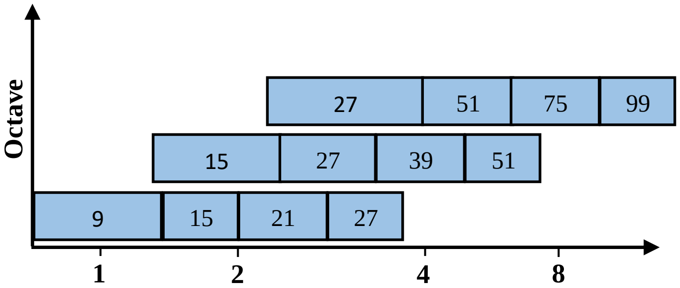


Figure 2.10: Graphical representation of box filter size for three octaves

Next step is to understand how the stable keypoints are detected from the generated box space in SURF.

Key point Localization:

Once the box space is generated, the next step is to localize stable key points from the box space. The procedure for detecting the keypoints can be explained in two major steps [30].

- Feature filtering: *this is accomplished using scale-normalized determinant of hessian.*
- Feature selection and location refinement: *this is accomplished with the combination of non-maxima suppression and thresholding.*

Feature filtering:

The objective of feature filtering is to locate more distinctive key points in a given image. Inspired from the robustness of the methods used in SIFT for localizing scale invariant features, SURF uses scale normalized determinant of hessian [30], to detect extreme points in the input image. The scale-normalized determinant of hessian is illustrated in Eq. (10).

$$DoH^{\mu}(I) := \frac{1}{\mu^4}\left(D^{\mu}_{xx}I . D^{L}_{yy}I - \left(wD^{\mu}_{xy}I\right)^2\right) \qquad (10)$$

The weighting factor, $w = 0.912$, is used to compensate the numerical approximation of the hessian determinant by box filters [31]. This is required to have the energy conservation between the gaussian kernels and their approximated counterparts (box filters). The next parameter that should be determined is the sampling frequency of the pixels. SURF suggests that it is not required to compute DoH^{μ} operator for each pixel at large filter scales [30]. Therefore, the pixel sampling frequency is determined based on the octave level o, in a way that the number of points sampled per octave is $\left\lceil \frac{M}{2^{o-1}} \right\rceil \times \left\lceil \frac{N}{2^{o-1}} \right\rceil$ for an image of size $M \times N$ [30]. This definition enables to have a sampling frequency of 2^{o-1}. It means the operator is applied after each 2^{o-1} number of pixels.

Feature selection and location refinement:

In SURF, localization of key points in $DoH^{\mu}(I)$, is performed in a similar way as in SIFT, by a non-maximum suppression in a $3 \times 3 \times 3$ neighborhood. This performs a comparison of each sampling point in discrete box space with its 26 nearest neighbours [30, 31]. The maximum point of this neighbourhood will be considered as a keypoint. Once a key point is detected, it is threshold based on a given threshold value as illustrated in Eq. (11).

$$DoH^{L}(I)(x, y) > t_H \tag{11}$$

The value of t_H can be determined by the user based on the application requirements. In this application value of t_H has been set to 500. Finally, eight images are available in box space with detected keypoints. Once the most stable keypoints are located, the algorithm finds their sub pixel accuracy in order to have a precise localization of its position.

Generation of SURF feature description:

The descriptor for a detected key point in SURF, describes the intensity content within the key point vicinity. This procedure is accomplished in a similar way followed in SIFT and its variants for the extraction of gradient information on a particular key point. The process of defining a unique identifier/descriptor in SURF is computed at two major steps.

- Orientation assignment around the interest point to achieve rotational invariance
- Generating SURF descriptor based on the orientation details

According to the information presented so far in the explanation of SURF, particular keypoint can be expressed as, $X_k = (x_k, y_k, \mu_k)$ which has the corresponding scale (σ_k), defined by the relationship, $\sigma_k = 0.4\mu_k$. In order to achieve rotational invariance for a detected keypoint, a dominant orientation must be computed relative to the local distribution of gradient orientation. The definition of a region for the computation of local gradients inside I, is accomplished by defining a circular neighbourhood, $B_{6\sigma_k}(x_k, y_k)$, with the radius of $6\sigma_k$ centered at the location of the desired keypoint X_k [31]. The computation of gradient information at inside $B_{6\sigma_k}(x_k, y_k)$, is accomplished by convolution with first order box filters, D_x^{μ} and D_y^{μ} [30, 31]. In order to minimize the impact of distant pixels, the gradient samples are weighted according to their distance from the desired keypoint using a discrete gaussian kernel (G_1) with a standard deviation of $2\sigma_k$ [30]. Eq. (12) illustrates how these weighted gradients are computed at a point (x, y).

$$\emptyset_k(x,y) = \begin{pmatrix} D_x^{L_k} \\ D_y^{L_k} \end{pmatrix} * I(x,y). G_1\left(\frac{x - x_k}{2\sigma_k}, \frac{y - y_k}{2\sigma_k}\right) \tag{12}$$

In SIFT, the dominant orientation is computed by creating a 36 bin histogram based on the local gradients and defining the orientation of the highest peak as the dominant orientation. SURF computes a score vector, $\emptyset_k(\theta)$, that sums all weighted gradients in the defined neighborhood, which have orientations that fall within a fixed tolerance of $\theta \pm \frac{\pi}{6}$, where θ is the angle of current gradient [30, 31]. This expression is illustrated in Eq. (13).

$$\emptyset_k(\theta) = \sum_{(x,y) \in B_{6\sigma_k}(x_k, y_k)} \emptyset_k(x,y) \times 1_{\left(\theta - \frac{\pi}{6}, \theta + \frac{\pi}{6}\right)}(\angle \emptyset_k(x,y)) \tag{13}$$

Here $\angle\emptyset \in [-\pi, \pi]$.

The global maximum of the orientation score function will be defined as the orientation of X_k [30, 31]. At this stage, the computed keypoints can be expressed in the form of, $X_k = (x_k, y_k, \mu_k, \theta_k)$. Now they are invariant to scale and rotation. However, they still do not possess a property that can make them unique. Then a unique identifier must be computed for the detected keypoints as accomplished in SIFT.

A SURF descriptor is a 16×4 vector, that consists of mean and absolute mean values of gradients extracted from a spatial grid (R) divided into 4×4 sub regions [30]. A detailed description on the computation of the SURF descriptor is provided in [30].

The detected key points by SURF, on several model images are illustrated in Figure 2.11.

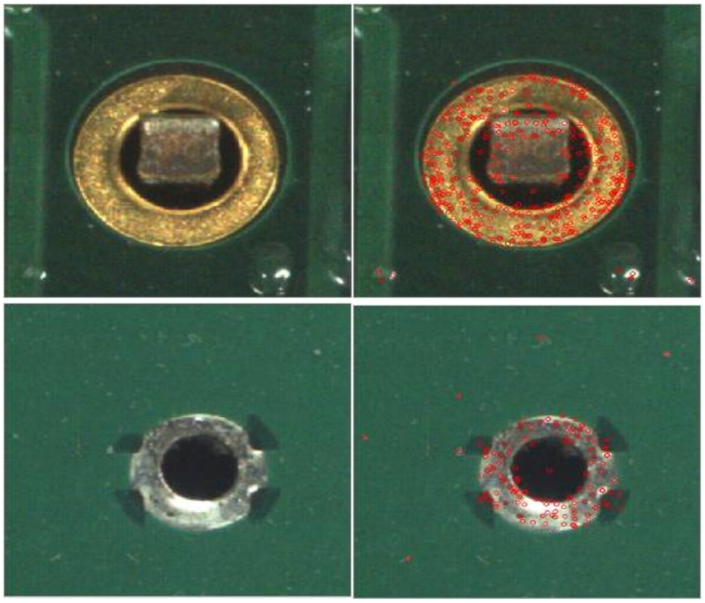


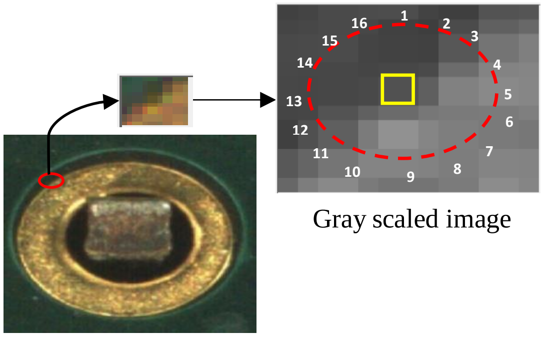
Figure 2.11: Detected key points on solder pads, vias and test pads on different colour PCBs at no of octaves = 4, threshold = 500, starting box filter size = 9 × 9 using SURF detector

According to Figure 2.11, it can be clearly seen that over 95% of detected key points lie on the object regions and distinctive areas in a given image. Section 2.3.3 explains the implementation of a fast feature detection algorithm that have been developed especially targeting for platforms with less computational power, but require high speed of feature detection.

2.3.3 Features from Accelerated Segment Test

FAST, Features from Accelerated Segment Test [32], is a fast corner detection algorithm developed by Edward Rosten and Tom Drummond. The main idea behind the development of FAST algorithm is to have a fast feature detector for real time applications, which are running on platforms with limited computational power. In this application, the speed of detecting key points does not play a significant role, since the fiducials of a given PCB are verified only once for that particular PCB. Therefore, The robustness and accuracy of this algorithm to locate most stable and unique features from a given input image, were given the highest priority.

In FAST, estimation of a desired pixel, p is whether a corner or not is accomplished based on its intensity level, I_p and its relationship to the 16 pixels, numbered in clockwise direction that lie inside a circular region that has the radius of 3 pixels [32]. Figure 2.12 illustrates how this neighbourhood is implemented over a given candidate pixel.



Gray scaled image

Original image

Figure 2.12: Selection of neighbouring pixels lie at candidate pixel P (Yellow Square) in FAST

A detailed explanation on how this neighbour comparison process is carried out is presented in [32]. Once the stable keypoints are detected as explained in [32], the algorithm must compute a unique descriptor to each detected keypoint as SIFT and SURF to make it distinctive. However, the authors of FAST do not provide their own descriptor for the detected features. Therefore, one of the well-known descriptor algorithms, Binary Robust Independent Elementary Features (BRIEF) [39] was used in this research. The SIFT and SURF use descriptors with the sizes of 128 and 64 dimensional vectors respectively, which lead to have memory constraints for the applications which have limited memory capacity. BRIEF comes into picture at such situations by providing binary strings for detected keypoints rather than computing individual descriptor to each of them. A detailed explanation on how the BRIEF descriptor works can be found in [39]. The detected key points on several model images by FAST algorithm is illustrated in Figure 2.13.

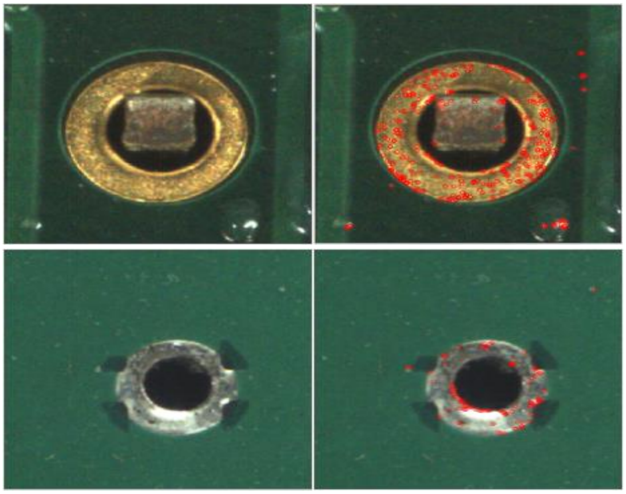


Figure 2.13: Detected key points on solder pads, vias and test pads on different colour PCBs at $I_t = 30$ with non-maximum suppression using FAST corner detector. Here I_t is a user defined threshold value

As stated earlier, FAST is not a scale and rotational invariant feature detector as SIFT or SURF. However, these two properties do not add a significant impact on the performance of the algorithm, since the PCB is placed on a positioning bed and the distance between the camera and the PCB is precisely controlled by the soldering system ensuring that no scale distortion or a slight displacement occurs in this application. Section 2.4 presents a detailed explanation on the performance evaluation of these three feature detection algorithms.

2.4 Performance Evaluation of Feature Extraction Algorithms

In Sections 2.3.1-2.3.3, the feature detection of a given image using SIFT, SURF and FAST algorithms has been discussed. In order to evaluate performance of these feature extraction algorithms, three main evaluation criteria's have been considered in this research.

- locations of detected keypoints
- repeatability of detected keypoints in different illumination conditions
- Successful and false detection rate at different illumination conditions

The illumination level was controlled digitally using a PWM controlled LED panel fixed at a distance. According to Figures 2.7, 2.11 and 2.13, it could be seen that majority of the detected keypoints lie inside the object regions in a given input image. Table 2.1 illustrates the results obtained based on the localized region (Foreground and Background) of keypoints and repeatability of detection at consecutive trials (50) using a sample of100 images (the size of the sample for each analysis carried out in this section) using these three algorithms. The average percentage of keypoints detected inside foreground region is computed by taking the ratio between the number of detected keypoints inside the user defined object region and the total number of keypoints detected.

Table 2.1: Average percentage of keypoints detected in foreground region and repeatability of detection using SIFT, SURF and FAST feature detectors

Criteria	**Detection rate of different feature detectors**		
	SIFT	**SURF**	**FAST**
Percentage of Keypoints Detected in Foreground	95%	98%	96%
Repeatability at Highly Illuminated Conditions	78.5%	82%	72%
Repeatability at Poorly Illuminated Conditions	86%	85%	76%

Conditions:

- Percentage of detected keypoints in foreground is calculated with respect to the total number of keypoints detected.
- Normal Illumination Level: (250 – 750) Lux, Highly Illuminated Level: (1000 – 1500) Lux and Poorly Illuminated Level: (75 –200) Lux

According to Table 2.1, it can be seen that all of these feature detectors are capable of detecting features inside foreground regions in a given image, even though SURF slightly outperforms other two algorithms in terms of accuracy in detecting keypoints. In addition to that, the requirement of analysis in different illumination levels is influenced by the evaluation of the capability of the vision system to operate in environments where no specific lighting is available. Even though, SURF could render a better outcome at highly illuminated environments, SIFT could outperform it by 1% at poorly illuminated environments.

Next, the robustness of detected keypoints to be identified in a located image (I_L) must be fairly evaluated. This task is accomplished with the use of a feature matching algorithm that is capable of matching descriptors of the detected features of a model image to the descriptors of detected features in a located image. In this application, two main feature matching algorithms have been evaluated for matching feature descriptors that have been generated for the detected features in I_M and I_L.

- BruteForce Matcher [25] (L1-norm, L2-norm and Hamming distance matching)
- Fast Library for Approximate Nearest Neighbour (FLANN) [25]

It is vital to evaluate the performance of these algorithms based on their stability and accuracy of matching features under varying illumination conditions, since this AOI system works without any enclosed chambers. This analysis was performed in two ways as listed below.

- Percentage of erroneous matching between given I_M and I_L
- Percentage of correct matching between given I_M and I_L

Here, the erroneous detection percentage means the ratio between the number of matched keypoints with a wrong I_L and the detected keypoints in I_M. The successful detection percentage means the ratio between the number of matched keypoints in given I_L of same model image and the detected keypoints in I_M. Tables 2.2-2.4 illustrate the erroneous detection rate of these three feature detectors together with feature descriptor matching algorithms at different illumination levels. Figures 2.14-2.16 illustrate several examples of the application of these feature detectors together with descriptor matching algorithms, acquired under normal illumination level.

Table 2.2: Percentage of erroneous matching between given model images and located images at (250-750) lux

Algorithm Name	**L1-Norm**	**L2-Norm**		**FLANN**
SIFT	6.93%	5.72%		0.13%
SURF	0%	0.09%		0.56%
Algorithm Name	**L1-Norm**	**L2-Norm**	**Hamming**	**FLANN**
FAST	12.66%	5.23%	7.33%	0.02%

Table 2.3: Percentage of erroneous matching between given model images and located images at (75-200) lux

Algorithm Name	**L1-Norm**	**L2-Norm**		**FLANN**
SIFT	10.05%	9.02%		2.1%
SURF	1.9%	1.3%		1.1%
Algorithm Name	**L1-Norm**	**L2-Norm**	**Hamming**	**FLANN**
FAST	19.5%	11.2%	18%	1.8.%

Table 2.4: Percentage of erroneous matching between given model images and located images at (1000-1500) lux

Algorithm Name	**L1-Norm**	**L2-Norm**		**FLANN**
SIFT	11.5%	10%		2.8%
SURF	1.2%	1.8%		1.3%
Algorithm Name	**L1-Norm**	**L2-Norm**	**Hamming**	**FLANN**
FAST	17.3%	16.8%	14.5%	4.8%

According to the results obtained in Tables 2.2-2.4, it can be seen that SURF is still able to hold the erroneous detection rate below 2% for each feature matching algorithm, even though FAST holds the lowest figure with FLANN in Table 2.2. Another interesting point that can be noticed with the obtained results is that FLANN feature matching library provides the best accuracy for all three feature detectors. Tables 2.5-2.7 illustrate the successful average detection percentage obtained by these algorithms.

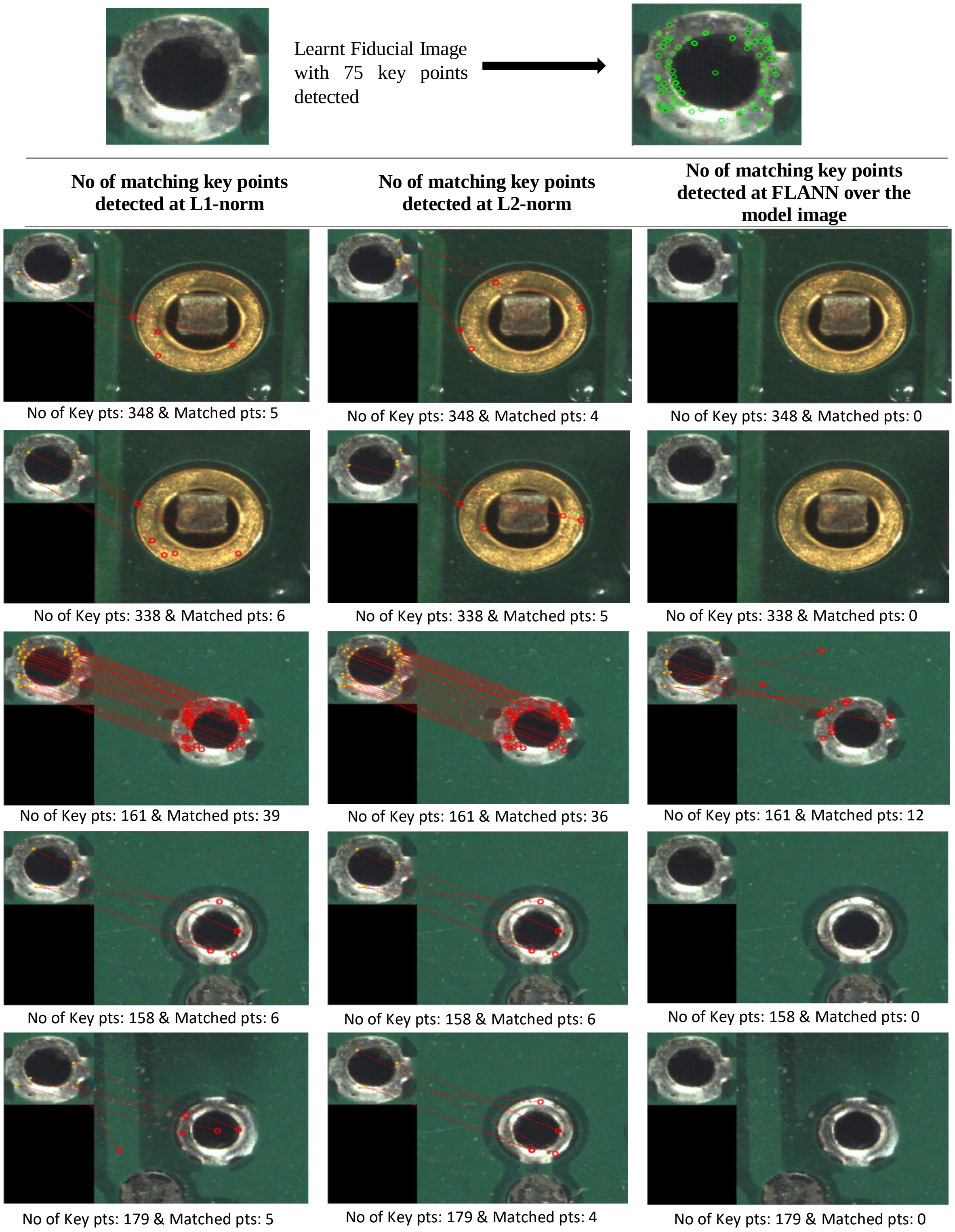
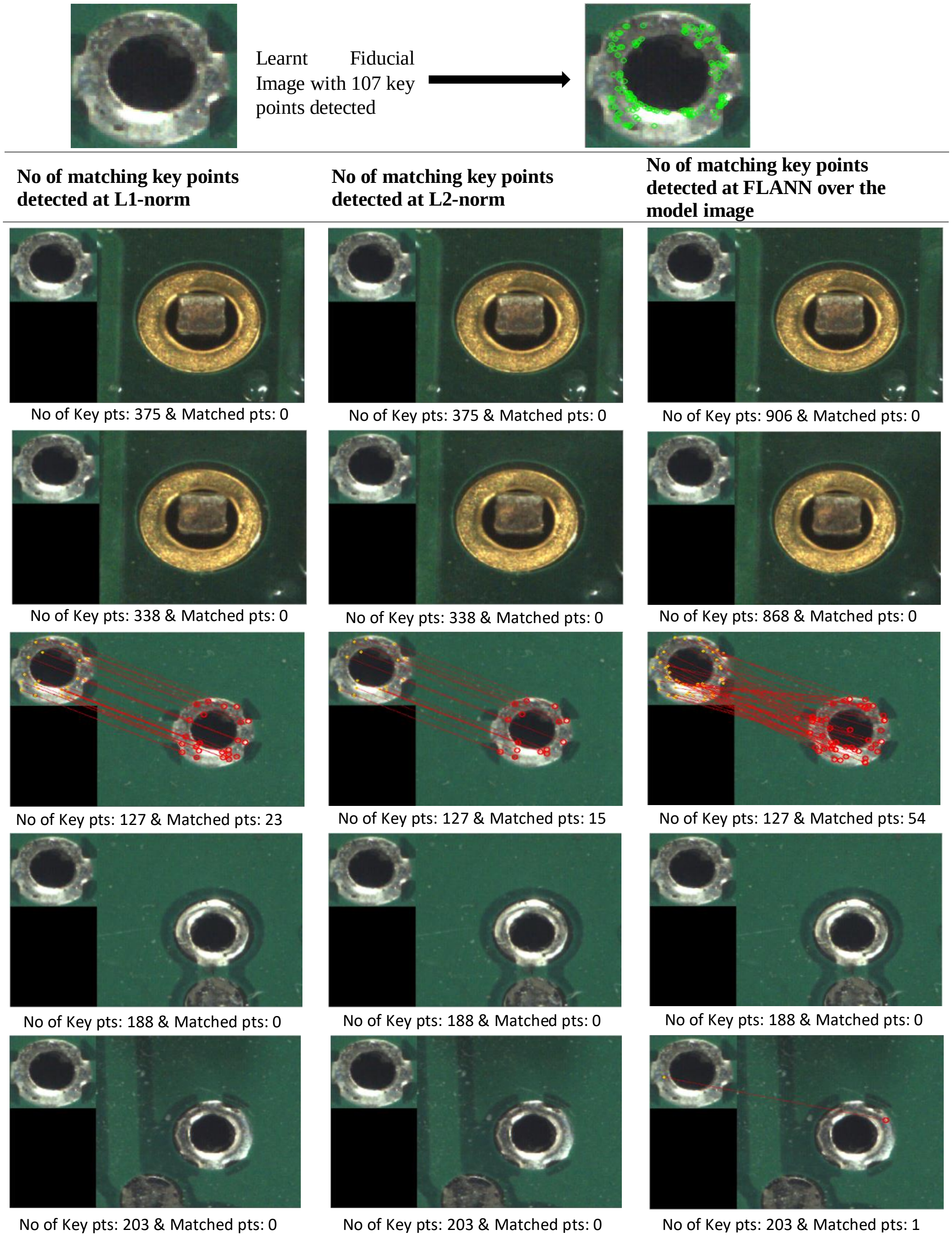

Learnt Fiducial Image with 75 key points detected
No of matching key points detected at L1-norm
No of matching key points detected at L2-norm
No of matching key points detected at FLANN over the model image
No of Key pts: 348 & Matched pts: 5
No of Key pts: 348 & Matched pts: 4
No of Key pts: 348 & Matched pts: 0
No of Key pts: 338 & Matched pts: 6
No of Key pts: 338 & Matched pts: 5
No of Key pts: 338 & Matched pts: 0
No of Key pts: 161 & Matched pts: 39
No of Key pts: 161 & Matched pts: 36
No of Key pts: 161 & Matched pts: 12
No of Key pts: 158 & Matched pts: 6
No of Key pts: 158 & Matched pts: 6
No of Key pts: 158 & Matched pts: 0
No of Key pts: 179 & Matched pts: 5
No of Key pts: 179 & Matched pts: 4
No of Key pts: 179 & Matched pts: 0

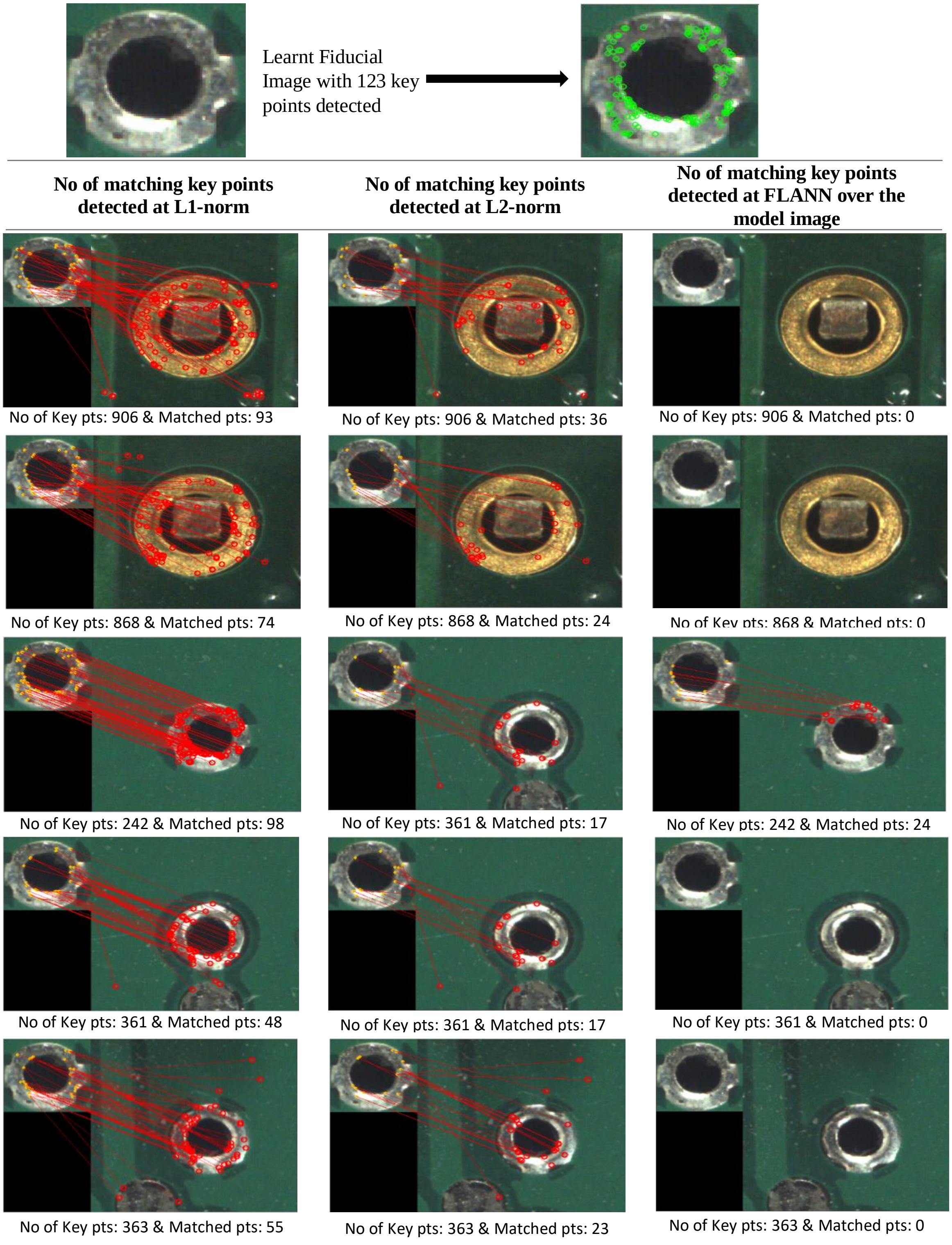
Figure 2.14: Feature matching between images using SIFT together with distance measurement algorithms



Learnt Fiducial Image with 107 key points detected

No of matching key points detected at L1-norm	No of matching key points detected at L2-norm	No of matching key points detected at FLANN over the model image
No of Key pts: 375 & Matched pts: 0	No of Key pts: 375 & Matched pts: 0	No of Key pts: 906 & Matched pts: 0
No of Key pts: 338 & Matched pts: 0	No of Key pts: 338 & Matched pts: 0	No of Key pts: 868 & Matched pts: 0
No of Key pts: 127 & Matched pts: 23	No of Key pts: 127 & Matched pts: 15	No of Key pts: 127 & Matched pts: 54
No of Key pts: 188 & Matched pts: 0	No of Key pts: 188 & Matched pts: 0	No of Key pts: 188 & Matched pts: 0
No of Key pts: 203 & Matched pts: 0	No of Key pts: 203 & Matched pts: 0	No of Key pts: 203 & Matched pts: 1

Figure 2.15: Feature matching between images using SURF together with distance measurement algorithms



Learnt Fiducial Image with 123 key points detected

No of matching key points detected at L1-norm

No of matching key points detected at L2-norm

No of matching key points detected at FLANN over the model image

No of Key pts: 906 & Matched pts: 93

No of Key pts: 906 & Matched pts: 36

No of Key pts: 906 & Matched pts: 0

No of Key pts: 868 & Matched pts: 74

No of Key pts: 868 & Matched pts: 24

No of Key pts: 868 & Matched pts: 0

No of Key pts: 242 & Matched pts: 98

No of Key pts: 361 & Matched pts: 17

No of Key pts: 242 & Matched pts: 24

No of Key pts: 361 & Matched pts: 48

No of Key pts: 361 & Matched pts: 17

No of Key pts: 361 & Matched pts: 0

No of Key pts: 363 & Matched pts: 55

No of Key pts: 363 & Matched pts: 23

No of Key pts: 363 & Matched pts: 0

Figure 2.16: Feature matching between images using FAST together with distance measurement algorithms

Table 2.5: Percentage of successful matching between given model images and located images at (250-750) lux

Algorithm Name	**L1-Norm**	**L2-Norm**		**FLANN**
SIFT	26%	24.2%		10.5%
SURF	18.5%	17.7%		41.1%
Algorithm Name	**L1-Norm**	**L2-Norm**	**Hamming**	**FLANN**
FAST	17.3%	8.4%	14.9%	8.2%

Table 2.6: Percentage of successful matching between given model images and located images at (75-200) lux

Algorithm Name	**L1-Norm**	**L2-Norm**		**FLANN**
SIFT	21.8%	20.1%		7.8%
SURF	15.2%	14.3%		36.5%
Algorithm Name	**L1-Norm**	**L2-Norm**	**Hamming**	**FLANN**
FAST	14.3%	5.6%	10.1%	6.5%

Table 2.7: Percentage of successful matching between given model images and located images at (1000-1500) lux

Algorithm Name	**L1-Norm**	**L2-Norm**		**FLANN**
SIFT	20.3%	19.3%		8.4%
SURF	14.5%	14.1%		35.3%
Algorithm Name	**L1-Norm**	**L2-Norm**	**Hamming**	**FLANN**
FAST	13.1%	4.8%	9.3%	6.2%

Even though, SIFT holds the highest detection rate for both L1-Norm and L2-Norm, SURF is able to provide the highest successful detection rate together with FLANN under different illumination levels according to the results obtained in Tables 2.5–2.7. Furthermore, SURF has been able to provide consistent rate above 30% in all situations. Figures 2.14 - 2.16 further reinforce this conclusion. Since the fiducial verification is carried out only once for a PCB, computation time is not a critical parameter. The results obtained in Tables 2.1-2.7, show that SURF together with FLANN render more stable and accurate detection rate for fiducial verification in this application.

The next step is the precise localization of model image area inside the located image once the proposed feature detection algorithm assures the existence of the model image inside the located image. The localized fiducial images are then used to calculate the relative distances to other points on the PUS reference to the distances from these points to system origin. Therefore, it is imperative to localize the exact area contained by the fiducial image very precisely. The performance of two methods was analysed in order to accomplish this task at highest accuracy as listed below.

- Using homography matrix
- Using template matching techniques

A homography is a perspective transformation of a plane that is a re-projection of a plane from one camera view into a different camera view subjective to change in the position and orientation of the camera. Homography matrix is a 3×3 transformation matrix that is used to perform this transformation between planes. It is required to have minimum of four points to find this transformation matrix. In this application, set of points that have been matched between I_M and I_L, were used and the corresponding perspective transformation matrix was found. This matrix is used to find the corners of the region in I_L that are corresponding to points in I_M. Figure 2.17 illustrates the results obtained with the use of a homography matrix generated from matched points between several images.

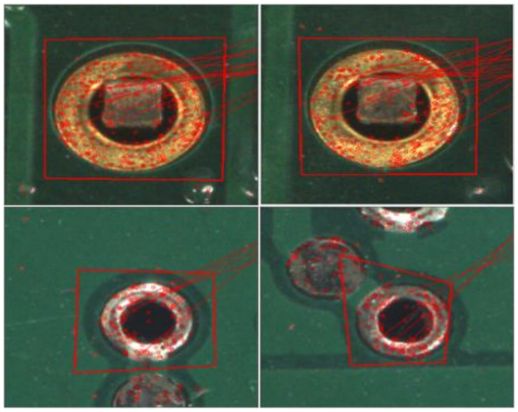


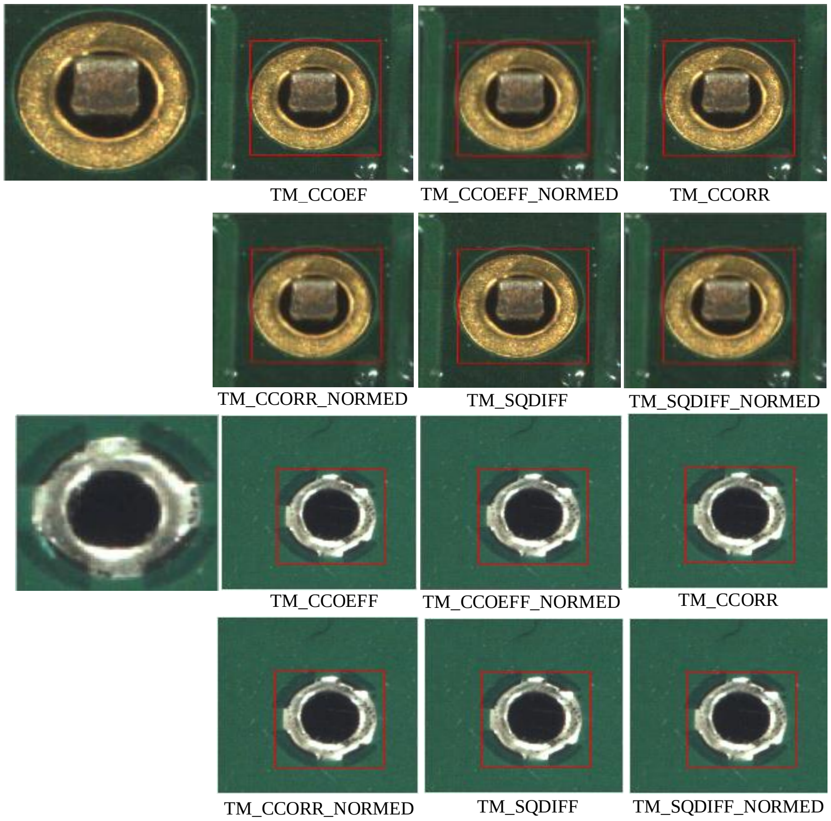
Figure 2.17: Falsely detected areas of the model image inside the input image using homography matrix

In Figure 2.17, it is obvious that the application of homography is not suitable, since the irregular shape can introduce offset between the centres of I_M and I_L. It leads to wrong computation of relative distances to other pads. It has been proved earlier in this thesis that template matching would not be a good choice to localize a model image inside a given input image, when the existence of model image is unknown. But at this stage of the application, the existence of I_M inside I_L has been confirmed with SURF. Therefore, there might be a possibility to have a precise localization of model image area inside located image using template matching algorithms. Section 2.5 presents a performance evaluation of six different template matching algorithms over a sample of 100 images.

2.5 Performance Evaluation of Template Matching Algorithms

Template matching is a high level machine vision technique that localizes the best matched location of a model image inside a given input image [22, 25]. Template matching techniques are flexible and relatively straightforward to use, which makes them one of the popular methods of object localization in the fields of surveillance, vehicle tracking, robotics, medical imaging, manufacturing etc. The occurrence of similarities, resulted from comparison between I_M and I_L, are reflected in the resulted image (I_R). In this research, six types of template matching techniques, TM_SQDIFF, TM_SQDIFF_NORMED, TM_CCORR, TM_CCORR_NORMED, TM_CCOEFF and TM_CCOEFF_NORMED [25] have been discussed and analysed for their accuracy in identifying the location of I_M inside I_L.

The basic concept of a template matching algorithm is to find objects in a given image, which have minimum error difference with a model image [25]. The complexity of these methods varies starting from simple squared difference of error between I_M and I_L to more complex processes like computing correlation and correlation coefficient between I_M and I_L. The performance of each template matching technique is evaluated to find out the best suited algorithm for the localization of fiducial area inside the I_L. Figure 2.18 illustrates the outcome of these algorithms over the feature matched images using SURF. Table 2.8 illustrates the acquired results over a sample of 100 different images. Here the successful detection rate means the Intersection over Union (IoU) [25] is greater than 0.8 between I_M and I_L.



TM_CCOEF TM_CCOEFF_NORMED TM_CCORR

TM_CCORR_NORMED TM_SQDIFF TM_SQDIFF_NORMED

TM_CCOEFF TM_CCOEFF_NORMED TM_CCORR

TM_CCORR_NORMED TM_SQDIFF TM_SQDIFF_NORMED

Figure 2.18: Model image localization inside given input image using template matching algorithms

Table 2.8: Successful detection rate of template matching algorithms over the feature matched located images

Template Matching Algorithm					
TM_CCOEFF	TM_CCOEFF_ NORMED	TM_CCORR	TM_CCORR_N ORMED	TM_SQDIFF	TM_SQDIFF_N ORMED
100%	100%	100%	100%	100%	100%

The results obtained in Table 2.8 shows that almost every algorithm showed the best performance in precise localization of the given model image. However, these results have been obtained assuming an ideal condition of a given PCB. It means that it is assumed that the PCB has no manufacturing impurities like, the size differences of objects in PCBs. The template matching algorithms are vulnerable to dimension changes of input images.

Therefore, it is required to find the most robust technique out of these algorithms, which is less vulnerable to slight dimension changes of located images. Therefore, the located images were downscaled by 2% from their actual size and verified with the template matching algorithms. Figure 2.19 illustrates the outcome of this process.

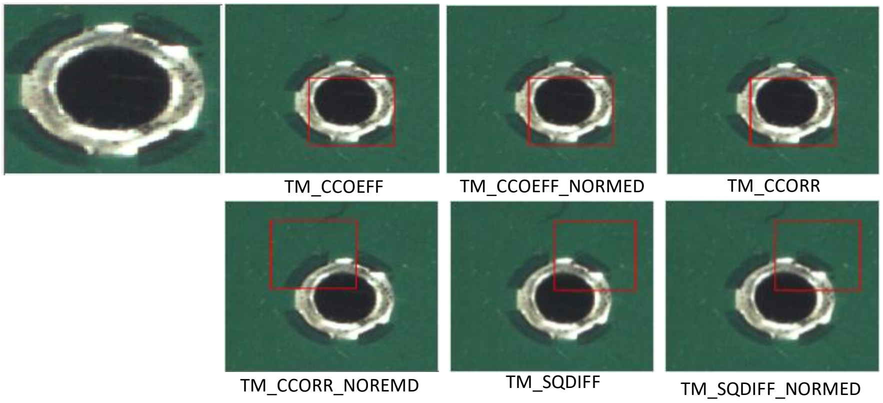


Figure 2.19: Results obtained from template matching algorithm over 2% scaled down images

According to Figure 2.19, it can be seen that TM_CCORR is capable of rendering most accurate results even with the slight variation of size of a located image. Table 2.9 contains the results obtained over a sample of 100 different scaled down images by a percentage of 2%.

Table 2.9: Successful detection rate of template matching algorithms over the scaled down located images by a percentage of 2%

Template Matching Algorithm	**Success Rate**
TM_CCOEFF	92%
TM_CCOEFF_NORMED	92.5%
TM_CCORR	95%
TM_CCORR_NORMED	80%
TM_SQDIFF	82%
TM_SQDIFF_NORMED	83%

Even though TM_CCORR gives the highest successful detection rate, it is still not perfect. However, the effect of this 5% error can be made further minimized with the procedures described in Chapter 3 that describes the precise identification of solder pad area from the PCB surface. According to the results presented so far in this section it should be clear that the proposed methodology could provide a stable platform for fiducial verification. Figure 2.20 illustrates how this fiducial verification process is implemented in this application.

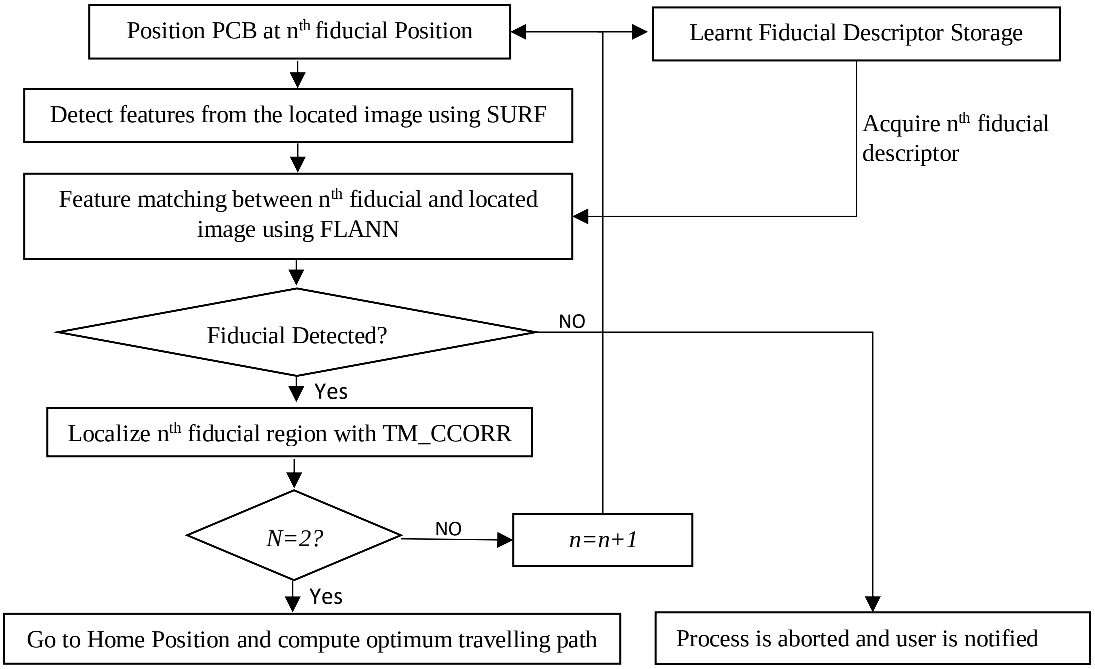

Position PCB at nth fiducial Position
Learnt Fiducial Descriptor Storage
Detect features from the located image using SURF
Acquire nth fiducial descriptor
Feature matching between nth fiducial and located image using FLANN
Fiducial Detected?
NO
Yes
Localize nth fiducial region with TM_CCORR
N=2?
NO
n=n+1
Yes
Go to Home Position and compute optimum travelling path
Process is aborted and user is notified

Figure 2.20: Process flow chart of the proposed fiducial verification process

The accurate identification of fiducials adds significant benefits to the overall performance of the soldering system as listed below. It provides:

- precise positioning of the points to be soldered;
- Enables to compute the optimum travelling path based on the computed relative distances to component solder pads and positions them automatically.

Once the fiducials are precisely localized by the vision system, the next step is to calculate the relative distances to the THT solder pads to be soldered and positions them accordingly. Chapter 3 presents how the implemented vision system identifies the area contained by the solder pad inside positioned image precisely.

3. Identification and Localization of Bare Solder Pad from PCB Surface

Commercial AOI systems virtually process an acquired image with reference to a learnt model image, whereas the implemented AOI system inspects each stage of the soldering process without prior teaching. Therefore, it is vital for the developed AOI system to identify the exact area contained by the solder pad, since it enables to perform:

- Real time position calibration by comparing the distance offset between identified solder pad centre and the centre of the camera Field Of View (FOV);
- Verification of solder bridging situations through size comparison;
- Segmentation of geographically important areas inside the detected solder pad like drill-hole and pad regions (Discuss in Chapter 5).

The complexity of this task is further increased due to different combinations of solder mask and pad colours. Therefore, a detailed analysis process was carried out to find the optimum way through analysing applicability of colour models on such different combinations. Chapter 3 discuss how the solder pads are precisely segmented from their background.

3.1 Overview on Colour Models

Colour is perceived by humans as a combination of three primary colours R (Red), G (Green) and B (Blue). Video monitors and televisions display colour images by modulating the intensity of RGB components at each pixel of the image. Even though the RGB model is suitable for colour displays, it is not suitable for colour image processing because of high correlation among R, G, and B components [40]. Several colour models like HSV, HSL, YIQ, YC_bC_r, $I_1I_2I_3$, RGI, CMYK, XYZ and Lab, are used in image processing applications, which can be derived using either linear or non-linear transformations from RGB colour space [40]. Selecting a best colour model that suits for an image processing application is a challenging task. Figure 3.1 illustrates how RGB colour model is represented in a three-dimensional space while Figure 3.2 and Figure 3.3 illustrate the mapping of colours for HSV and HSL colour models respectively.

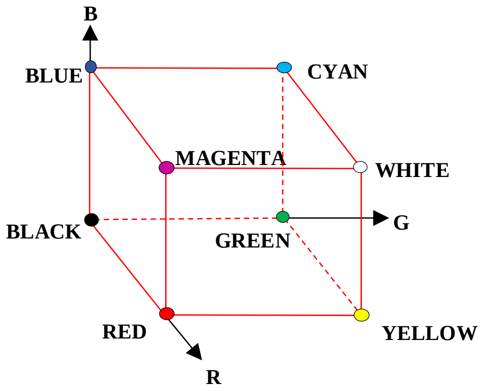


Figure 3.1: RGB colour

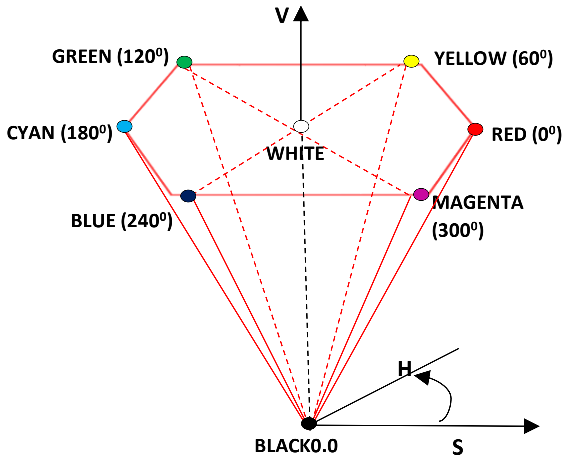


Figure 3.2: HSV colour

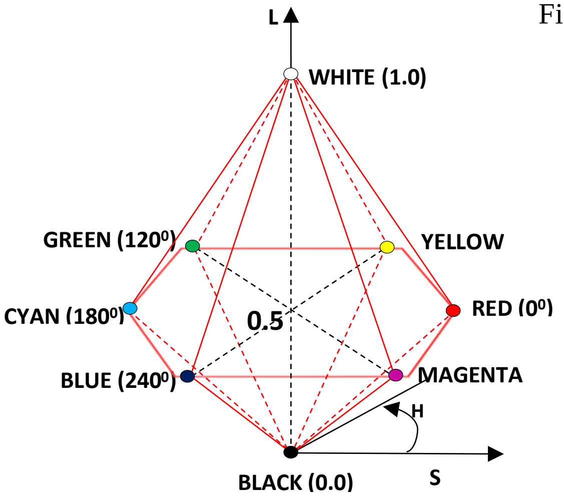


Figure 3.3: HSL colour model

HSV is a general purpose colour model where colour is represented in terms of hue (H), saturation (S) and brightness (V) [41]. The colour transformation from RGB colour model to HSV colour model is carried out through Eq. (14), Eq. (15) and Eq. (16).

$$H = 60^0 * H' \tag{14}$$

$$S = \begin{cases} 0, & if\ C = 0 \\ \frac{C}{M}, & if\ C \neq 0 \end{cases} \tag{15}$$

$$V = M \tag{16}$$

Where,

$$H' = \begin{cases} 0^0, & if\ C = 0 \\ \frac{(G-B)}{C}\ mod6, & if\ M = R \\ \frac{(B-R)}{C} + 2, & if\ M = G \\ \frac{(R-G)}{C} + 4, & if\ M = B \end{cases}$$

$M = \max(R, G, B)$, $m = \min(R, G, B)$ and $C = M - m$

In the HSV colour model hue (H) is expressed as an angle from 0^0 to 360^0. Hue of red, yellow, green, cyan, blue and magenta is represented as 0^0, 60^0, 120^0, 180^0, 240^0 and 300^0 respectively [41]. Saturation (S) is the depth or purity of the colour which is represented as the distance from the vertical axis. Distance along the vertical axis represents the brightness value (V). HSL is another general purpose colour model where colour is represented in terms of hue (H), saturation (S) and lightness (L). Transformation of RGB colour model to HSL colour model is carried out through Eq. (17), Eq. (18) and Eq. (19). The computation of *H* is carried in a similar way as defined in Eq. (14).

$$S = \begin{cases} 0 & , \Delta = 0 \\ \Delta/1 - |2L - 1| & , \Delta \neq 0 \end{cases} \tag{17}$$

$$I = \frac{(C_{max} + C_{min})}{2} \tag{18}$$

$$\Delta = C_{max} - C_{min} \tag{19}$$

Where,

$C_{max} = max(R', G', B')$ and $C_{min} = min(R', G', B')$

$R' = R/255$, $G' = G/255$ and $B' = B/255$

Similar to HSV model, in the HSL colour model *H* is also expressed as an angle from 0^0 to 360^0. Hue of red, yellow, green, cyan, blue and magenta is represented as 0^0, 60^0, 120^0, 180^0, 240^0 and 300^0 respectively. *S* is represented as the distance from the vertical axis and distance along the vertical axis represents the lightness *L*.

From Figures 3.1-3.3, it can be observed that colours are mapped to different points in 3-D Cartesian coordinate system by applying HSV and HSL colour transformation. For an example, in the original RGB model, yellow colour is represented by coordinate (0, 1, 0) and blue colour is represented by coordinate (0, 0, 1). The vector length between these two colours is $\sqrt{3}$. In HSV model, yellow and blue colour is represented by coordinates (60^0, 1, 1) and (240^0, 1, 1) respectively. The vector length between these two colours in HSV model is 2, which is larger than the vector length in the original RGB model. Further in the HSL model, yellow and blue colour is represented by coordinates (60^0, 1, 0.5) and (240^0, 1, 0.5) respectively. The vector length between these two colours in HSV model is 2. Effectiveness of extraction of certain colour features of an image can be improved by maximizing the vector length. Hence colour transformation plays an important role in extracting features of a colour image. There are many more colour transformations such as YIQ, YC_bC_r, $I_1I_2I_3$, XYZ, Lab etc.

YIQ is the colour model used to encode colour information in TV signal for NTSC TV system and it is intended to take advantage of human colour-response characteristics [41]. Transformation of RGB colour model to YIQ colour model is carried out through Eq. (20), Eq. (21) and Eq. (22).

$$Y = 0.299 * R + 0.587 * G + 0.114 * B \tag{20}$$

$$I = 0.596 * R - 0.275 * G - 0.321 * B \tag{21}$$

$$Q = 0.212 * R - 0.523 * G + 0.311 * B \tag{22}$$

The Y component, is a measure of the luminance of the colour, and is a prominent candidate for edge detection in a colour image while I and Q components describe the hue and saturation of the image [40].

YC_bC_r Model is used as a part of the colour image pipeline in video and digital photography systems. Transformation of RGB colour model to YC_bC_r colour model is carried out through Eq. (23), Eq. (24) and Eq. (25).

$$Y \quad = 0.299 * R + 0.587 * G + 0.114 * B \tag{23}$$

$$C_b \quad = -0.169 * R - 0.331 * G + 0.5 * B \tag{24}$$

$$C_r \; = 0.5 * R - 0.419 * G - 0.081 * B \tag{25}$$

The Y component is the luma component and *C_b* and *C_r* are the blue difference and red difference chroma components [41].

$I_1I_2I_3$ is another linear colour transformation from RGB space used in image segmentation applications [41]. Transformation of RGB colour model to $I_1I_2I_3$ colour model is carried out through Eq. (26), Eq. (27) and Eq. (28).

$$I1 \; = (R + G + B)/3 \tag{26}$$

$$I2 \; = (R - \; B)/2 \tag{27}$$

$$I3 \; = (2 * G - R \; - \; B)/4 \tag{28}$$

Tz-Sheng Peng and Chiou-Shann Fuh described that modified version of $I_1I_2I_3$ colour space rendered better results for solder joint segmentation among other models [42]. Transformation from RGB colour model to modified $I_1I_2I_3$ colour model is carried out through Eq. (29), Eq. (30) and Eq. (31).

$$CH0 \; = (R + G + B)/3 \tag{29}$$

$$CH1 \; = (R - \; B) \tag{30}$$

$$CH2 \; = (2 * G - R \; - \; B) \tag{31}$$

XYZ colour model, which is one of the most important colour spaces defined by the Commission Internationale de I'Elcairage (CIE). This colour space is the most common way in technical colorimetric work to describe the colour of light. Transformation from RGB colour model to XYZ colour model is carried out through Eq. (32).

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = [M] \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{32}$$

Where,

$$[M] = \begin{bmatrix} S_rX_r & S_gX_g & S_bX_b \\ S_rY_r & S_gY_g & S_bY_b \\ S_rZ_r & S_gZ_g & S_bZ_b \end{bmatrix}$$

$$X_r = \frac{x_r}{y_r}$$

$$Y_r = 1$$

$$Z_r = (1 - x_r - y_r)/y_r$$

$$X_g = x_g/y_g$$

$$Y_g = 1$$

$$Z_g = (1 - x_g - y_g)/y_g$$

$$X_b = x_b/y_b$$

$$Y_b = 1$$

$$Z_b = (1 - x_b - y_b)/y_b$$

$$\begin{bmatrix} S_r \\ S_g \\ S_b \end{bmatrix} = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix}^{-1} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix}$$

Here, *(x_r , y_r), (x_g , y_g), (x_b , y_b)* are the chromacity coordinates of a given RGB system and *(X_w , Y_w, Z_w)* are the reference white of these coordinates.

Lab colour space is designed to approximate human vision where colour is represented in terms of lightness (L) while a and b components describe colour component dimensions based on non-linearly compressed coordinates. This colour space is useful for sharpening images and removing artifacts in images from digital cameras and scanners. Transformation from RGB colour model to Lab colour model is carried out through Eq. (33), Eq. (34) and Eq. (35).

$$L = 116 * f(Y/Y_n) - 16 \tag{33}$$

$$a = 500\,[f(X/X_n) - f(Y/Y_n)\,] \tag{34}$$

$$b = 500\,[f(Y/Y_n) - f(Z/Z_n)\,] \tag{35}$$

Where,

$$f(t) = \begin{cases} t^{1/3} & , if\ t > \left(\frac{6}{29}\right)^3 \\ \frac{1}{3}\left(\frac{29}{6}\right)^2 t + \frac{4}{29} & , \quad otherwise \end{cases}$$

Here X_n, Y_n and Z_n are the CIE XYZ tristimulus values of the reference white point.

3.2 Performance Analysis of Colour Models for Solder Pad Identification

Even though these colour models have been proven to render better results over many applications throughout the years, their robustness, stability and accuracy must be verified with a mathematical approach for solder pad segmentation. Suitability of these colour models for solder pad segmentation may vary due to the varying parameters of solder pads, PCB surface finish, PCB colour and silk screen layer. For accurate solder pad segmentation, a colour model which maximize the vector distance between the solder pad area (foreground) and immediately surrounding PCB area (background) is better suited. Typical solder pad foreground and background for two different PCB types are shown in Figure 3.4.

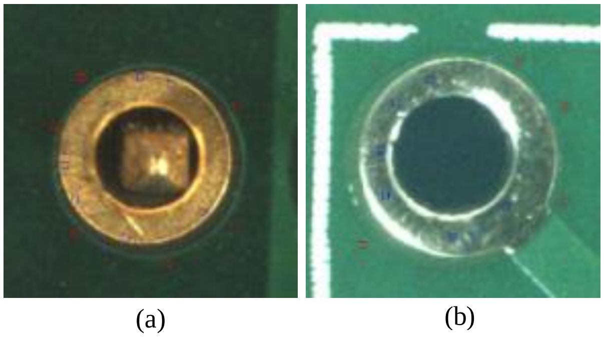


Figure 3.4: Selected foreground and background regions of a given solder pad. (a). Gold plated solder pad on green PCB surface. (b). Tin plated solder pad on green PCB surface

Even though the gold-plated solder pad shows a significant colour difference between foreground and background (Figure 3.4 (a)), the situation is not similar for tin plated solder pad (Figure 3.4 (b)). The Average (AVG) and the Standard Deviation (STD) of the vector distance between the foreground and background of following PCB solder mask and pad plating colours were analysed (refer Table 3.1 for the acquired results for nine colour models).

- Green PCBs with gold-plated and tin-plated solder pads
- Blue PCBs with gold-plated solder pads
- Red PCBs with tin-plated solder pads
- Black PCBs with tin-plated solder pads

In this analysis process, background and foreground were carefully selected with minimum noise components.

Table 3.1: Data Analysis for Different Combinations of Solder Mask and Solder Pad Colours

Solder Mask and Pad Colour Combination	**Calculate Value**	**Colour Models**								
		RGB	**HSV**	**HSL**	**YIQ**	**$I_1I_2I_3$ Modified**	**YC_bC_r**	**XYZ**	**$I_1I_2I_3$**	**Lab**
Green PCB-Gold Plated Pads	AVG	165.6	134.1	127	**215.5**	128.4	105.9	152.9	168.7	120.8
	STD	20.9	17.9	29.5	**9.4**	17	13.1	24.6	26	14.2
Blue PCB-Gold Plated Pads	AVG	179.9	144.1	134.9	**298.4**	154.3	131.4	164.9	170.3	126.6
	STD	30.4	11.8	17.7	**12.8**	12.9	14.9	26.2	16.2	16.2
Green PCB-Tin Plated Pads	AVG	43.1	75.1	59.3	27.9	63.9	**187.9**	25.4	35.8	21.8
	STD	14.2	8.3	7.08	5.3	7.9	**38.9**	16.9	33.6	4.3
Red PCB-Tin Plated Pads	AVG	104.6	**176.2**	150.6	74.6	63.4	56.3	38.8	62.8	61.3
	STD	3.9	**22.2**	23.5	3.1	6.2	3.1	2.01	26.5	1.3
Black PCB-Tin Plated Pads	AVG	**112**	74.1	69.3	76.9	48.6	69.4	108.6	72.2	72.4
	STD	**20.1**	11.9	10.9	8.9	8.7	12.4	20.3	8.6	12.2

Even though Tz-Sheng Peng and Chiou-Shann Fuh described that modified $I_1I_2I_3$ model rendered better results in solder joint segmentation [42], it is noted that other colour models record higher average of vector length between foreground and background as observed in Table 3.1. Figure 3.5 (a) illustrates the output of modified $I_1I_2I_3$ over these different PCB types. According to Table 3.1, YIQ colour model gives an optimum result with a larger vector distance between foreground and background of a solder pad with a low standard deviation with respect to the other colour models for green PCBs with gold plated solder pads. The Result originated from the YIQ colour space can also be compared with colour perception wheel which is illustrated in Figure 3.5 (b).

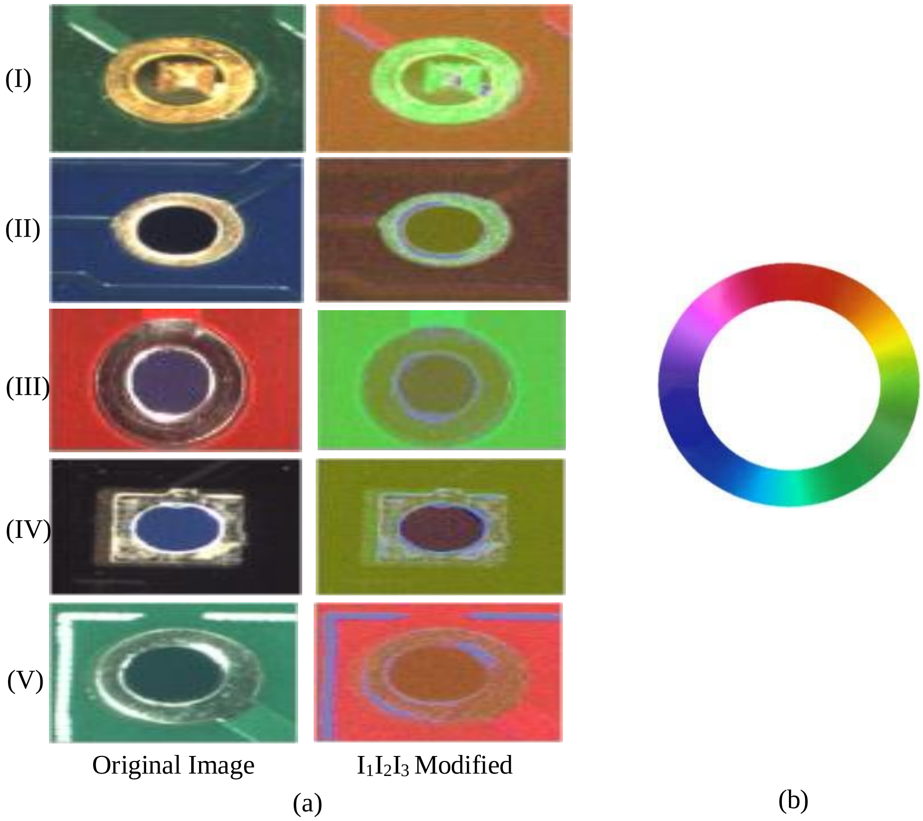


Figure 3.5: Visual comparison between colour transformed images using modified $I_1I_2I_3$ colour model. (I) Green PCB-Gold Plated. (II) Blue PCB-Gold Plated. (III) Red PCB-Tin Plated. (IV) Black PCB-Tin Plated. (V) Green PCB-Tin Plated. (a). Resulted images after transforming to modified $I_1I_2I_3$ colour model from RGB colour space. (b). Colour perception wheel

It can be observed that the distance between the positions of background colour and the foreground colour of the YIQ transformed image is much wider comparing to other colour models in the colour wheel (refer Figure 3.5 (b) for colour wheel). Figure 3.6 (a) illustrates first four colour models that give the maximum vector length difference between background and foreground. Furthermore, Figure 3.6 (b) illustrates few colour transformed solder pads using YIQ colour models at different conditions.

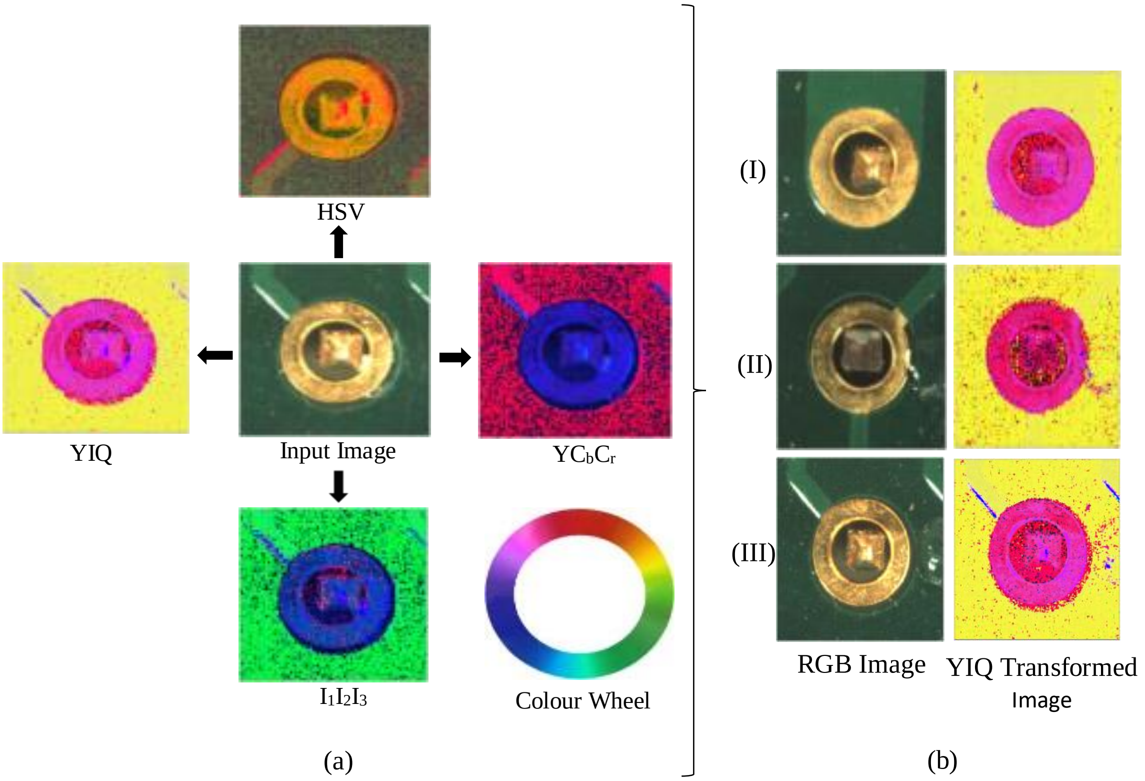


Figure 3.6: Colour space transformation of a gold-plated solder joint on green PCB. (a). Results from four prominent models for this combination. (b). Resulted YIQ colour transformed images at different conditions. (I). Solder pad placed on a different colour path. (II). Solder pad with flux. (III) Solder pad with an illuminated PCB track

According to Figure 3.6 (b) (II) and (III), it can be clearly seen that the application of YIQ colour model diminishes the impact of highly illuminated PCB tracks and applied flux by mapping those areas to separate colour components. However, such a separation is not possible with gray-scale image. This is one of the major drawbacks in the research described in [12]. YIQ colour model is also capable of giving a distinguishable outcome for blue PCBs with gold plated solder pads according to Table 3.1. The mapped colours of both foreground and background areas of the PCB, lie at almost opposite directions of the colour wheel. Figure 3.7 (a) shows the first four colour models that give the highest vector length difference for blue colour PCB. Furthermore, Figure 3.7 (b) illustrates few colour transformed solder pads using YIQ colour models at different conditions.

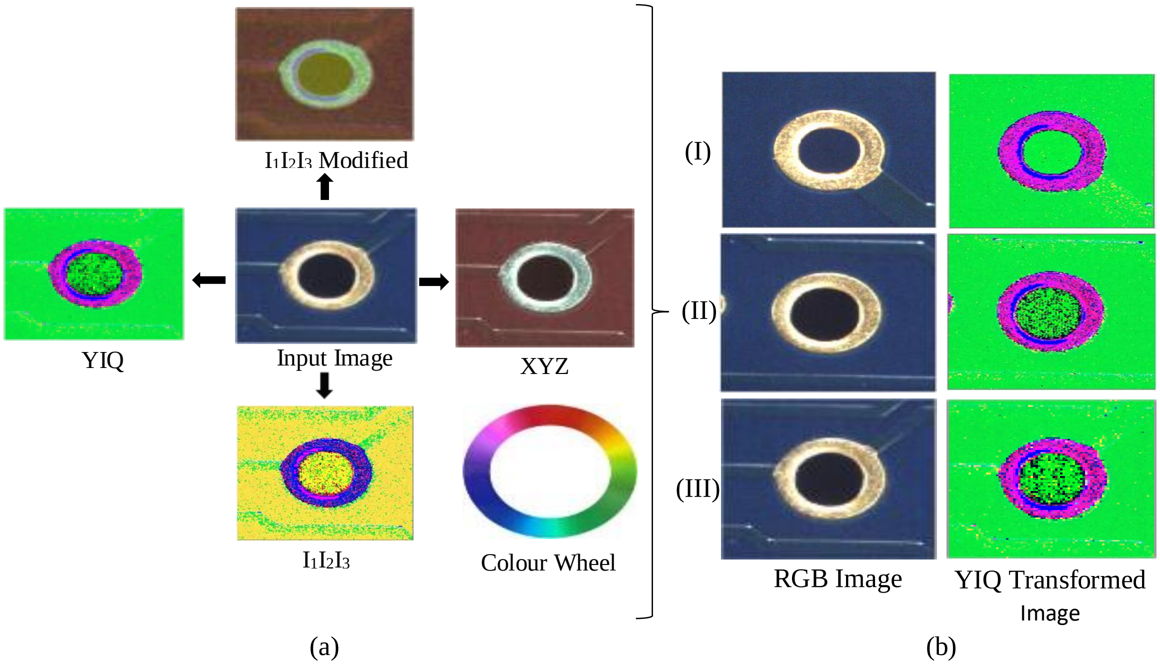


Figure 3.7: Colour space transformation of a gold-plated solder joint on blue PCB. (a). Results from four prominent models for this combination. (b). Resulted YIQ colour transformed images at different conditions. (I). Solder pad placed with a PCB track. (II). Solder pad without PCB track. (III) Solder pad with an illuminated PCB track

According to Table 3.1, YC_bC_r colour model gives distinguishable vector length difference for Green PCB with tin plated solder pads. The mapped colours lie at much wider distance of the colour wheel when compared to other colour models as illustrated in Figure 3.8 (a). Furthermore, Figure 3.8 (b) illustrates few colour transformed solder pads using YC_bC_r colour models at different conditions. The application of gray-scale image of input images (refer Figure 3.8 (b)) as described in [12] can lead to a false identification of pad area due to insignificant difference between foreground and background regions, if it is used for segmentation.

According to Table 3.1, HSV colour model gives the best vector length difference between background and foreground for tin plated solder pads on a red colour PCB. Figure 3.9 (a) shows the first four colour models that give the highest vector length difference for red colour PCB. Furthermore, Figure 3.9 (b) illustrates few colour transformed solder pads using HSV colour models at different conditions.

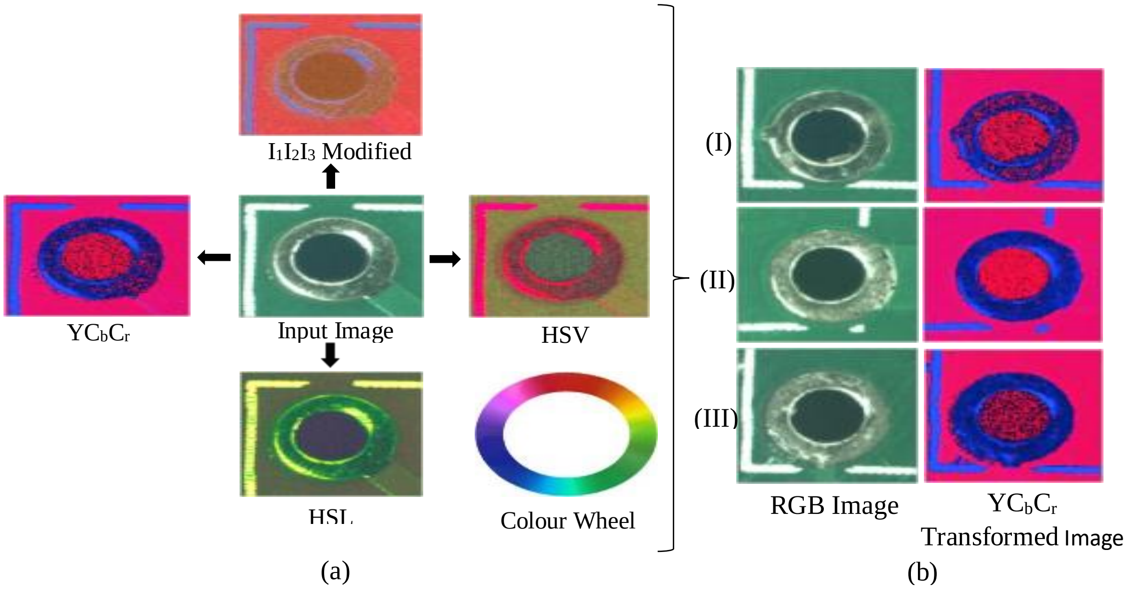


Figure 3.8: Colour space transformation of a tin-plated solder joint on green PCB. (a). Results from four prominent models for this combination. (b). Resulted YC_bC_r colour transformed images at different conditions. (I) Tin plated solder pad with darker colour pad closer to silk screen marking. (II) Tin plated solder pad with PCB track closer to silk screen marking. (III) Tin plated solder pad with light colour pad closer to a silk screen marking.

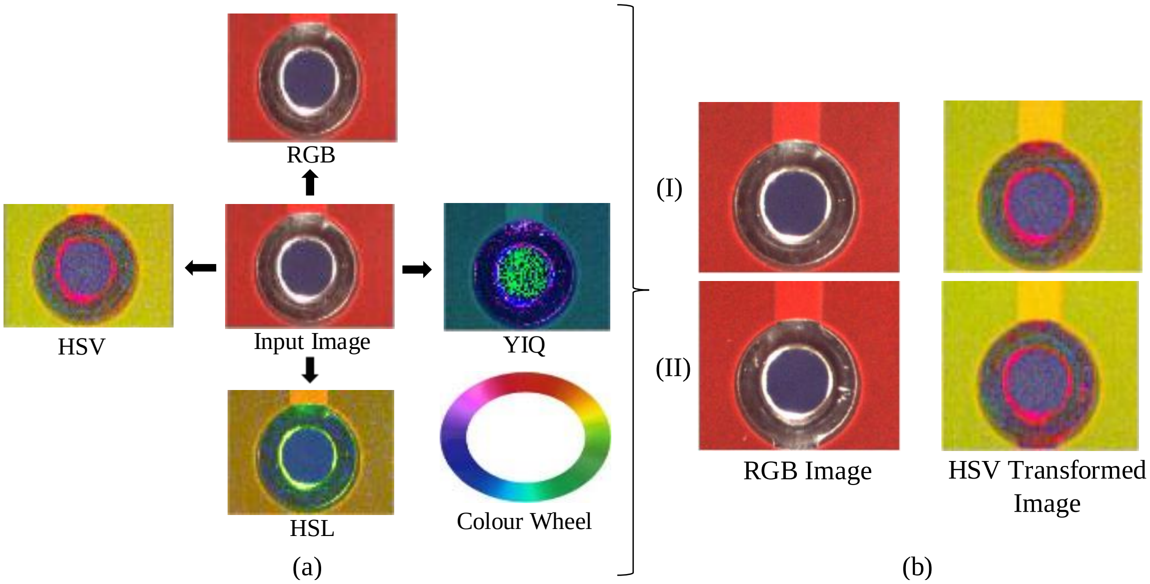


Figure 3.9: Colour space transformation of a tin-plated solder pad on red PCB. (a). Results from four prominent models for this combination. (b). Resulted HSV colour transformed images at different conditions. (I). Tin plated solder joint on a red colour PCB under less illuminated environment. (II). Tin plated solder joint on a red colour PCB under illuminated environment

According to Table 3.1, both RGB and XYZ colour spaces provide approximately similar results on tin plated solder joints on a black colour PCB. Figure 3.10 illustrates the outcome of colour models for a tin-plated solder pad on a black PCB.

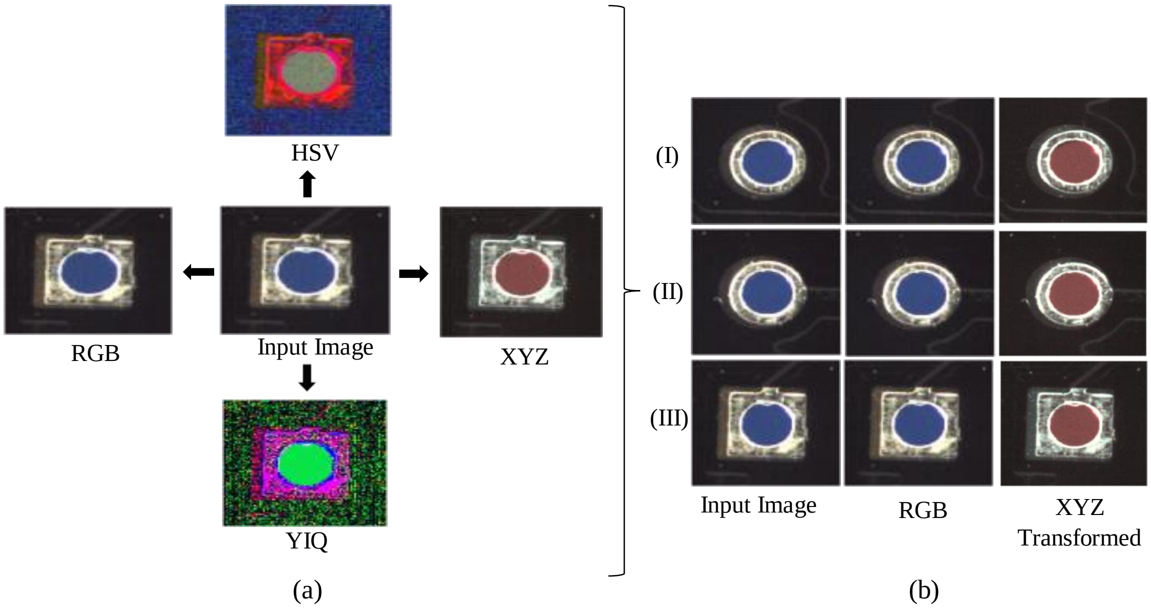


Figure 3.10: Colour space transformation of a tin-plated solder pad on black PCB. (a). Results from four prominent models for this combination. (b). Resulted images after RGB and XYZ colour transformation. (I). Solder pad without a PCB track. (II). Solder pad with a PCB track. (III). Solder pad with an offset with the PCB surface

Since the processing time is one of the most critical parameters, RGB image was taken as the processing model in this research for image segmentation.

3.3 Image Type Verification for Image Segmentation

The next step is to decide the appropriate image type (colour image or gray scale image) of colour transformed image to perform image segmentation. Same analysis procedure was carried out on gray scale images on each channel of colour models to compute the vector length difference between foreground and background. Table 3.2 illustrates the acquired results for separate channels of gray scale images and for a colour image itself for green PCBs with gold plated solder joints.

Table 3.2: Acquired results for average and standard deviation for separate channels of gray scale images and for a colour image of solder pads on green PCB gold-plated

Colour Model	**CH0**		**CH1**		**CH2**		**Colour Image**	
	AVG	**STD**	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**
RGB	32.7	7.5	90.3	12.2	159	14.7	186	16.8
HSV	48.8	2.18	56.2	10.7	132	13.8	153	14
HSL	48.8	2.17	87	9.57	104	26.7	145	21.9
YIQ	105	10.5	191	6.97	7.71	7.87	218.3	7.49
$I_1I_2I_3$ Mod.	70.6	7.01	120	10.6	15.6	12.7	140	11.8
YC_bC_r	104	10.5	13.4	2.3	99.2	46.6	148.5	32.9
XYZ	38.5	9.5	80.5	11.8	150	12.5	152.9	24.6
$I_1I_2I_3$	108	11.8	35.6	11.3	85	25.1	172	27.3
Lab	40.5	21.8	65.8	12.9	70.5	17.8	120.8	14.2

According to Table 3.2, it is clear that gray scale images cannot even provide significant vector length difference even for gold plated solder pads. Figure 3.11 reveals that no single channel in gray scale image provides a distinct colour difference between the illuminated PCB track and solder joint, whereas a significant colour difference is noticeable in colour image. These evidences further confirm the drawbacks of [12] which were stated in Section 1.3.

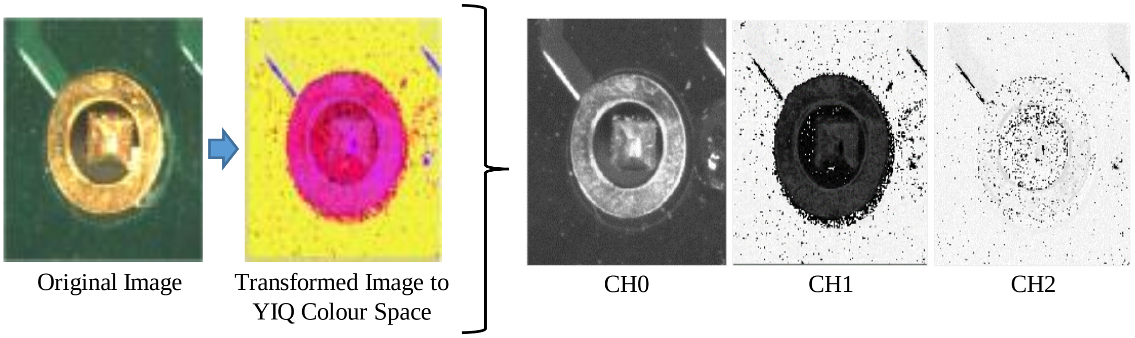


Figure 3.11: Channels of YIQ colour model

Consequently, colour image processing is used for the segmentation of solder pad from its immediate background. Section 3.4 provides a detailed overview how this task has been accomplished in this research.

3.4 Solder Pad Segmentation

Image segmentation is considered as one of the most prominent operations for meaningful analysis and interpretation of acquired image. Even though, it is one of the most difficult tasks, its stability and precise operation determine the quality of final outcome of any image processing application. There is no unique image segmentation technique that can be applicable to each application. Many studies are extensively carried out in the field of colour image processing, in order to find out more robust and adaptive algorithm for image segmentation. Recent works included variety of techniques: for example, stochastic model based approaches [43, 44], morphological watershed based region growing [45], energy diffusion [46], graph partitioning [47], Quantitative evaluation methods [48], K-Means clustering [49-52], Fuzzy-C Means clustering [53-55], Mean-Shift [56] etc.

The requirement of an efficient colour image segmentation algorithm is vital at this stage of application to reduce the processing time on a solder pad. The colour model analysis process described in Section 3.3 lessens the complexity of the solder pad segmentation process for different PCB surface and pad colour combinations. Therefore, the colour clustering algorithm, K-Means, was chosen to segment solder pads in the colour transformed image as illustrated in Figure 3.12.

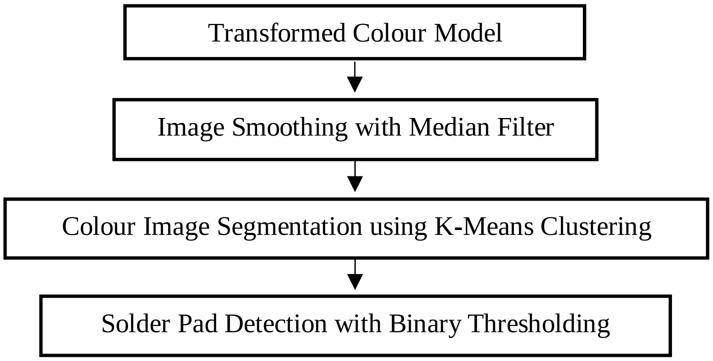

Transformed Colour Model
Image Smoothing with Median Filter
Colour Image Segmentation using K-Means Clustering
Solder Pad Detection with Binary Thresholding

Figure 3.12: Structure for Image Segmentation Process

The reason for thresholding the colour clustered image is to accurately identify the pad region under varying illumination levels.

Before going deeper into the algorithm, it would be better to understand the size of the acquired image. The size of the raw image from the camera is 1920×1080. Then a rectangular area of size 460×460, co-centred with the centre of the acquired image is defined as the Region of Interest (ROI). This is the standard size of the input image to the implemented vision system.

For effective image segmentation, removal of noise while preserving the colour boundaries is very important. Median filtering [25] was used to accomplish this task in this application. Figure 3.13 illustrates the effect of median filter on different PCB images.

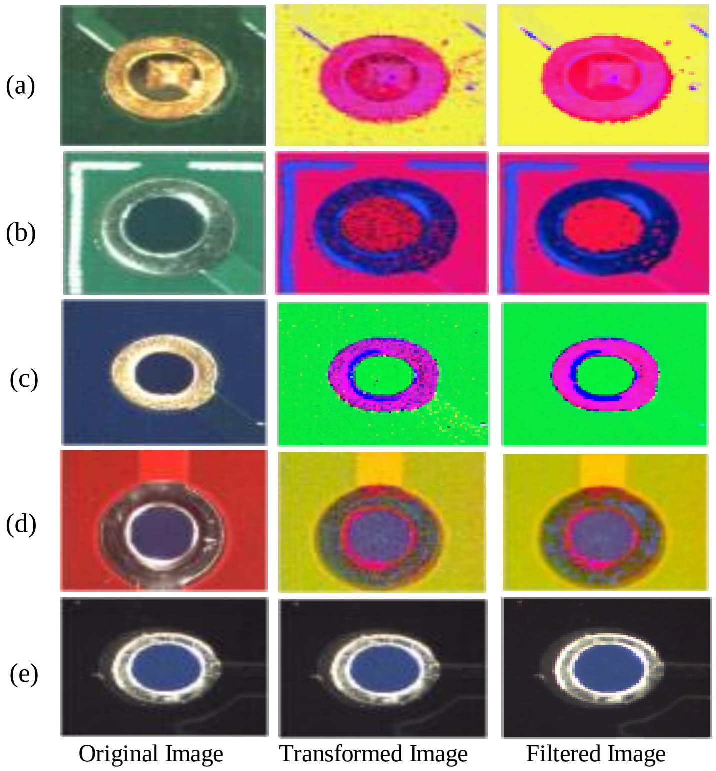


Figure 3.13: Noise filtered colour transformed images using median filter. (a) Output of median filter on gold-plated solder pad on green PCB. (b) Output of median filter on tin-plated solder pad on green PCB. (c) Output of median filter on gold-plated solder pad on blue PCB. (d) Output of median filter on tin-plated solder pad on red PCB. (e) Output of median filter on tin-plated solder pad on black PCB

The main step of image segmentation of a colour image is to classify each pixel in a given image into one of a discrete number of colour classes. In order to segment the image, colours in the image can be coarsely quantized to reduce the computational complexity.

The main concept of K-Means clustering is to define *K* number of centres derived from a given data set, where each cluster contains group of elements that have the minimum distance to that particular centre [25, 49, 51]. In a general segmentation process of an image with a distinct object area and the surrounding background, only two centres are required that differentiate pixels lie on the foreground and background. Algorithm randomly chooses two centroids C_1 and C_2 out of image data that possess the maximum vector length difference. Figure 3.14 illustrates the selection of two centroids in a data set which have distributed colour values over a limited range.

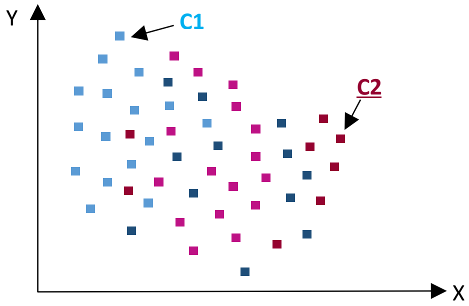


Figure 3.14: Selection of centroid on a given data set in K-means clustering

Then the distance from each point to both centroids is calculated and if the calculated distance is much closer to C_1, it is labelled as '1' and else it is labelled as '2'. Once the algorithm iterates through all the pixels in a given image, the two centroids are re-calculated based on the average colour level within that particular cluster. Figure 3.15 illustrates the outcome of this process.

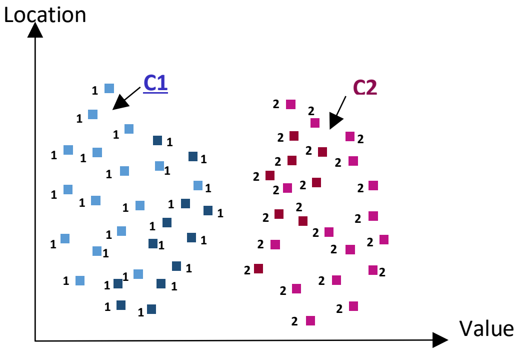


Figure 3.15: Clustered data set in K-means clustering

This Process is iterated until both centroids are converged to fixed points. The final output value for these two centroids are such that sum of distance between each pixel value of the input image and their corresponding centroids are minimum. Figure 3.16 illustrates the effect of this colour segmentation algorithm on different PCB types.

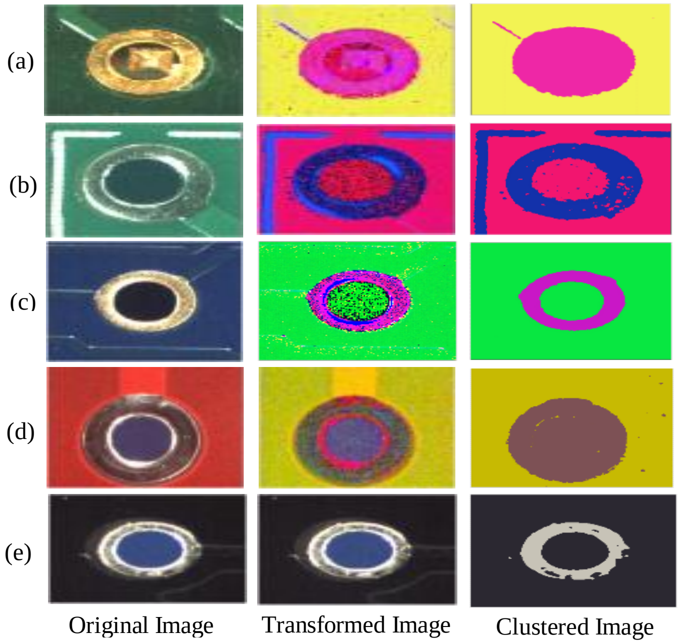


Figure 3.16: Colour clustered images using k-means colour clustering algorithm. (a) Output of k-means clustering on gold plated solder joint on green PCB. (b) Output of k-means clustering on tin plated solder joint on green PCB. (c) Output of k-means clustering on gold plated solder joint on blue PCB. (d) Output of k-means clustering on tin plated solder joint on red PCB. (e) Output of k-means clustering on tin plated solder joint on black PCB

Figure 3.16 shows that the colour clustered image produces an output which has a minimum effect due to uneven lighting, flux and other residuals on the PCB. Then the clustered image is thresholded against a predetermined threshold value acquired empirically as depicted in Eq. (36) in order to precisely identify the pad region.

$$P(x,y) = \begin{cases} 255, & if\ P(x,y) \geq TH \\ 0, & if\ P(x,y)\ < TH \end{cases} \quad (36)$$

TH: Predetermined threshold value which is a combination of different hue and value

Figure 3.17 illustrates the outcome of this algorithm over different PCB types.

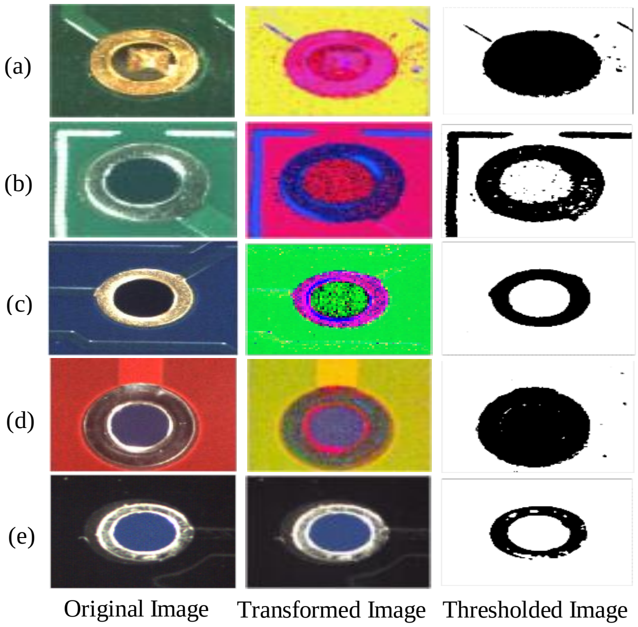


Figure 3.17: Thresholding of colour clustered images for solder pad segmentation. (a) Thresholded gold plated solder joint on green PCB. (b) Thresholded tin plated solder joint on green PCB. (c) Thresholded gold plated solder joint on blue PCB. (d) Thresholded tin plated solder joint on red PCB. (e) Thresholded tin plated solder joint on black PCB

Next, the binary image is processed with morphological filtering [25] to close open holes and reduce distributed noise inside the binary image. Then the contour finding algorithm, which is the method of identification of connected regions inside a binary image, is used to segment discontinuous areas on the binary image. It iterates through all the connected regions and define a bounding rectangle that contains the boundary pixels belongs to that region. Then the algorithm finds the largest and second largest regions from the detected contours, because the solder pad area should be contained by any of these two regions. Finally, the algorithm compares the shape and dimension of these regions with the respective information from the CAD file to find the contour that contains the solder pad. Figure 3.18 illustrates the detected solder pads over several images. Table 3.3 summarizes the performance analysis of this methodology for different PCB types at different conditions (Each criteria contains 25 different samples). Here, the successful detection rate means the capability of the algorithm to identify the pad area with an IoU greater than 0.85 with reference to manually define bounding box for the solder pad**.**

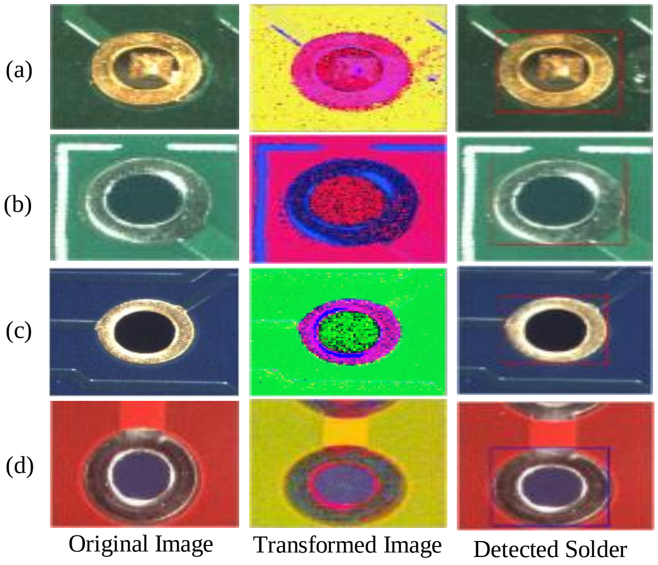


Figure 3.18: Segmented solder pads on different pad and PCB surface colours. (a) Detected gold-plated solder joint on green PCB. (b) Detected tin plated solder joint on green PCB. (c) Detected gold-plated solder joint on blue PCB. (d) Detected tin plated solder joint on red PCB. (e) Detected tin plated solder joint on black PCB

Table 3.3: Successful detection rate of colour image segmentation process on selected colour models over different PCB types at different controlled conditions

PCB Type	Controlled Situation	YIQ	HSV	YC_bC_r	RGB
Gold Plated Solder Pad on Green Colour PCB	Normal Condition (without any defect)	80%	78%	60%	72%
	Connected with PCB Track/ Highly Illuminated	100%	75%	0%	70%
	Flux Applied	88%	75%	92%	75%
Gold Plated Solder Pad on Blue Colour PCB	Normal Condition (without any defect)	100%	90%	72%	80%
	Connected with PCB Track/ Highly Illuminated	72%	65%	72%	62%
	Flux Applied	68%	60%	88%	55%
Tin Plated Solder Pad on Green Colour PCB	Normal Condition (without any defect)	16%	88%	100%	76%
	Connected with PCB Track/ Highly Illuminated	32%	100%	100%	20%
	Flux Applied	32%	80%	92%	72%
Tin Plated Solder Pad on Red Colour PCB	Normal Condition (without any defect)	32%	100%	68%	82%
	Connected with PCB Track/ Highly Illuminated	32%	96%	20%	50%
	Flux Applied	0%	80%	60%	68%
Tin Plated Solder Pad on Black Colour PCB	Normal Condition (without any defect)	16%	60%	96%	100%
	Connected with PCB Track/ Highly Illuminated	16%	56%	92%	100%
	Flux Applied	32%	40%	72%	80%

Manufacturing defects on the PCBs, like offsets between the drill hole pad and the plated pad area, lead to false detection of solder pads in YIQ colour model for green colour PCBs with gold-plated solder pads as illustrated in Figure 3.19.

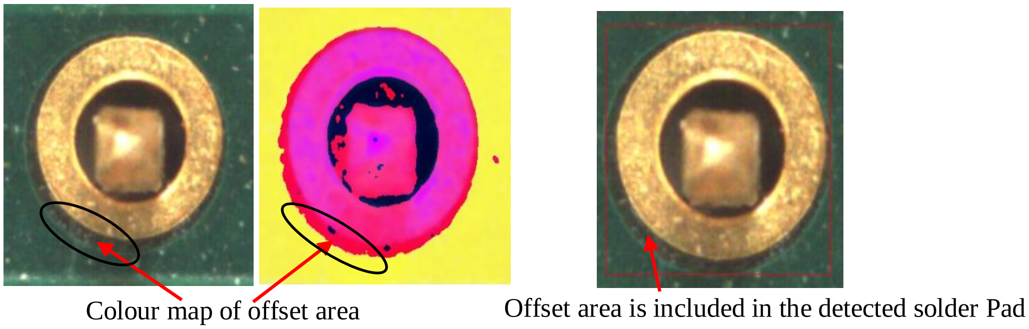

Colour map of offset area
Offset area is included in the detected solder Pad

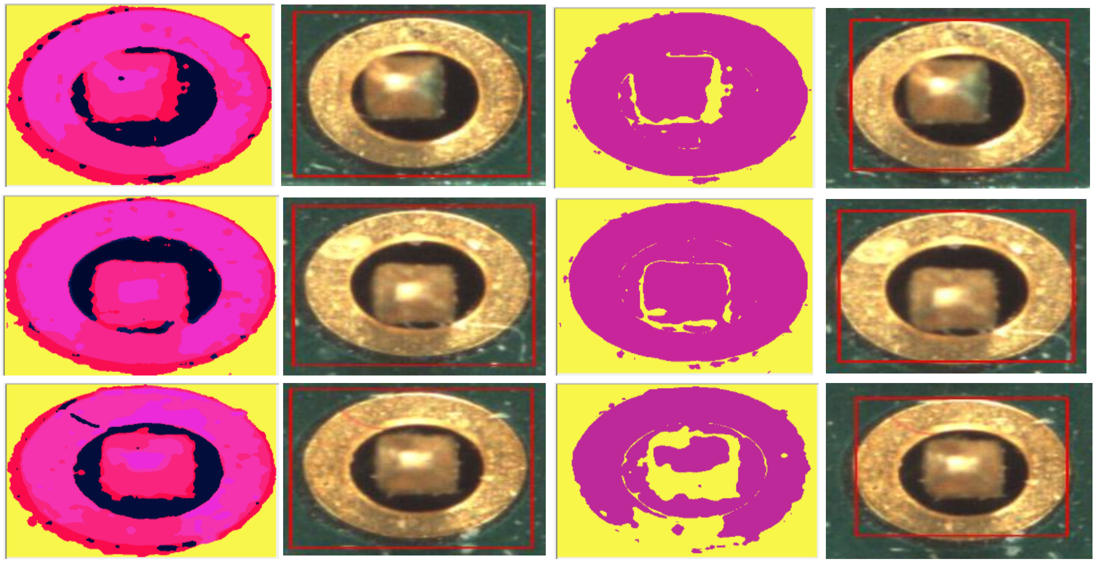
Figure 3.19: Effect of offset issue on green colour PCBs with gold plated solder pads

Figure 3.19 illustrates that the offset area has been mapped to a colour which has the minimum vector length difference to the colour mapped for gold plated solder pad resulting a much larger solder pad area. This offset area must be carefully eliminated from the detected solder pad to have a precise segmentation. This is accomplished by re-quantizing the identified region is into six different colours which include colours mapped for PCB background, solder pad, drill-hole, offset area, highly illuminated areas on PCB background. Then the vector distances to each colour vector was experimentally obtained over a sample of 100 gold-plated solder pads on green colour PCBs with reference to the origin of the colour model (0, 0, 0) as outlined in Table 3.4.

Table 3.4: Experimentally obtained colour vector distances for quantized colours on the detected object area

Region of interest	**Experimentally obtained vector distance (x) range for different areas on the detected solder pad**
Solder pad area	$285 \leq x \leq 330$
Offset area	$255 \leq x \leq 280$
Drill-Hole area	$0 \leq x \leq 100$
Background area	$350 \leq x \leq 370$

Then the algorithm replaces the colour of the pixels whose vector distances fall to the regions other than pad and drill-hole regions with the colour obtained for PCB background. Ultimately, image segmentation process is repeated to locate the actual solder pad area from the detected object as illustrated in Figure 3.20.



Object area of the YIQ colour transformed image

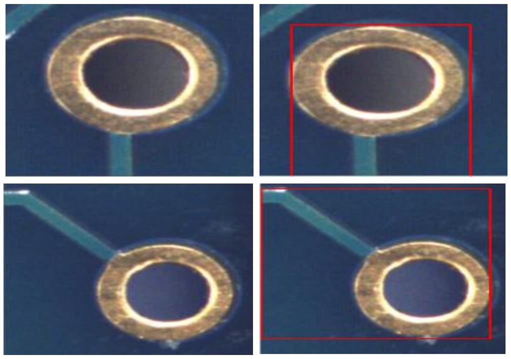
Falsely detected solder pad due to pad offset area

Colour quantized image after remapping offset area colour to background area colour

Detected solder pad after remapping of offset area to background area colour

Figure 3.20: Result of offset area removing algorithm

In addition to above effect, the performance of the YIQ colour model is considerably affected by gold plated solder pads with PCB tracks connected on blue colour PCBs as illustrated in Figure 3.21.



Original Image

Detected Solder Pad

Figure 3.21: False detection of solder pad areas due to light colour PCB tracks on gold-plated solder pads on blue colour PCBs

PCB tracks with light blue are mapped to a colour with minimum vector distance to the colour mapped for pad resulting a larger area for solder pad as illustrated in Figure 3.22.

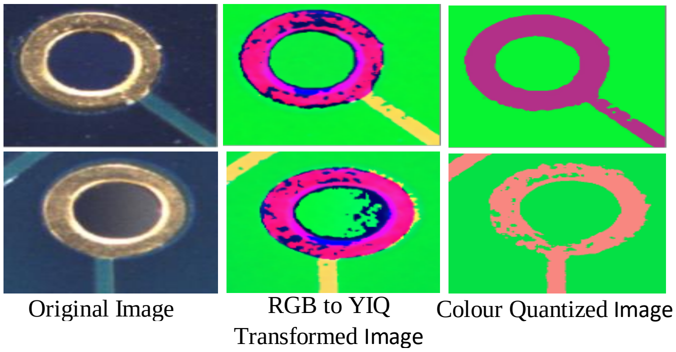


Figure 3.22: Mapped colours of light colour PCB tracks during colour transformation and colour quantization processes for gold plated solder pads on blue colour PCBs

The same procedure described earlier for eliminating pad offsets, is used to eliminate the effect of light colour PCB tracks (Vector distance range for light colour PCB tracks : 345 ≤ x ≤ 370). Figure 3.23 illustrates the outcome of this methodology over different solder pads that connect with light colour PCB tracks on blue colour PCBs.

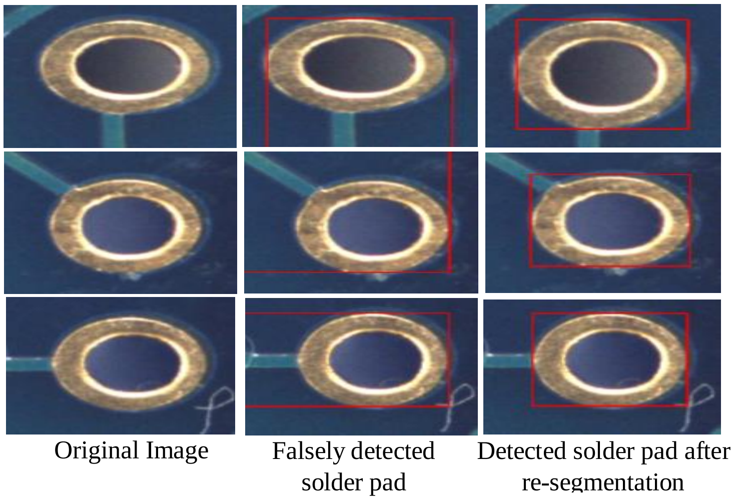

Original Image
Falsely detected solder pad
Detected solder pad after re-segmentation

Figure 3.23: Accurate detection of solder pads after removing the effect of PCB tracks on blue colour PCBs with gold-plated solder pads

Table 3.5 summarizes the success rate for YIQ colour model after using the developed methodology for removing the effect of pad offsets and light colour PCB tracks.

Table 3.5 depicts that the new algorithm successfully eliminates the effect of manufacturing defects and surface finish of blue and green colour PCBs with gold plated solder pads. Even though the modified version of $I_1I_2I_3$ colour space, was stated to provide better outcome for solder pad segmentation in [42], it could be shown that a single colour model is not capable of giving a more precise image segmentation for different PCB colours with different solder pads according to the results obtained in Sections 3.2-3.4.

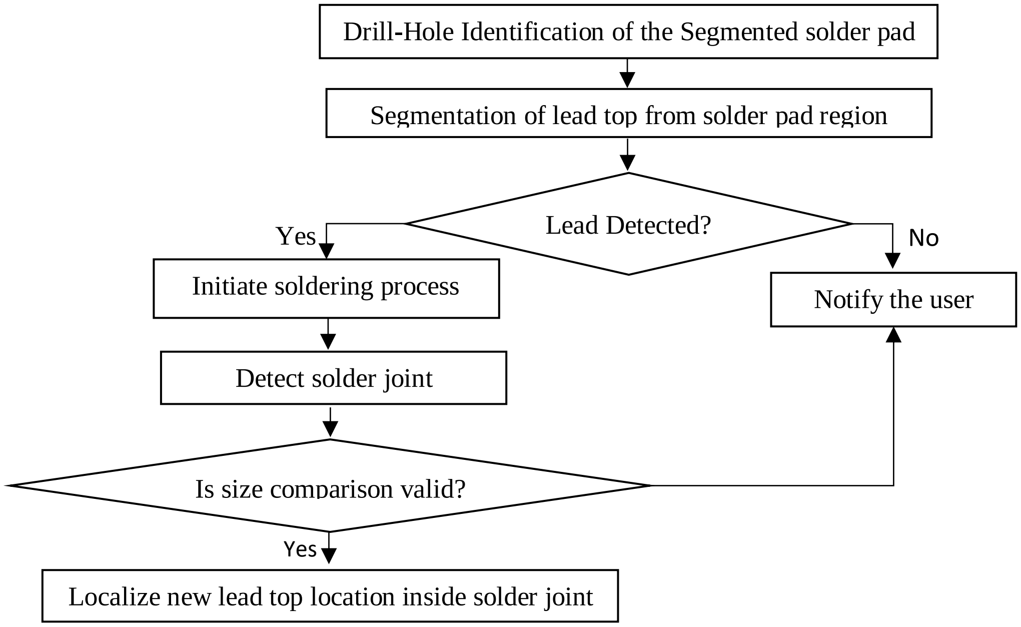
Table 3.5: Successful detection rate after removing the effect of pad offsets and tracks

PCB Type	**Controlled Situation**	**Detection rate of before removing the effect of pad offsets & PCB tracks**	**Detection rate after removing the effect of pad offsets & PCB tracks**
Gold Plated Solder Joint on Green Colour PCB	Normal Condition (without any defect)	90%	98%
	Connected with PCB Track / Highly Illuminated	100%	100%
	Flux Applied	88%	88%
Gold Plated Solder Joint on Blue Colour PCB	Normal Condition (without any defect)	100%	100%
	Connected with PCB Track / Highly Illuminated	72%	100%
	Flux Applied	68%	80%

Since there is no requirement of externally applied flux other than the flux available in solder wire, the negative impact of flux was not given a higher priority. Furthermore, the precise identification of solder pad from its background further enables to have real time calibration of positioning accuracy of the robotic system. In this research, it could be identified that the component lead plays a significant role when classifying the quality of the solder joint. Therefore, several methods were developed to precisely segment the component lead from the detected solder pad prior to soldering and finds its new location inside the solder joint once the soldering process is completed. Chapter 4 describes the implementation of this methodology in a detailed way.

4. Identification and Localization of THT Component Lead

The results acquired in first two stages of the AOI system, show that a significant success could be acquired. Even though, the first impression in someone's mind may be to start soldering once a pad is identified, it could be revealed that the component lead top must be prior identified and localized inside solder joint to provide robust quality inspection capability. This chapter describes how this task can be effectively accomplished. Figure 4.1 illustrates the proposed system architecture.



Drill-Hole Identification of the Segmented solder pad
Segmentation of lead top from solder pad region
Lead Detected?
Yes
No
Initiate soldering process
Notify the user
Detect solder joint
Is size comparison valid?
Yes
Localize new lead top location inside solder joint

Figure 4.1: The implemented methodology to detect and localize lead top of a component

Section 4.1, explains the importance of identifying and localizing the component lead inside a solder joint.

4.1 Importance of Identification and Localization of THT Component Lead Top inside a Solder Joint

The shape of a solder joint does not possess a consistent appearance. Therefore, it increase the complexity of the solder quality assurance process. Once a pad is soldered, it will be completely a new object to the vision system, except the component lead top that provides only relationship to the joint's previous state. Further, the location of the component lead top governs the shape of the solder joint and illumination distribution across it. According to the observations carried out on 500 different solder joints, it could be observed that the detection and localization of THT component lead top is important in three main aspects.

- Verification of component availability can be performed by verifying the existence of component lead prior to soldering.
- Higher probability of detecting lead tops which possess a darker colour comparing to solder paste, as a not soldered area at solder quality classification stage.
- The location of lead top has a direct impact on the shape of the solder joint and the illumination distribution across it. Since the AOI system analyse the light distribution across the solder joint surface for the determination of amount of applied solder, the knowledge of its exact location is vital.

Figure 4.2 provides examples for above discussed situations. In Figure 4.2 (b), it can be seen that there is a high possibility to identify the lead top as a soldering defect because of its much darker colour comparing to its background. Therefore, it is clear that lead top must be prior identified and localized in order to enhance the accuracy and robustness of the vision system.

The human vision system identifies the shape of an object, based on the distribution of reflected light over its surface. The shape of the object adds a significant impact over the light intensity distribution, when it is being viewed in orthogonal direction. In this application, the direction of the camera is perpendicular and concentric to the solder joint being inspected. Therefore, the distribution of light captured from the camera, depends on the shape of the solder joint. Figure 4.2 (c) provides an example of a solder joint with excess solder.

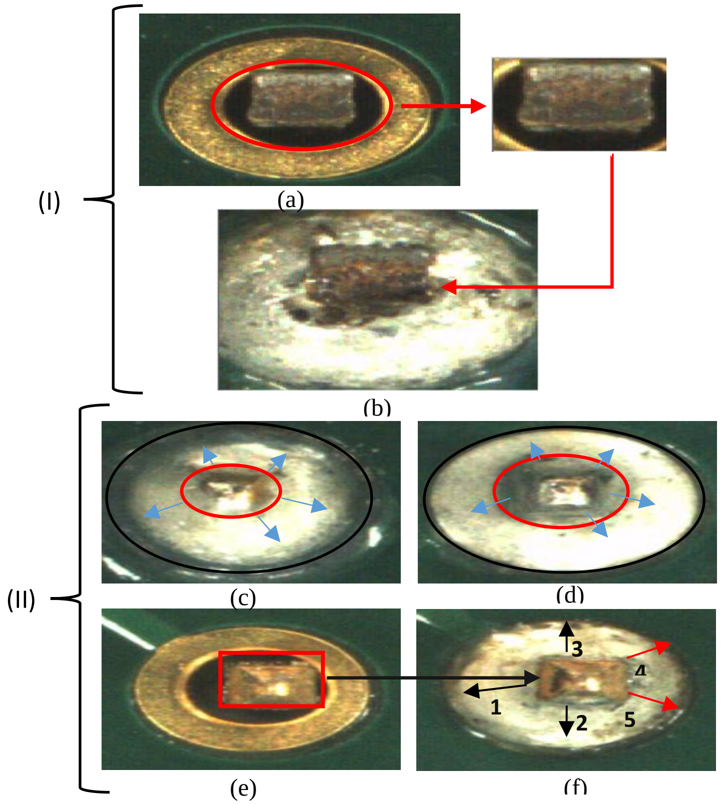


Figure 4.2: Importance of localizing the component lead top inside the solder joint. (I). Component with darker lead. (II). The impact of component lead on the light distribution across the solder joint. (a). Bare solder pad and drill-hole region. (b). Existence of darker area inside the solder joint. (c). Solder joint with excess solder. (d). Good solder joint. (e). Solder pad with colourful component lead. (f). Distribution of light across the solder joint

In Figure 4.2 (II) (c) and (f), it can be seen that the light intensity decreases from lead top to PCB surface in an over soldered joint while it is in opposite direction for a good solder joint. In Figure 4.2 (II) (f), it can be seen that the distribution of light intensity from top to bottom is consistent (bright) along in the directions denoted by “1”, “2” and “3”, whereas the intensity decreases along in the directions denoted by “4” and “5”. Therefore, this AOI system localizes it precisely to provide a robust solder quality classification process. However, the vison systems described in [12] and [13] are not capable of taking the advantage of this parameter since they inspect the solder quality in different platforms rather than the platform that soldered them. Section 4.2 provides a detailed explanation on the implementation of the methods to identify component lead.

4.2 Segmentation of THT Component Lead Top from Solder Pad Region

In Section 4.1, a detailed explanation has been given on the importance of identifying and localizing the component lead. Its segmentation from bare solder pad can be accomplished either of following three ways prior to the soldering process.

1. Process the entire solder pad area and identify the lead top out of it.
2. Extracts the drill-hole region based on CAD data and segments the lead top.
3. Automatically identifies the drill-hole and then segments the lead top.

The first method is not robust enough at situations where the colour dissimilarity between the component lead and the solder pad is minimum. This impact would be detrimental, when the component lead touches the solder pad as illustrated in Figure 4.3. No algorithm will not be capable enough to identify the lead from its background without introducing heavy computational complexity to the segmentation process.

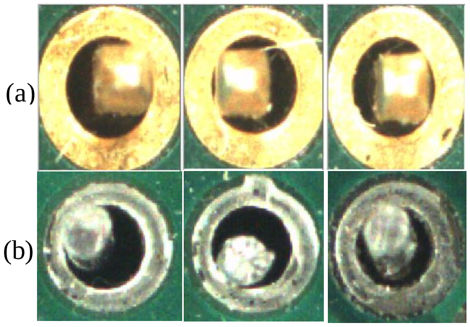


Figure 4.3: Impact of colour similarity between solder pad and component lead on segmentation accuracy. (a). For gold plated solder pads . (b). For tin plated solder pads

Therefore, the priority was given to 2nd and 3rd methods. Eventually the 3rd was chosen because of the below disadvantages of the 2nd method.

- The resolution computed during camera calibration procedure can contain μm level error and it is proportional to the size of the solder pad. Therefore, that error can either introduce some pad area as the drill-hole or results in loss of some portion of drill-hole.

- Slight offsets occurred due to imperfections of PCB drilling process usually result in mismatched dimensions to that of CAD information as illustrated in Figure 4.4 (Majority of the cases actual drill-hole is quite large than the actual size).

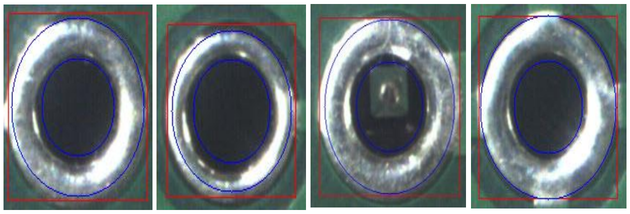


Figure 4.4: The offset between the computed size (based on CAD data) and the actual size of the drill-hole. Small and large blue colour circles denote computed drill-hole and pad dimensions based on CAD data. Red colour bounding box denotes the segmented solder pad

Section 4.2.1 describes how the drill-hole region is automatically identified.

4.2.1 Automatic Computation of Drill-hole Region

The following algorithm is developed to segment the area contained by the drill-hole region as depicted in steps 1-11 (refer Figure 4.5 for the structure of the algorithm).

Step 1: Segmented solder pad is transformed into HSV colour space. (see Figure 4.6)

Step 2: Colour transformed image is clustered into six colour classes. (see Figure 4.7 (b)).

Step 3: Colour clustered image is thresholded using a predetermined threshold value. (see Figure 4.7 (c)).

Step 4: Find the largest contour, C_L and compare its size and distance from the image centre against predetermined threshold values $A_{TH_{Min}}$ and D_{TH}. If the result is false, user is alarmed.

Step 5: The second largest contour area, C_M, is found by the algorithm, if the resulted area of C_L is below a predetermined threshold, A_{TH_Max}. Unless the algorithm jumps to Step 6.

Step 6: Set all the pixels to zero, that are not contained by the selected contours. Then generate the binary image I_S that contain the required information to approximate drill-hole region. (see Figure 4.7 (d)).

Step 7: External boundary points of I_S, in all four directions, top-centre, left-centre, bottom-centre and right-centre are computed according to Eq. (37)-(40).

$$P_{LC}(X) = (p(x-1,y) == 0 \;\&\&\; p(x,y) == 255\;)?\, P_{LC} \cup p(x,y) : NOP \;\forall (x,y) \in \big((0, W/2), (0, H)\big) \tag{37}$$

$$P_{BC}(X) = (p(x,y+1) == 0 \;\&\&\; p(x,y) == 255\;)?\, P_{BC} \cup p(x,y) : NOP \;\forall (x,y) \in \big((0, W), (H, H/2)\big) \tag{38}$$

$$P_{RC}(X) = (p(x+1,y) == 0 \;\&\&\; p(x,y) == 255\;)?\, P_{RC} \cup p(x,y) : NOP \;\forall (x,y) \in \big((W/2, W), (0, H)\big) \tag{39}$$

$$P_{UC}(X) = (p(x,y-1) == 0 \;\&\&\; p(x,y) == 255\;)?\, P_{UC} \cup p(x,y) : NOP \;\forall (x,y) \in \big((0, W), (0, H/2)\big) \tag{40}$$

Where,

NOP: No operation

$P_{LC}(X)$, $P_{BC}(X)$, $P_{RC}(X)$ and $P_{UC}(X)$ are the list of boundary points in left-centre, bottom-centre, right-centre and top-centre directions respectively.

W and H are the width and height of I_S respectively.

Step 8: The average of $P_{LC}(X)$, $P_{BC}(X)$, $P_{RC}(X)$ and $P_{UC}(X)$ is computed and find the lists whose sizes are larger than the average and combine them to $P(X)$. (see Figure 4.8 (c)).

Step 9: Find the point $P_1(X)$ in $P(X)$ based on the list that contains the highest average (Ex: If $P_{LC}(X)$ holds the highest average, then finds the point with minimum *x* coordinate). Then finds the point $P_2(X)$ in $P(X)$ that holds the maximum distance to $P_1(X)$.

Step 10: Find the point, $P_P(X)$, which have maximum perpendicular distance to the line formed by $P_1(X)$ and $P_2(X)$ according to Eq. (41). (see Figure 4.8 (d)).

$$P_P(X) = (d_x == D_{Max})?\; p(x,y) : NOP \;\forall p(x,y) \in P(X) \tag{41}$$

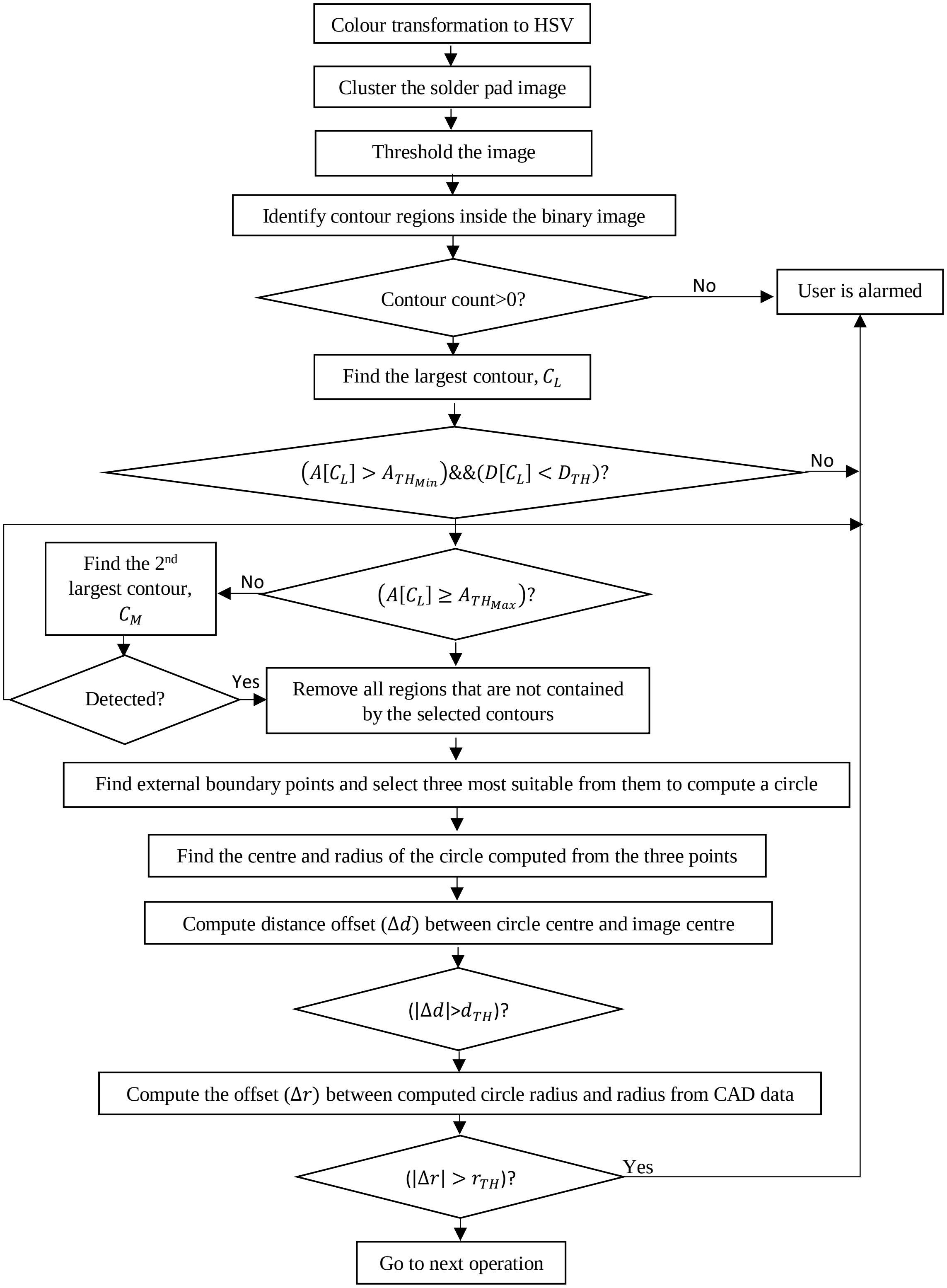

Colour transformation to HSV
Cluster the solder pad image
Threshold the image
Identify contour regions inside the binary image
Contour count>0?
No
User is alarmed
Find the largest contour, C_L
$(A[C_L] > A_{TH_{Min}})\&\&(D[C_L] < D_{TH})$?
No
Find the 2nd largest contour, C_M
No
$(A[C_L] \geq A_{TH_{Max}})$?
Detected?
Yes
Remove all regions that are not contained by the selected contours
Find external boundary points and select three most suitable from them to compute a circle
Find the centre and radius of the circle computed from the three points
Compute distance offset (Δd) between circle centre and image centre
($|\Delta d|$>d_{TH})?
Compute the offset (Δr) between computed circle radius and radius from CAD data
($|\Delta r| > r_{TH}$)?
Yes
Go to next operation

Figure 4.5: Structure of the drill-hole segmentation algorithm

Where,

d_x: perpendicular distance from point (m, n) to the line $Ax + By + C = 0$, denoted by: $d_x = \frac{|Am+Bn+C|}{\sqrt{A^2+B^2}}$ and D_{Max} is the maximum computed distance.

Step 11: Compute the circle that goes through $P_1(X)$, $P_2(X)$ and $P_P(X)$ according to Eq. (42), since the drill-hole is a circular shape. (see Figure 4.8 (e)).

$$\begin{bmatrix} C_x \\ C_y \\ R \end{bmatrix} = \begin{bmatrix} x_{P_{x\_Min}(X)} & y_{P_{x\_Min}(X)} & 1 \\ x_{P_{y\_Max}(X)} & y_{P_{y\_Max}(X)} & 1 \\ x_{P_P(X)} & y_{P_P(X)} & 1 \end{bmatrix}^{-1} \begin{bmatrix} -\left(x^2_{P_{x\_Min}(X)} + y^2_{P_{x\_Min}(X)}\right) \\ -\left(x^2_{P_{y\_Max}(X)} + y^2_{P_{y\_Max}(X)}\right) \\ -\left(x^2_{P_P(X)} + y^2_{P_P(X)}\right) \end{bmatrix} \tag{42}$$

Where,

C_x, C_y, R : *x* and *y* coordinates of the centre and radius of the circle that goes through points $P_{x\_Min}(X)$, $P_{y\_Max}(X)$ and $P_P(X)$ respectively

In Step 1, the segmented solder pad is colour transformed. Even though, a significant colour dissimilarity can be observed between the drill-hole and its immediate background, RGB colour model is not a suitable input for a colour clustering algorithm, because of high correlation among R, G and B components. Hence, an experimental basis analysis was carried out based on the procedure described in Section 3.2 over a sample of 200 solder pads to find the best colour model. It could be observed that the HSV colour model is a suitable candidate. Figure 4.6 illustrates the outcome of several colour models that rendered a distinctive outcome over a bare solder pad.

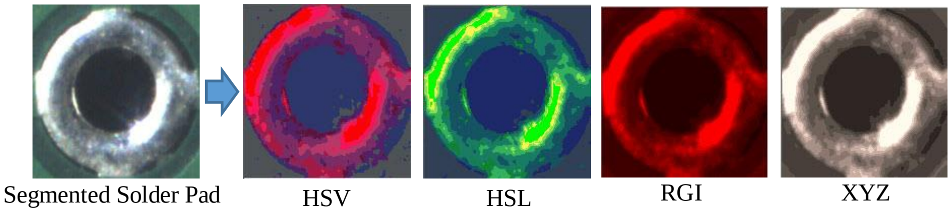


Figure 4.6: Results from colour transformation using several colour models over a segmented bare solder pad

In Step 2, the colour transformed image is segmented using a colour clustering algorithm. Since, the drill-hole has a significant colour dissimilarity to its immediate background, k-means clustering algorithm is used. However, there is a possibility for the existence of oxidized regions on the pad area that lead to a colour similarity with the drill-hole (refer Figure 4.7(a)). Therefore, Gaussian smoothing function is used to further minimize the effect of such regions. In order to further increase the stability, an analysis was carried out over a sample of 200 solder pads, to determine the number of colour clusters required to map such regions into different clusters to that of drill-hole. It could be observed that clustering into six clusters could diminish the effect of such areas.

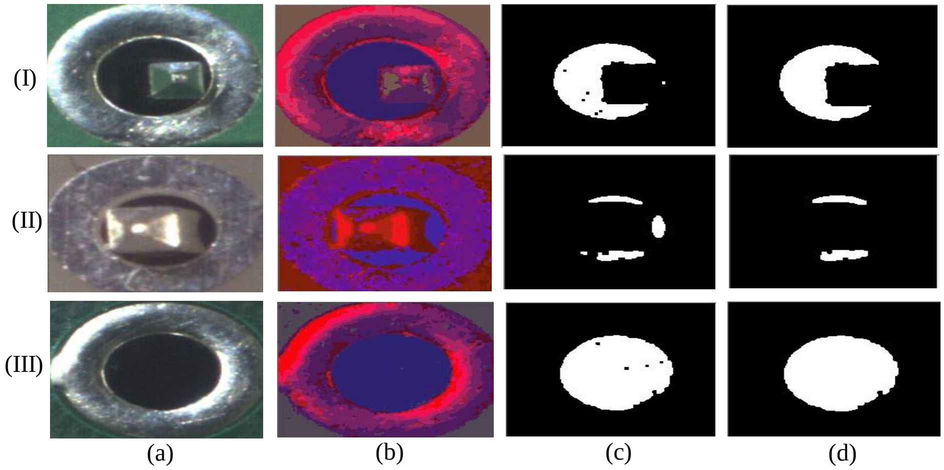


Figure 4.7: Generation of a binary image to compute area covered by drill-hole. (a). Input image. (b). Colour clustered image. (c). Thresholded image. (d). Generated binary image for drill-hole boundary definition

According to Figure 4.7 (b), it can be seen that the colours of oxidized areas have been mapped to clusters with different hues than that of drill-hole.

In Steps 3-4, the clustered image is thresholded based on a predetermined threshold value and finds the largest contour, $C_L$based on following outlines.

- Largest among the detected contours (C_L) must be contained by the drill-hole.
- The existence of the largest contour must be closer to the centre of the image (since the centre of the drill-hole coincide with the image centre with minimum offset).

The size of C_L and distance offset ($D[C_L]$) are normalized to make them independent from sizes of solder pad and drill-hole. Furthermore, this comparison minimizes the likelihood of selecting an area on the solder pad as C_L at situations where a substantial area of the drill-hole is covered by the component lead and the existence of darker regions on the solder pad. If no contour can be found that meet above conditions, the user is notified.

In Step 5, the second largest contour area is found, if the area of C_L is below a predetermined threshold, A_{TH_Max}. This is performed based on following outlines.

- It must be larger than a predetermined threshold
- The distance difference between the centres of C_L and image centre, must be lower than a predetermined threshold value

A contour that satisfies both of above conditions is selected as the second largest contour, C_M. If no contour can be found that meets above conditions, the user is notified.

In Step 6, The binary image (I_S) is computed by setting the value of pixels that are not contained by these contours to zero (refer Figure 4.7 (d)). I_S now holds the useful regions to approximate the area contained by the drill-hole.

In Step 7, the outer boundary of the contour regions in I_S is computed in four directions top-centre, left-centre, bottom-centre and right-centre. This is because, the outer boundary pixels represent the boundary between the drill-hole and solder pad.

In Step 8, the average of the boundary points acquired in Step 7 is computed to determine the most dominant areas where the contours lie. This is because, if there are open regions in I_S as in Figure 4.7 (II), there is a likelihood that the boundary points being detected in inner boundaries of combined contour regions. That will lead to a false computation in Step 11.

In Steps 9-10, three points, $P_1(X)$, $P_2(X)$ and $P_P(X)$, are computed to find the circle that goes through them. First two points, $P_1(X)$ and $P_2(X)$, are computed based on the nature of distribution of contour region in order to provide a significant distance difference between them to increase the accuracy of detecting $P_P(X)$. Finally, in Step 11, the centre point and the circle radius is computed that goes through these points (see Figure 4.8(e)).

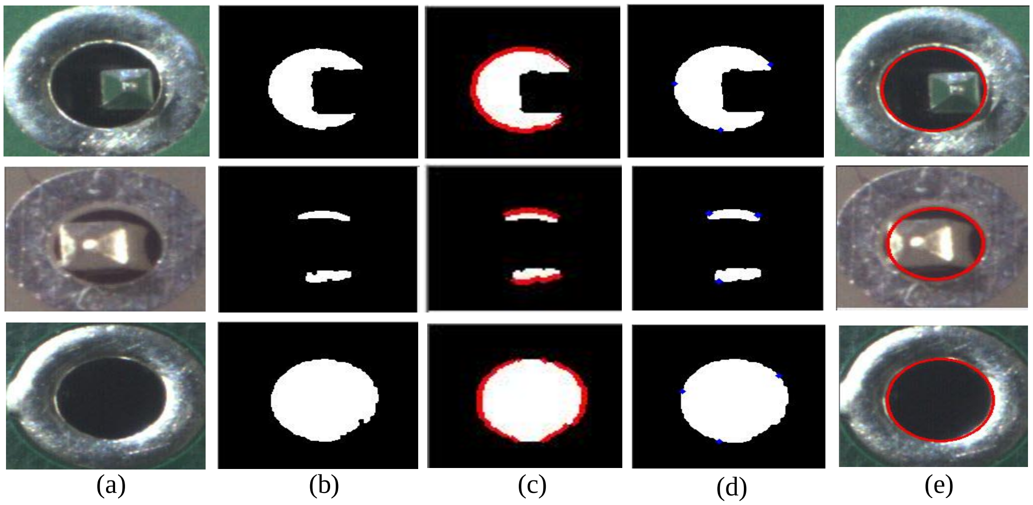


Figure 4.8: Automatic computation of the drill-hole. (a). Input image. (b). Computed binary image for drill-hole approximation. (c). Computed boundary points (red colour). (d). Selected three boundary points to compute drill-hole (blue colour). (e). Computed drill-hole

The action represented by "User is alarmed" in Figure 4.5, occurs when the vision system cannot acquire enough information from the segmented solder pad to generate the circular region containing the drill-hole. This usually happens at situations when:

- the component lead tightly fits inside the drill-hole, minimizing hole visibility;
- the reflection of light inside the drill-hole alters its colour significantly;
- the distance offset ($|\Delta d|$) between the centres of the image and the computed circle is beyond the defined threshold value, d_{TH};
- the offset ($|\Delta r|$) between the radius of computed circle and radius of drill-hole acquired from CAD file is beyond the defined threshold, r_{TH}.

In such a case, the vision system maps the dimension from the CAD file on the segmented solder pad and notify the user asking the permission to proceed. The performance of this algorithm was evaluated under different conditions as tabulated in Table 4.1. Here the successful detection rate means the identification of drill-hole region with less than 3% error in both dimensions (height and width) comparing to visible size of the drill hole measured in pixel count.

Table 4.1: Successful detection of drill-hole region of a solder pad

Lighting Condition	Segmentation of drill-hole using CAD data		Segmentation of drill-hole using proposed algorithm	
	Successful Detection Rate	False Detection Rate	Successful Detection Rate	False Detection Rate
Highly Illuminated	88%	12%	97%	3%
Normally Illuminated	88%	12%	100%	0%
Poorly Illuminated	88%	12%	92%	8%

Finally, the AOI segments finds the existence of the component lead inside this region.

4.2.2 Segmentation of Component Lead Top

Since the colour of the component lead accommodates a significant number of different colours comparing to the colour of drill-hole, a colour image segmentation algorithm [47, 57-61] or colour clustering algorithm would be a good choice to segment it from its background. After having some studying on graph-cut based colour image segmentation algorithm [58], it could be concluded that colour clustering based algorithms would be a good choice at this stage of this application due to following reasons.

- The lead is also segmented into different regions since a graph-cut based image segmentation algorithm segments a given image into several homogenous regions (see Figure 4.9).

- Computation time is comparatively higher than the colour clustering algorithms. This will be much higher, when it is required to deploy another algorithm to verify the first point.

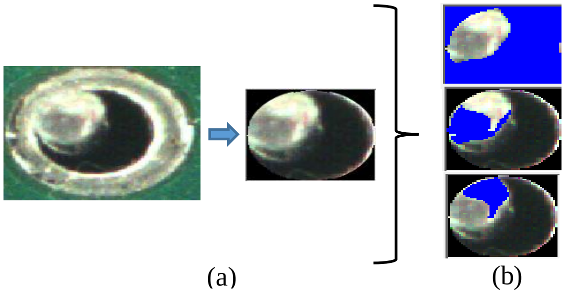


Figure 4.9: Results from a graph-cut based image segmentation algorithm. (a). Segmented solder pad and drill-hole region. (c). Segmented regions after the graph-cut based image segmentation

Even though, the distinct difference of colour between drill-hole and solder pad in Section 4.2.1 reinforces the application of K-Means for image segmentation, the situation is not the same when the drill-hole area is considered alone. Some regions in the component lead top contains similar colour to that of its surrounding drill-hole. Therefore, the probability of mapping such regions into the same colour class of drill-hole, is higher. Hence, the performance of three colour clustering algorithms, K-means, Fuzzy C-means (FCM) [25] and mean-shift [25] was analysed in terms of:

- the Euclidean distance of clustered colours inside lead top and drill-hole
- computation time
- repeatability of mapping colours into same colour clusters over consecutive trials
- successful detection rate of the component lead

The successful detection rate of an algorithm is computed based on the percentage of the detected area compared to the actual size of the component lead top. The actual size of the component lead top is calculated manually selecting the area contained by the lead top. The number of required colour clusters can be prior defined in both k-means and FCM colour clustering algorithms (five colour clusters have defined in this application), whereas mean-shift algorithm works with quantizing available colours into minimum possible number of clusters. Tables 4.2 presents the acquired results of these algorithms over a sample of 200 solder pad images.

Table 4.2: Average euclidean distance of clustered colours inside lead top and drill-hole region for gold-plated component leads with reference to the origin of RGB colour cube

Parameter	**Colour Clustering Algorithm**					
	K-means/units		**FCM/units**		**Mean-shift**	
	Lead Top	Drill-hole	Lead Top	Drill-hole	Lead Top	Drill-hole
Average Euclidean distance	248.3	60	196.5	27.3	NA	NA
Computation time / ms	69.68		562.36		985	
Repeatability / %	98%		98%		92%	
Successful Detection Rate / %	98%		96%		NA	

The results obtained in Table 4.2 reveals that K-Means clustering algorithm outperforms other two algorithms in all criterias. Figure 4.10 illustrates few colour clustered images using the two main candidates.

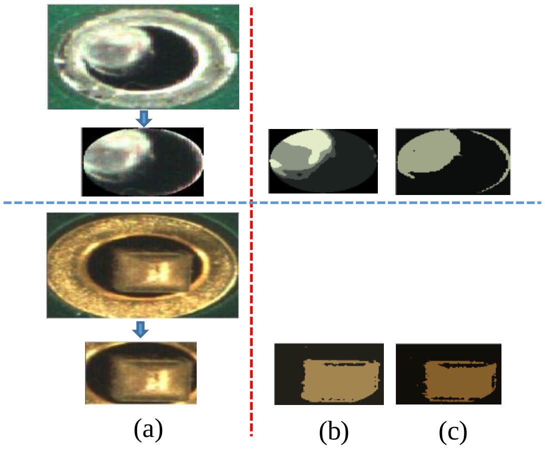


Figure 4.10: Resulted images from colour clustering algorithms for lead detection. (a). Detected solder pad image and filtered out drill-hole region. (b). Resulted image from k-means clustering. (c). Resulted image from FCM

Next, the lead top must be extracted from its background. This task is accomplished by thresholding the resulted image from k-means colour clustering against a predetermined threshold value calculated based on the results acquired from a sample of 200 images. Figure 4.11 illustrates the segmented component lead top of several solder pads. However, Figure 4.11 (V) shows that the reflected light resulted from highly illuminated regions inside the solder pad lead to false detections in component availability verification (refer Figure 4.11 (V)(e)). These false detections must be eliminated effectively. Following algorithm is implemented to eliminate such effects.

Step 1: The size of the largest contour (C_{L-Lead}) is compared against a predetermined threshold value (depends on the size of the drill-hole). If the result is true then Image ROI is set to C_{L-Lead} and jump to Step 2, otherwise the user is notified.

Step 2: The external boundary points of the contour is computed according to Eq. (43).

$$B(X) = B(X) \cup P(X) \tag{43}$$

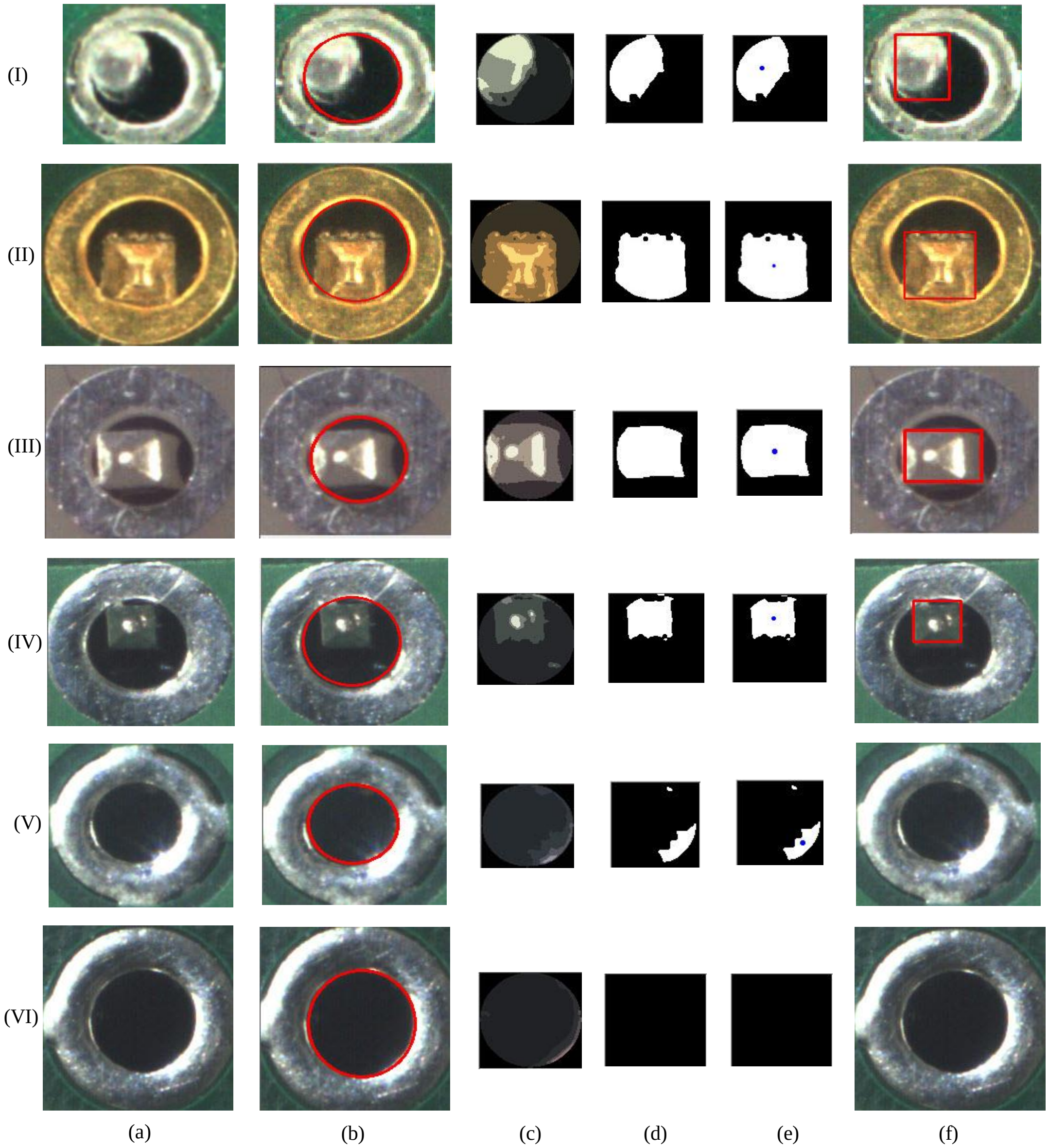


Figure 4.11: Detected component lead top inside the drill-hole region of the solder pad. (a). Extracted solder pad region. (b). Identified drill-hole region. (c). Resulted image after colour clustering. (d). Resulted image after thresholding and morphological filtering. (e). Computed centre point of the largest contour. (f). Solder pad image with detected component lead top. (I). Tin plated lead with ellipsoidal shape. (II). Gold plated lead. (III). Lead with highly illuminated areas. (IV). Lead with darker colour. (V). Empty solder pad with reflected light inside drill-hole. (VI). Empty solder pad with no reflected light

Where,

$$P(X) = \big((p => p.x == Min[C(X_i).X]) \; || \; (p => p.x = \\ = Max[C(X_i).X])\big)? \; C(X_i): \{\} \; \forall X_i(x,y) \in C(X)$$

$$C(X) = ((p(x-1,y) == 0 \; || \; p(x+1,y) == 0) \; \&\& \; p(x,y) = \\ = 255))? \, C(X) \cup p(x,y): NoP \; \forall (x,y) \in \big((0,W),(0,H)\big)$$

Here W and H are the width and height of the ROI set image respectively.

Step 3: The centre point, $C_{L-Lead}(X)$, of C_{L-Lead} is computed according to Eq. (44). (See Figure 4.11 (e).)

$$C_{L-Lead}(X) = \left(\frac{\mu_{1,0}}{\mu_{0,0}}, \frac{\mu_{0,1}}{\mu_{0,0}}\right) \tag{44}$$

Where,

$$\mu_{0,0} = \sum_{x=0}^{W}\sum_{y=0}^{H}(p(x,y) == 255)? \, 1: 0$$

$$\mu_{1,0} = \frac{\sum_{x=0}^{W}\sum_{y=0}^{H}(p(x,y) == 255)? \, x: 0}{\mu_{0,0}}$$

$$\mu_{0,1} = \frac{\sum_{x=0}^{W}\sum_{y=0}^{H}(p(x,y) == 255)? \, y: 0}{\mu_{0,0}}$$

Step 4: The distances between $C_{L-Lead}(X)$ and each point in $B(X)$ is calculated and the Standard Deviation (*SD*) of the distribution is computed.

Step 5: The computed *SD* is normalized with respect to R and compared against a predetermined threshold. If it is higher, then C_{L-Lead} is discarded and user is notified.

In Step 1, the image ROWis set to C_{L-Lead}. This is done in order to avoid the impact of other smaller contour regions in the original image.

In Step 2, each boundary point of C_{L-Lead} is computed to provide a reasonable approximation to its shape.

In Step 3, the centre point, $C_{L-Lead}(X)$, is calculated based on the moments of C_{L-Lead}. Because $C_{L-Lead}(X)$ is geometrically important, when predicting the contour distribution.

In Step 4, the distances from each boundary point to $C_{L-Lead}(X)$ is calculated to approximate the distribution of the contour. If C_{L-Lead} is contained by a region other than the lead top, then it usually possesses a narrow shape with a wider distribution. This results in a higher *SD,* since the variance of distance differences between C_{L-Lead} and boundary is higher. If C_{L-Lead} is contained by the component lead, then the variance of the distance differences are minimum, resulting a lower *SD*.

In Step 5, the computed *SD* is normalized with respect to *R* since the sizes of the component lead and the drill-hole are approximately proportional.

The implementation of this algorithm could significantly reduce the impact of the light reflections inside the drill-hole (Refer Figure 4.11 (V)-(VI)). If no lead is detected, the user is notified robotic and no soldering is performed. The automatic assurance of the component availability increases the reliability and the robustness of this vision system. In Section 4.3, the suitability of several methods to localize the lead top inside the solder joint once the soldering is performed by the robotic system, was evaluated.

4.3 Localization of THT Component Lead inside Solder Joint

The requirement of precisely localizing the component lead top inside the solder joint is important because of slight position drifts of lead occurred due to force applied from solder iron. In Chapter 2, the suitability of SIFT, SURF and FAST algorithms was analysed to localize the lead top inside the solder joint. Figure 4.12 illustrates the outcome of these algorithms on a component lead that have distinctive colour compared to the colour of solder at different threshold levels.

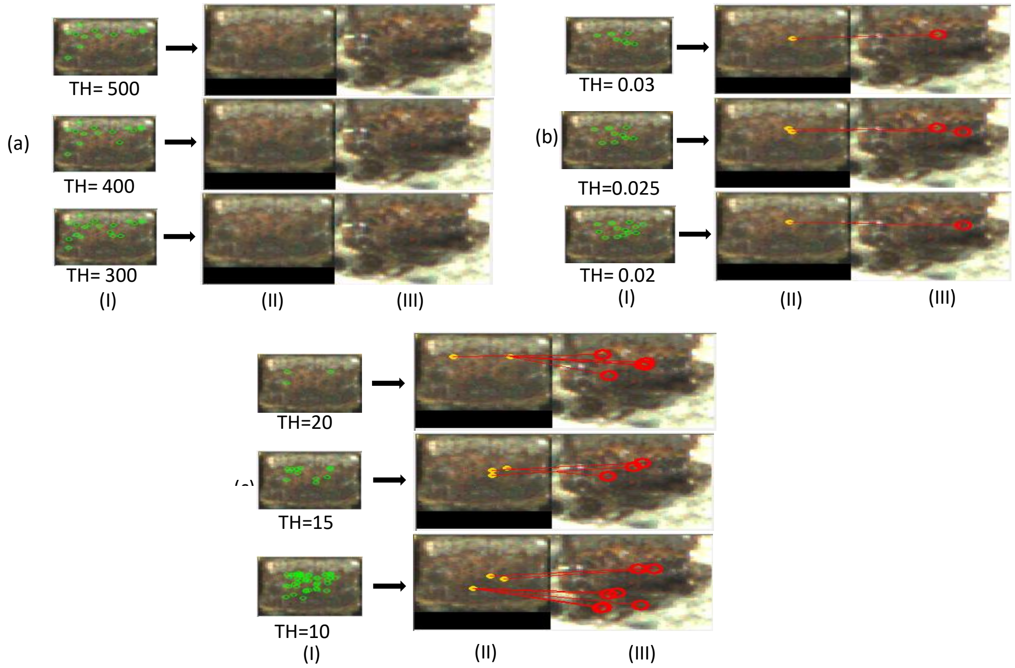


Figure 4.12: Application of feature extraction algorithms for lead top localization. (I). Feature detected lead top image. (II) Feature matched lead top image. (III). ROI set solder joint. (a). Application of SURF feature detector. (b). Application of SIFT feature detector. (c). Application of FAST feature detector

According to Figure 4.12, it can be seen that no feature detector is neither capable of identifying significant number of key-points nor performing consistent matching. Moreover, the change of appearance in lead top (common for THT component lead) after the soldering can negatively impact the performance of feature detectors. Since the availability of the component is known, the suitability of template matching algorithms (See Chapter 2) were verified as illustrated in Table 4.3, over a sample of 200 solder joints.

Table 4.3: Successful and erroneous detection rate of lead top inside the solder joint using template matching algorithms with respective average computation time

Algorithm Name	**Success Rate**	**Erroneous rate**	**Computation time/*ms***
TM_SQDIFF	95%	5%	31.9
TM_SQDIFF_NORMED	90%	10%	31.9
TM_CCORR	85%	15%	30.5
TM_CCORR_NORMED	75%	25%	30.5
TM_CCOEFF	70%	30%	29.7
TM_CCOEFF_NORMED	75%	25%	31

Here, the successful rate means the localization of lead top with an IoU greater than 0.8 with respect to manually define lead top area. Results in Table 4.3, illustrates that TM_SQDIFF outperforms other peer algorithms. Figure 4.13 illustrates the results obtained using SQDIFF algorithm at several situations.

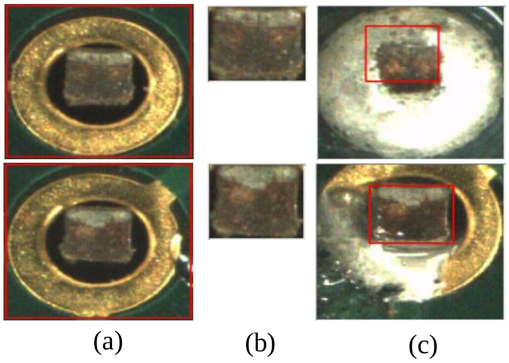


Figure 4.13: Localized THT component lead top with higher colour dissimilarity to solder paste on several solder joints using SQDIFF template matching algorithm. (a). Detected solder pad area. (b). Detected lead top using k-means clustering. (c). Localized lead top inside solder joint using SQDIFF template matching algorithm

Even though the outcome of TM_SQDIFF is accurate, its performance varies when the colour dissimilarity between the lead top and solder paste is minimum. Figure 4.14 illustrates the poor outcome of SQDIFF algorithm over several solder joints with tin plated leads where the colour dissimilarity is minimum. Since the processing time is a critical parameter, the application of a complex algorithm is not a viable option to solve this problem. Therefore, it is required to have a simple method that is capable of rendering the expected accuracy level with less computation time.

Inspired from the work carried out in Chapter 3, the performance of colour models was evaluated over a sample of 200 solder joints together with the above described template matching algorithms as illustrated in Table 4.4. According to those results, it can be seen that HSL and HSV colour models were able to provide a higher accuracy level together with SQDIFF template matching algorithm comparing to other combinations of colour models and template matching algorithms.

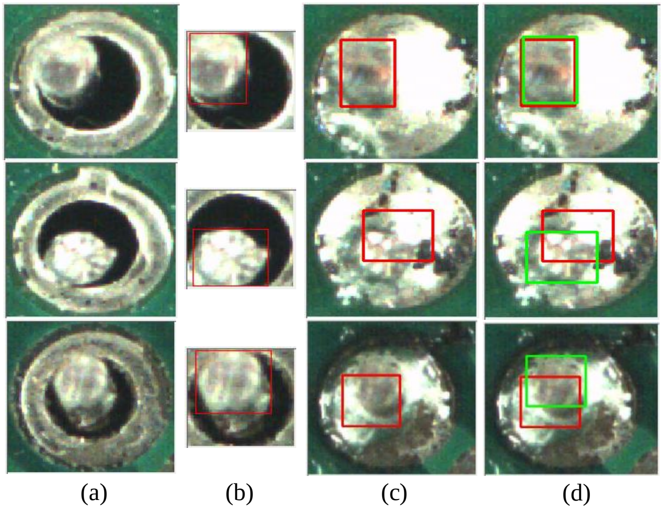


Figure 4.14: Localized THT component lead top with lower colour dissimilarity to solder paste on several solder joints using SQDIFF template matching algorithm. (a). Solder joints with tin plated component leads. (b). Detected component lead. (c). Localized lead top using SQDIFF algorithm defined by red colour rectangular boundary. (d). Deviation between localized (red colour) and the actual location (green colour) of lead tops

Table 4.4: Successful detection rate of lead top inside the solder joint using template matching algorithms over different colour models

Colour Model	**SQDIFF**	**SQDIFF_ NORMED**	**CCORR**	**CCORR_ NORMED**	**CCOEFF**	**CCOEFF_ NORMED**
HSV	98%	90%	70%	50%	50%	45%
HSL	99%	90%	60%	55%	50%	50%
YIQ	70%	65%	51%	54%	58%	51%
RGI	85%	75%	50%	55%	55%	55%
CMYK	80%	85%	97%	94%	90%	85%
Lab	90%	85%	90%	80%	80%	80%
$I_1I_2I_3$	85%	75%	80%	80%	70%	75%
$I_1I_2I_3$ Mod.	62%	58%	60%	55%	60%	66%
YC_bC_r	51%	45%	50%	46%	45%	48%

Figure 4.15 illustrates the outcome of the colour models that possess the successful percentage greater than 75%. The lead top typically possess the highest position inside the solder joint. This geographical location further enables it to contain the highest illumination level. This fuels the success rate achieved by both HSL and HSV colour models since they map intensity value of light to a separate channel (Value (*V*) for HSV

and Lightness (*L*) for HSL).This light distribution pattern on the lead top does not undergo a significant difference between post and prior stages of soldering unless it is heated for a longer time. However, HSL colour model was selected since it slightly outperforms HSV by 1% of successful detection. Figure 4.16 illustrates the localized lead tops in several solder joint images using this methodology.

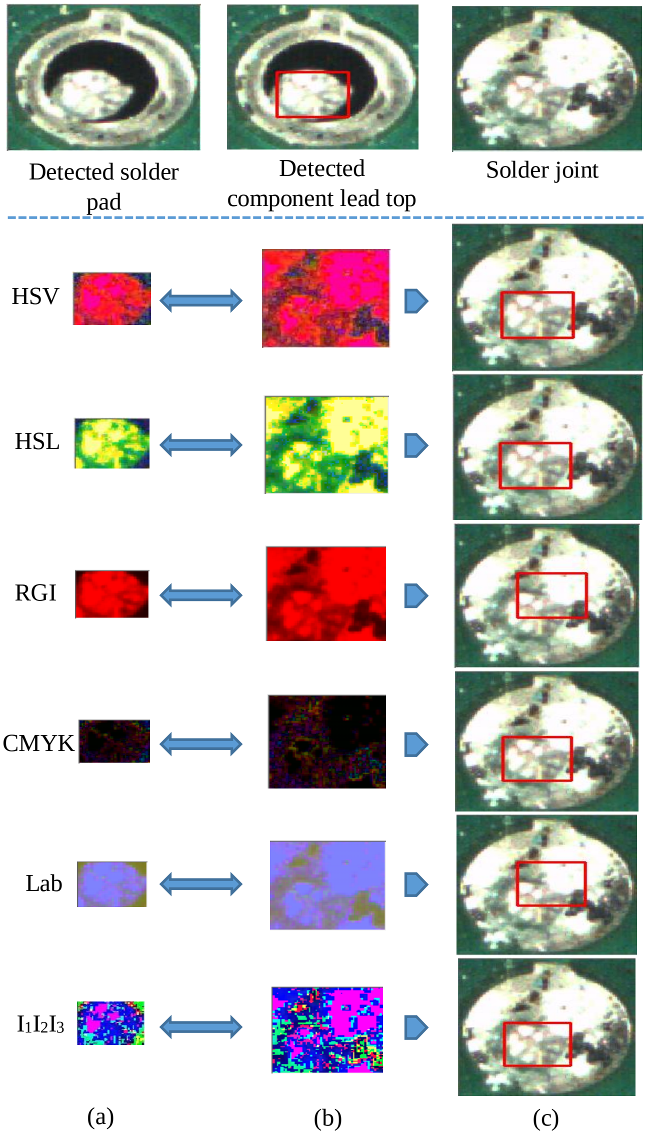


Figure 4.15: Localized lead top using SQDIFF algorithm over several colour models. (a). Colour space transformed lead top image. (b). Colour space transformed ROI set solder joint. (c). Lead top localized solder joint

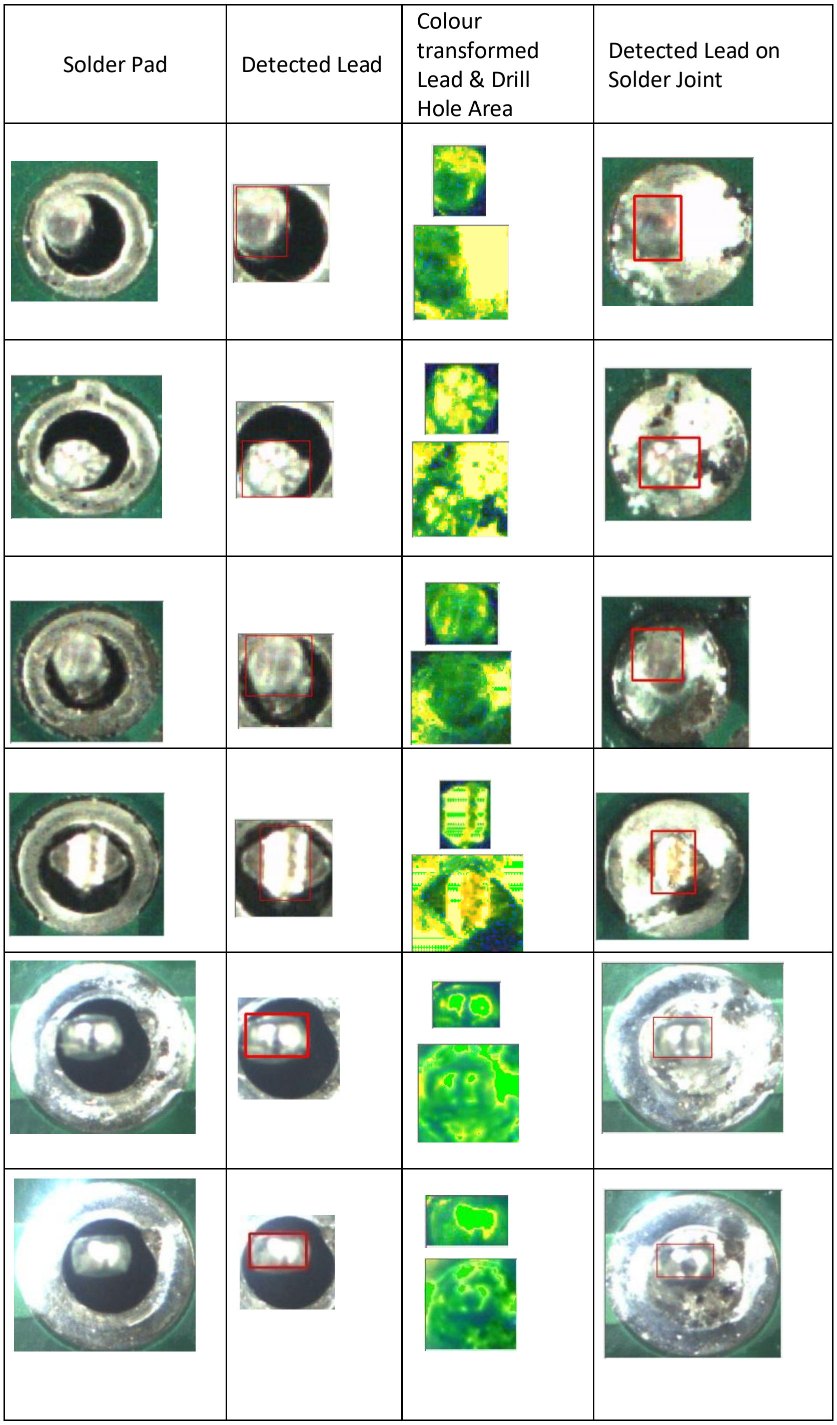

Solder Pad
Detected Lead
Colour transformed Lead & Drill Hole Area
Detected Lead on Solder Joint

Figure 4.16: Detected THT component lead top on several HSL colour transformed solder joints using SQDIFF template matching algorithm

These results have been obtained under different controlled situations like, highly illuminated conditions (2nd, 5th and 6th rows), over heated conditions where the colour of the component lead has been faded (1st and 3rd rows) and when considerable area of the component lead is covered by the solder (4th row). According to the results presented so far in Chapter 4, it could be seen that the proposed methods have provided a higher accuracy level in identifying and localizing the THT component lead top at a very low computational cost. As described earlier in Section 4.1, the precise implementation of this stage lays a strong foundation for the operation of solder quality inspection platform. Chapter 5 describes the implementation of solder quality classification stage of the implemented AOI system.

5. Classification of the Quality of THT Solder Joints

The primary task of an AOI system is to provide a non-contact testing capability over a given PCB to inspect for soldering defects occurred during an assembling process. This is the core of any vision system deployed in EMS industry. As described in Chapter 1, the implemented vision system inspects every action performed by the robotic system, whereas other commercially available systems perform the inspection on already soldered joints.

In Chapters 2-4, a detailed explanation on the implementation of the supporting stages for the classification of the quality of THT solder joints has been provided. These chapters sequentially described the importance of each stage and their impact on the robustness and accuracy of the solder quality assurance process. This Chapter describes the implementation of most critical task involved with the AOI system, soldering quality classification. In Chapter 5, a detailed overview has been provided on a set of new algorithms that have been developed to identify and localize various types of soldering defects precisely. It would be better to understand the types of soldering defects that may occur during a soldering process before going deeper into the methods developed to identify them. Section 5.1 presents a detailed overview on several soldering defect types that have been primarily focused in this study.

5.1 Types of Soldering Defects on THT Solder Joints

Soldering defects are the irregular distributions of applied solder by a human operator or a soldering mechanism across a bare solder pad during the soldering process. The precise identification of these defects is highly preferred inside a standard assembling process, since prior identification of such defects leads to minimize the deficiencies within the manufacturing process. In this research, an automatic solder quality inspection mechanism that is integrated with a THT soldering system, has been implemented. The soldering defects that are more likely to occur inside a solder joint, can be summarized as given below.

- Solder bridging (*Figure 5.1(a)*):
- Not soldered areas (Voids) inside the drill-hole region (*Figure 5.1(b)*):
- Voids on solder pad area (*Figure 5.1(c)*)
- Excess solder on solder joints (*Figure 5.1(d)*)

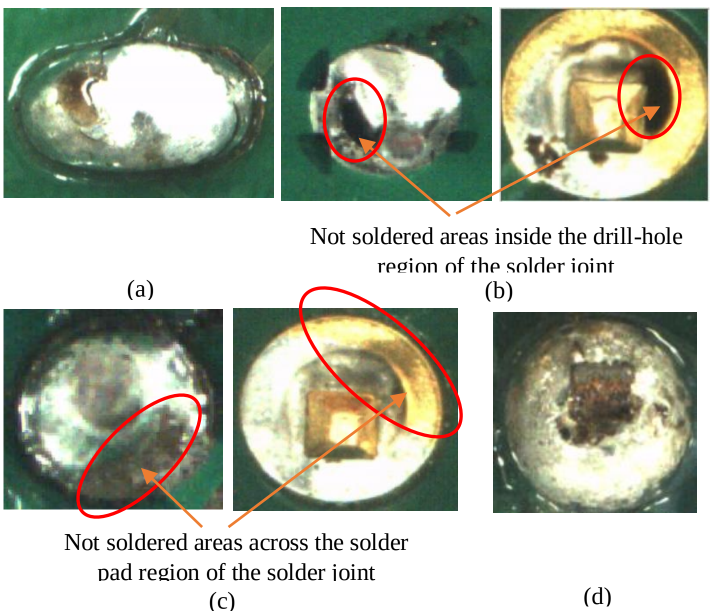

Not soldered areas inside the drill-hole region of the solder joint
(a)
(b)
Not soldered areas across the solder pad region of the solder joint
(c)
(d)

Figure 5.1: Different types of soldering defects. (a) Solder bridging. (b) Voids inside drill-hole region. (c) Voids on the pad region. (d) Excess solder on the joint

The proposed AOI system is capable of identifying above solder defects precisely for different solder mask colours, pad colours, lead colours and shape of the solder joint under varying illumination levels. All these features are implemented using the orthogonally acquired images from a camera placed perpendicular to the placed PCB. Automatic classification of soldering defects adds a huge advantage for any soldering robotic system in terms of providing the possibility of automatic soldering defect identification and reworking over defective solder joints. Sections 5.2-5.5 describe the algorithms that can be effectively used to detect four major types of soldering defects and evaluate their performance under different practical and environmental conditions.

5.2 Identification the Defect of Solder Bridging

Solder bridges as shown in Figure 5.1 (a) are created due to spreading out of accumulated solder on the soldering iron tip to neighbouring solder pads. Solder bridges are identified by comparing area of the solder pad and solder joint. In Chapter 2, colour model which maximizes the vector distance between the solder pad area (foreground) and immediately surrounding PCB area (background) is presented for different PCB solder mask colours and for different pad plating types. The same technique is proposed to identify the solder joint. Table 5.1 depicts the mean (μ) and standard deviation (σ) of the vector distance between a foreground and the background of the soldered joint for nine different colour models and four different solder mask colours. These results were acquired over a sample of 150 solder joints for each solder mask colour.

Table 5.1: Mean (μ) and standard deviation (σ) of the vector distance between the foreground and background of the soldered joint for nine different colour models and four different solder mask colours

Colour Model	**Green Solder Mask**			**Blue Solder Mask**			**Red Solder Mask**			**Black Solder Mask**		
	μ	σ	$\frac{\mu}{\sigma}$	μ	σ	$\frac{\mu}{\sigma}$	μ	σ	$\frac{\mu}{\sigma}$	μ	σ	$\frac{\mu}{\sigma}$
RGB	289.55	40.86	7.09	324.15	41.2	7.87	104.589	18.868	5.54	112.04	20.103	5.57
HSV	205.23	16.15	12.71	235.1	5.99	39.25	196.187	14.163	13.85	108.06	9.911	10.90
HSL	231.18	37.2	6.21	237.3	32.15	7.38	150.620	23.463	6.42	69.38	10.967	6.33
YIQ	290.12	40.15	7.23	307	28.2	10.89	74.644	9.0967	8.21	76.99	9.900	7.78
$I_1I_2I_3$ Modified	134.2	12.1	11.09	324.15	41.2	7.87	63.439	10.217	6.21	48.57	12.678	3.83
YC_bC_r	206.32	25.15	8.20	235.1	5.99	39.25	56.369	11.131	5.06	69.39	12.411	5.59
XYZ	279.13	43.4	6.43	237.3	32.15	7.38	38.843	10.006	3.88	108.56	20.320	5.34
$I_1I_2I_3$	259.1	43.5	5.96	307	28.2	10.89	62.759	26.469	2.37	62.22	9.614	6.47
Lab	161.23	20.4	7.90	324.15	41.2	7.87	61.287	12.306	4.98	72.44	12.15	5.96

From Table 5.1, it is clear that HSV colour model gives the highest value for the $\frac{\mu}{\sigma}$ ratio for all PCB solder mask types. Hence, HSV colour space is used with the k-means colour clustering algorithm to extract solder joints. The result generated by the k-means colour clustering algorithm is shown in Figure 5.2 (c) for the solder bridge show in Figure 5.1 (a). Then the colour clustered image is converted into a binary image by thresholding to compare the size of the solder joint with respect to the corresponding solder pad size (refer Chapter 3). The detected solder bridge is shown in Figure 5.2 (d). Since this is the easiest type of defects to be identified, a deep analysis on the classification of this defect is not provided.

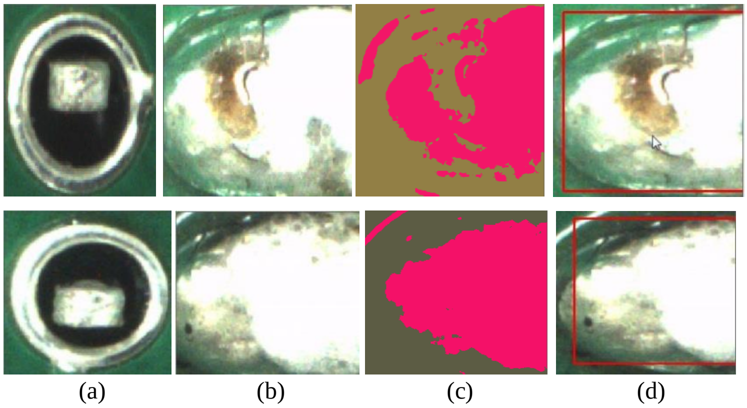


Figure 5.2: Identification of Solder bridging among different solder pads. (a) Solder pad with component lead. (b) Solder bridge. (c) Result from k-means algorithm. (d) Detected solder bridge shown by red box

5.3 Identification of Not Soldered Areas inside the Drill-Hole Region of a Solder Joint

Solder voids as shown in Figure 5.1 (b) are created due to insufficient solder feeding, low tip temperature and poor tip conditions. In manual soldering, solder voids due to these imperfections is usually corrected by the operator then and there. There is a high probability of getting solder voids due to these imperfections in a soldering system which lacks the human skills. The only solution to overcome this problem is to identify these defects using an AOI system and carry out rework accordingly. These defects can be identified using a colour image segmentation algorithm. As the defect to be analysed is within the drill-hole region, a masking image is generated based on the computed drill-hole as described in Chapter 4, for image size reduction. Figure 5.3 illustrates how the region contained by the drill-hole is filtered from solder joint.

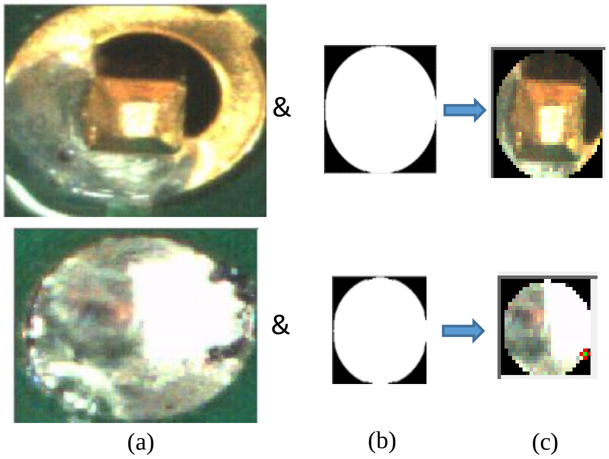


Figure 5.3: Image size reduction for solder void detection inside drill-hole. (a) Solder joint. (b) Binary masking image. (c) ROI

In order to identify the solder defects inside the drill-hole region, the suitability of following approaches was studied.

- colour clustering using k-means, mean-shift and fuzzy c-means algorithms
- graph-cut based image segmentation algorithm [57-61]

Colour clustering is a process that reduces the number of distinct colours used in an image into homogenous colour clusters. Although, colour clustering is robust and efficient in many fields of image processing, it did not provide better results compared to graph-cut algorithm for the detection of solder voids inside the drill-hole region. The accuracy of these colour clustering algorithms was severely affected by the uneven illumination distribution, shape of the solder joint, shadows on the solder joint and flux residues.

Graph-based image segmentation techniques models a given input image (I) in terms of a graph defined by Eq. (45) [58].

$$G = (V, E) \tag{45}$$

Where,

G : Undirected graph of I

V : is a collection of nodes ($v_i \in V$), where each node (v_i) corresponds to each pixel in the image. Here *i=1, 2,...,*$(m \times n)$ *for* $(m \times n)$ image

E : is a collection of edges ($e_{ij} \in E$), where each edge (e_{ij}) connects certain pairs of neighboring pixels (v_i and v_j)

The input image I is prior smoothed by gaussian function to eliminate effects due to uneven illumination distribution. The collection of edges E is constructed by taking all nearest neighbour combinations around a given pixel *P(x, y)*. A weight w_{ij} is assigned to each e_{ij} which represents the dissimilarity (e.g., the difference in intensity, colour, motion, location or some other local attribute) between the two pixels v_i and v_j connecting the edge e_{ij} [58]. Although HSV colour model was better suited to identify solder joint against solder mask in the background, it could be found that the RGB colour model was better suited to identify the solder voids inside the drill-hole region by defining the weight w_{ij} as defined in Eq. (46).

$$w_{i,j} = \left(\frac{d_{P_{(x,y)}} + d_{P_{(x',y')}}}{d_{max}}\right) \times 100 \tag{46}$$

Where,

$d_{P_{(x,y)}}$: $\sqrt{R^2_{(x,y)} + G^2_{(x,y)} + B^2_{(x,y)}}$

$d_{P_{(x',y')}}$: $\sqrt{R^2_{(x',y')} + G^2_{(x',y')} + B^2_{(x',y')}}$

d_{max} : $\sqrt{R^2_{Max} + G^2_{Max} + B^2_{Max}}$

The value of $w_{i,j}$ exists within the range of 0 to 200.

In a graph-based segmentation, V is clustered into a set of distinctive components C. Each component $C_l \in C$ is represented by a set of vertices V_l of the connected sub graph G_l [58]. A binary predicate $D(C_A, C_B)$ is defined as in Eq. (47) to evaluate whether vertices v_i and v_j connected by the edge e_{ij} belongs to two different components $C_A \in C$ and $C_B \in C$ or not.

$$D(C_A, C_B) = \begin{cases} true; \ \ if\ Dif(C_A, C_B) > Mint(C_A, C_B) \\ false; \ otherwise \end{cases} \tag{47}$$

Where,

$$Dif(C_A, C_B)\ : \underset{v_i \in C_A, v_j \in C_B, e_{ij} \in E}{min} w_{ij}$$

$$Mint(C_A, C_B) = \min(Int(C_A) + \tau(C_A), Int(C_B) + \tau(C_B))$$

$$Int(C_l) = \underset{e \in MST(G_l)}{max} w(e)$$

$$\tau(C_l) = k/|C_l|$$

In Eq. (47), $Dif(C_A, C_B)$ represents minimum edge weight that connects two different components C_A and C_B. Further, $Int(C_l)$ represents largest weight in the minimum spanning tree (MST) of G_l corresponding to component C_l. $Mint(C_A, C_B)$ denotes the minimum internal difference between components C_A and C_B. The k is a user defined constant and $|C_l|$ is the size of the component C_l.

Even though the kruskal's algorithm [62] is used in [58] to obtain the MST corresponding to each component, prim's algorithm [63] was used in this study. Because it can be easily implemented using matrix operations. The image is segmented using the following algorithm which is depicted in Figure 5.4.

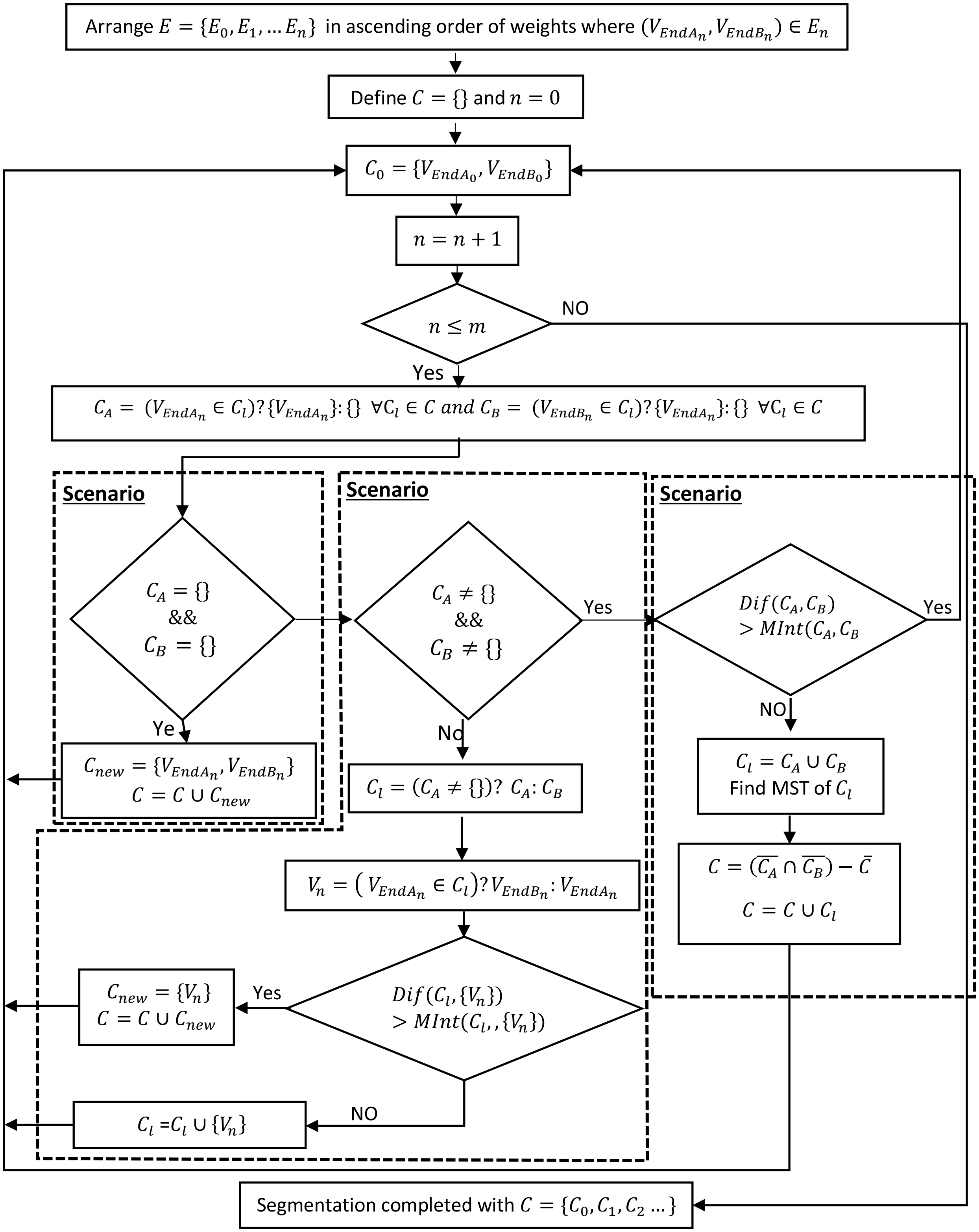

Arrange $E = \{E_0, E_1, \dots E_n\}$ in ascending order of weights where $(V_{EndA_n}, V_{EndB_n}) \in E_n$
Define $C = \{\}$ and $n = 0$
$C_0 = \{V_{EndA_0}, V_{EndB_0}\}$
$n = n + 1$
$n \leq m$
NO
Yes
$C_A = \ (V_{EndA_n} \in C_l)?\{V_{EndA_n}\}:\{\}\ \forall C_l \in C\ and\ C_B = \ (V_{EndB_n} \in C_l)?\{V_{EndA_n}\}:\{\}\ \forall C_l \in C$
Scenario
$C_A = \{\}$ && $C_B = \{\}$
Ye
$C_{new} = \{V_{EndA_n}, V_{EndB_n}\}$
$C = C \cup C_{new}$
Scenario
$C_A \neq \{\}$ && $C_B \neq \{\}$
Yes
No
$C_l = (C_A \neq \{\})?\ C_A : C_B$
$V_n = \left(V_{EndA_n} \in C_l\right)? V_{EndB_n} : V_{EndA_n}$
$Dif(C_l, \{V_n\}) > MInt(C_l,, \{V_n\})$
Yes
$C_{new} = \{V_n\}$
$C = C \cup C_{new}$
NO
$C_l = C_l \cup \{V_n\}$
Scenario
$Dif(C_A, C_B) > MInt(C_A, C_B$
Yes
NO
$C_l = C_A \cup C_B$
Find MST of C_l
$C = (\overline{C_A} \cap \overline{C_B}) - \bar{C}$
$C = C \cup C_l$
Segmentation completed with $C = \{C_0, C_1, C_2 \dots\}$

Figure 5.4: Structure of the graph-cut image segmentation algorithm

5.3.1 Graph-cut based image segmentation algorithm

Step 1: Arrange E in ascending order and define component set $C = \{\}$. Take the first edge in E and assign two vertices of selected edge v_{EndA_0} and v_{EndB_0} to C_0 where C_0 ∈ C.

Step 2: Select next edge in E and execute the applicable Scenario in Step 3.

Step 3:

Scenario 01:

Check whether the vertices v_{EndA_n} and v_{EndB_n} of selected *n*th edge are not contained in any component of C. Then add these two vertices to a new component C_{new} where C_{new} ∈ C.

*Scenario 02***:**

Check whether either vertices v_{EndA_n} or v_{EndB_n} is contained in any component of C. Then compute the binary predicate $D(C_A, C_B)$ by taking the vertice v_{EndA_n} or v_{EndB_n}not contained in any component of C, and the component which contains the vertices v_{EndA_n}or v_{EndB_n}. If $D(C_A, C_B)$ is false merge the vertices v_{EndA_n} or v_{EndB_n}not contained in any component of C with the component which contains the vertice v_{EndA_n} or v_{EndB_n} and find the new MST. Otherwise add the vertice v_{EndA_n} or v_{EndB_n} not contained in any component of C to a new component C_{new} where C_{new} ∈ C.

*Scenario 03***:**

Check whether both V_{EndA_n} and V_{EndB_n} are contained in two different components (C_A and C_B) of C. Then compute the binary predicate ($D(C_A, C_B)$). If $D(C_A, C_B)$ is false, merge C_A and C_B to a new component C_{new} where C_{new} ∈ C. Then find the MST of merged component.

Step 4: Repeat Step 2 and 3 until all the edges in E are considered.

Figure 5.5 illustrates the results obtained by above algorithm for a good solder joint and solder joint with void inside its drill-hole. The area of the detected lead top inside the solder joint is masked out prior to segmentation process to reduce the computational overhead. Further, the number of pixels contained by the homogeneous regions is normalized with respect to the size of drill-hole region. The normalized value is compared against an experimentally computed threshold value. This comparison is performed in order to minimize the impact of smaller residues and darker regions inside the drill-hole. Figure 5.5 contains the regions whose sizes are larger than to this threshold value.

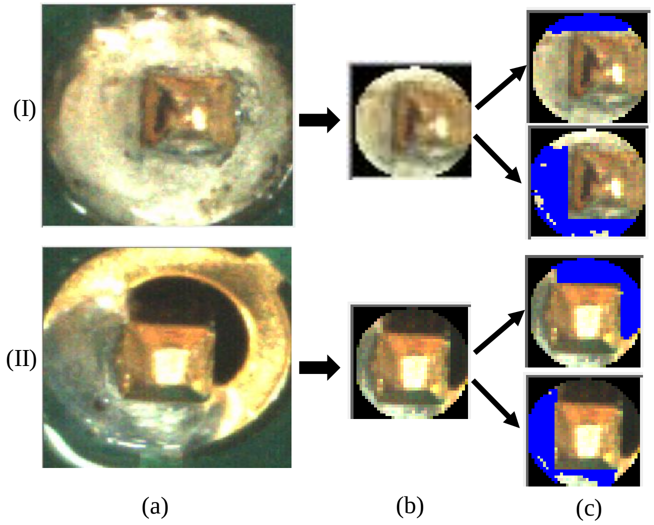


Figure 5.5: Results from graph-cut based image segmentation over two solder joint types. (a) Input image. (b) ROI. (c) Large homogeneous regions (shown in blue) detected by graph-cut algorithm. (I) Good solder joint. (II) Solder joint with defect

In order to classify each identified component as a defect or not, the average value of the MST of respective component was computed and compared against a user defined threshold. All components where average MST value is below the threshold is classified as a defect as illustrated in Figure 5.6. This classification may create false defects due to shadows and dark spots created by flux residues (See Figure 5.6 (II) (b)).

False identification of defects due to shadows and dark spots can be minimized by inspecting the adjoining solder pad area. When there is a void inside the drill-hole, there is a high probability to get a solder void in the adjoining solder pad area (See Figure 5.7).

In order to identify adjoining solder pad area also contains voids, this area must be extracted precisely using a masking image.

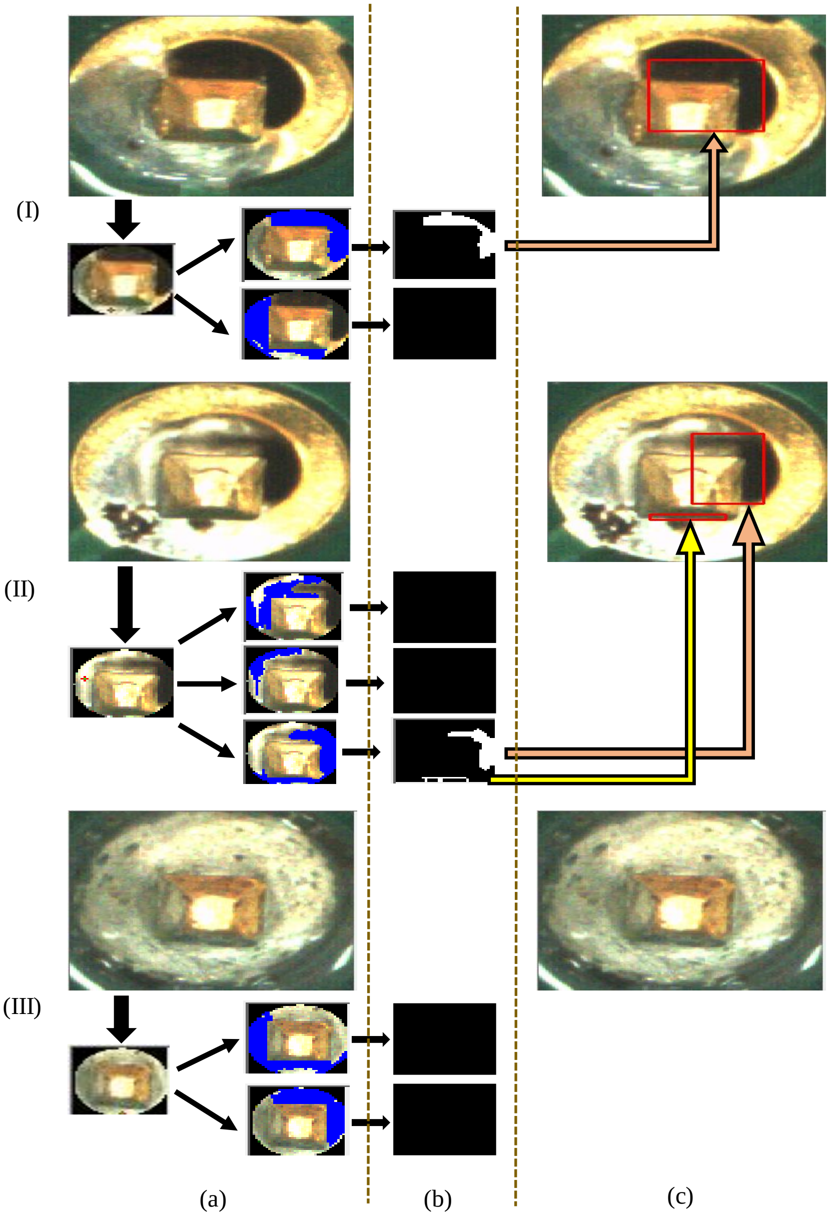


Figure 5.6: Detected faults inside the drill-hole region for three solder joints types. (a) Input image and the segmented homogeneous regions. (b) Thresholded homogeneous regions. (c) Detected faults shown by red box. (I) No false detection. (II) False detection due to a shadow. (III) Perfect joint

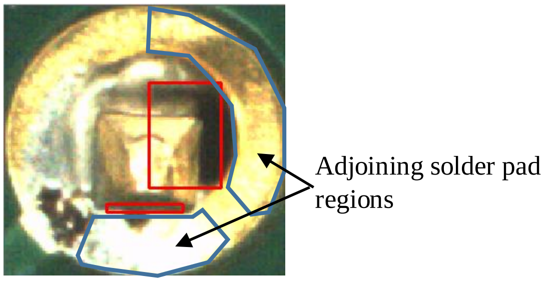


Figure 5.7: Adjoining solder pad regions (blue colour) corresponding to detected voids inside drill –hole (red colour)

5.3.2 Adjoining solder pad region extraction algorithm

Step 1: Get the binary image that contains vertices in the given component C_l, where $C_l \in C$. Compute the line L_1 connecting the image centre point (x_c, y_c) with the top left corner (x_{L_1}, y_{L_1}) and the line L_2 connecting the image centre point (x_c, y_c) with the bottom right corner (x_{L_2}, y_{L_2}) of the bounding rectangle R that contains vertices in C_l as given in Eq. (48) and (49) respectively. (See Figure 5.8)

$$L_1 \rightarrow y_1 = \frac{(y_{L_1} - y_c)}{(x_{L_1} - x_c)} x_1 \tag{48}$$

$$L_2 \rightarrow y_2 = \frac{(y_{L_2} - y_c)}{(x_{L_2} - x_c)} x_2 \tag{49}$$

Step 2: Determine the intersect points of lines L_1 and L_2 with the outer boundary of solder pad as given in Eq. (50) and (51).

$$x_{IP\_L_x} = \begin{cases} x_c - \left|r_p * \cos \alpha_{L_x}\right|, x_c > x_{L_x} \\ x_c + \left|r_p * \cos \alpha_{L_x}\right|, x_c < x_{L_x} \end{cases} \tag{50}$$

$$y_{IP\_L_x} = \begin{cases} y_c - \left|r_p * \sin \alpha_{L_x}\right|, y_c > y_{L_x} \\ y_c + \left|r_p * \sin \alpha_{L_x}\right|, y_c < y_{L_x} \end{cases} \tag{51}$$

Where,

$(x_{IP\_L_x}, y_{IP\_L_x})$: intersect points of line L_x, *(x=1, 2)* with the outer boundary of the pad

r_p : radius of the solder pad region

α_{L_x} : gradient of the line L_x in radians

Step 3: Determine the intersect points of lines L_1 and L_2 with the outer boundary of drill-hole as given in Eq. (52) and (53).

$$x_{ID\_L_x} = \begin{cases} x_c - \left| r_d * \cos \alpha_{L_x} \right|, x_c > x_{L_x} \\ x_c + \left| r_d * \cos \alpha_{L_x} \right|, x_c < x_{L_x} \end{cases} \tag{52}$$

$$y_{ID\_L_x} = \begin{cases} y_c - \left| r_d * \sin \alpha_{L_x} \right|, y_c > y_{L_x} \\ y_c + \left| r_d * \sin \alpha_{L_x} \right|, y_c < y_{L_x} \end{cases} \tag{53}$$

Where,

$(x_{ID\_L_x}, y_{ID\_L_x})$: intersect points of line L_x, *(x=1, 2)* with the outer boundary of the drill-hole

r_d : radius of the drill-hole

α_{L_x} : gradient of the line L_x in radians

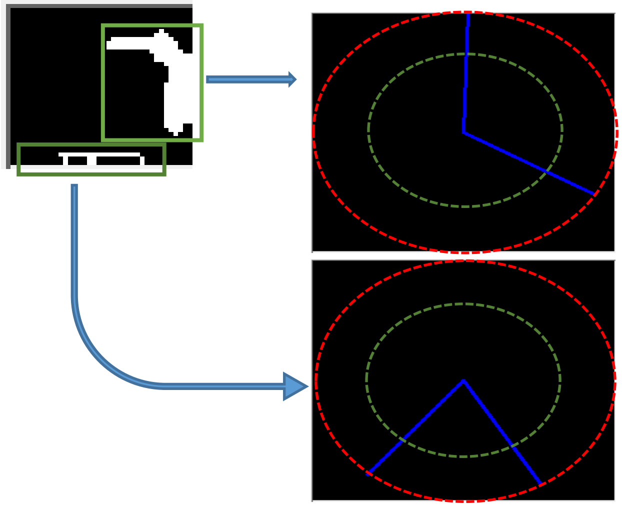


Figure 5.8: Intersection of line segments L_1 and L_2 (blue lines) with outer boundary of solder pad (red dash circle) and outer boundary of drill-hole (green dash circle)

Step 4: Defines the masking region T covering the area defined by lines L_1, L_2 and the pad arc segment between intersect points $(x_{IP\_L_1}, y_{IP\_L_1})$ and $(x_{IP\_L_2}, y_{IP\_L_2})$ and the drill-hole arc segment between intersect points $(x_{ID\_L_1}, y_{ID\_L_1})$ and $(x_{ID\_L_2}, y_{ID\_L_2})$.

Step 5: Extract the image of the adjoining solder pad region corresponding to detected void inside the drill-hole by masking the solder joint image region with T.

Figure 5.9 depicts images corresponding to key steps of the above algorithm.

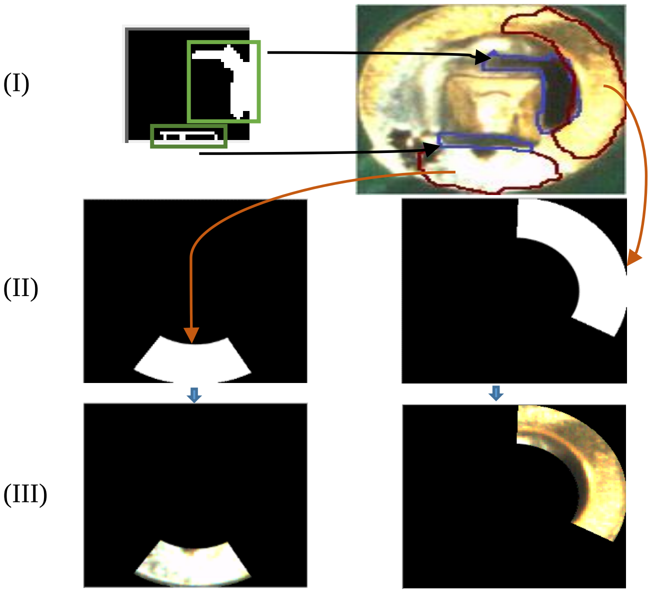


Figure 5.9: Extracting adjoining solder pad region corresponding to detected voids inside the drill-hole region. (I) Detected voids and their adjoining solder pad regions. (II) Region T. (III) Extracted image of the adjoining solder pad region

5.3.3 Verification of the availability of applied solder on adjoining solder pad region

In a solder joint with voids on the adjoining solder pad, there is a considerable colour hue difference between the solder pad (gold and tin plated) and solder in majority of the cases. Even though, the RGB colour model rendered better outcome for gold-plated solder pads due to significant colour dissimilarity comparing to solder, experimental results showed that the HSL colour model was better suited for tin-plated pads.

The suitability of k-means, FCM and mean-shift colour clustering algorithms were analysed to decide whether extracted regions contain voids. Table 5.2 illustrates the average euclidean distance, average computation time and repeatability for gold-plated and tin-plated solder pads. These values are calculated using 200 different solder joint images of size (45×45) for each pad types. As Mean-shift algorithm quantizes available colours into minimum possible number of colour clusters, the euclidean distance for mean-shift algorithm is not included in Table 5.2.

Table 5.2: Performance evaluation of clustering algorithms for solder void confirmation on adjoining solder pad to a detected void in drill-hole region

Algorithm	**Pad Type**	**Average Euclidean Distance/units**		**Average Computation time/ms**	**Repeatability**
		Solder Pad	**Solder**		
k-means	Gold plated	317.5	422.2	35.28	97%
	Tin plated	120.3	235.2	60.57	96%
FCM	Gold plated	324.5	422.2	103.28	97.2%
	Tin plated	118.5	235.4	159.28	96%
Mean-Shift	Gold plated	NA	NA	209.51	96%
	Tin plated	NA	NA	188.72	95%

From Table 5.2, it can be seen that k-means and FCM provides similar average euclidean distance. All three algorithms have similar values for repeatability. But k-means clustering algorithm significantly outperforms FCM and mean-shift in terms of computation time. Since the computation time is a crucial parameter in this application, k-mean was chosen as the clustering algorithm to find voids on the adjoining solder pad.

Once the filtered-out images are colour clustered as illustrated in Figure 5.10, they are thresholded against a predetermined threshold value (different threshold values for tin and gold-plated solder pads) that has been computed experimentally. Then the resulted binary image is processed to compute the area of not soldered. Finally, the size of the computed area is compared against an adaptive threshold value that depends on the size of the adjoining pad (refer Section 8.4.1). Table 5.3 illustrates the successful (identification of >95% of defective area) and erroneous detection rate (identification of <95% defective area or falsely identified regions) of the graph-cut algorithm with and without the inspection of adjoining solder pad region, based on the results obtained over a sample of 200 solder joints. Table 5.3 demonstrates that the inspection of adjoining pad region of the detected void inside the drill-hole, minimizes the erroneous detection rate significantly (by 11%). Figure 5.11 illustrates the results obtained over several solder joints using the proposed algorithm. The adjoining pad verification is performed only for the detected voids whose size is below 60% of the size of the corresponding drill-hole. For others, the entire solder pad is processed to determine whether the defect is real or not.

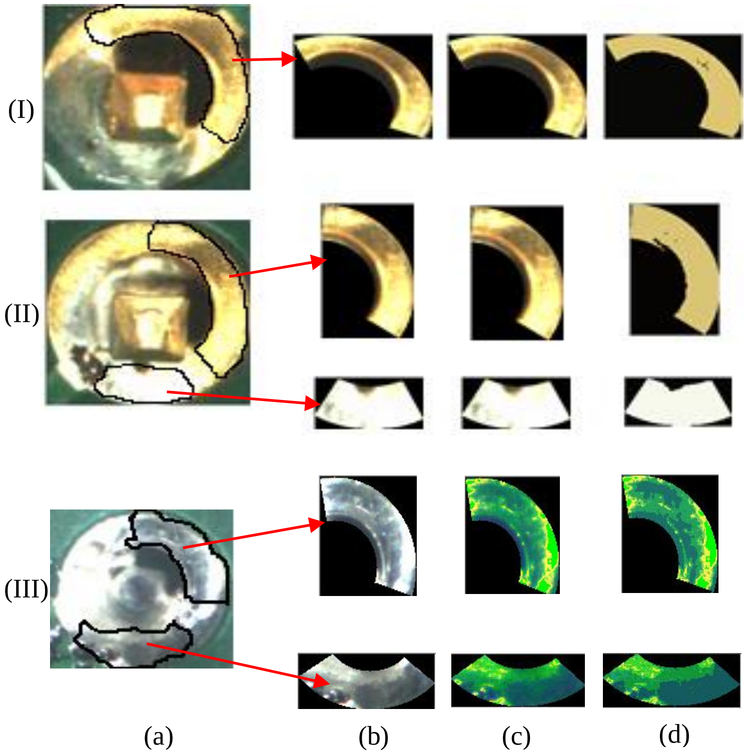


Figure 5.10: Resulted images from colour clustering algorithms for adjoining solder pad verification. (a) Solder joint. (b) Adjoining solder pad areas corresponding to detected voids. (c) Colour transformed image. (d) Colour clustered image

Table 5.3: Successful and Erroneous detection rate of void detection algorithm inside drill-hole region of the solder joint

Algorithm	Successful Detection Rate on defective solder joints	Erroneous Detection Rate on good solder joints.
Implemented algorithm without the inspection of adjoining solder pad region	100%	12%
Implemented algorithm with the inspection of adjoining solder pad region	100%	1%

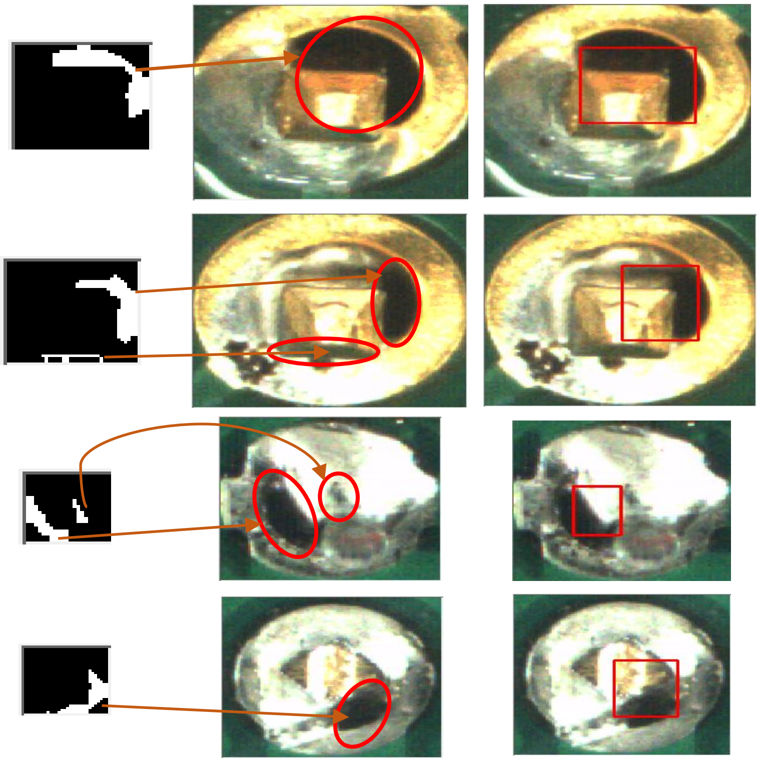


Figure 5.11: Detected voids inside the drill-hole region of a solder joint using the proposed algorithm. (a) Binary image from the graph-cut algorithm. (b) Detected voids inside drill-hole region. (c) Positively detected voids inside the drill-hole region after the inspection of adjoining solder pad region

5.4 Identification of Not Soldered Areas across the Solder Pad Region of a Solder Joint

Solder voids on solder pad region as shown in Figure 5.1 (c) are created due to insufficient solder feeding, low tip temperature and poor tip conditions. If the AOI system does not identify any voids inside the drill-hole region, then it inspects voids on the solder pad region as depicted in Figure 1.1. This section describes the implementation of the algorithms that can be effectively used to identify voids on the solder pad. As this defect exists only on the solder pad region, the solder pad region is filtered out by defining a binary masking image using the approximated drill-hole region (refer Chapter 4) and CAD data (for the dimensions of solder pad) for creating a ROI as depicted in Figure 5.12.

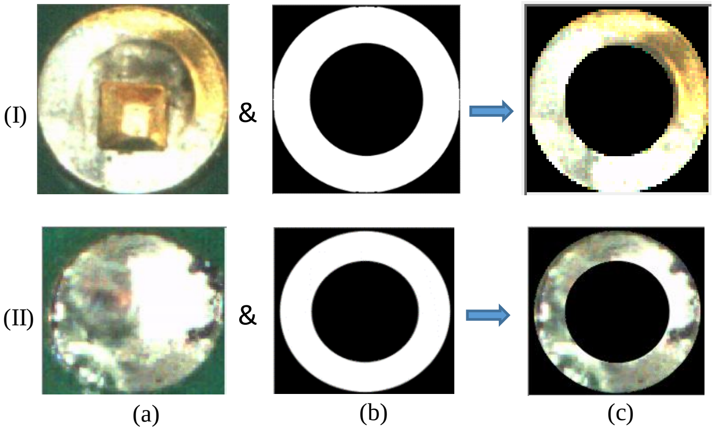


Figure 5.12: Image size reduction for solder void detection on solder pad. (a) Solder joint. (b) Binary masking image. (c) ROI

Although the colour hue difference between gold plated solder pad and solder in Section 5.3 was taken into account in determining adjoining solder pad is a void or not, when checking for voids on the solder pad, the colour hue is severely affected by uneven illumination distribution, shadows on the solder joint and solder pad oxidizations during PCB storage (see Figure 5.13). In order to minimize the effect due to these imperfections, following algorithm is proposed.

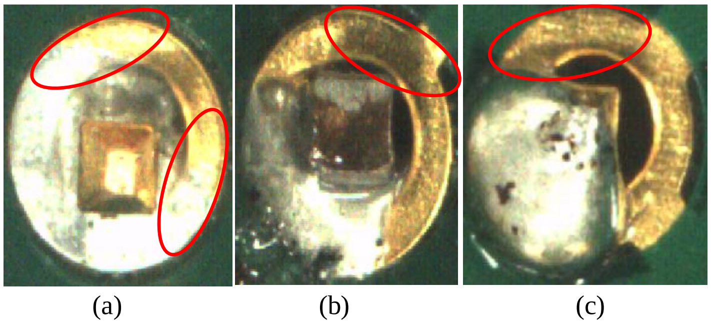


Figure 5.13: Imperfections on solder pad regions marked in red boundary. (a) Uneven illumination. (b) Shadows. (c) Oxidizations

Voids on solder pad detection algorithm

Step 1: Segments the solder pad region into set of components C using the graph-cut based image segmentation algorithm described in Section 5.3.1. (See Figure 5.14 and Figure 5.15 (I) for an illustration.)

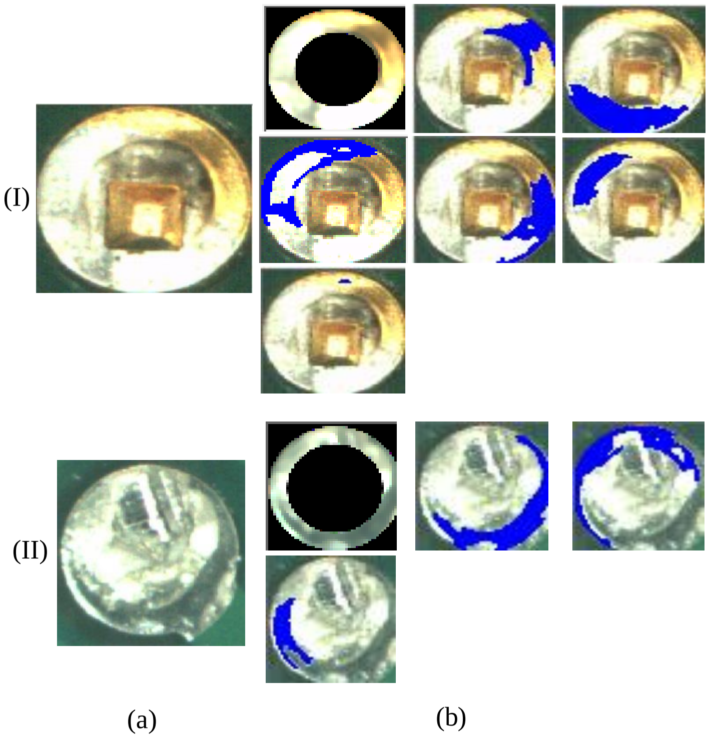


Figure 5.14: Image segmentation using graph-cut algorithm (Blue highlighted regions represents the segmented regions). (a) Solder joint. (b) Segmented regions

Step 2: Find the number of vertices $|C_l|$, the average MST of the graph corresponding to C_l $AVG(MST_{C_l})$ and binary predicate $R(C_l)$ for each component $C_l \in C$. Use Eq. (54) and (55) for gold plated and tin-plated pads respectively for determining $R(C_l)$. Remove all components where binary predicate is true. (See Figure 5.15 (II).)

$$R(C_l) = \begin{cases} true; if(|C_l| < T_A \quad || \, T_{UG} \geq \; AVG(MST_{C_l})) \\ false; otherwise \end{cases} \tag{54}$$

$$R(C_l) = \begin{cases} true; if(|C_l| < T_A \quad || \, AVG(MST_{C_l}) \geq T_{UT}) \\ false; otherwise \end{cases} \tag{55}$$

Where,

T_A : User defined threshold value for the number of vertices inside a given C_l

T_{UG} and T_{UT}: User defined upper threshold value for average MST value of a given C_l for gold plated and tin-plated pads respectively

Step 3: Find a set of ROIs of the input image by obtaining the bounding box covering the area contained by all vertices of each component $C_l \in C$. (See Figure 5.15 (II).)

Step 4: Transform each ROI of the input image using modified version of $I_1I_2I_3$ colour model described in Chapter 2 for gold plated solder pads and HSL colour model for tin plated solder pads, See Figure 5.15 (III). Cluster each ROI using k-means colour clustering algorithm, See Figure 5.15 (IV).

Step 5: Identify defective regions by comparing the hue value of each cluster against user defined upper and lower thresholds. (See Figure 5.15 (V).) (Note: Upper & lower thresholds are different for gold-plated and tin-plated solder pads and defective regions are within the thresholds.). Figure 5.15 (VI) illustrates the identified void areas on the solder pad after executing above algorithm.

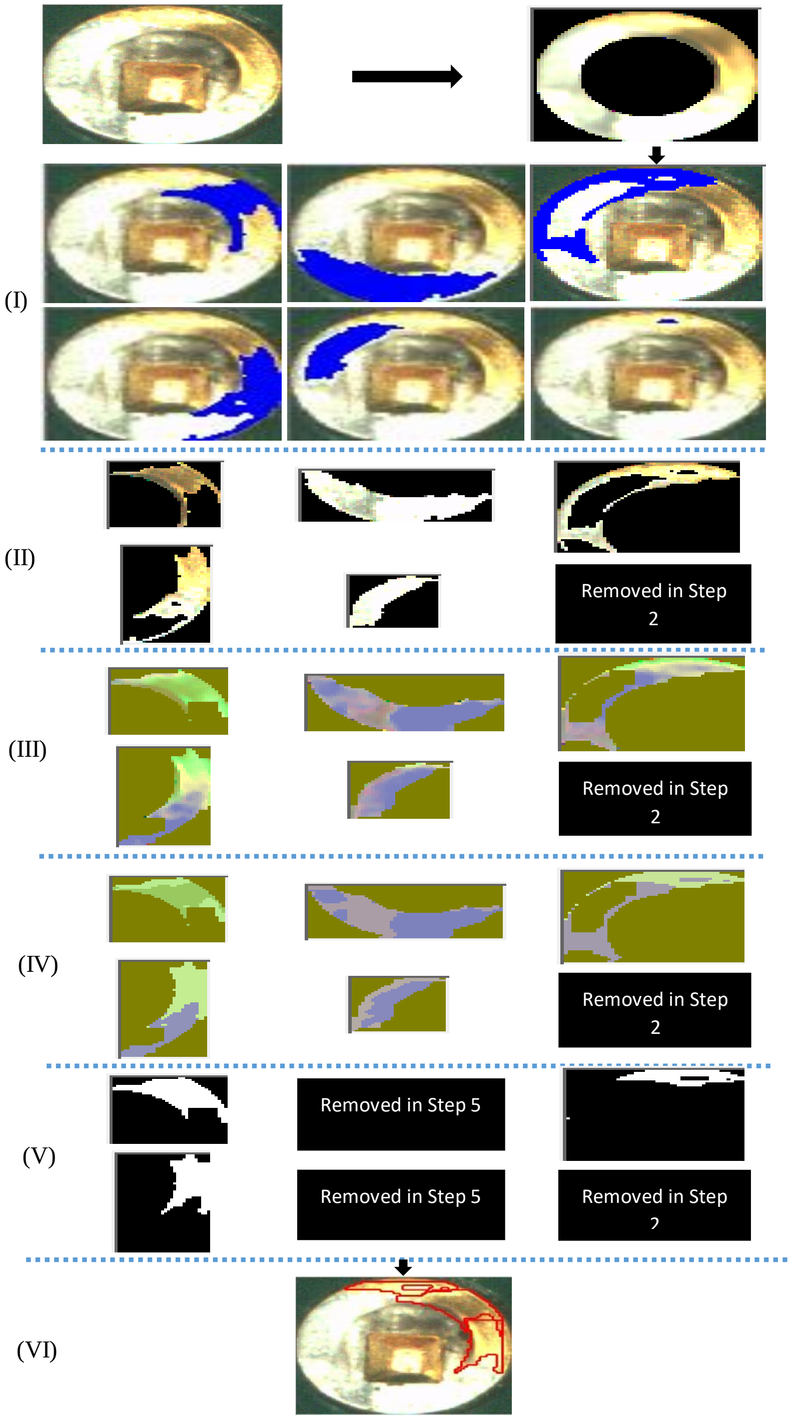


Figure 5.15: Step wise illustration of proposed algorithm for void detection on solder pads. (I) Segmented regions - Step 1. (II) ROI corresponding to segmented regions – Steps 2 & 3. (III) Colour transformed images - Step 4. (IV) Identified defective regions - Step 5 & 6. (V) Identified voids on the solder pad region - Step 7. (VI) Identified void regions shown by red boundaries

In Step 1, graph-cut algorithm is used to segment the input image using RGB colour model. Further in Step 4, k-means clustering algorithm is used together with modified version of $I_1I_2I_3$ colour model for gold plated solder pads and HSL colour model for tin plated solder pads respectively. This approach is used to minimize the possibility of segmenting highly illuminated bare solder pad and soldered area into one defective region. To reinforce this concept, the successful (identifies >95% of defective regions) and erroneous detection percentage (<95% of defective regions or false detections) of the proposed algorithm with the graph-cut based segmentation algorithm and k-means colour clustering were evaluated using a sample of 300 solder joints. The results are tabulated in Table 5.4 and some sample images of the successful void detection are depicted in Figure 5.16.

Table 5.4: Successful and erroneous detection rate in identifying voids inside the solder pad region of a solder joint over gold-plated and tin-plated solder pads

Solder Pad Type	**Detection Rate**	**Applied Method**		
		Only with k-means	**Only with Graph-cut based algorithm**	**Combination of graph-cut based algorithm and k-means**
Gold-Plated	Successful detection rate over defective solder joints	52%	81%	100%
	Erroneous detection rate over good solder joints	49%	55%	<1%
Tin-Plated	Successful detection rate over defective solder joints	41%	73%	100%
	Erroneous detection rate over good solder joints	85%	65%	2.8%

In one of the publications made on this research, a statement has been made that the RGI colour model can be effectively used for the void identification on tin-plated solder pads. However, further analysis showed that the application of HSL colour model can render better outcome than the RGI colour model over the varying appearance of tin-plated solder pad surface.

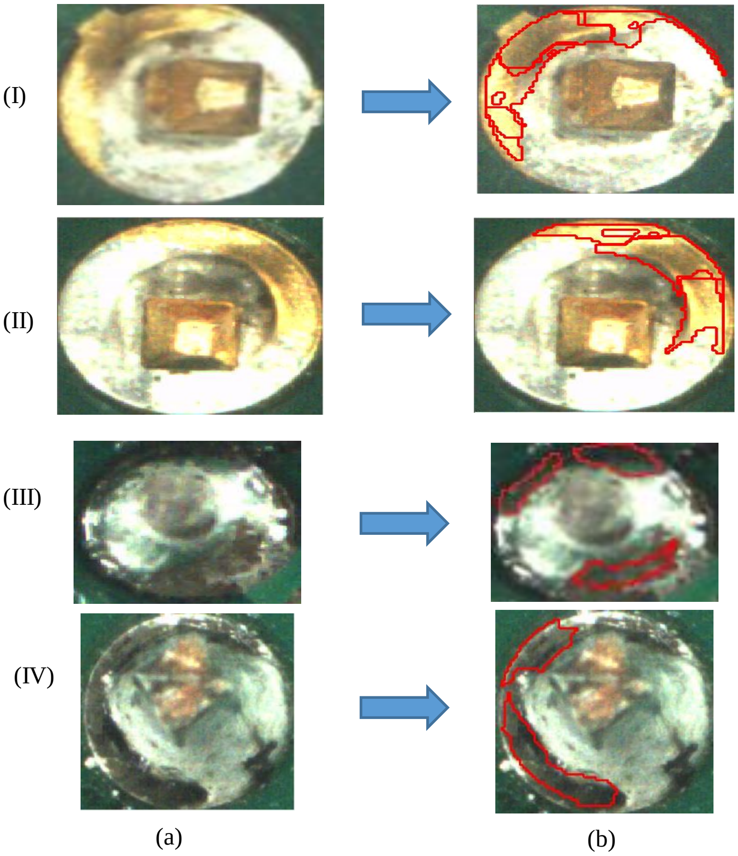


Figure 5.16: Detection of voids on tin and gold plated solder pad regions. (a) Solder joint. (b) Identified voids on (I) & (II) gold plated solder pads (III) & (IV) tin plated solder pads

5.5 Computation of Solder Coverage on the Pad

In Institute for IPC standards, it is stated that at least 75% of solder landing area (solder pad area) must be covered with solder paste for all IPC classes (IPC class I, II and III) as a guide line [15, 16]. The solder joints with voids inside the drill-hole region are treated as bad solder joints (See Figure 1.1). Therefore, it is important to compute the solder coverage on the solder pad area.

In order to calculate the solder coverage on the pad, first count the number of pixels in each defective region A_i as identified in Step 5 of voids on solder pad extraction algorithm.

Then count the number of pixels in the solder pad region A_S. Now the solder coverage on the pad SC can be calculated by Eq. (56).

$$SC = (\sum_{i=1}^{n} A_i \Big/ A_S) * 100 \tag{56}$$

Figure 5.17 depicts some outputs generated by this computation.

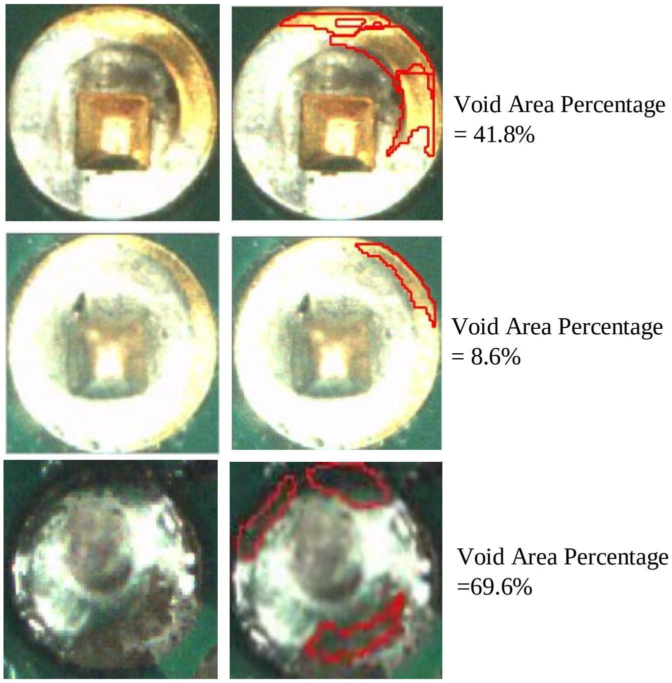


Figure 5.17: Evaluation on the solder coverage for the solder joints with detected voids on the solder pads. (a) Solder joint. (b) Detected voids and the solder coverage percentage

5.6 Identification of Over Soldered Situation Applied on a Solder Joint

Excess solder on solder joint as shown in Figure 5.1 (d) are created due to excessive solder feeding, low tip temperature and poor tip conditions. Identification of excess solder on a solder joint is more challenging comparing to the classification of defects identified in previous sections. As described in Chapter 4, the difference of light distribution between a good solder joint and a joint with excess solder is the only significant property that can be used to determine the amount of solder applied. In order to use this property effectively, it is vital to use a colour model that is capable of mapping lightness of a given image into a main colour channel. Hence, HSV and HSL colour models are suitable candidates. Colour transformed images for a good and excess solder joint are depicted in Figure 5.18.

In Figure 5.18, it can be seen that the most significant colour difference between the highly illuminated regions and the rest of the areas inside the solder joint is seen in HSL colour model. Hence, the following algorithm is used in this application to detect excess solder on a solder joint.

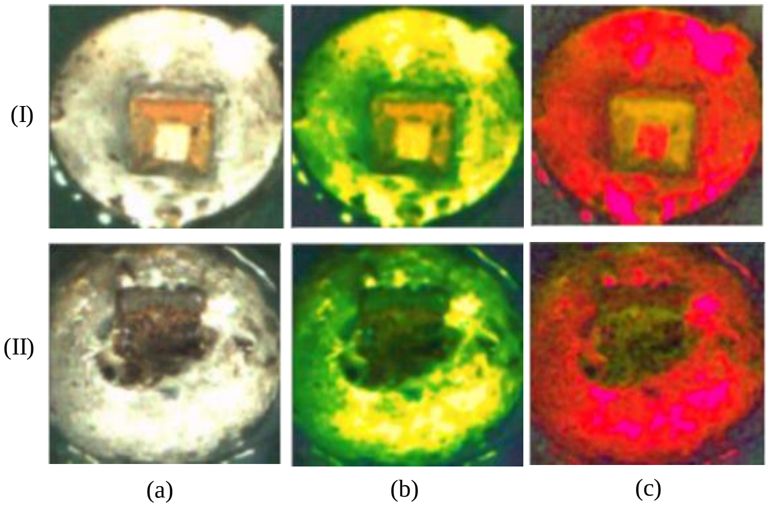


Figure 5.18: Colour transformation from RGB to HSL and RGB to HSV colour models for the classification of excess solder. (a) Solder joint. (b) RGB to HSL. (c) RGB to HSV. (I) Good joint. (II) Defective joint

Excess solder detection algorithm

Step 1: Transform colour space from RGB to HSL. (See Figure 5.18 (b)).

Step 2: Cluster the transformed image using k-means colour clustering algorithm. Then generates the binary image I_{TH} against a user defined hue threshold. (See Figure 5.19 (II) and (III)).

Step 3: Find each connected region C_{Cont_i} in I_{TH} and add them to a set C_{Cont} where $C_{Cont_i} \in C_{Cont}$.

Step 4: Find the largest connected region C_{Cont_L} in C_{Cont}. Then compare the percentage of the area contained by C_{Cont_L} inside the solder joint against a user defined threshold value. If the result is true then define the solder joint as good, unless go to Step 5.

Step 5: Check whether the location of C_{Cont_L} exists inside the location of component lead. If the result is true, then define the solder joint as good. Unless finds the second largest connected region C_{Cont_M} in C_{Cont} and go to Step 6.

Step 6: Make a copy of I_{TH} (I_{TH_C}). Then assign C_{Cont_L} as the desired connected region C_{Cont_R}.

Step 7: Remove all other connected regions in I_{TH} that are not contained by C_{Cont_R}. (See Figure 5.19 (IV).)

Step 8: Find horizontal boundary pixel set $P_{Horiz.}(B)$ and vertical boundary pixel set $P_{Vert.}(B)$ in C_{Cont_R} using Eq. (57) and (58).

$$P_{Horiz.}(B) = \Big(\big(I_{TH\_X}(x-1,y) == 0 \parallel I_{TH\_X}(x+1,y) == 0\big) \,\&\&\, p(x,y) == 255\Big)?\ P_{Horiz.}(B) \cup p(x,y): No\ Operation \quad \forall(\mathrm{x,y}) \in C_{Cont.R} \tag{57}$$

$$P_{Vert.}(B) = \Big(\big(I_{TH\_X}(x,y-1) == 0 \parallel I_{TH\_X}(x,y+1) == 0\big) \,\&\&\, p(x,y) == 255\Big)?\ P_{Vert.}(B) \cup p(x,y): No\ Operation \quad \forall(\mathrm{x,y}) \in C_{Cont.R} \tag{58}$$

Step 9: Find boundary pixel set $P_{Bound.}(B)$ by selecting distinct vertices in $P_{Horiz.}(B)$ and $P_{Vert.}(B)$. (See Figure 5.19 (V).)

Step 10: Compute the line segment L_{TC} that connects center point of the lead top region and center point of the bounding box of C_{Cont_R}. (See Figure 5.19 (VI).)

Step 11: Find outer boundary pixels $p(x,y)$ within a user defined cone angle $\emptyset_R$ centered to line segment L_{TC}. Then determine the distance $D_{Boundary}$, between the outer boundary pixels $p(x, y)$ and the center point of the lead top. (See Figure 5.20.)

Step 12: Compute average distance difference D_{AVG} between each outer boundary pixel $p(x, y)$ with solder pad radius r_P. Classify the solder joint by comparing D_{AVG} with respect to user defined upper and lower thresholds T_U and T_L as Good, Bad or Undefined using Eq. (59).

$$R(X) = \begin{cases} Good, & D_{AVG} \leq T_L \\ Undefined, & T_L < D_{AVG} \leq T_U \\ Bad, & D_{AVG} > T_U \end{cases} \tag{59}$$

If classification is either ‘Good’ or ‘Bad’, then exit the loop, otherwise go to Step 13.

Step 13: Assign C_{Cont_M} as the desired connected region C_{Cont_R}. Remove all other connected regions that are not contained by C_{Cont_R} in I_{TH_C} and repeat steps 8 - 12. If classification is ‘Undefined’, alarm for manual checking.

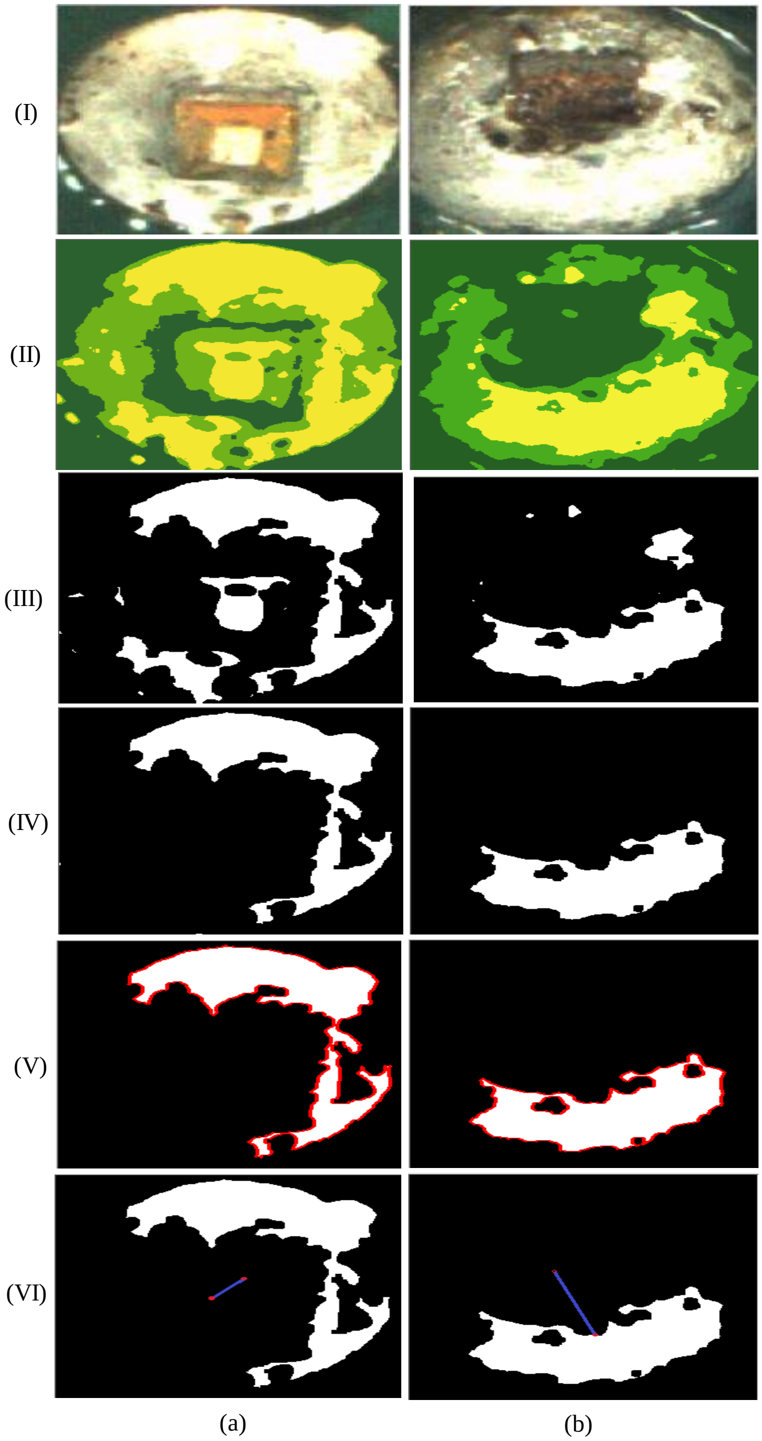


Figure 5.19: Boundary detection of segmented illuminated regions of a good (a) and excess soldered (b) solder joints. (I) Solder joint. (II) Colour clustered image. (III) Thresholded image. (IV) Filtered out largest contour area. (V) Detected boundary points. (VI) Line segment L_{TC} (blue colour)

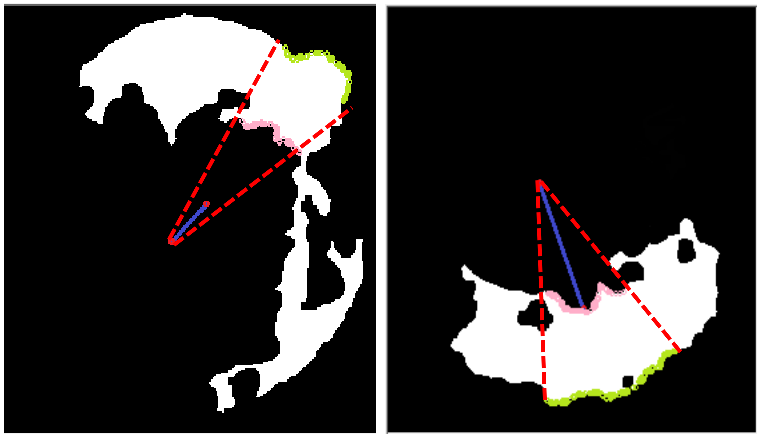


Figure 5.20: Selection of boundary pixels within the offset angle $\emptyset_R$ marked by red dash lines. Pink and green dots show the inner and outer boundary pixels inside $\emptyset_R$ respectively

In Step 4, the largest connected region is selected and compared its size against a user defined threshold. This is because, when the solder joint contains an ideal shape, it has a better reflection of light towards the camera which is being held on top of the solder joint (See Figure 5.21).

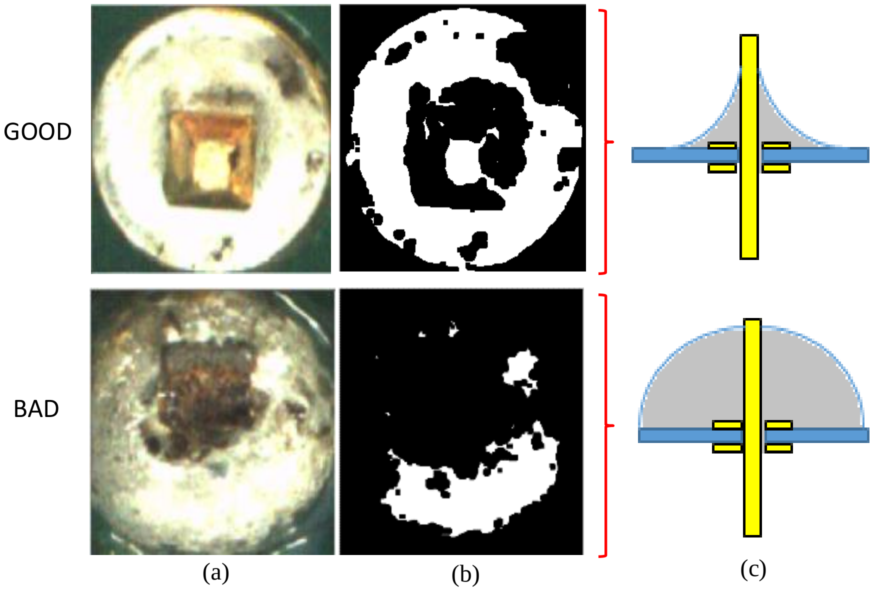


Figure 5.21: Comparison of the size of the connected region inside the resulted binary image with respect to shape of the solder joint for excess solder detection. (a) Solder joint. (b) Resulted binary image after the thresholding. (c) Cross section of a solder joint

In Step 5, the locations of C_{Cont_R} and the lead top are compared. This is because, the top of the component lead holds a significant probability of being illuminated than other areas of the solder joint.

In Steps 10 - 11, line segment L_{TC} that connects center point of the lead top region and centre point of bounding box of C_{Cont_R} is computed and then defines a cone angle $\emptyset_R$ centred to line segment L_{TC} (The magnitude of $\emptyset_R$ depends on the size of the lead top). This is because, it could be observed that the light distribution across the solder joint mainly spreads along a line that connects image centre and centre of the component lead. The results acquired using a sample of 300 solder joints are tabulated in Table 5.5 and some sample images of the successful excess solder detection are depicted in Figure 5.22. It could be observed that erroneous detection of excess solder was due to poor illumination.

Table 5.5: Successful and erroneous detection rate of proposed algorithm for the detection of excess solder on a solder joint

Successful Detection Rate on soldered joints with excess solder	**Erroneous Detection Rate on good solder joints**
94.2%	5.8%

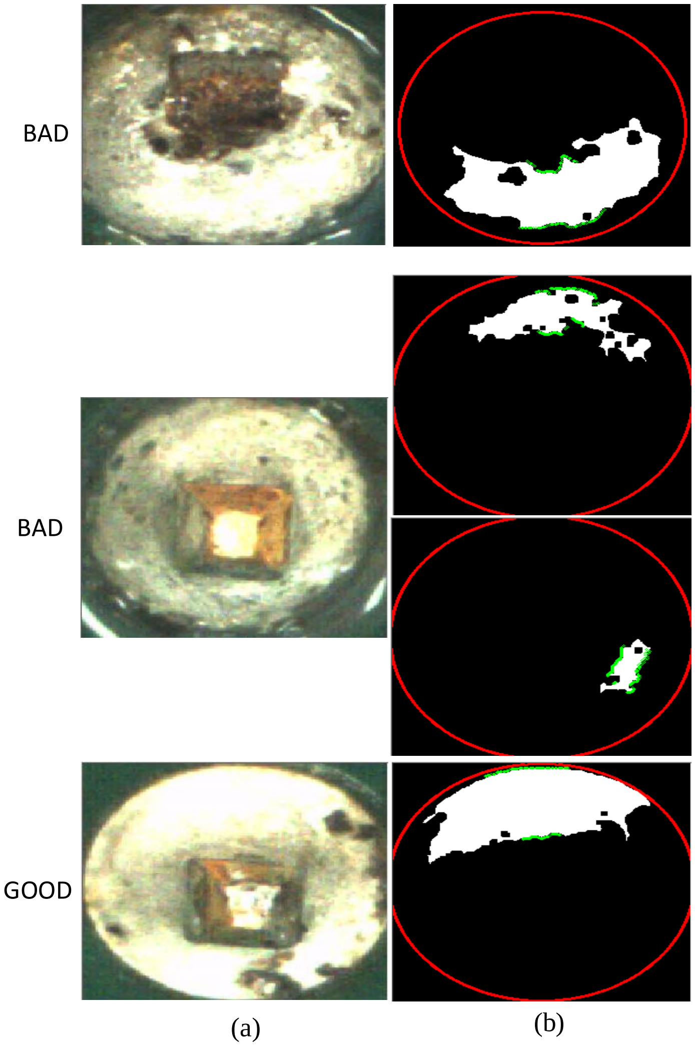


Figure 5.22: Connected regions and outer boundary pixels for excess solder detection. (a) Solder joint. (b) Outer and inner boundary pixels of the connected regions (green) and the solder joint boundary (red)

The implementation of algorithms that can be used to precisely identify four major types of soldering defects inside a THT solder joint have been described in Chapter 5. The robustness and accuracy of each algorithm have been tested under various illumination conditions and practical imperfections like applied flux, burnt flux residues, illuminated PCB tracks and shadows. According to the analysis results presented in Chapters 2-5, the implemented AOI system provides a very precise and robust platform to classify quality of THT solder joints. Figure 5.23 illustrates few examples that present the overall functionality of the implemented AOI system which operates according to flow chart depicted in Figure 1.1.

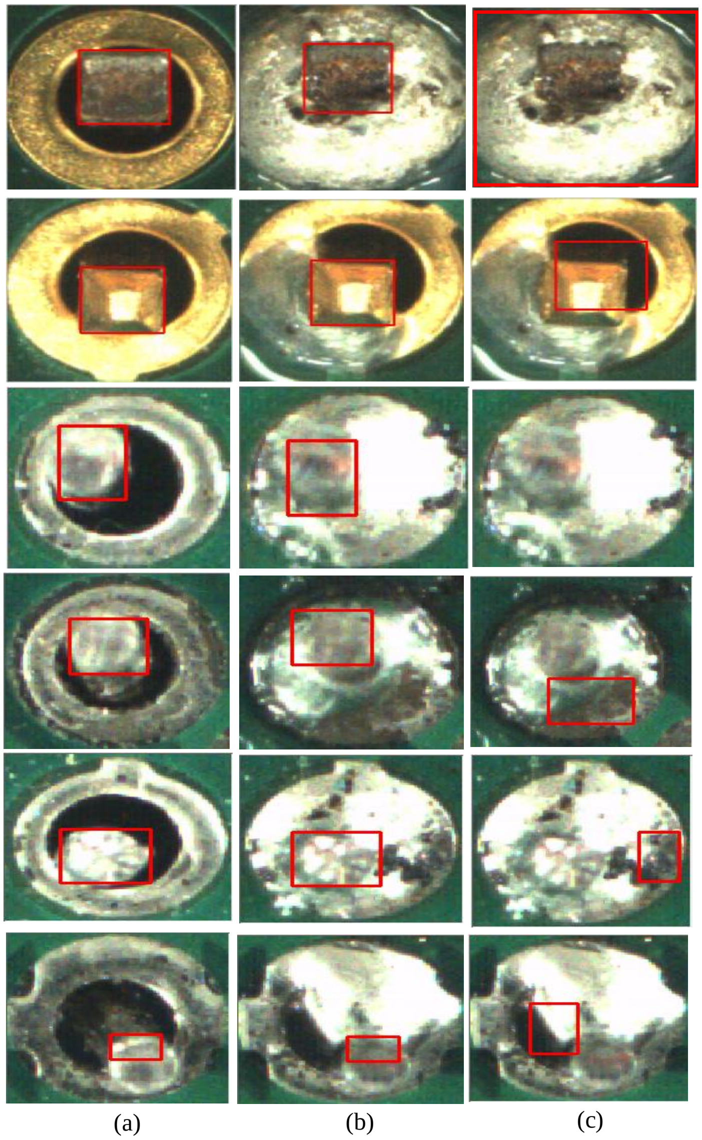


Figure 5.23: Performance of the AOI system for solder quality classification. (a) Solder pad region and detected THT component lead top shown in red box. (b) Localized THT component lead top after soldering shown in red box. (c) Identified defects shown in red box

In manual soldering, usually 8 seconds is allowed for a THT joint and on the average of 6 seconds is needed for quality inspection. Hence, the robotic soldering system with AOI should complete the soldering and quality inspection within 14 seconds. The analysis which was carried in this research, shows that the average positioning and soldering time needed for robotic soldering system is around 8 seconds. Therefore, the automatic quality inspection must be carried out within 6 seconds. The current implementation of the vision system from solder pad segmentation, real time position calibration, component availability verification, lead localization and quality classification consumes 12 seconds. The largest processing time is acquired by the graph-cut based image segmentation algorithm. However, the operation of this algorithm can be made at least 5 times faster than the current speed by implementing it on a powerful Graphical Processing Unit (GPU). Further, in some critical applications it is necessary to identify the fillet angle of the solder joint. As the solder joint is inspected in orthogonal view, the implemented AOI system is not capable of identifying the fillet angle. A detailed explanation is provided about the additional countermeasures taken to solve some practical problems encountered during the THT soldering system operation in Chapter 8.

In Chapters 2-5, a detailed overview has been presented on how the implemented vision system functions covering the entire soldering process carried out by a soldering robotic system. According to the results acquired during the implementation, it could be proved that the implemented AOI system is a viable option to be integrated with a soldering robotic system. The application of CAD system has been discussed in several occasions in the previous chapters. Chapter 6 informatively presents the implementation of the CAD system, which has been integrated with the THT soldering system.

6. Computer Aided Design Tool

PCBs are designed on specialized Electronic Design Automation (EDA) platforms like Altium, Eagle, Mentor etc. or CAD systems like AutoCAD, Solid works etc. Each of these systems outputs PCB fabrication data formatted into a specific format (CAD File) that is unique to a particular system. This data is generally used for the fabrication of a PCB. A CAD file contains all the necessary information to manage the manufacturing process of a particular PCB effectively like component orientation, assembling layer, physical dimensions of components and their pads, netlist information, locations of test pads, vias and mounting holes. This information can be easily used by an automated system to provide an efficient service than a manual worker. Therefore, the extraction of these details is very useful for an automated system like the developed system in this research.

6.1 CAD System Implementation

Different EDA and CAD systems produce different types of CAD file formats like Cadence, Cadif, Gencad, PADS, P-Cad etc. Even though some CAD systems are commercially available they are application specific and cannot provide all the information required by this application like solder pad/hole dimension, width of PCB tracks, easy selection of components to be soldered directly on the GUI and the heights of nearby components. One of the core concepts in this design is to have a skill independent operation. Hence such CAD systems require expertise knowledge of operation, they are not suitable for this application. The developed CAD system provides,

- Easy selection of the components to be soldered through an interactive GUI
- Efficient orientation adjustment of the PCB without repeated teaching of locations
- Automatic extraction of physical parameters of solder pads for the AOI operation
- Automatic computation of pre-heat time based on PCB track width and pad size
- Easy definition of fiducial points from multiple options like vias, pads,.. etc.
- Automatic calculation of amount of solder for a solder pad based on its size

- Automatic definition of soldering iron touching direction for a solder pad based on the heights of neighbouring objects

Figure 6.1 illustrates the structure of the developed CAD system.

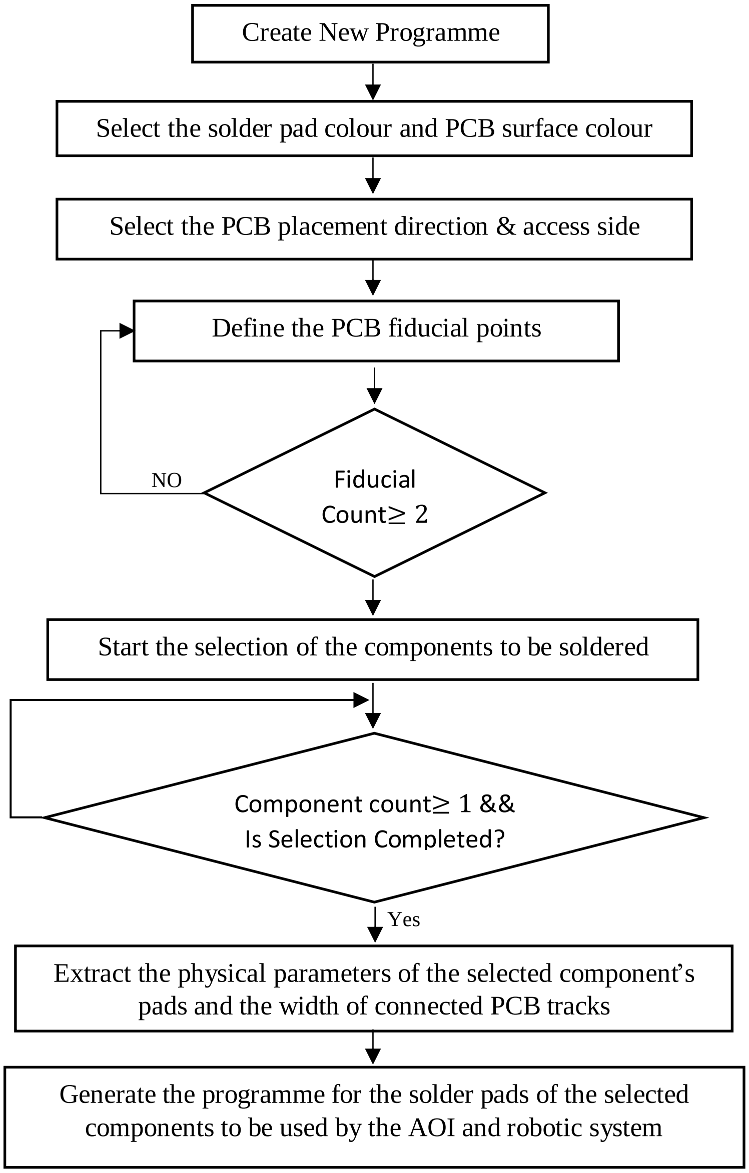
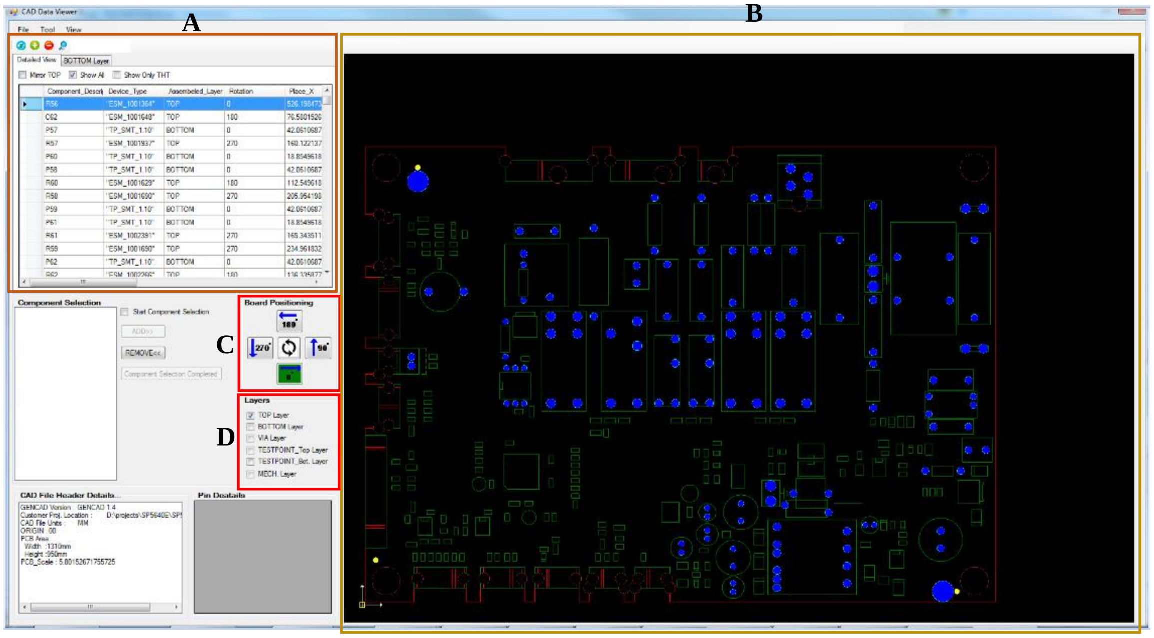

Create New Programme
Select the solder pad colour and PCB surface colour
Select the PCB placement direction & access side
Define the PCB fiducial points
NO
Fiducial Count≥ 2
Start the selection of the components to be soldered
Component count≥ 1 && Is Selection Completed?
Yes
Extract the physical parameters of the selected component's pads and the width of connected PCB tracks
Generate the programme for the solder pads of the selected components to be used by the AOI and robotic system

Figure 6.1: Structure of the implemented CAD tool

This CAD system provides an elegant GUI that makes the system skill independent as illustrated in Figure 6.2. Figure 6.3 illustrates how the geometrical information of a component pin is formatted in this CAD system.



□ Mirror TOP ☑ Show All □ Show Only THT

	Component_Descri	Device_Type	Assembeled_Layer	Rotation	Place_X
▸	R56	"ESM_1001364"	TOP	0	526.190473
	C62	"ESM_1001648"	TOP	180	76.5801526
	P57	"TP_SMT_1.10"	BOTTOM	0	42.0610687
	R57	"ESM_1001937"	TOP	270	160.122137
	P60	"TP_SMT_1.10"	BOTTOM	0	18.8549618
	P58	"TP_SMT_1.10"	BOTTOM	0	42.0610687
	R60	"ESM_1001629"	TOP	180	112.549618
	R58	"ESM_1001690"	TOP	270	205.954198
	P59	"TP_SMT_1.10"	BOTTOM	0	42.0610687
	P61	"TP_SMT_1.10"	BOTTOM	0	18.8549618
	R61	"ESM_1002391"	TOP	270	165.343511
	R59	"ESM_1001690"	TOP	270	234.961832
	P62	"TP_SMT_1.10"	BOTTOM	0	42.0610687
	R62	"ESM_1002266"	TOP	180	136.335877

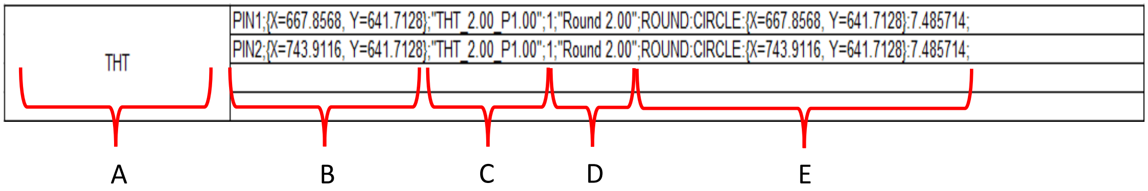
□ Start Component Selection

☑ TOP Layer
□ BOTTOM Layer
□ VIA Layer
□ TESTPOINT_Top Layer
□ TESTPOINT_Bot. Layer
□ MECH. Layer

```
GENCAD Version : GENCAD 1.4
Customer Proj. Location :      D:\projects\SP5640E\SP
CAD File Units :    MM
ORIGIN : 00
PCB Area
  Width  :1310mm
  Height :950mm
PCB_Scale : 5.80152671755725
```

Legend	Description
A	The extracted information related to each component on the PCB (locations and physical dimensions of THT component pads,vias, test points, mechanical holes and width of connected PCB tracks)
B	The main graphical visualization area that contains the drawn PCB
C	The area for PCB orientation definition
D	The area that facilitates the user to switch between different layers of the PCB

Figure 6.2: GUI of the implemented CAD tool



```
PIN1;{X=667.8568, Y=641.7128};"THT_2.00_P1.00";1;"Round 2.00";ROUND:CIRCLE:{X=667.8568, Y=641.7128}:7.485714;
PIN2;{X=743.9116, Y=641.7128};"THT_2.00_P1.00";1;"Round 2.00";ROUND:CIRCLE:{X=743.9116, Y=641.7128}:7.485714;
```

Legend	Description
A	Component type (Whether SMD or THT component)
B	Pin designator and its actual location
C	The description is given in the format <Pad Type>**_**<Actual Pad Diameter>**_P**<Actual Drill-Hole Diameter>". All the dimensions are in mm
D	Pad geometric type. It means whether it is circular, ellipsoidal or rectangular. The notation "Round 2.00". It means this pad is a circular pad with the diameter of 2mm.
E	Summary of the information provided in A, B, C and D. Here the dimensions are denoted in number of pixels

Figure 6.3: The formatted geometrical information related to the pins of a particular component

The PCB is up-scaled or downscaled based on its dimensions in a way that fit to the visualization area (‘B’ in Figure 6.2). Therefore, the dimensions provided in ‘*E*’ are in number of pixels, not in mm. Figure 6.4 illustrates few examples of PCB assembling layouts. Blue colour circles denote the THT solder pads. This feature provides a clear picture to the user about the actual locations of the THT components and their orientations. Hence, it eliminates the requirement of the real PCB to develop the programme for soldering. The region ‘C’ in Figure 6.2 allows user to define the orientation of the PCB. Figure 6.5 illustrates the alternated orientations of the PCBs shown in Figure 6.4. Figure 6.6 illustrates the several layers of the PCB illustrated in Figure 6.4 (b).

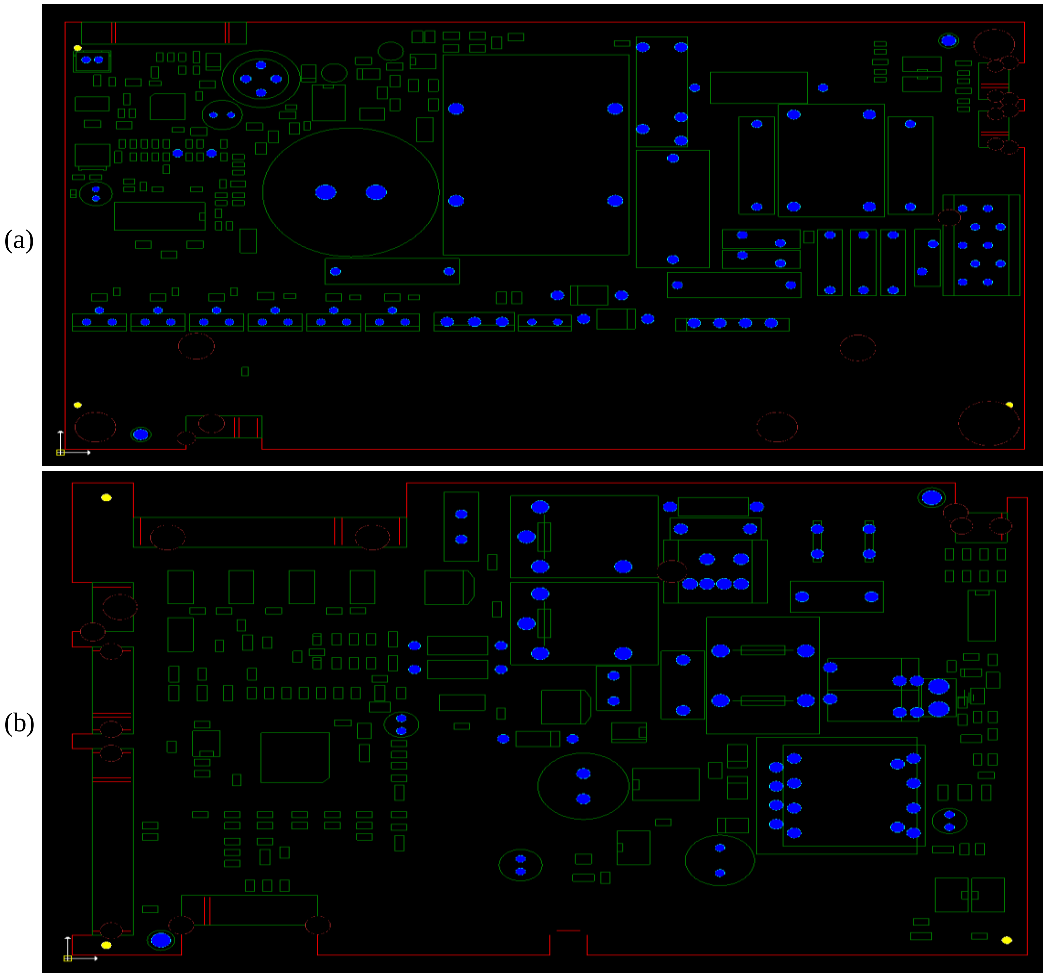
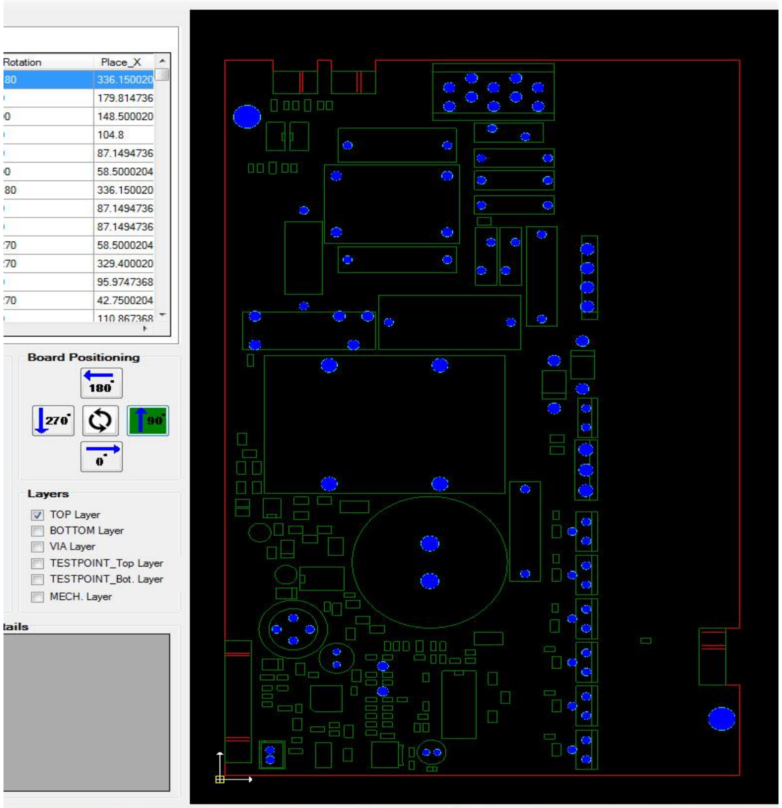


Figure 6.4: Drawn PCB layout on the CAD system GUI based on the information acquired from the respective CAD file

(a)



Rotation	Place_X
80	336.150020
	179.814736
0	148.500020
	104.8
	87.1494736
0	58.5000204
80	336.150020
	87.1494736
	87.1494736
70	58.5000204
70	329.400020
	95.9747368
70	42.7500204
	110.867368

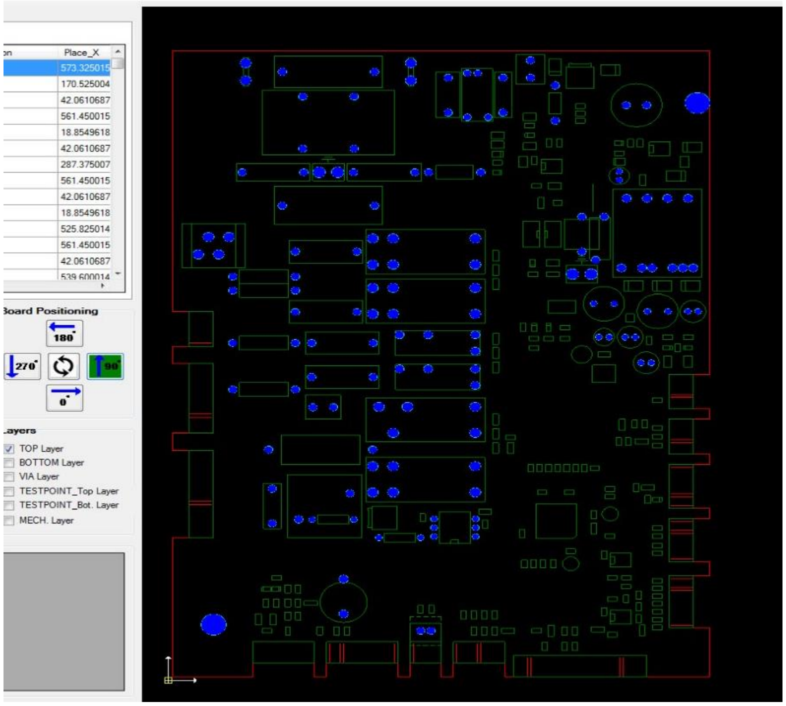
Board Positioning

Layers

☑ TOP Layer
□ BOTTOM Layer
□ VIA Layer
□ TESTPOINT_Top Layer
□ TESTPOINT_Bot. Layer
□ MECH. Layer

tails

(b)



on	Place_X
	573.325015
	170.525004
	42.0610687
	561.450015
	18.8549618
	42.0610687
	287.375007
	561.450015
	42.0610687
	18.8549618
	525.825014
	561.450015
	42.0610687
	539.600014

oard Positioning

ayers

☑ TOP Layer
□ BOTTOM Layer
□ VIA Layer
□ TESTPOINT_Top Layer
□ TESTPOINT_Bot. Layer
□ MECH. Layer

Figure 6.5: Multiple orientations of the PCB layouts

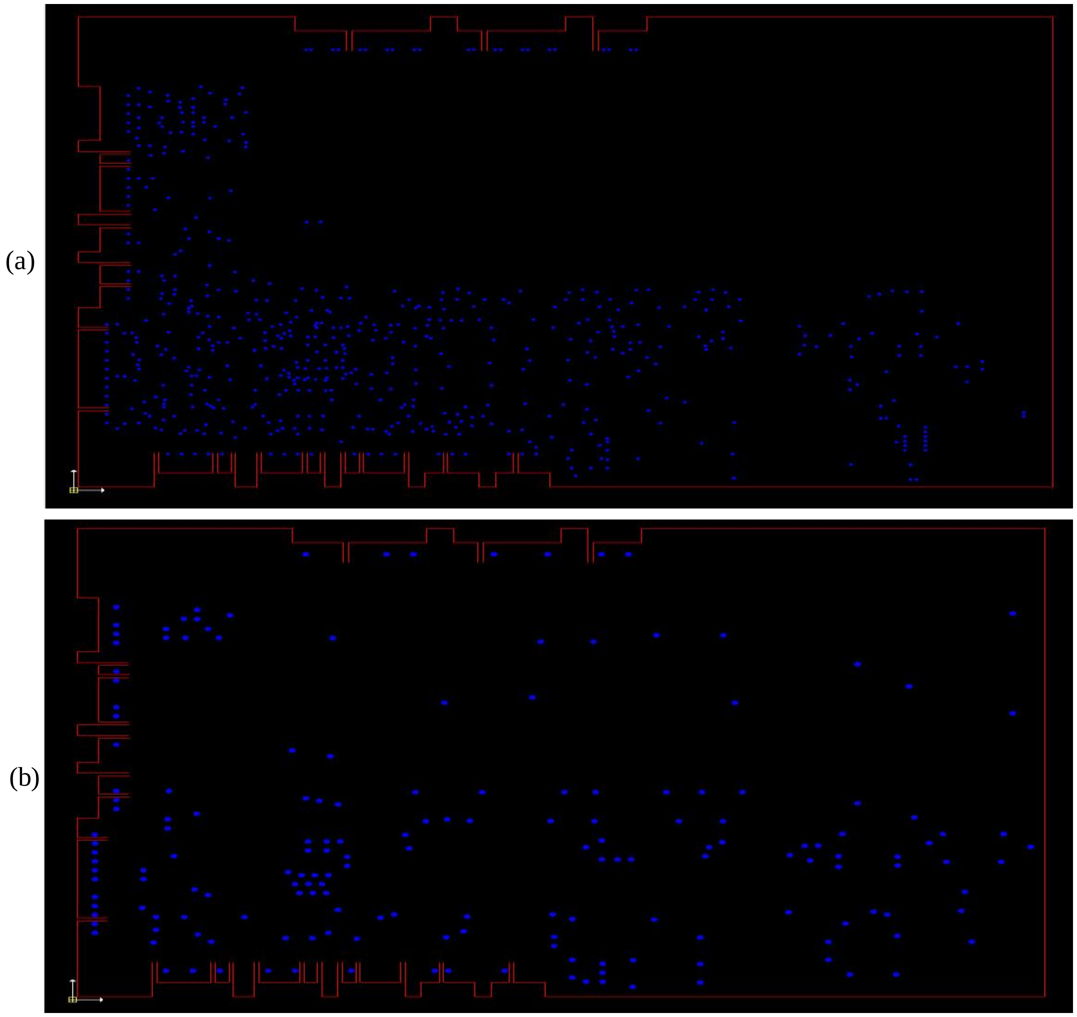


Figure 6.6: Separate layers of the drawn PCB. (a). Via layer. (b). Test point layer

Section 6.2 describes how a soldering programme is generated for a new PCB using this CAD tool.

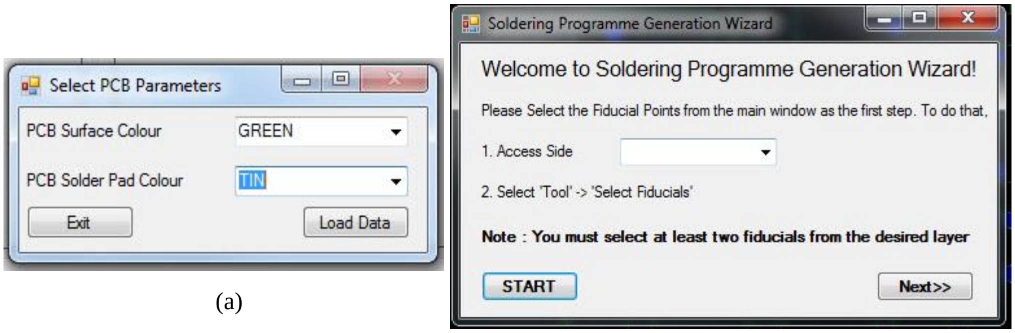
6.2 Generating a Programme for a New PCB

A New programme must be generated for each different PCB type based on respective CAD file. The generated programme includes,

- information on PCB fiducials;
- geometrical information about the pads to be soldered;
- the accessible direction of the solder iron to touch the pad;

- Pre-heat time for a solder pad based on its area and size of the connected track;
- Amount of solder that must be applied on a particular solder pad based on its size.

As illustrated in Figure 6.1, the initial steps of soldering programme generation include the definition of PCB access side and colour for PCB surface and pad. Figure 6.7 illustrates how the user is automatically directed by the system software to define these parameters.



Select PCB Parameters

PCB Surface Colour: GREEN

PCB Solder Pad Colour: TIN

(a)

Soldering Programme Generation Wizard

Welcome to Soldering Programme Generation Wizard!

Please Select the Fiducial Points from the main window as the first step. To do that,

1. Access Side

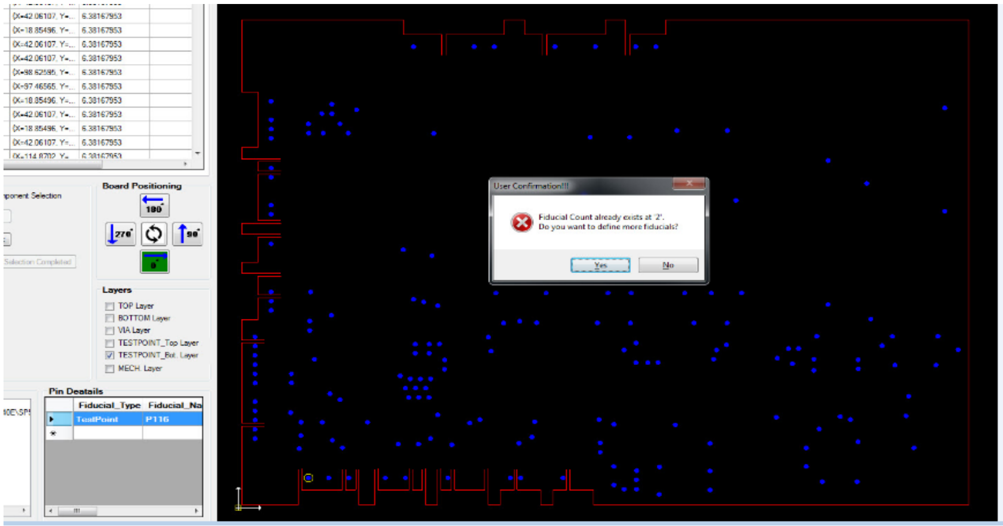
2. Select 'Tool' -> 'Select Fiducials'

Note : You must select at least two fiducials from the desired layer

(b)

Figure 6.7: User confirmation windows that accepts the user inputs to define PCB surface colour, solder pad colour and PCB access side. (a). User inputs are requested to define PCB surface colour and solder pad colour. (b). User input is requested to define the PCB access side

Next, the PCB fiducial points can be defined either from via, test point, mechanical or from the fiducial layers. This fiducial definition process is illustrated in Figure 6.8.



(X=42.06107, Y=...	6.38167953
(X=18.85496, Y=...	6.38167953
(X=42.06107, Y=...	6.38167953
(X=42.06107, Y=...	6.38167953
(X=98.62595, Y=...	6.38167953
(X=97.46565, Y=...	6.38167953
(X=18.85496, Y=...	6.38167953
(X=42.06107, Y=...	6.38167953
(X=18.85496, Y=...	6.38167953
(X=42.06107, Y=...	6.38167953
(X=114.8702, Y=...	6.38167953

Layers

□ TOP Layer
□ BOTTOM Layer
□ VIA Layer
□ TESTPOINT_Top Layer
☑ TESTPOINT_Bot. Layer
□ MECH. Layer

Pin Deatails

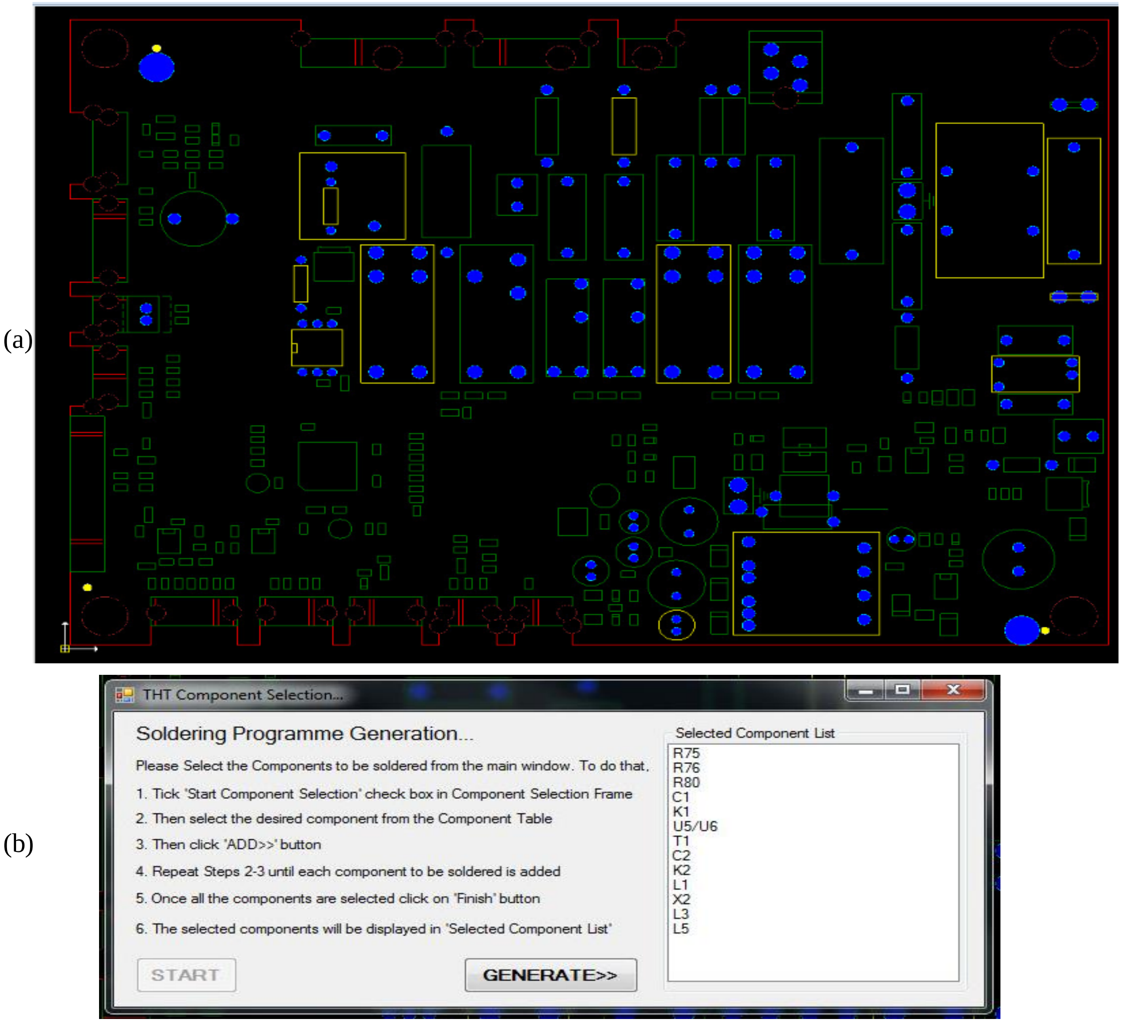
Fiducial_Type	Fiducial_Na
TestPoint	P116

User Confirmation!!!

Fiducial Count already exists at '2'.
Do you want to define more fiducials?

Figure 6.8: Defined fiducial point from the via layer

Eventually, the user is asked to select and add the components for soldering using the GUI (‘A’ in Figure 6.2). Once this done, the CAD system automatically highlight the location of that component as illustrated in Figure 6.9 (a).



Soldering Programme Generation...

Please Select the Components to be soldered from the main window. To do that,

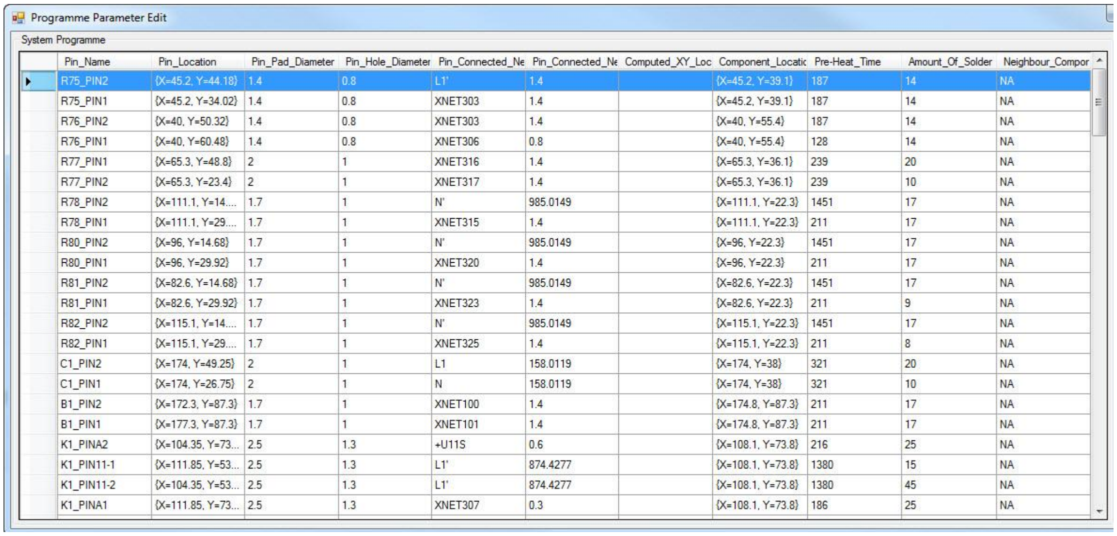
1. Tick 'Start Component Selection' check box in Component Selection Frame
2. Then select the desired component from the Component Table
3. Then click 'ADD>>' button
4. Repeat Steps 2-3 until each component to be soldered is added
5. Once all the components are selected click on 'Finish' button
6. The selected components will be displayed in 'Selected Component List'

Selected Component List

R75
R76
R80
C1
K1
U5/U6
T1
C2
K2
L1
X2
L3
L5

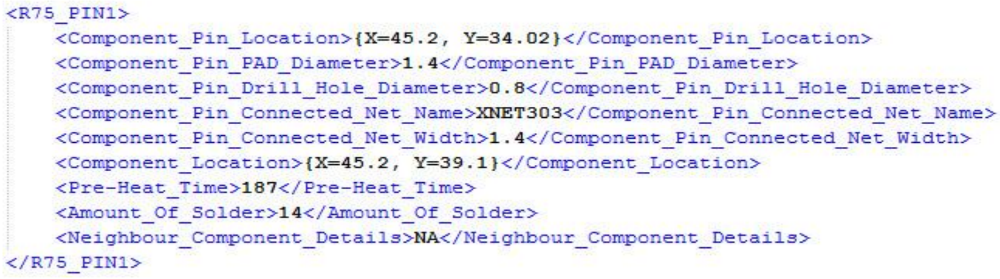
Figure 6.9: Illustration of selected components on the GUI of CAD tool. (a). Illustration of selected components on the GUI with boundary highlighted by yellow. (b). List of selected components

Finally, the programme for soldering can be generated using the ‘Generate’ button as illustrated in Figure 6.9 (b). Figure 6.10 (a) illustrates the GUI for accessing the generated programme for the solder pads of the selected components illustrated in Figure 6.9 (b). This programme is stored as a xml file as illustrated in Figure 6.10 (b).



Pin_Name	Pin_Location	Pin_Pad_Diameter	Pin_Hole_Diameter	Pin_Connected_Ne	Pin_Connected_Ne	Computed_XY_Loc	Component_Locatic	Pre-Heat_Time	Amount_Of_Solder	Neighbour_Compor
R75_PIN2	{X=45.2, Y=44.18}	1.4	0.8	L1'	1.4		{X=45.2, Y=39.1}	187	14	NA
R75_PIN1	{X=45.2, Y=34.02}	1.4	0.8	XNET303	1.4		{X=45.2, Y=39.1}	187	14	NA
R76_PIN2	{X=40, Y=50.32}	1.4	0.8	XNET303	1.4		{X=40, Y=55.4}	187	14	NA
R76_PIN1	{X=40, Y=60.48}	1.4	0.8	XNET306	0.8		{X=40, Y=55.4}	128	14	NA
R77_PIN1	{X=65.3, Y=48.8}	2	1	XNET316	1.4		{X=65.3, Y=36.1}	239	20	NA
R77_PIN2	{X=65.3, Y=23.4}	2	1	XNET317	1.4		{X=65.3, Y=36.1}	239	10	NA
R78_PIN2	{X=111.1, Y=14....	1.7	1	N'	985.0149		{X=111.1, Y=22.3}	1451	17	NA
R78_PIN1	{X=111.1, Y=29....	1.7	1	XNET315	1.4		{X=111.1, Y=22.3}	211	17	NA
R80_PIN2	{X=96, Y=14.68}	1.7	1	N'	985.0149		{X=96, Y=22.3}	1451	17	NA
R80_PIN1	{X=96, Y=29.92}	1.7	1	XNET320	1.4		{X=96, Y=22.3}	211	17	NA
R81_PIN2	{X=82.6, Y=14.68}	1.7	1	N'	985.0149		{X=82.6, Y=22.3}	1451	17	NA
R81_PIN1	{X=82.6, Y=29.92}	1.7	1	XNET323	1.4		{X=82.6, Y=22.3}	211	9	NA
R82_PIN2	{X=115.1, Y=14....	1.7	1	N'	985.0149		{X=115.1, Y=22.3}	1451	17	NA
R82_PIN1	{X=115.1, Y=29....	1.7	1	XNET325	1.4		{X=115.1, Y=22.3}	211	8	NA
C1_PIN2	{X=174, Y=49.25}	2	1	L1	158.0119		{X=174, Y=38}	321	20	NA
C1_PIN1	{X=174, Y=26.75}	2	1	N	158.0119		{X=174, Y=38}	321	10	NA
B1_PIN2	{X=172.3, Y=87.3}	1.7	1	XNET100	1.4		{X=174.8, Y=87.3}	211	17	NA
B1_PIN1	{X=177.3, Y=87.3}	1.7	1	XNET101	1.4		{X=174.8, Y=87.3}	211	17	NA
K1_PINA2	{X=104.35, Y=73....	2.5	1.3	+U11S	0.6		{X=108.1, Y=73.8}	216	25	NA
K1_PIN11-1	{X=111.85, Y=53....	2.5	1.3	L1'	874.4277		{X=108.1, Y=73.8}	1380	15	NA
K1_PIN11-2	{X=104.35, Y=53....	2.5	1.3	L1'	874.4277		{X=108.1, Y=73.8}	1380	45	NA
K1_PINA1	{X=111.85, Y=73....	2.5	1.3	XNET307	0.3		{X=108.1, Y=73.8}	186	25	NA

(a)



```
<R75_PIN1>
    <Component_Pin_Location>{X=45.2, Y=34.02}</Component_Pin_Location>
    <Component_Pin_PAD_Diameter>1.4</Component_Pin_PAD_Diameter>
    <Component_Pin_Drill_Hole_Diameter>0.8</Component_Pin_Drill_Hole_Diameter>
    <Component_Pin_Connected_Net_Name>XNET303</Component_Pin_Connected_Net_Name>
    <Component_Pin_Connected_Net_Width>1.4</Component_Pin_Connected_Net_Width>
    <Component_Location>{X=45.2, Y=39.1}</Component_Location>
    <Pre-Heat_Time>187</Pre-Heat_Time>
    <Amount_Of_Solder>14</Amount_Of_Solder>
    <Neighbour_Component_Details>NA</Neighbour_Component_Details>
</R75_PIN1>
```

(b)

Figure 6.10: Generated programme for soldering. (a). Physical parameters of each pin of the selected components illustrated in Figure 6.9 (b). (b). Xml file that contains the generated programme data

According to Figure 6.10, it can be seen that the generated programme contains all the information required for a fully automated operation of a soldering robotic system. Except,

- Pre-heat time;
- Amount of solder;
- Neighbour component details for the computation of solder iron touching direction.

Next paragraph describes how the above parameters are computed by the CAD system.

Pre-heat time for a solder pad is imperative to provide a decent solder filling. Because, the degree of heat transferred to the solder pad directly governs the distribution of soldering across it. This parameter is generally controlled by four factors.

- Width of the connected PCB track / copper plane
- Size of the other solder pads connected to the same net
- Size of the component lead
- Fabricated material of the component lead

The CAD system acquires the information related to the first two parameters directly from the respective CAD file and computes the adequate pre-heat time with respect to an empirically acquired model value. The other two parameters are not known by the algorithm at the time of programme generation. However, the impact from these two parameters is not significant when the assembled component is not purely metal (For an example several occasions could be identified during the system implementation where metal lugs are assembled to connect earth cables. Consequently, lugs resulted in a decline of the heat transferred to the pad by absorbing a significant amount of it).

The amount of applied solder is computed based on the size of the solder pad with respect to an empirically acquired model value. The distance and angle of neighbouring components pads are taken into account when they exist in a close proximity to the desired solder pad for determining solder iron touching direction for a solder pad. Once the positioned solder pad is identified and component availability is verified by the AOI, the arm rotates in this computed direction to perform soldering.

A detailed explanation was provided in Chapter 6 on the implementation of CAD system and how it operates. Chapter 7 presents a detailed overview on how the THT soldering system hardware and mechanical structure have been developed in this research.

7. Implementation of 4-Axis THT Soldering System

Soldering robots automate the soldering process by replicating a human operator using an automatic wire feeder and soldering iron for soldering on an automatically positioned solder pad. A wide range of soldering robotic systems are developed by leading manufacturers like Japan Unix, Unitechnologies, Thermaltronics etc. The principle movements performed by such systems are virtually the same. Such a system,

- positions the soldering arm/PCB in *x*-*y* directions
- positions and rotates the soldering arm/PCB in horizontal and vertical planes
- performs the feeding of the required amount of soldering wire automatically
- performs the cleaning of soldering iron tip within defined time intervals
- performs the automatic heat control of the soldering iron tip when it is necessary

The implemented THT soldering system in this research provides additional features like,

- automatic soldering quality classification and rework on defective areas
- automatic verification of the existence of the components to be soldered
- real time calibration of positioning accuracy with the use of AOI system
- automatic computation of the amount of solder required for a solder pad
- automatic definition of soldering iron touching direction for a solder pad
- automatic computation of optimum travelling path for the selected pads
- automatic computation of preheat time based on the width of connected PCB track

Section 7.1 describes the implementation of electronics platform of the soldering system.

7.1 Electronics System Operation of THT Soldering System

The controlling system of the soldering platform controls the each action performed by the THT soldering system. It consists of four individual systems as described in Sections 7.1.1-7.1.4.

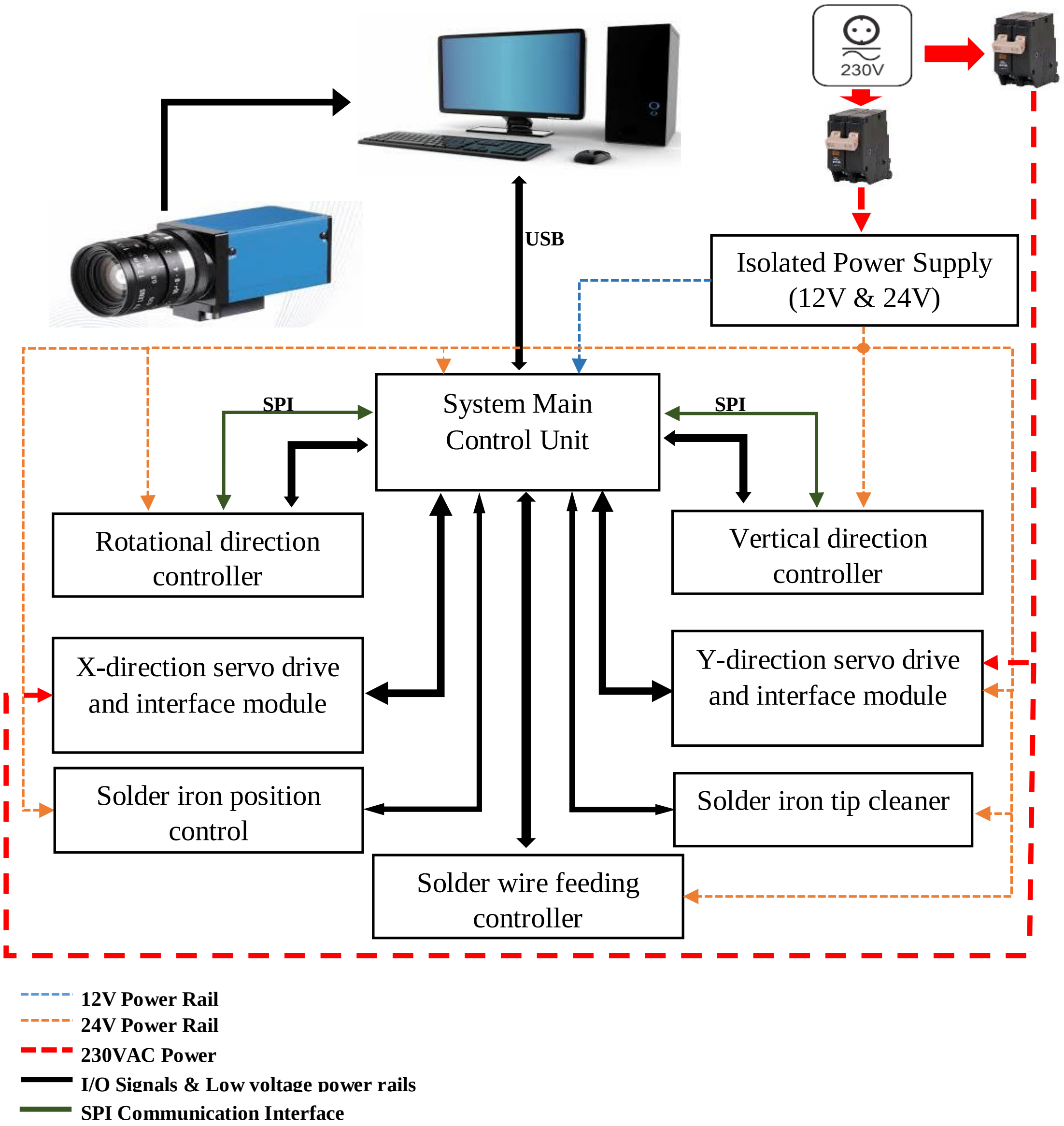

230V
USB
Isolated Power Supply (12V & 24V)
SPI
SPI
System Main Control Unit
Rotational direction controller
Vertical direction controller
X-direction servo drive and interface module
Y-direction servo drive and interface module
Solder iron position control
Solder iron tip cleaner
Solder wire feeding controller
12V Power Rail
24V Power Rail
230VAC Power
I/O Signals & Low voltage power rails
SPI Communication Interface

Figure 7.1: Robotic system operational block diagram

7.1.1 System Main Controlling Unit

This is the main controlling unit of the robotic system. This unit,

- controls the x-y directions position control servo drivers;
- controls the function of vertical and rotational direction control drivers over SPI;
- controls the power management of the robotic system;

- controls the function of wire feeder unit;
- controls the position of the solder iron and solder wire feeder pneumatically;
- performs the power switching to soldering iron and fume extractor;
- communicates with the system software over USB;
- provides error handling on individual power rails of the system;
- controls the robotic system cooling unit and the camera light panel intensity.

The ATXMEGA128A1U microntroller running at 32MHz clock frequency functions as the Module Control Unit (MCU). All the time critical functions are served by the built in Direct Memory Access (DMA) controller of the MCU. The system provides easy diagnostic at power failures since the MCU controls each power rail independently. Figure 7.2 illustrates the PCB layout of the main controlling unit.

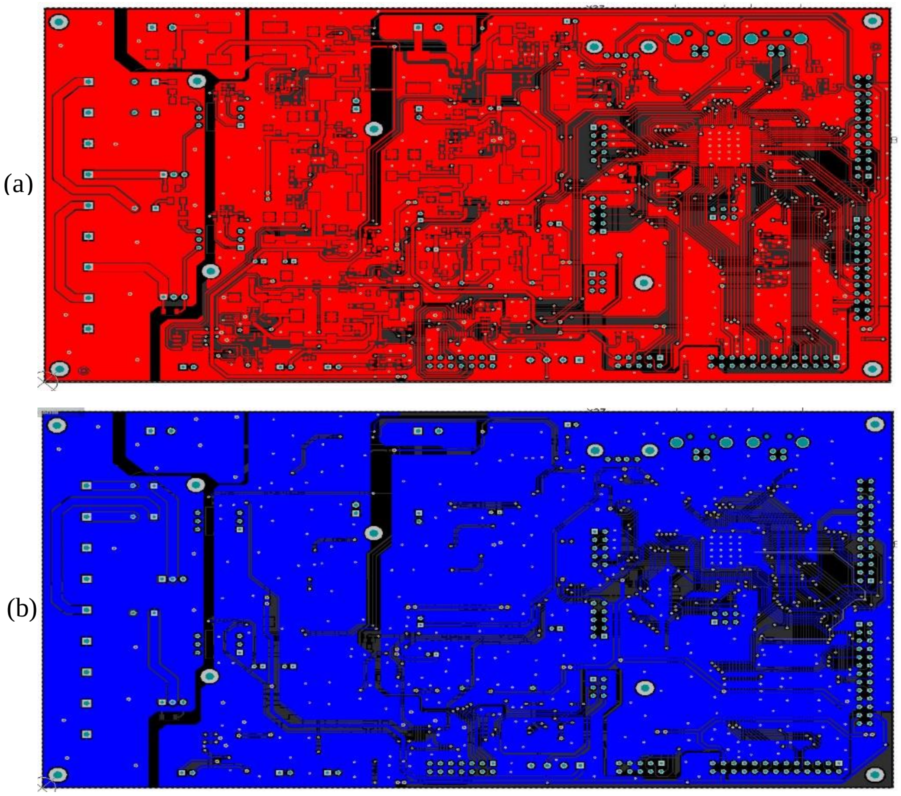


Figure 7.2: PCB layout of the main controlling unit. (a). Top layer (b). Bottom layer

The operational block diagram of the main controlling unit is illustrated in Figure 7.3.

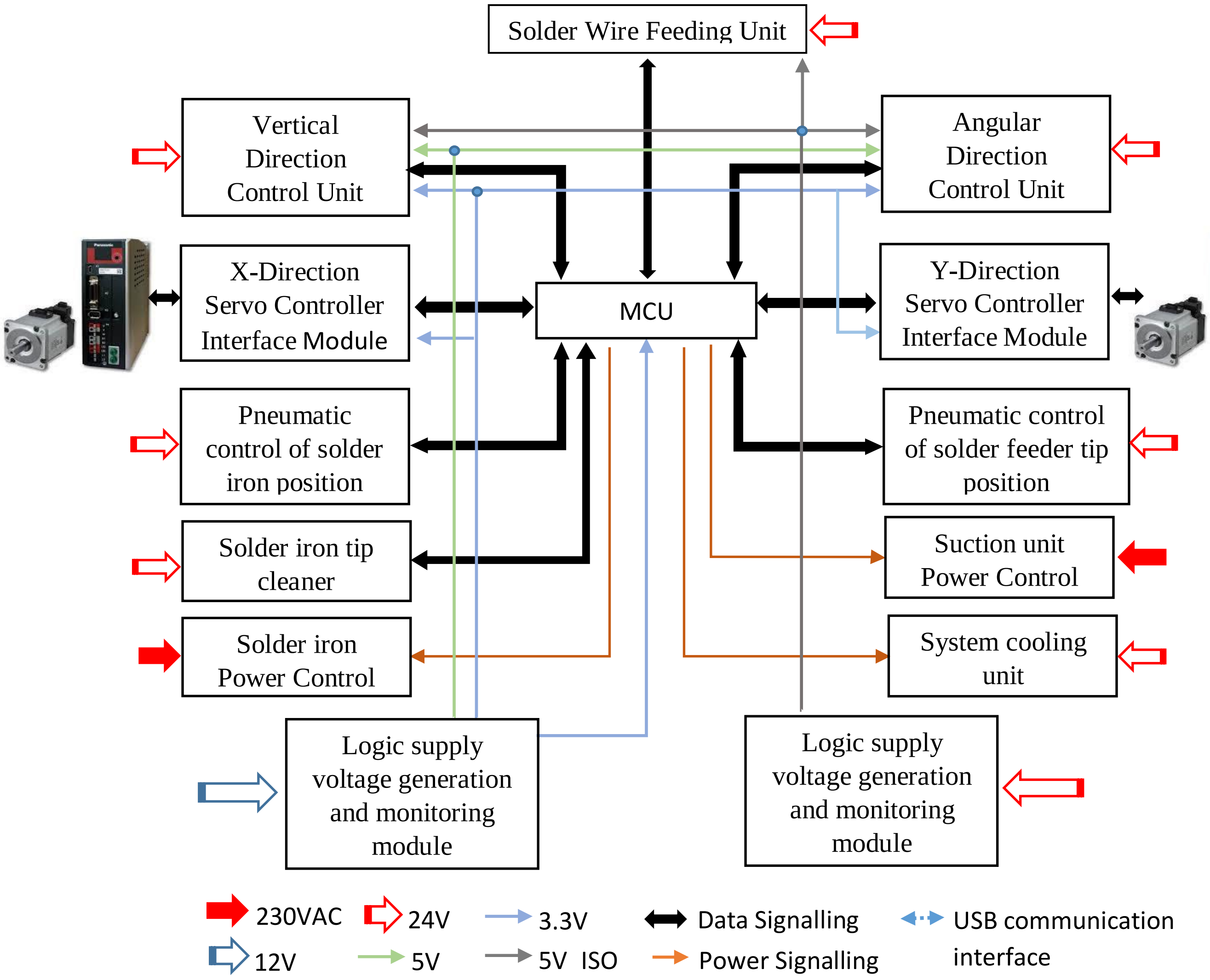

Solder Wire Feeding Unit
Vertical Direction Control Unit
Angular Direction Control Unit
X-Direction Servo Controller Interface Module
MCU
Y-Direction Servo Controller Interface Module
Pneumatic control of solder iron position
Pneumatic control of solder feeder tip position
Solder iron tip cleaner
Suction unit Power Control
Solder iron Power Control
System cooling unit
Logic supply voltage generation and monitoring module
Logic supply voltage generation and monitoring module
230VAC
24V
3.3V
Data Signalling
USB communication interface
12V
5V
5V ISO
Power Signalling

Figure 7.3: Operational block diagram of the main controlling unit

The MCU controls the speed of the servo motors in a T-Profile as illustrated in Figure 7.4.

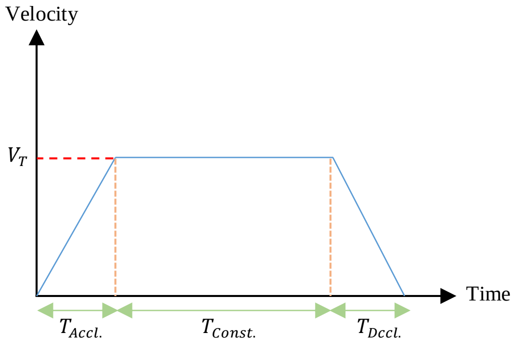


Figure 7.4: Speed control profile for servo drivers

Both $T_{Accl.}$ and $T_{Dccl.}$ are equal in size and it is 20% of the time required to reach the destination. The target velocity (V_T) is defined by the system software relative to the travelling distance as denoted by Eq. (60).

$$V_T = V_L + \left(\frac{V_U - V_L}{D_M}\right) d \tag{60}$$

Where,

V_U, V_L : upper and lower speed values
D_M : maximum travelling distance
d : distance to travel

7.1.2 Vertical Direction Controlling Unit

The primary function of this module is to position the robotic arm in vertical direction according to the instructions provided by the MCU (refer Figure 7.5). This controller can function in two modes of operations.

- Stand-alone mode: *Uses its hardcoded parameters to drive the stepper motor.*
- Configurable mode: *The MCU downloads values for torque, decay mode, blanking time, OFF time, Over Current Protection level and thresholds for stall detection.*

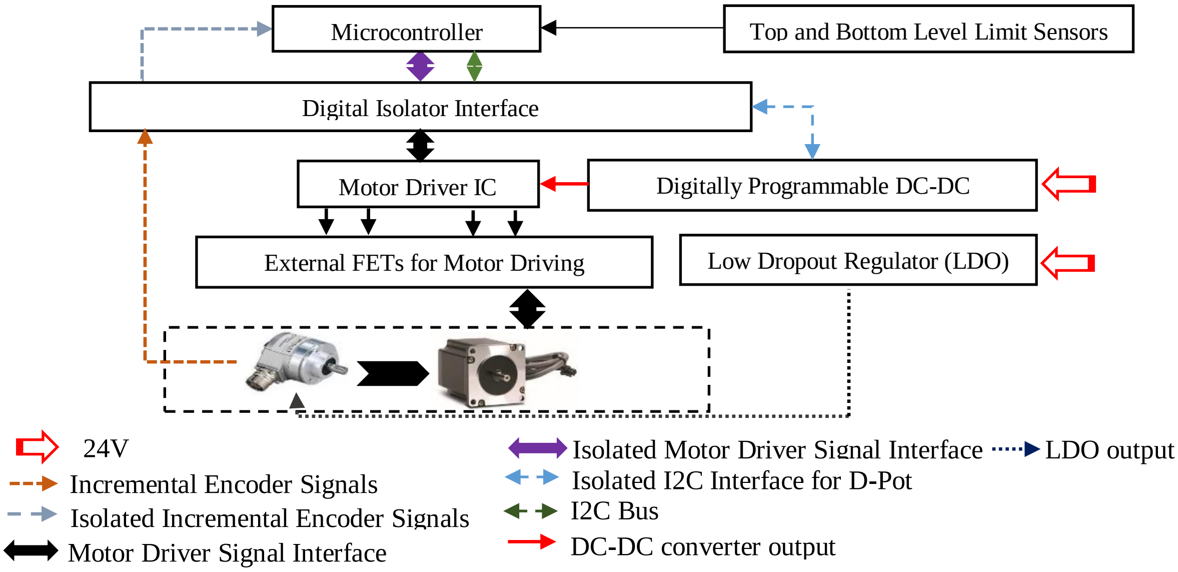

Microcontroller
Top and Bottom Level Limit Sensors
Digital Isolator Interface
Motor Driver IC
Digitally Programmable DC-DC
External FETs for Motor Driving
Low Dropout Regulator (LDO)
24V
Incremental Encoder Signals
Isolated Incremental Encoder Signals
Motor Driver Signal Interface
Isolated Motor Driver Signal Interface
LDO output
Isolated I2C Interface for D-Pot
I2C Bus
DC-DC converter output

Figure 7.5: Operational block diagram of the vertical direction controller

The motor shaft is connected to a gear mechanism (100:1) to mitigate the load introduced by the weight of the soldering arm. Consequently, the motor sees virtually no load during start-up. The speed control profile of the motor is illustrated in Figure 7.6.

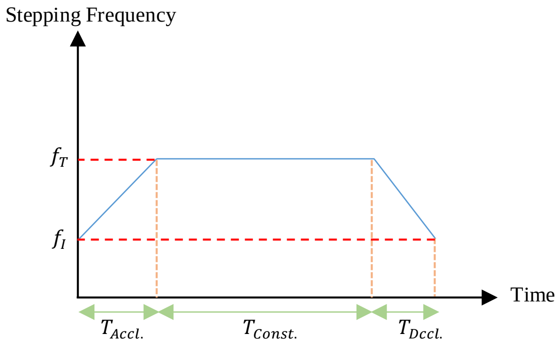


Figure 7.6: Speed loop for stepper motor speed control

Here, f_I is the initial frequency which the motor can start at a given load, T_L. f_I can be computed based on the torque introduced by the load as defined in the particular motor datasheet and the target frequency (f_T) is computed in a similar way as defined in Eq. (60). The width of both $T_{Accl.}$ and $T_{Dccl.}$ are equal to 20% of the total time required to reach the destination. Figures 7.7 illustrate the PCB layout of the designed motor driver.

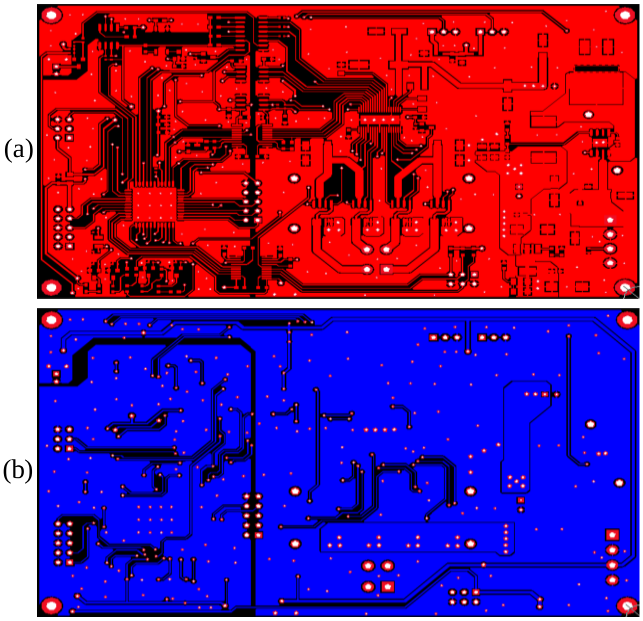


Figure 7.7: PCB layout of vertical direction controlling unit. (a). Top layer (b). Bottom layer

7.1.3 Angular Direction Controlling Unit

The primary function of this module is to position the robotic arm in angular direction according to the instructions provided by the MCU (refer Figure 7.2). This module drives a bipolar stepper motor interfaced with an absolute encoder. This controller also uses a similar speed control procedure as with vertical direction controller (refer Figure 7.6). Figure 7.8 illustrates the PCB layout of the vertical direction controlling unit.

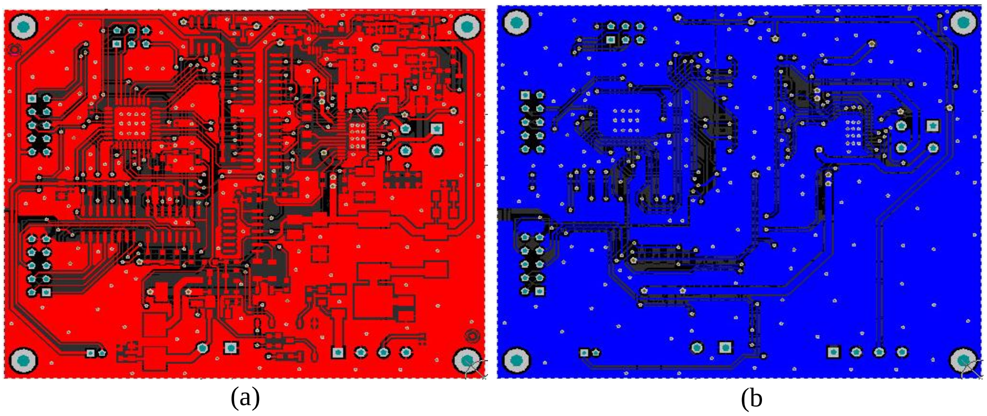

(a) (b

Figure 7.8: PCB layout of angular direction controlling unit. (a). Top layer (b). Bottom layer

Figure 7.9 illustrates the operational block diagram of the angular direction controlling unit.

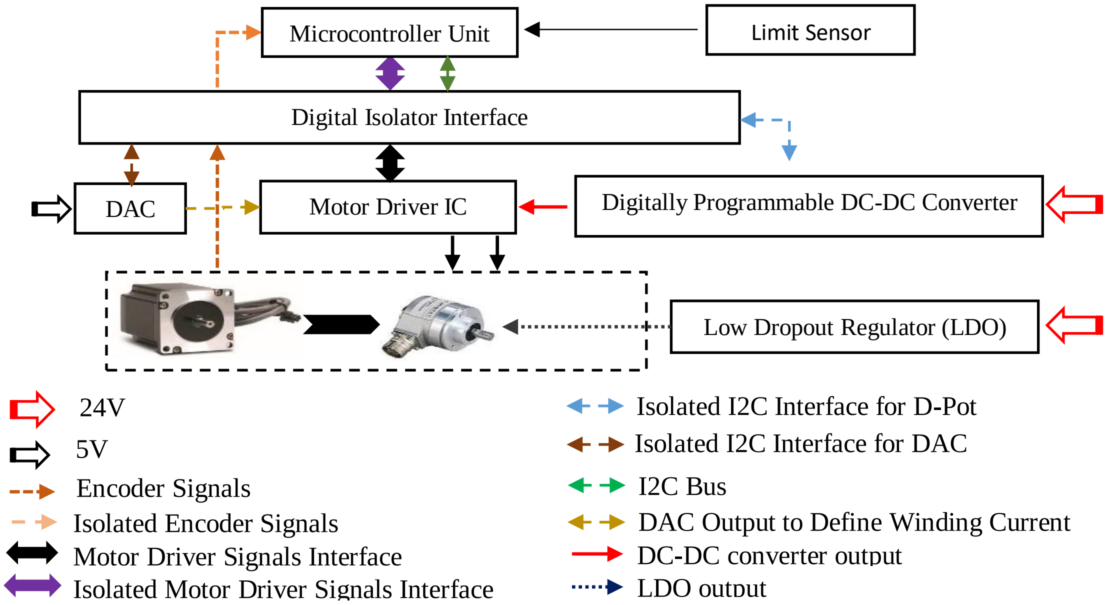

Microcontroller Unit
Limit Sensor
Digital Isolator Interface
DAC
Motor Driver IC
Digitally Programmable DC-DC Converter
Low Dropout Regulator (LDO)
24V
5V
Encoder Signals
Isolated Encoder Signals
Motor Driver Signals Interface
Isolated Motor Driver Signals Interface
Isolated I2C Interface for D-Pot
Isolated I2C Interface for DAC
I2C Bus
DAC Output to Define Winding Current
DC-DC converter output
LDO output

Figure 7.9: Operational block diagram of the angular direction controller

7.1.4 Servo Driver Interface Module

The primary function of this module is to provide an isolation between the system main controller and the servo driver. This further provides 230VAC switching functionality to both servo driver and cooling fan unit. Figure 7.10 illustrates the PCB layout of this module.

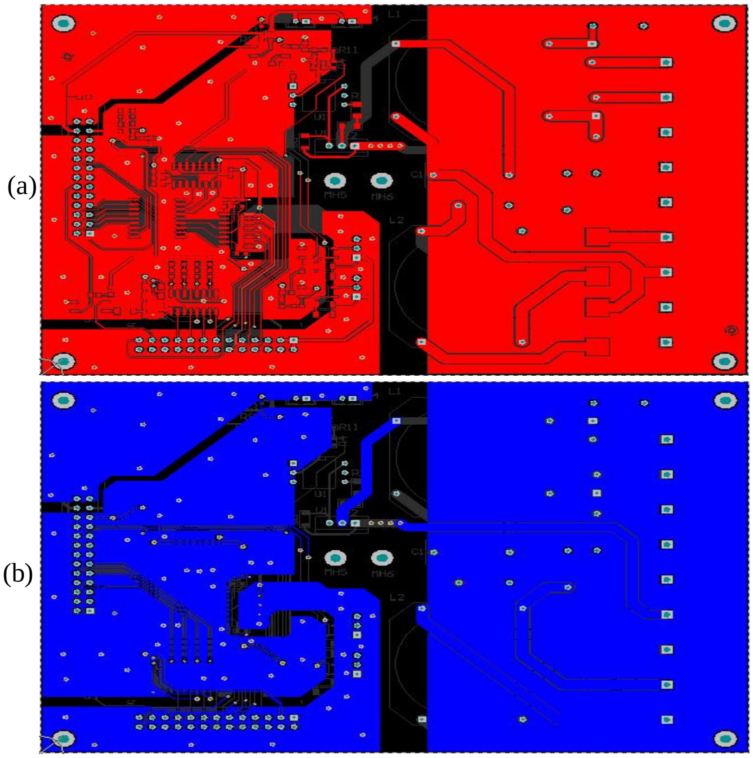


Figure 7.10: PCB layout of servo driver interface module. (a). Top layer (b). Bottom layer

7.2 Implementation of Robotic System

The implementation of the robotic system mechanical structure consists of moving mechanisms that facilitates to operate the system in x-y directions, vertical direction and angular direction. The reliability and the robustness of these mechanisms are vital since the robotic system must operate in um level accuracy. This platform mainly consists of three main mechanisms as illustrated in Figure 7.11.

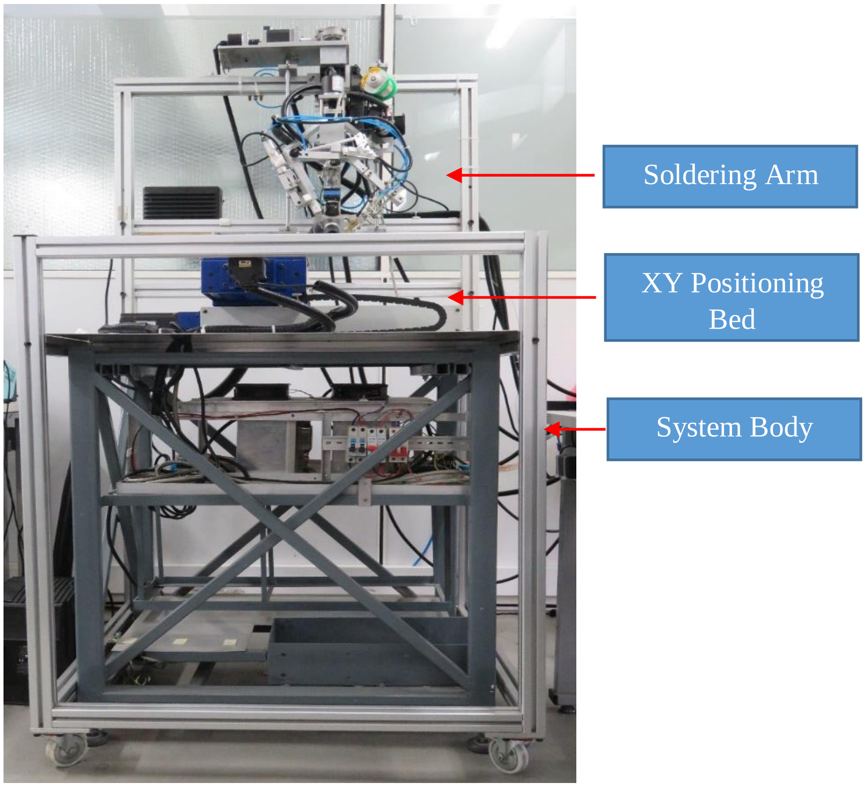


Figure 7.11: THT Soldering System

Sections 7.2.1-7.2.2 describes the implementation of these systems.

7.2.1 XY Positioning System

This unit was purchased during the system implementation. This contains four limit sensors where two sensors per each axis to make sure the safe operation. Figure 7.12 illustrate how the XY positioning system looks like.

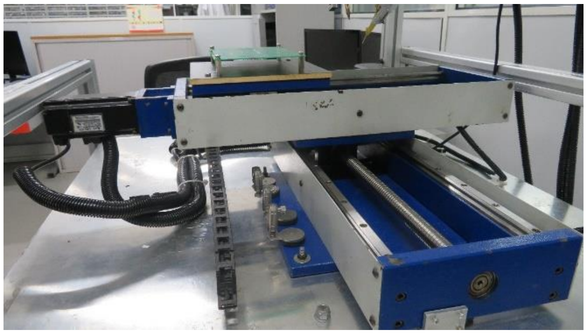


Figure 7.12: *XY* Positioning system

The positioning accuracy of the system is 10μm with a movable distance of 30*cm* in each axis. However, positioning accuracy calibration has to be performed by any robotic system when it is expected to operate continuously. Many commercial systems accomplish this by executing a calibration procedure with the help of a skilled operator, whereas the integrated AOI performs real time positioning calibration as explained in Section 7.2.1.1.

7.2.1.1 Real Time Calibration of Positioning Accuracy

In Section 4.3, it was stated that the AOI performs online positioning accuracy calibration. This is accomplished by computing the distance offset between the centres of camera FOV and identified solder pad. The system automatically corrects this offset, if there is any. Figure 7.13 illustrates an example how this is accomplished in this application.

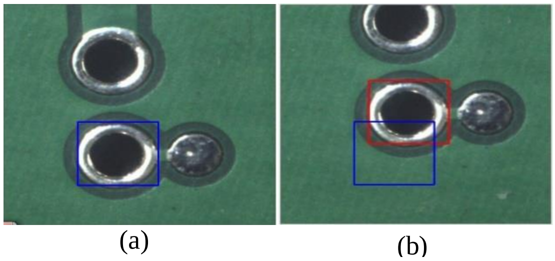


Figure 7.13: Real time correction of positioning offsets. (a). The position of initially detected solder pad (blue). (b). The position of localized solder pad after repositioning (red)

Finally, the vision system finds the new location of the solder pad using template matching considering the initially detected solder pad as the model image. This real time calibration procedure eliminates the requirement of a skilled operator to perform the calibration as with other soldering robotic systems. Further, this procedure makes sure that the soldering iron will exactly touch the solder pad without damaging the PCB surface. Hence, the reliability and robustness of the robotic system can be significantly enhanced.

7.2.1.2 Computation of Optimum Travelling Path

In Section 2.5, it was stated that a precise fiducial verification process enables to compute optimum travelling path for the selected components (refer Chapter 6). This section describes how the optimum travelling path is computed in this application.

Step 1: Compute relative distances to each solder pad from system origin based on the coordinates of the localized fiducials.

Step 2: Arrange the computed distances in ascending order and update the list L_P. Repeat Steps 3-8 until all the pads are covered.

Step 3: Initialize threshold value, TH_S and compute the distances from the solder pad, P_i to other solder pads in L_P.

Step 4: Perform comparison according to Eq. (61).

$$E(P_i) = \left(0 < d(P_{ij}) \leq TH_S\right)?\, d\left(P_{ij}\right) \cup E(P_i) : \infty \cup E(P_i)\ \forall P_i \in L_P \tag{61}$$

Where,

$E(P_i)$: list of distances to points that satisfies Eq. (61)

$d(P_{ij})$: distance between the points P_i and P_j

Step 5: If $|E(P_i)|$==0, then TH_S+= 10 and go to Step 4. Unless, $E(P) = E(P) \cup E(P_i)$ and move to next pad.

Step 6: Repeat Steps 3-5 until all the pads are considered.

Step 7: Finally, the distance matrix $E(P)$ has been computed which has columns and rows equal to number of pads in L_P. The two pads which are not connected in a list $E(P_i)$ contains negative values in the corresponding positions of $E(P)$.

Step 8: Perform Prim's algorithm on $E(P)$ [63] to find the MST, L_{MST}.

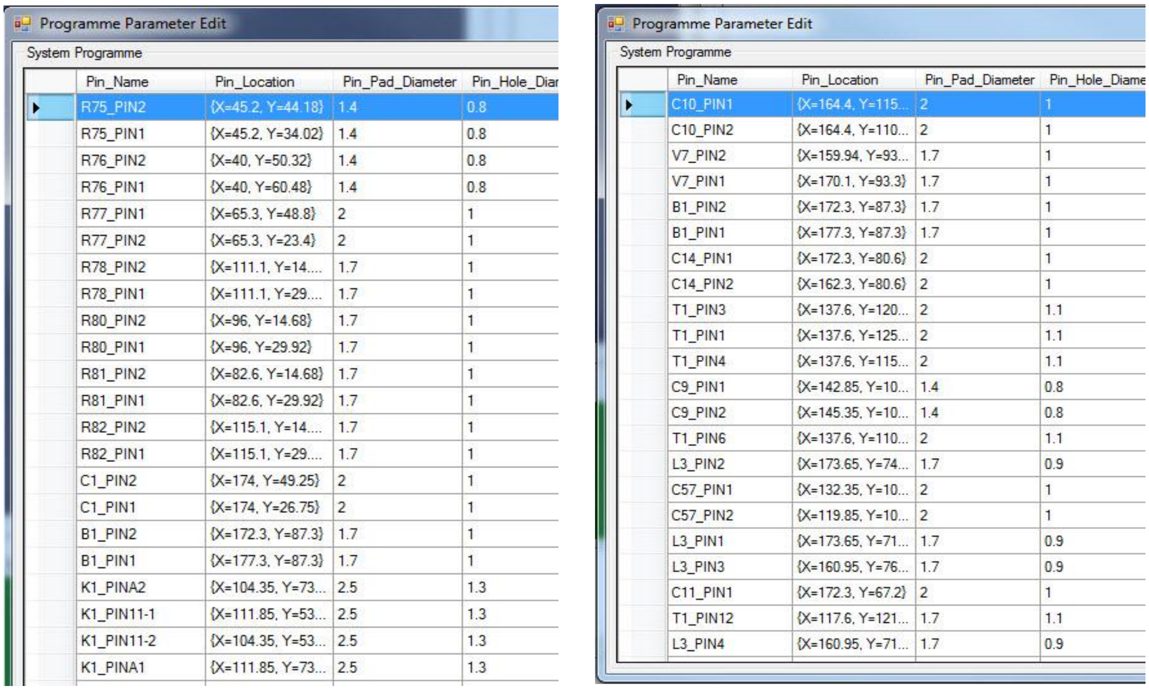
Step 9: Find the pads that are not contained by the MST of $E(P)$. If the count is non-zero, then include them to a new list L_{P_New} and arrange it in ascending order. Else jump to Step 12.

Step 10: Initialize TH_S and double its value. Then follow the similar procedure as defined in Steps 3-8 and combine the two MSTs to form L_{MST}.

Step 11: Find the pads of the same component which are five indexes away in L_{MST}. If any, then the pad that exist far away is re-indexed.

Step 12: L_{MST} holds the optimum travelling path.

Even though many algorithms are available to find the shortest possible path of a connected data set, they did not give a better outcome when they are used alone on this not connected data set. Even though the proposed algorithm might not give the most optimum outcome, it gave a significant improvement over the existing algorithms. Figure 7.14 illustrates how the computed travelling path reorganizes the selected component pads shown in Figure 6.10 (a).



Pin_Name	Pin_Location	Pin_Pad_Diameter	Pin_Hole_Diar
R75_PIN2	{X=45.2, Y=44.18}	1.4	0.8
R75_PIN1	{X=45.2, Y=34.02}	1.4	0.8
R76_PIN2	{X=40, Y=50.32}	1.4	0.8
R76_PIN1	{X=40, Y=60.48}	1.4	0.8
R77_PIN1	{X=65.3, Y=48.8}	2	1
R77_PIN2	{X=65.3, Y=23.4}	2	1
R78_PIN2	{X=111.1, Y=14....	1.7	1
R78_PIN1	{X=111.1, Y=29....	1.7	1
R80_PIN2	{X=96, Y=14.68}	1.7	1
R80_PIN1	{X=96, Y=29.92}	1.7	1
R81_PIN2	{X=82.6, Y=14.68}	1.7	1
R81_PIN1	{X=82.6, Y=29.92}	1.7	1
R82_PIN2	{X=115.1, Y=14....	1.7	1
R82_PIN1	{X=115.1, Y=29....	1.7	1
C1_PIN2	{X=174, Y=49.25}	2	1
C1_PIN1	{X=174, Y=26.75}	2	1
B1_PIN2	{X=172.3, Y=87.3}	1.7	1
B1_PIN1	{X=177.3, Y=87.3}	1.7	1
K1_PINA2	{X=104.35, Y=73...	2.5	1.3
K1_PIN11-1	{X=111.85, Y=53...	2.5	1.3
K1_PIN11-2	{X=104.35, Y=53...	2.5	1.3
K1_PINA1	{X=111.85, Y=73...	2.5	1.3

(a)

Pin_Name	Pin_Location	Pin_Pad_Diameter	Pin_Hole_Diame
C10_PIN1	{X=164.4, Y=115...	2	1
C10_PIN2	{X=164.4, Y=110...	2	1
V7_PIN2	{X=159.94, Y=93...	1.7	1
V7_PIN1	{X=170.1, Y=93.3}	1.7	1
B1_PIN2	{X=172.3, Y=87.3}	1.7	1
B1_PIN1	{X=177.3, Y=87.3}	1.7	1
C14_PIN1	{X=172.3, Y=80.6}	2	1
C14_PIN2	{X=162.3, Y=80.6}	2	1
T1_PIN3	{X=137.6, Y=120...	2	1.1
T1_PIN1	{X=137.6, Y=125...	2	1.1
T1_PIN4	{X=137.6, Y=115...	2	1.1
C9_PIN1	{X=142.85, Y=10...	1.4	0.8
C9_PIN2	{X=145.35, Y=10...	1.4	0.8
T1_PIN6	{X=137.6, Y=110...	2	1.1
L3_PIN2	{X=173.65, Y=74...	1.7	0.9
C57_PIN1	{X=132.35, Y=10...	2	1
C57_PIN2	{X=119.85, Y=10...	2	1
L3_PIN1	{X=173.65, Y=71...	1.7	0.9
L3_PIN3	{X=160.95, Y=76...	1.7	0.9
C11_PIN1	{X=172.3, Y=67.2}	2	1
T1_PIN12	{X=117.6, Y=121...	1.7	1.1
L3_PIN4	{X=160.95, Y=71...	1.7	0.9

(b)

Figure 7.14: Reorganized component pads according to the computed optimum travelling path. (a). The programme before the computation of the optimum travelling path. (b). Reorganized component pads according to the optimum travelling path

7.2.2 Implementation of Soldering Arm

The soldering arm is the vital mechanical part implemented in this system. Its stability, robustness and precise operation controls the overall effectiveness of the soldering system. The implemented soldering arm accommodates a camera, soldering iron, soldering wire feeder and fume extraction unit as illustrated in Figure 7.15. The soldering iron and the wire feeder are pneumatically controlled in the directions illustrated by the purple lines in Figure 7.15.

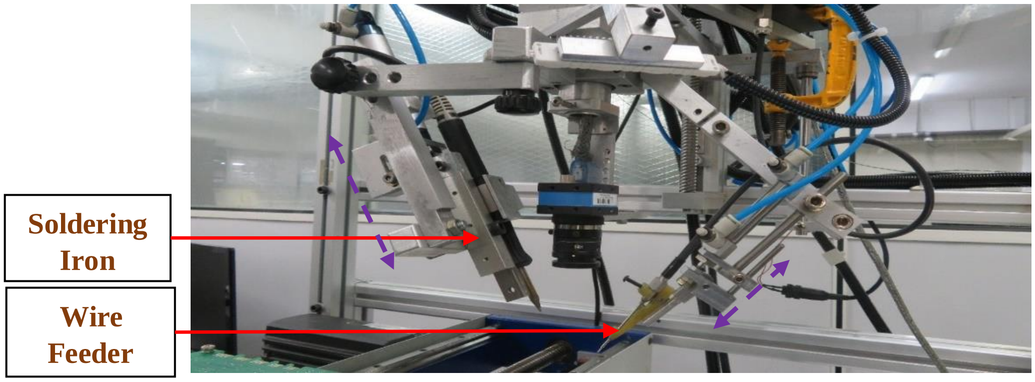
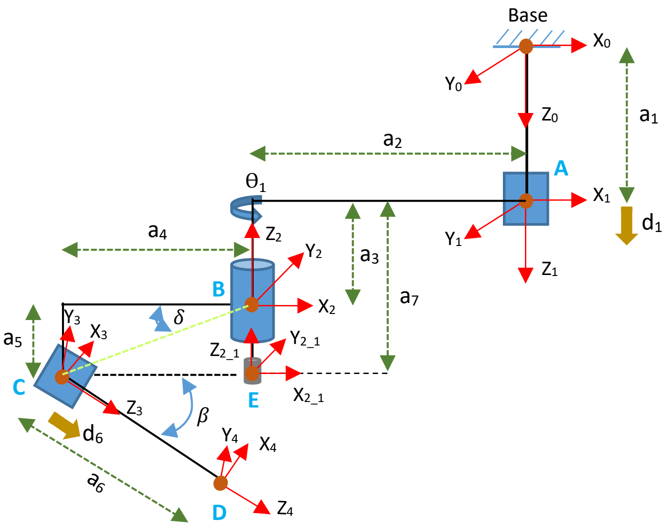


Figure 7.15: Implemented soldering arm

Figure 7.16 and Table 7.1 illustrate the kinematic diagram of the robot arm and DH parameters table respectively.



A	**Prismatic joint that moves arm in vertical direction**
B	**Revolute joint that rotates arm**
C	**Prismatic joint that moves soldering iron**
D	**Soldering iron tip (End-effector 1)**
E	**Camera (End-effector 2)**

Figure 7.16: Kinematic diagram of soldering arm

Table 7.1: DH Parameters for Soldering Arm

Frame	$\boldsymbol{\theta}$	$\boldsymbol{\alpha}$	$\boldsymbol{r}$	$\boldsymbol{d}$
1	0	0	0	$a_1 + d_1$
2	0	180	a_2	a_3
3	$\theta_1 + 90$	$90 + \beta$	a_4	$\sqrt{{a_4}^2 + {a_5}^2}\sin(\beta + \delta)$
4	0	0	0	$a_6 + d_6$

The coordinates for the end-effector (soldering iron tip) can be obtained using homography transformation matrix as illustrated in Eq. (62)-(63). Here $n = 1 \dots 4$.

$$H_n^{n-1} = \begin{bmatrix} \cos\theta_n & -\sin\theta_n\cos\alpha_n & \sin\theta_n\sin\alpha_n & r_n\cos\theta_n \\ \sin\theta_n & \cos\theta_n\cos\alpha_n & -\cos\theta_n\sin\alpha_n & r_n\sin\theta_n \\ 0 & \sin\alpha_n & \cos\alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{62}$$

$$H_4^0 = H_1^0 . H_2^1 . H_3^2 . H_4^3 \tag{63}$$

Any point on the end effector can now be easily computed with reference to base frame, since H_4^0 is known. Table 7.2 illustrates the DH parameters table for camera.

Table 7.2: DH Parameters for Camera

Frame	$\boldsymbol{\theta}$	$\boldsymbol{\alpha}$	$\boldsymbol{r}$	$\boldsymbol{d}$
1	0	0	0	$a_1 + d_1$
2_1	0	180	a_2	a_7

The position of the camera (Frame 2_1) with respect to base frame can be obtained similarly as illustrated in Eq. (63). This can be used to find the transformation between real world coordinates and image coordinates. Figure 7.17 illustrates how this transformation is accomplished in a thin film camera model [64].

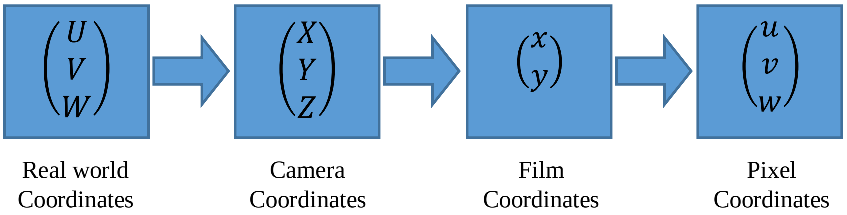

$\begin{pmatrix}U\\V\\W\end{pmatrix}$
$\begin{pmatrix}X\\Y\\Z\end{pmatrix}$
$\begin{pmatrix}x\\y\end{pmatrix}$
$\begin{pmatrix}u\\v\\w\end{pmatrix}$
Real world Coordinates
Camera Coordinates
Film Coordinates
Pixel Coordinates

Figure 7.17: Forward projection of coordinate frames between real world and image coordinate systems

Eq. (64) illustrates the homogeneous form of the pin-hole model of a camera that performs the coordinate transformation between camera coordinates and film coordinates [64].

$$\begin{pmatrix}\tilde{x}\\ \tilde{y}\\ \tilde{z}\end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0\\ 0 & f & 0 & 0\\ 0 & 0 & 1 & 0\end{pmatrix}\begin{pmatrix}X\\ Y\\ Z\\ 1\end{pmatrix} \tag{64}$$

Where,

$x = \tilde{x}/Z$, $y = \tilde{y}/Z$ and f is the focal length of the camera.

Figure 7.18 illustrates projection of an object inside the camera at a focal length *f*.

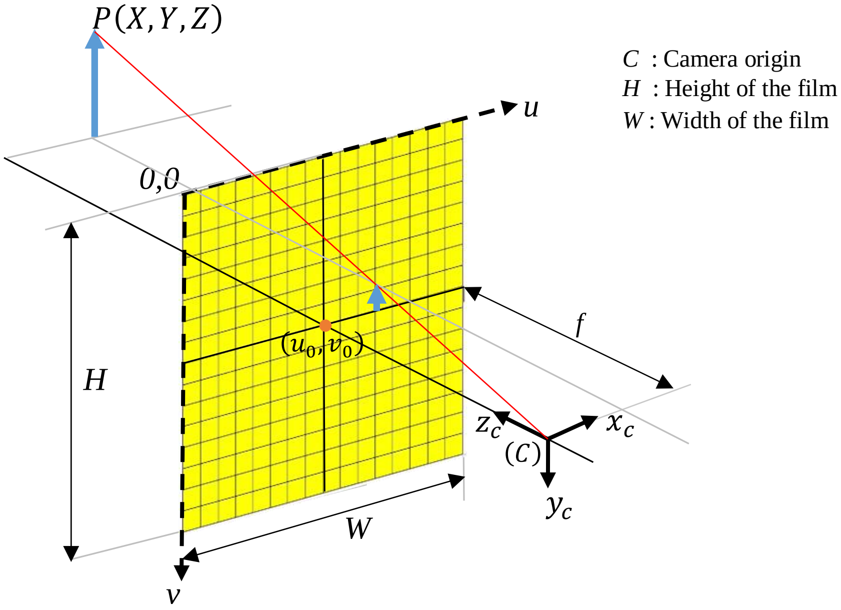


Figure 7.18: Projection of an image inside camera at a focal length, *f*

The coordinates of the image frame starts at upper left corner of the film (refer Figure 7.18). Eq. (65) denotes how the transformation is performed between $(\tilde{x}, \tilde{y}, \tilde{z})$ and (u, v) frames [64].

$$\begin{pmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{pmatrix} = \begin{pmatrix} \frac{1}{p_u} & 0 & u_0 \\ 0 & \frac{1}{p_v} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{pmatrix} \tag{65}$$

Where, p_u and p_v are pixel densities in *u* and *v* directions respectively.

Therefore, the coordinate transformation between the robot coordinate frame to image coordinate frame can be accomplished using Eq. (66).

$$\begin{pmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{pmatrix} = \begin{pmatrix} \frac{1}{p_u} & 0 & u_0 \\ 0 & \frac{1}{p_v} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} {H_{2\_1}^{0}}^{-1} \begin{pmatrix} U \\ V \\ W \end{pmatrix} \tag{66}$$

In addition to the coordinate transformation between different frames, the angle between the iron and vertical plane possess ($(90 - \beta)$ in Figure 7.16) a significant importance. Unless it is maintained at an optimum value, the melted soldering paste tries to move towards the iron tip when it is lifted, resulting deformed solder joints as illustrated in Figure 7.19.

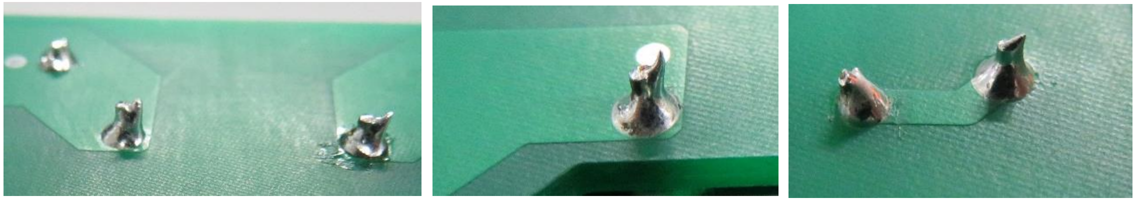


Figure 7.19: Deformations occurred on solder joints due to the wrong angle between soldering iron and vertical plane

This deformation can be mitigated by minimizing, the angle between the iron and the vertical plane. Then the melted soldering rises up along the component lead, rather than in the direction of the iron tip when the soldering iron is lifted up.

Some soldering robotic systems like Weller WTBR1000, the soldering wire is fed to the soldering iron tip, rather than on the pad to reduce the pre-heat time. However, this degrades the soldering quality and becomes worst, when the solder pad is connected to a large ground or power plane. It could be identified that feeding the wire directly into the space between the lead and the solder pad, could result in good quality soldering. Figure 7.20 illustrates how the soldering iron and the soldering wire touches the solder pad.

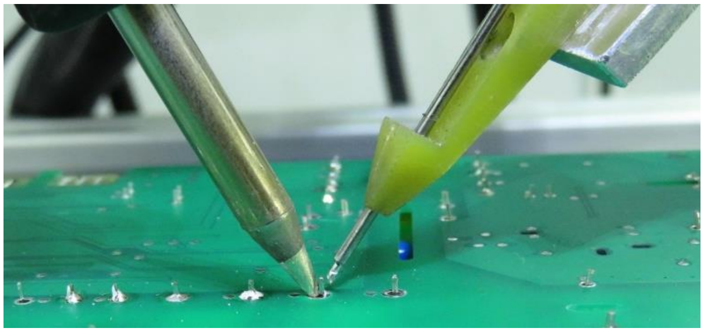
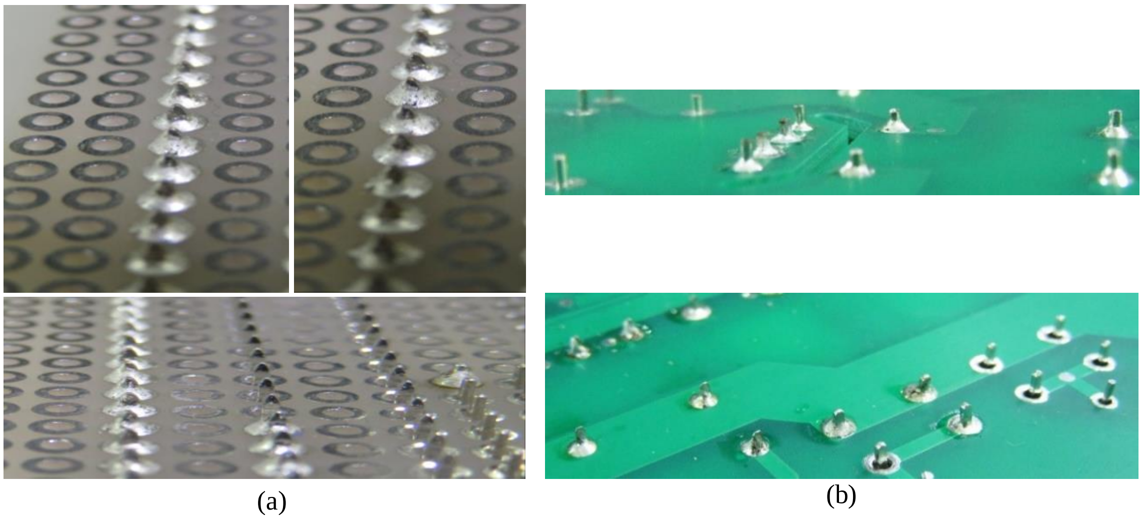


Figure 7.20: The angle which the soldering iron and wire feeder points to particular solder pad

This mechanism renders a significant improvement in soldering quality as illustrated in Figure 7.21. It can be seen that the implemented soldering system can perform a good quality soldering even on pads connected to large copper plane (refer Figure 7.21 (b)).



(a)

(b)

Figure 7.21: Solder joints performed by the THT soldering system. (a). Soldering performed on a regular dot board. (b). Solder joints performed on an industrial PCB

The soldering arm can be controlled in both horizontal plane and vertical plane independently as depicted in Figure 7.22 (refer Figure 7.16 kinematic diagram).

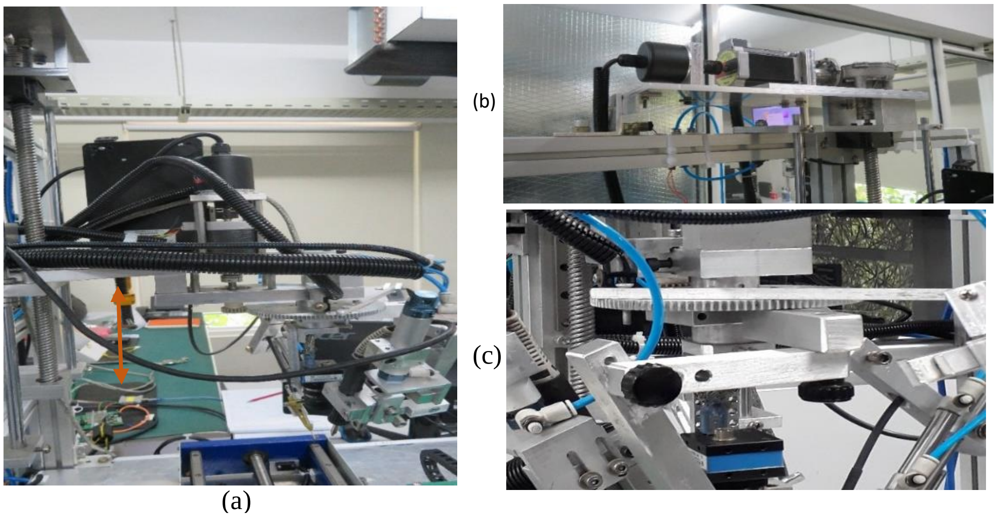


Figure 7.22: Structure of the soldering arm. (a). Moving directions (Brown arrow: For vertical direction, Red arrow: For horizontal plane). (b). Mechanism for controlling the soldering arm in vertical direction. (c). Mechanism for controlling the soldering arm in horizontal plane

The soldering arm possesses 100μm and 0.5^0 accuracy in vertical and horizontal directions respectively. Soon after the system is initialized, the soldering arm is driven in vertical direction until it reaches the distance defined in the respective soldering programme and that height is maintained as the flying height for that PCB.

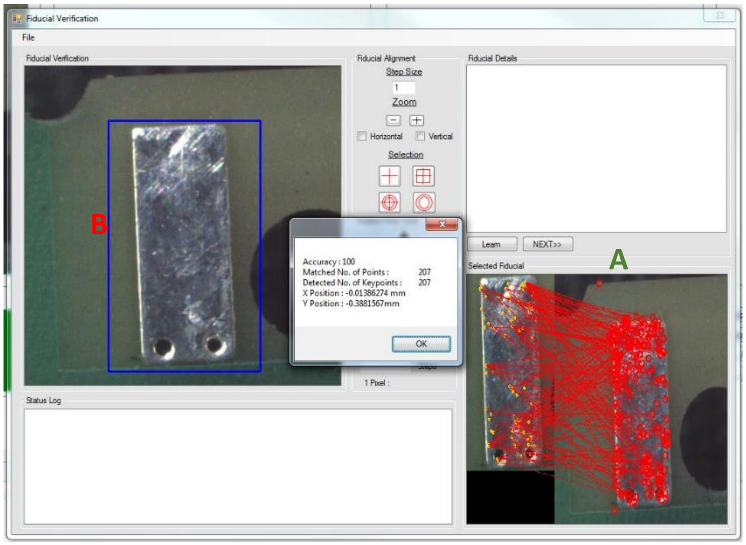
Chapters 2-7 describe how the THT soldering system with integrated AOI has been implemented in this research. Even though, the current system does not accommodate significant robotic operations, it provides unique features combined with AOI and CAD systems that are not available in commercially available systems. Chapter 8 discuss about the encountered problems during the operation of each stage and the countermeasures taken to provide a precise operation through minimizing their impact.

8. Results and Discussions

The information provided in Chapters 2-7 reveals that the implemented AOI, CAD and robotic systems have successfully solved each problem described in Section 1.3. However, Sections 8.1-8.5 describe how the stability of the implemented system varies at different stages of soldering during the operation in practical environment.

8.1. Discussion: Implementation of Automatic PCB Fiducial Alignment Process

The results tabulated in Tables 2.1 - 2.9 show that the implemented fiducial alignment process can provide a significant improvement over the Hough transform [25] based PCB alignment method described in [12]. Figure 8.1 illustrates how the robotic system controlling software performs the fiducial verification process.



Accuracy : 100
Matched No. of Points : 207
Detected No. of Keypoints : 207
X Position : -0.01386274 mm
Y Position : -0.3881567mm

Figure 8.1: Fiducial verification process carried using the implemented AOI

Here, ‘A’ depicts the matched keypoints and ‘B’ depicts the localized model image (blue rectangle). However, the performance of this verification process is affected by,

- the similarity between the defined master points in different PCB types
- multiple appearances of the same master point in different PCBs

Sections 8.1-8.2 discuss how the effect of these problems were mitigated by the system.

8.1.1 Minimizing the effect of similarity between the defined fiducial points in different PCB types

Occasionally, master points with similar appearance exists in different PCB types. Some PCBs come into production only with a slight change in layout (Ex: a component orientation change, new component added,..etc.), but as a different PCB type (refer Figure 8.2). Manual segregation of such PCBs requires skilled operators and hence automatic differentiation between such PCBs is vital for automated systems.

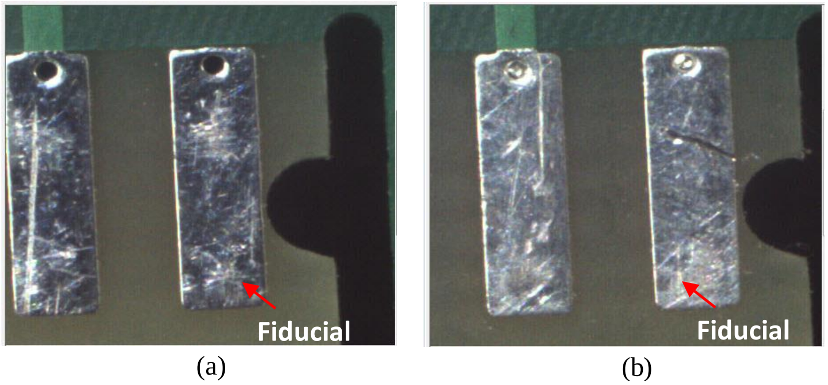
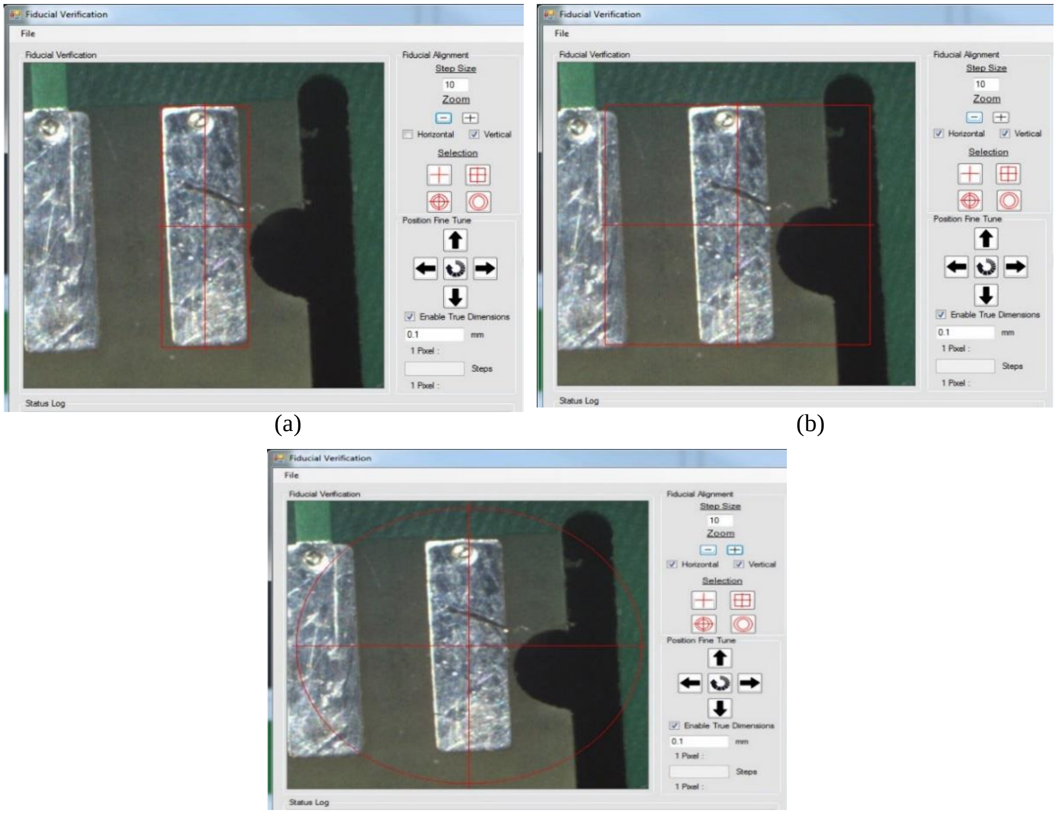


Figure 8.2: The impact of slight PCB layout changes over the robustness of the fiducial verification process. (a). The original PCB with the defined fiducial point. (b). The new PCB with a layout change of shifting the fiducial point (solder pad) by $400\mu m$ to the right

However, the existence of similar background around the same fiducial point has a lower probability to occur (refer Figure 8.2). Therefore, the system software allows the user to define multiple images for the same fiducial point in different shapes and sizes to detect the background variation during fiducial learning as illustrated in Figures 8.3-8.4. However, it is the sole responsibility of the user to define these subsequent images in a way that they contain meaningful regions.

The performance of two methods were evaluated in this research to measure the dissimilarity between the located image and the respective subsequent images.

- BGR histogram comparison technique [65-67]
- Background Subtraction algorithm [68-70]



(a)

Step Size 10

☐ Horizontal ☑ Vertical

☑ Enable True Dimensions

0.1 mm

Steps

(b)

Step Size 10

☑ Horizontal ☑ Vertical

☑ Enable True Dimensions

0.1 mm

Steps

(c)

Step Size 10

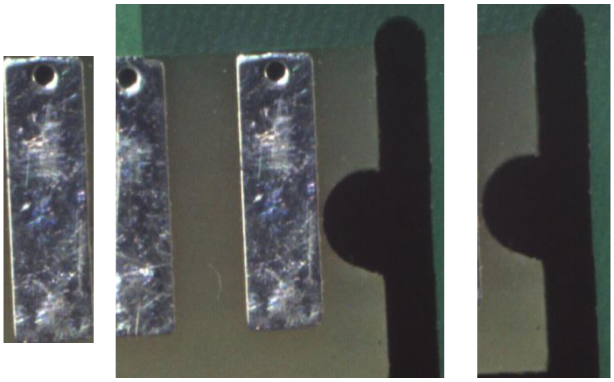
☑ Horizontal ☑ Vertical

☑ Enable True Dimensions

0.1 mm

Steps

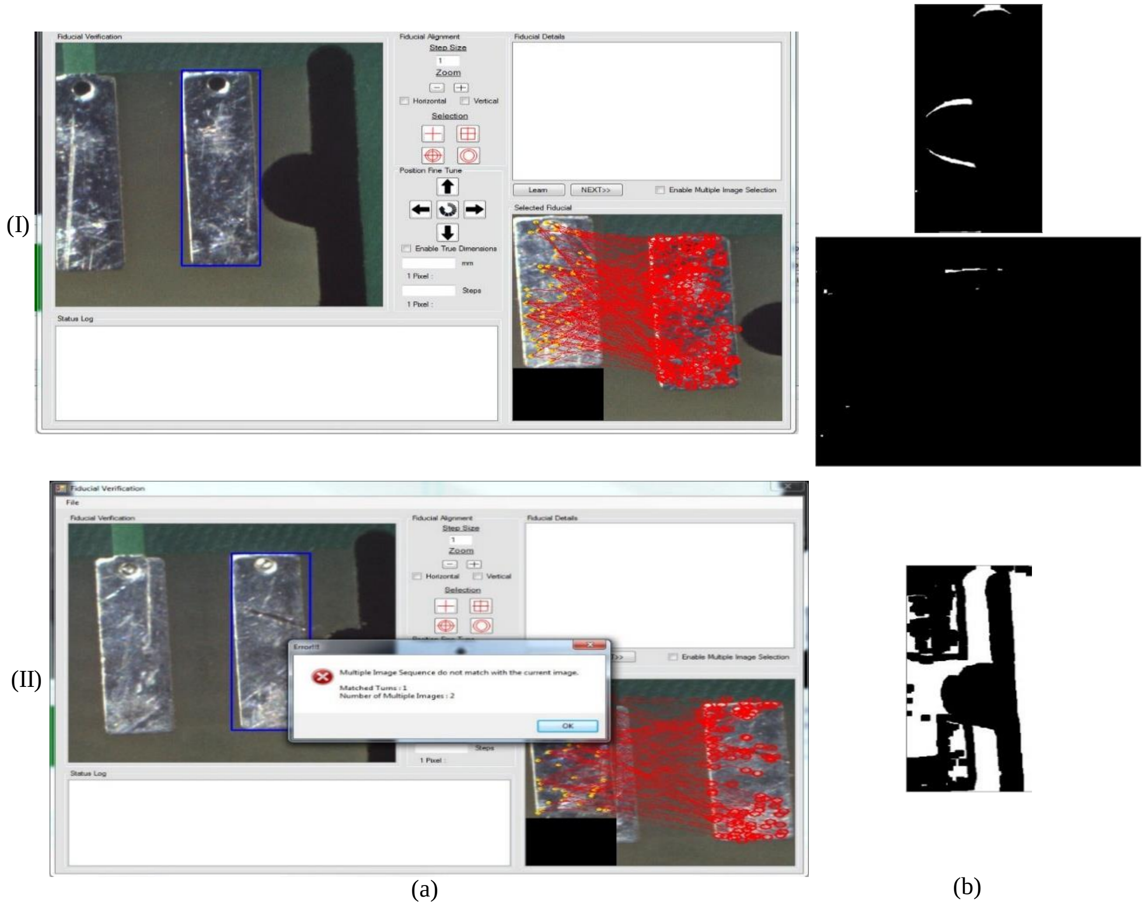
Figure 8.3: Definition of subsequent images in different shapes and sizes. (a). Definition of fiducial point. (b). Subsequent images can be defined with variable dimensions in a way that they include distinguishable areas (Rectangular regions). (c). Subsequent images can be defined with variable dimensions in a way that they include distinguishable areas (Circular regions)



(a) (b)

Figure 8.4: Definition of multiple subsequent images for a fiducial point. (a). Defined fiducial point. (b). Defined subsequent images containing unique areas

Experimental results show that the background subtraction method [25] outperforms histogram matching algorithm significantly. Figure 8.5 illustrates how the system effectively identifies the difference between the two PCB types.



Multiple Image Sequence do not match with the current image.

Matched Turns : 1
Number of Multiple Images : 2

Figure 8.5: Verification of the existence of fiducial point using background subtraction method (I) Successful detection of original PCB using background subtraction method. (II). Precise identification of the incorrect PCB using background subtraction method. (a). The system response for different PCB types. (b). Resulted binary image from background subtraction

According to Figure 8.5 (II) (b), it can be clearly seen that the vision system identifies the change in the background of the PCB with modified layout. The size of the largest contour region in the resulted binary image is proportional to the degree of change in the background. If the system detects a change of background greater than the user defined threshold value for a particular subsequent image, the user is alarmed and the robotic system is re-initialized after aborting the fiducial verification process.

8.1.2 Minimizing the effect of multiple appearances of the same fiducial point in different PCBs

The fiducial points described in this research are not larger than *3mm* of length in each dimension in majority of the cases. However, few occasions could be identified where a dissimilarity in the appearance of same master point exists as a result of oxidisation due to long time storage and physical damages. The user is authenticated to manually confirm the existence of master points in such situations. Table 8.1 tabulates how the performance of the fiducial verification process varies with these additional challenges.

Table 8.1: Performance evaluation of the fiducial verification process along with the background verification methodology

Current appearance of the PCB sample	**Successful detection rate of fiducial verification using SURF and Template Matching**	**Successful detection rate of fiducial verification using SURF, Template matching and background verification**
PCBs with unique fiducial points (80 PCBs with 320 fiducial points)	98%	98%
PCBs with similar fiducial points on different PCB types (8 PCBs with 16 fiducial points)	12.5%	100%
PCBs of same type but with damaged or heavily corroded fiducial points (12 PCBs with 36 fiducial points)	8.3%	32%

Table 8.1 illustrates that the vision system shows very poor performance when the fiducial points are heavily corroded or physically damaged. Even, the application of Deep Neural Networks (DNN) [71-74] is not a viable option to solve this issue because of the longer training time and the varying appearance of fiducial points.

8.2. Discussion: Implementation of Solder Pad and Component Lead Segmentation Process

Sections 8.2.1-8.2.2 discuss the encountered problems and the countermeasures taken to overcome them prior to the soldering process in practical environment.

8.2.1 The Accuracy of Localizing the Actual Pad to be Soldered by The Vision System as Defined in the Respective Programme for Soldering

Chapter 3 described how the solder pad region is precisely extracted from the PCB surface. Occasionally, solder pads with same dimensions exists inside the camera FOV. The situation becomes critical when such situations occur with a positioning error as illustrated in Figure 8.6. Therefore, the AOI must be capable enough to identify the exact pad to be soldered out of the objects inside the ROI and performs real time position calibration without any error.

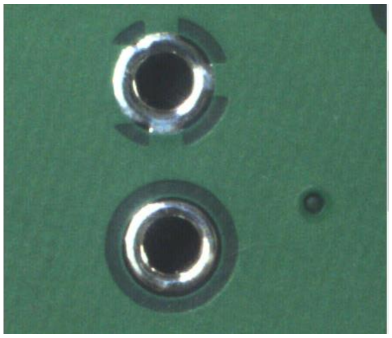


Figure 8.6: Existence of the objects with similar dimensions inside the camera FOV

The implemented method in Chapter 3 is not able to distinguish between the pads illustrated in Figure 8.6. However, following ways can be used to mitigate such impact.

- If either of pads is not included in the soldering programme, exact pad can be identified by verifying the component availability (refer Chapter 4)
- Learning of the immediate background of the pad to be soldered using DNN

The application of DNN requires extensive teaching of individual joints and its outcome is unpredictable at such situations due to the similarity of pad surface. However, a careful observation shows that the geometrical locations of these objects relative to each other, can be effectively used to learn the background of a solder pad. Therefore, the system software allows the user to execute a diagnostic procedure to perform a simple teaching process to accomplish this task based on his requirement as illustrated in Figure 8.7.

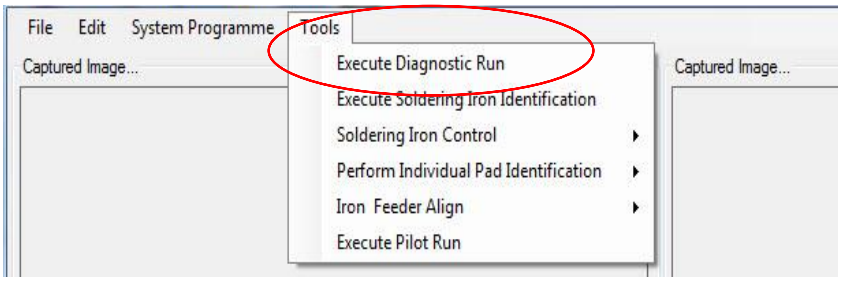


Figure 8.7: Executing diagnostic programme for solder pads to perform a precise real time calibration process

The system goes through each pad in the programme and identifies objects inside the camera FOV for each positioning (refer Figure 8.8) without performing further action.

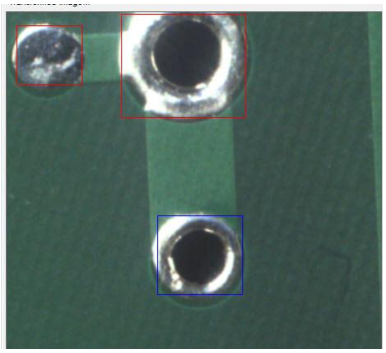


Figure 8.8: Identified objects inside the camera FOV. Blue highlighted pad holds the highest probability of being the solder pad to be soldered. Red highlighted objects depict the neighbouring objects

In diagnostic process, the robotic system software performs following steps sequentially for learning the background of a solder pad.

Step 1: Run the fiducial verification and computes the optimum travelling path.

Step 2: Segments each object inside the camera FOV that are larger than a predetermined threshold using the AOI system when a new pad is positioned.

Step 3: Highlight the object in blue colour which has the highest possibility to be the solder pad based on its dimensions acquired from CAD file and highlight other objects in red colour as illustrated in Figure 8.8.

Step 4: The user can either stay with the current selection or define the pad by clicking inside the object region based on CAD information or a sample PCB. Once the pad is defined, it is highlighted in purple (refer Figure 8.9) and the centre point is computed.

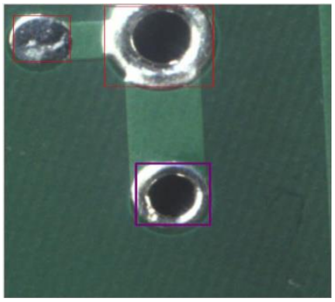


Figure 8.9: Defined solder pad by the user during the operation of diagnostic process

Step 5: Then the user can select neighboring objects. The distance and angle between the defined neighbour and the pad is computed for each definition (refer Figure 8.10).

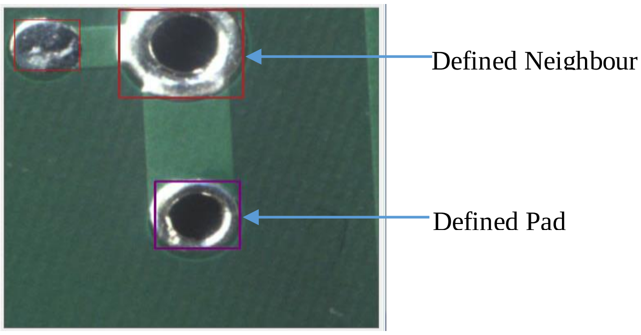


Figure 8.10: Selected candidate objects (highlighted in brown colour) and defined solder pad (highlighted in purple colour)

Step6: Steps 2-5 are repeated, until all the solder pads are defined with their neighbours.

The robotic system software operates according to the below procedure based on the availability of diagnostic data for each positioning in normal run.

Step 1: Segments the objects in camera FOV and add their bounding boxes to a list R_L.

Step 2: Select all objects whose sizes are greater than an adaptive threshold value relative to the solder pad size and update the list O_L with their bounding boxes.

Step 3: Iterates through each object C_x in O_L and compute distances and relative angles to the identified objects in R_L.

Step 4: Then finds the object in O_L that has similar distances and angles combinations to that of the learnt background of the solder pad in diagnostic data. If a perfect matching is found, real time position calibration is executed, unless user is notified.

Step 5: Repeat Steps 1-4 for each positioning.

This procedure further enables to have a second verification stage for PCBs with same fiducial points at identical locations but with different layouts. Table 8.2 illustrates the performance analysis of diagnostic procedure against the normal size comparison method.

Table 8.2: Performance evaluation of the size comparison method and the background learning methodology for identifying the exact solder pad to be soldered

Used methodology	**Successful detection rate**
Definition of solder pad based on the size comparison	94%
Background learning methodology using diagnostic run	100%

However, the performance of this methodology primarily depends on how well the neighbours to a pad are defined by the user. The user can decide whether he needs a diagnostic run or not based on the immediate neighbourhood of solder pads.

8.2.2 The accuracy of localizing the drill-hole region of the solder pad

Section 4.2 describes the implementation of an algorithm that automatically identifies drill-hole region of a solder pad. However, the performance of this algorithm is slightly affected by the amount of drill-hole area covered by the component lead (10 solder pads out of 500 solder pads could be found that covers more than 90% of the drill-hole region) and the camera frame rate. They can result in either of following false identifications.

- The computed drill-hole region contains a portion of pad (refer Figure 8.11 (a))
- The size difference between the computed drill-hole and the actual information in CAD file (refer Figure 8.11 (b))

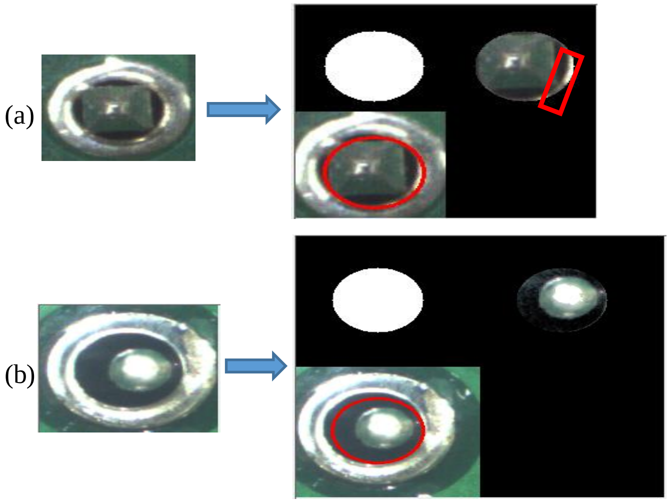


Figure 8.11: The erroneous computation of drill-hole region. (a). False detection of drill-hole due to unavailability of empty area for processing. (b). Image blurring due to camera frame rate results in false identification of drill-hole region

Drill-hole contains a slight amount of the pad: *This is not an easy situation to be improved due to dominating lead size. The current version of the vision system is not able to deal with such situations efficiently.*

Size difference between the computed drill-hole comparing to CAD data: *If the radius of the computed drill-hole is not within an adaptive threshold range related to the dimension defined in CAD data, the user confirmation is requested to proceed.*

Table 8.3 illustrates the performance variation of this algorithm over a sample of 500 solder pads in practical environment.

Table 8.3: Performance evaluation of automatic drill-hole computation algorithm during the operation of the THT soldering system

Outcome of the drill-hole segmentation algorithm	**Percentage of Occurrence**
Precise computation of drill-hole region	98%
False hits performed by the algorithm	2%
User confirmation required	1.4%

It could be observed that the frame rate (Current frame rate is 8 frames/sec) of the current camera has a significant impact over the difference of the results acquired in Table 4.1

and Table 8.3. This frame rate introduces a slight amount of blurring when the positioning is done over a short distance. Even the amount of noise introduced is less, it becomes significant when the component lead covers more than 90% of the drill-hole. A camera with higher resolution and frame rate will reduce this error rate.

8.2.3 The accuracy of segmentation of the component lead from the solder pad

Section 4.2 described how the component lead is precisely segmented from the drill-hole. However, the colour of the component lead and its shape plays a significant role on the segmentation accuracy in practical environment.

Colour of the component lead:

Even though, the component leads that have virtually the same colour to that of drill-hole could not be found in this research, there might be such situations when this system is used in a real environment. Figure 8.12 illustrates few component lead tops identified by the AOI that hold a darker colour. As a preventive action, a combination of gamma correction [75-76] and histogram equalization [77-79] based image enhancement methodology is proposed. However, the successful detection rate was not affected by the integration of this procedure for the current version of the AOI.

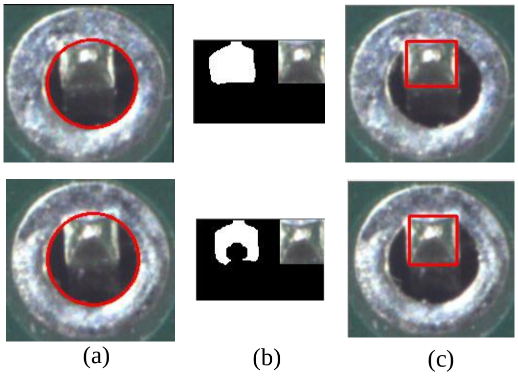


Figure 8.12: The computation of drill-hole and identification of component lead. (a). Segmented solder pad with computed drill-hole. (b). Thresholded image for the identification of component lead (left-binary image, right-segmented lead top). (c). Solder pad with identified lead

Shape of the component lead:

The algorithm for component lead top segmentation, computes the SD of the distances between boundary points and centre of detected objects to mitigate the impact of light reflection. However, leads with irregular shapes (refer Figure 8.13) degrades the accuracy of the algorithm.

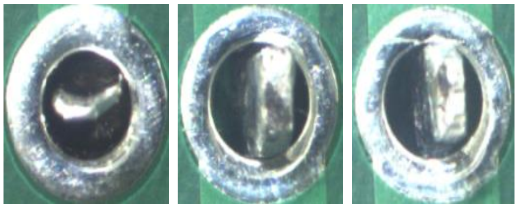
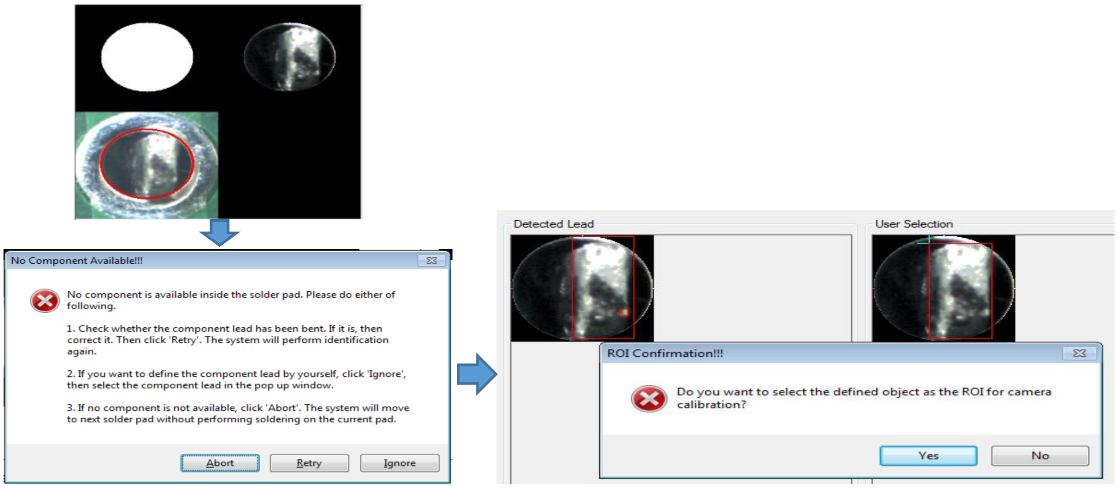


Figure 8.13: Different shapes of component leads

The vision system may conclude that there is no component inside the drill-hole at such situations. Hence, the user confirmation is requested by the system software based on the orientation and SD of identified region as illustrated in Figure 8.14. As an improvement to the implemented vision system, the system will learn the geometrical dimensions of such leads in diagnostic run and use that data to uplift the reliability of the system.



No Component Available!!!

No component is available inside the solder pad. Please do either of following.

1. Check whether the component lead has been bent. If it is, then correct it. Then click 'Retry'. The system will perform identification again.

2. If you want to define the component lead by yourself, click 'Ignore', then select the component lead in the pop up window.

3. If no component is not available, click 'Abort'. The system will move to next solder pad without performing soldering on the current pad.

ROI Confirmation!!!

Do you want to select the defined object as the ROI for camera calibration?

Figure 8.14: User confirmation for the pads with distributed shapes

8.3 Discussion: The accuracy of solder quality classification of the solder joint

Chapter 5 described how the solder quality is classified by the AOI. Sections 8.3.1- 8.3.3 discuss problems encountered and the solutions provided at different quality classification stages like voids in drill-hole, voids on solder pad and excess solder during the operation of the THT soldering system.

8.3.1 The classification of voids inside the drill-hole

Section 5.3 described how the voids inside the drill-hole region are classified by the AOI. However, the classification accuracy is affected by the prevailing illumination level and the surface condition of the solder pad. The availability of illuminated regions on the pad minimizes the colour dissimilarity between pad and solder (refer Figure 8.15). Therefore, it affects the algorithm accuracy since it verifies solder availability on adjoining pad when a void is detected inside drill-hole. The impact is significant when the pad is tin-plated.

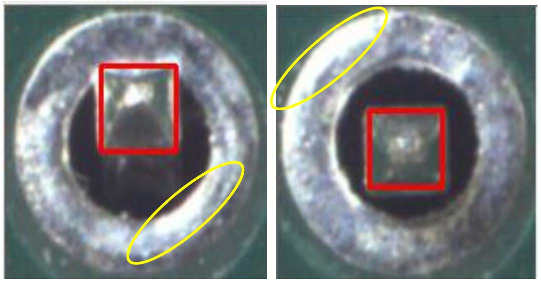


Figure 8.15: The impact of illuminated areas on the solder pad for the stability of classifying voids inside the drill-hole

This effect can be mitigated if such regions are prior identified. Hence, the system software allows the user to activate this feature based on his preference. If it is activated, the AOI identifies such regions prior to soldering. The first step involves with clustering colour transformed solder pad region (HSL colour transformation). The resulted image is thresholded based on predetermined value. Finally, the resulted binary is processed using morphological filtering to remove small regions. The adjoining pad area is AND operated with this binary image to filter valid regions for the verification of solder availability (refer

Figure 8.16). If the filtered out image does not contain any region, the user confirmation is requested to proceed.

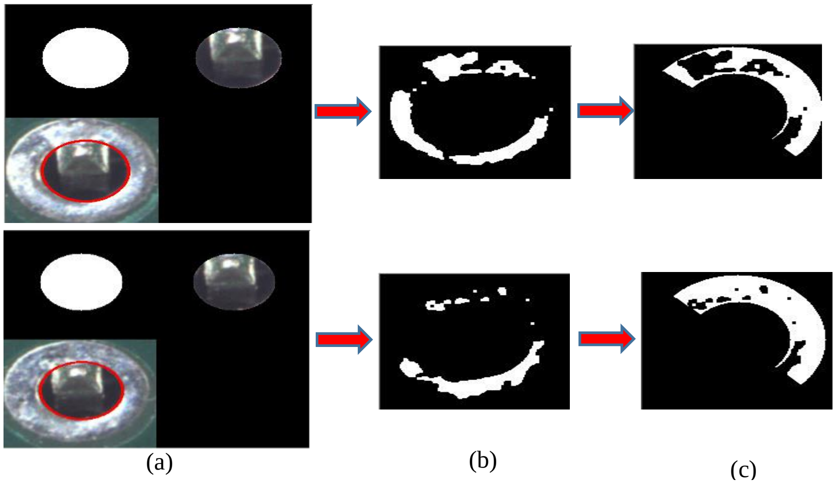


Figure 8.16: The removal of illuminated regions on the adjoining solder pad corresponding to a detected void inside the drill-hole. (a). Segmented pad (bottom-left) and drill-hole regions (top). (b). Extracted illuminated regions on the bare solder pad. (c). Filtered out regions on a computed adjoining pad region after removing illuminated regions on the solder pad

The removal of illuminated regions is not possible if the AOI system is used for the classification of already soldered joint, whereas this AOI system has the advantage of gathering information related to the prior stages of soldering. Table 8.4 illustrates the impact of this methodology together with the developed algorithm for void detection inside drill-hole over a sample of 200 solder joints with illuminated pad regions.

Table 8.4: Performance evaluation of the algorithm for classifying voids inside the drill-hole region of the solder joint

Used methodology	**Hit rate**
Application of algorithm without removing illuminated regions	90%
Application of algorithm with the removal of illuminated regions	98%
Requirement of user confirmation on identified voids with the application of the removal of illuminated regions	2%

The results obtained in Table 8.4 confirms that the effect of illumination can be successfully mitigated by the proposed methodology. Figure 8.17 illustrates some of the classifications carried out by activating this feature.

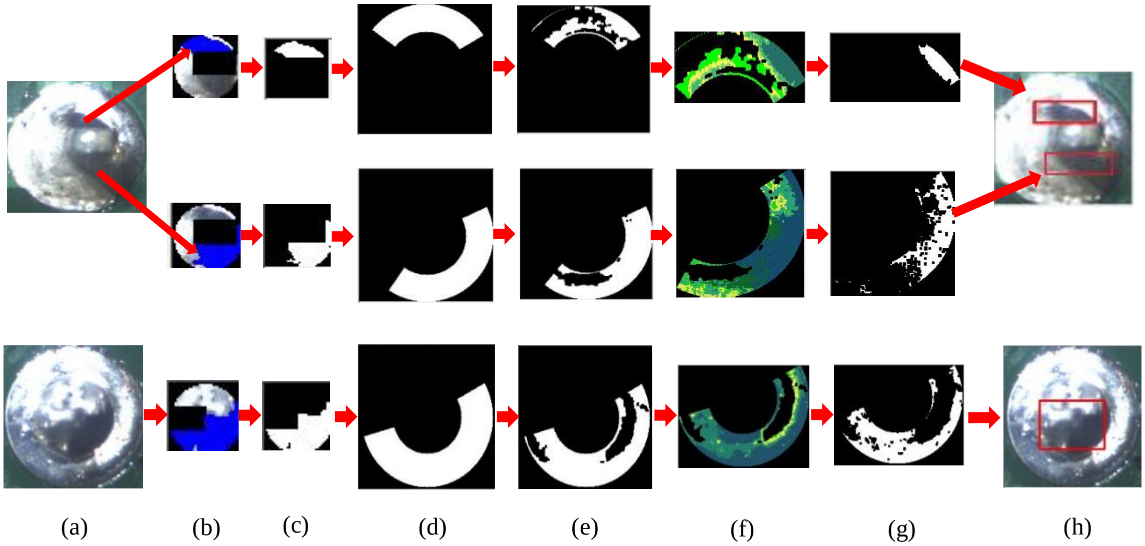
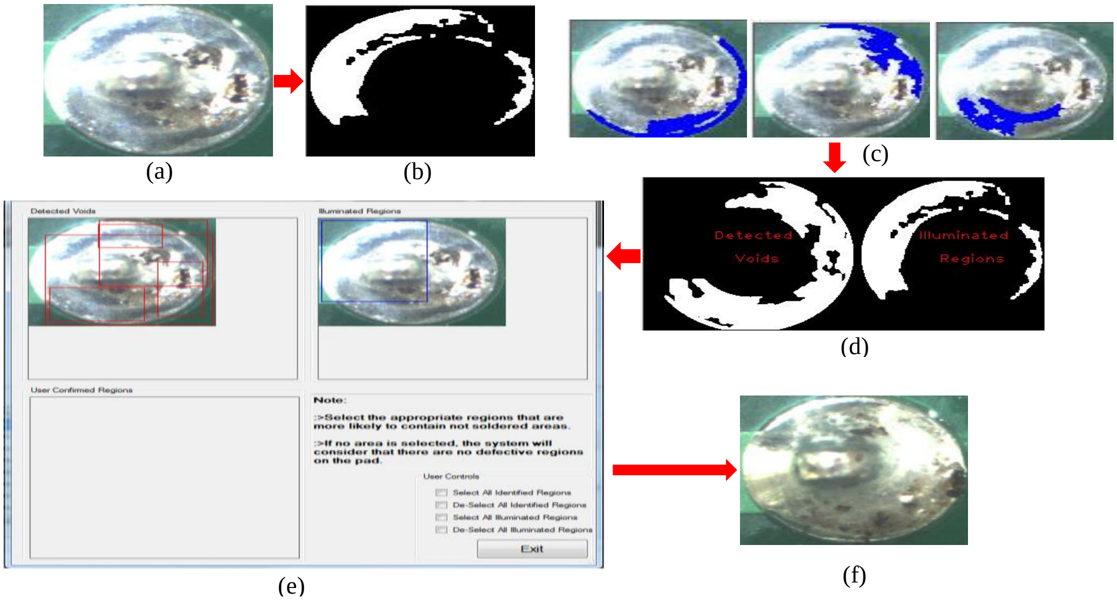


Figure 8.17: The soldering defects classification of voids inside the drill-hole, with the integration of the removal of illuminated pad regions. (a). Identified solder joint. (b). Segmented regions. (c). Identified voids from the segmented homogeneous regions. (d). Computed adjoining pad area. (e). Adjoining solder pad regions with removed illuminated pad areas. (f). Clustered colour transformed image (HSL) image. (g). Identified not soldered areas on the solder pad. (h). Detected void inside the drill-hole region of the solder pad

8.3.2 The classification of voids on the solder pad

Section 5.4 described how the voids on the solder pad region is classified by the AOI. The highly illuminated pad regions and reflective nature of solder joint surface put a detrimental impact on the accuracy of the developed algorithm. Section 8.4.1 describes how the vision system identifies the illuminated regions on the solder pad prior to the soldering. If the AOI detects illuminated pad regions, user authentication is requested to proceed as depicted in Figure 8.18. Then the user can define the defective region within several mouse clicks and authenticate to rework them.

Furthermore, the amount of heat applied brings a negative impact on the algorithm accuracy. The computed pre-heat time relative to the size of the connected PCB track is proportional to the degree of solder joint surface discolour level as illustrated in Figure 8.19 (I). Such a discoloured surface leads to misidentifications perform by the vision system as illustrated in Figure 8.19 (II). However, the required pre-heat time by the solder iron can be reduced by deploying a pre-heater prior to loading of the PCB to the system.

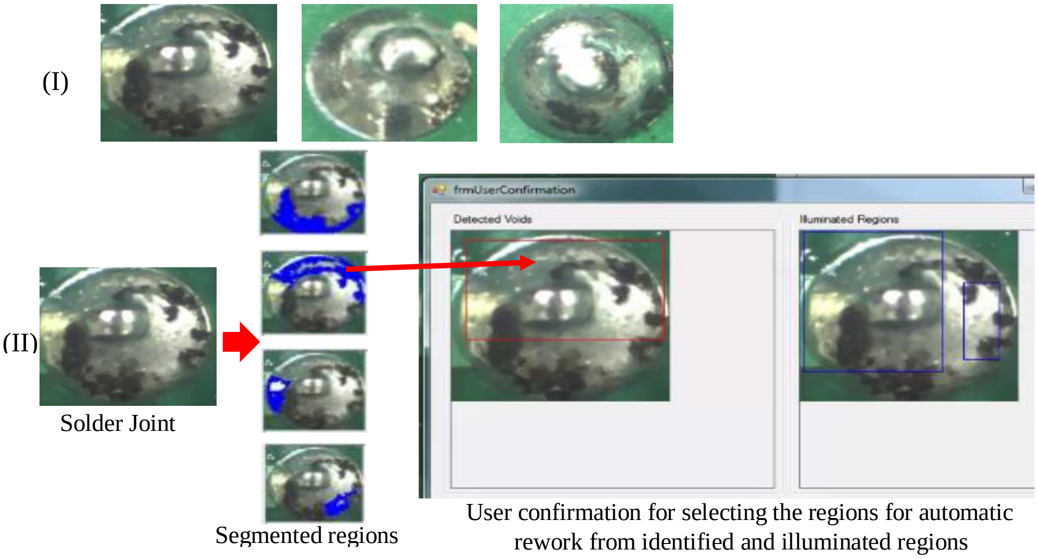

(a)
(b)
(c)
Detected Voids
Illuminated Regions
(d)
Detected Voids
Illuminated Regions
User Confirmed Regions
User Controls
□ Select All Identified Regions
□ De-Select All Identified Regions
□ Select All Illuminated Regions
□ De-Select All Illuminated Regions
Exit
(e)
(f)

Note:

:>Select the appropriate regions that are more likely to contain not soldered areas.

:>If no area is selected, the system will consider that there are no defective regions on the pad.

Figure 8.18: Effect of illuminated region removal on voids detection on the solder pad. (a). Segmented solder joint. (b). Segmented illuminated regions. (c). Segmented regions on pad. (d). Identified defective regions and illuminated pad regions. (e). User confirmation on the selection of the defective regions from the identified regions and illuminated regions. (f). The reworked solder joint based on the user defined regions



Solder Joint

Segmented regions

User confirmation for selecting the regions for automatic rework from identified and illuminated regions

Figure 8.19: Impact of longer pre-heat time required for pads connected to large copper planes on the accuracy of voids detection on solder pad. (I). Discoloured solder joints due to the longer pre-heat time. (II). Falsely detected discoloured regions as voids on the solder pad

8.3.3 The classification of excess solder

The occurrence of discoloured regions on solder joint surface (refer Figure 8.19) can also degrade the performance of excess solder detection algorithm by minimizing the availability of illuminated regions on the joint surface as illustrated in Figure 8.20. The occurrence of such regions can be mitigated by using a pre-heater as proposed in Section 8.3.2.

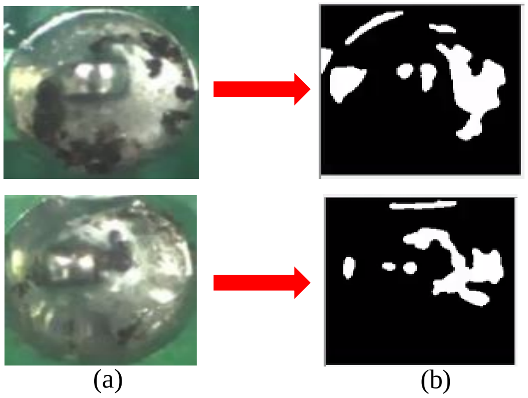


Figure 8.20: Impact of longer pre-heat time required by the solder pad on the distribution of light on the solder joint surface. (a). Solder pad connected to a larger PCB track or copper plane. (b). Identified illuminated regions on the solder joint surface

8.4 Discussion: The stability of the THT soldering system hardware

The implemented system is an electromechanical platform which has four axis of movement. Hence, the mechanical stability and precise positioning capability are vital to endow a reliable and robust operation. Sections 8.4.1-8.4.2 discuss about the problems encountered during the implementation of the robotic system.

8.4.1 The stability of the soldering wire feeder

This unit was one of the most difficult part to be manufactured because it requires high accuracy level. The feeder unit used in the prototype push the soldering wire from the top. This mechanism was not stable enough to render the required accuracy in this application.

Because, the force applied from the top is decayed at the end of the feeder tip when the feeding wire is longer. This resulted in two major problems during the system operation.

- Difference occurred between the applied solder and the required amount of solder
- The solder wire tends to bend when the solder pad is poorly heated up

These two problems can be solved when the feeder unit is a pull type. It means the feeding mechanism will be placed closer to the tip of the wire feeder. Hence the applied force is much higher since the travelling distance is minimum.

8.4.2 The stability of the soldering iron

The stability of the soldering iron plays a vital role in this application. Slight height inconsistency of the mounted PCB causes the delamination of surrounding PCB surface of a solder pad due to the touches of soldering iron tip at a high pressure. The current version of the robotic system controls the iron pneumatically between two positions. This mechanism introduces an uncontrolled pressure on the PCB surface when some height deviations exists. This effect can be mitigated by introducing a closed loop motor driver to control the soldering iron. Furthermore, the deployment of a spring tension on the rear end of the iron can provide a room for the iron to move in backward direction when a pressure is applied on the iron tip. This modification on the soldering iron positioning mechanism will uplift its reliability.

9. CONCLUSION

The primary aim of this research was to implement a vision system that can be integrated to a THT soldering system. The core functionalities of the implemented system are to provide automatic vision capability to minimize user mistakes, online classification of soldering quality and rework defective soldering joints. This research provides:

- An AOI that monitors every action performed by the THT soldering system;
- A CAD system that extract geometrical information of the components;
- A 4-axis soldering system that performs automatic soldering of THT solder joints.

Even though, the primary aim is to implement a fully automated system that endows skill independent soldering process, the current version of this system requires user assistance at three occasions.

- Confirmation for component availability when the lead contains different shapes
- Confirmation for voids on pad when the pad contains illuminated regions
- Confirmation of excess solder on discoloured solder joint surfaces

Even though, the commercial AOI systems also require user confirmation to verify the components and classify soldering quality, the primary aim of the AOI to provide 100% autonomous operation for a soldering robotic system, could not be achieved as discussed in Chapter 8. However, the developed AOI outperforms commercial systems by:

- Eliminating the prior teaching of individual pads and providing online calibration;
- Eliminating the need of specific lighting environments within enclosed chambers;
- Providing online classification of component availability and soldering quality classification of THT solder joints.

The modified version of the AOI (as suggested in Chapter 8) together with controlled illumination level can significantly reduce the rate of user confirmation required.

Even though, the application of machine learning approaches based on DNN and Convolution Neural Networks (CNN) are gaining recognition by leaps and bounds in some industrial applications, their usability is not convenient in this research due to following reasons.

- Neural network based algorithms require huge amount of data (sometimes millions of labelled samples) for training and a longer time for learning to provide accurate results.
- The varying appearance of solder joint surface and its shape lessens the robustness of neural network based algorithms.
- The automatic verification of component availability cannot be successfully achieved as with the implemented algorithm, if neural network is used, because of high variability of lead colours and reflected light inside the drill-hole.
- The reason for the output of a neural network is unknown and hence troubleshooting or backward adjustment is not possible. The impact is detrimental when the pad contains illuminated regions as described in Section 8.3.1-8.3.2.
- The output of a neural network is the set of probabilities that an input data can have. In that way, one of the primary tasks of this AOI, automatic rework based on the size of the defective region cannot be accomplished.

Furthermore, a CAD system could also be developed with an interactive GUI that reduces the programme development time for a PCB from several engineering hours to maximum of 15 minutes for a PCB with 120 solder joints. This CAD system provides all the geometrical information required by the AOI and other relevant information like pre-heat time and solder iron touching direction for a solder pad required by the soldering system for its proper operation.

The goal of implementing a THT soldering system with automatic rework capability on defective solder joints, automatic real time calibration, automatic definition of iron tip cleaning period based on the deployed preheat time and amount of applied solder, could be achieved successfully during this research.

However, the implemented THT soldering system can be further improved by,

- replacing the existing solder wire feeder (push type) with a pull type feeder;
- modifying the soldering iron in a way that it absorbs pressure applied on the iron tip rather than pressuring the PCB surface;
- altering the soldering iron positioning mechanism from pneumatically to motor driven;
- replacing the existing soldering iron tip cleaning mechanism with the automatic rolling mechanism;
- moving the soldering arm in *XY* directions keeping the PCB stationary.

The development of the commercial version of this robotic system has been already started to proceed with these modifications. However, this is the first soldering robotic system of this kind in the market today which has its own CAD system and an integrated automatic vision system that monitors each action performed by the soldering system.

Reference List

[1] C.E Barateiro, “Industrial Internet of Things and Automation for Process Control: Pursing the Operational Excellence,” *Rio Oil & Gas Expo and Conference* 2018, Available at: https://www.academia.edu/37718722/4.0_INDUSTRIAL_INTERNET_OF_THINGS_AND_AUTOMATION_FOR_PROCESS_CONTROL_PURSING_THE_OPERATIONAL_EXCELENCE?auto=download

[2] iQor, Consumer Electronics. Available at: https://www.iqor.com/industries/consumer-electronics, Accessed on 12 December 2018

[3] J. Brooks and L. Brooks, “Automation in the medical field,” *IEEE Engineering in Medicine and Biology Magazine*, Vol. 17, no. 4, P. 76, July-Aug. 1998.

[4] S. Tavagad, S. Bhosale, A. P Singh and D. Kumar, “Survey Paper on Smart Surveillance System,” *International Research Journal of Engineering and Technology (IRJET)*, Vol. 3, no. 2, Feb. 2016

[5] Japan Unix and Soldering Robotic Systems, Available at: https://www.japanunix.com/en/products/automation/, Accessed on 22 June 2013

[6] Kurtz Ersa and Wave Soldering Systems, Available at: https://www.kurtzersa.com/electronics-production-equipment/soldering-machines/wave-soldering-systems/produkt-details/powerflow-air-1.html, Accessed on 09 May 2013

[7] Nordson, Selective Soldering Systems, Available at: http://www.nordson.com/en/divisions/select/products/selective-soldering-systems?con=t, Accessed on 12 June 2013

[8] T. C Chang and R. A Wysk, “Computer-Aided Manufacturing”, *Association for Computer Machinery*, 1997

[9] K. Mitzner, “Introduction to PCB Design and CAD,” *Complete PCB Design Using OrCAD Capture and PCB Editor*, pp. 1-13, Dec. 2009

[10] Unitechnologies, Microflame Soldering Technology, Available at: https://unitechnologies.com/en/product/microflame/, Accessed on 28 December 2018

[11] Fancort Industries Inc., Laser Soldering, Available at: https://fancort.com/Products/Robotic-Soldering/Laser-Soldering.aspx, Accessed on 28 December 2018

[12] N. S. S Mar, Masters by Research dissertation, "Vision-based Classification of Solder Joint Defects," School of Engineering Systems, Queensland University of Technology, Brisbane, Australia , 2010.

[13] T. Y Ong, Z. Samad and M. M Ratnam, "Solder joint inspection with multi-angle imaging and an artificial neural network," *International Journal of Advanced Manufacturing Technology*, vol. 38, no. (5-6), pp. 455-462, Aug. 2008.

[14] Kuk Won Ko and Hyung Suck Cho, "Solder Joints Inspection Using a Neural Network and Fuzzy Rule-Based Classification Method", *IEEE Transactions on Electronics Packaging Manufacturing*, vol. 23, no. 2, Apr. 2000

[15] IPC. 2014. IPC. Available at: http://shop.ipc.org/IPC-A-610F-English-P , Accessed on 15 April 2015

[16] IPC, Association Collecting Electronics Industries, http://www.ipc.org/ContentPage.aspx?pageid=J-STD-001, Accessed 05 May 2016

[17] Direct Industry, SMT pick-and-place machine, Available at: http://www.directindustry.com/industrial-manufacturer/smt-pick-and-place-machine-210849.html, Accessed on 28 December 2018

[18] SPEA, Flying Probe Testers, Available at: http://www.spea.com/BoardTest/ProductsbyFunction/FlyingProbeTesters/tabid/426/language/en-US/Default.aspx, Accessed on 28 December 2018

[19] Yush Electronic Technology CO.Ltd, Available at: http://www.hk-yush.com, Accessed on 28 December 2018

[20] R. Katukam and P. Sindhoora, "Image Comparison Methods & Tools: A Review," *National Conference on, Emerging Trends in Information Technology[ETIT]*, 28th December 2015, pp. 35-42

[21] R. Brunelli, "Template Matching Techniques in Computer Vision- Theory and Practice," *WILEY*, 2009

[22] N. Perveen, D. Kumar and I. Bhardwaj, "An Overview on Template Matching Methodologies and its Applications," *International Journal of Research in Computer and Communication Technology*, Vol. 2, no. 10, Oct. 2013

[23] A. Banharnsakun and S. Thanathong, “Object detection based on template matching through use of best-so-far ABC,” *Journal of Computer Intelligence and Neuroscience*, Vol. 2014, no. 7, Jan. 2014

[24] L. Lamberti, M. T Ling and C. Ferlong et. Al,” Advancement of Optical Methods & Digital Image Correlation in Experimental Mechanics,” *Springer Proceedings of Annual Conference on Experimental and Applied Mechanics*, Vol. 3, 2018

[25] G. Bradski and A. Kaehler, “Learning OpenCV,” *O’Reilly*, Oct. 2008

[26] A. Aichert, “Feature Extraction Techniques”, *CAMP Medical Seminar WS0708*, 9 Jan. 2008

[27] T. Kadir and M. Brady,” Saliency, Scale and Image Description,” *International Journal of Computer Vision*, Vol. 45, no. 2, pp. 83-105, Nov. 2001

[28] M. Bicego, A. Lagorio, E. Grosso and M. Tistarelli, “On the Use of SIFT Features for Face Authentication,” *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, 17-22 Jun. 2006, DOI: https://ieeexplore.ieee.org/document/1640475

[29] D.G. Lowe,” Distinctive Image Features from Scale-Invariant Keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, Nov. 2004

[30] E. Oyallen and J. Rabin, “An Analysis of the SURF Method,” *IPOL Journal*, vol. 5, pp. 176-218, Jul. 2015

[31] H. Bay, T. Tuytelaars and L. V. Gool, “SURF: Speeded Up Robust Features,” *European Conference on Computer Vision*, vol. 3951, pp. 404-417, 2006

[32] E. Rosten and T. Drummond, “Machine Learning for High-Speed Corner Detection,” *European Conference on Computer Vision, Springer*, vol. 3951, pp. 430-443, DOI: https://doi.org/10.1007/11744023_34

[33] H. Abdi and L. J Williams, “Principal component analysis,” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433-459, 15 Jul. 2010

[34] A. Hyvarinen and E. Oja, “Independent component analysis: algorithms and applications,” *ELSEVIER- Neural Networks*, vol. 13, no. 4-5, pp. 411-430, Jun. 2000

[35] G. E. Sotak and K. L. Boyer, “The laplacian-of-gaussian kernel: A formal analysis and design procedure for fast, accurate convolution and full-frame output,” *ELSEVIER:*

Computer Vision, Graphics, and Image Processing, vol. 48, no. 2, pp. 147-189, Nov. 1989

[36] Y. Xiaoxia, W.H Brian, H. Jing, Z. Yanchun and A. Derek, "Accurate Image Analysis of the Retina Using Hessian Matrix and Binarisation of Thresholded Entropy with Application of Texture Mapping", *PLOS*, April 2014, DOI: https://doi.org/10.1371/journal.pone.0095943

[37] B. S.Morse, "Differential Geometry", Lecture 11, Brigham Young University 1998-2000

[38] D. E. Brown, The Hessian matrix: Eigenvalues, concavity, and curvature, *BYU Idaho Dept. of Mathematics*, April 2014

[39] M. Calonder, V. Lepetit, C. Strecha and P. Fua, "BRIEF: Binary Robust Independent Elementary Features", *CVLab, EPFL*, Lausanne, Switzerland

[40] H.D Heng, X.H Jiang and Y. Sun et.al," Color image segmentation: advances and prospects," *ELSEVIER- Pattern Recognition*, vol. 34, no. 12, pp. 2259-2281, Dec. 2001

[41] N. A. Ibraheem, M. M. Hassan, R. Z. Khan and P. K. Mishra, "Understanding Color Models: A Review," *ARPN Journal of Science and Technology*, vol. 2, no. 3, Apr. 2012

[42] T. S. Peng and C. S. Fuh, "Color-based Printed Circuit Board Solder Segmentation," Dept. of Computer Science and Information Engineering, National Taiwan University

[43] Y. Akbulut, Y. Guo, A. Sengur and M. Aslan, "An effective color texture image segmentation algorithm based on hermite transform," *ELSEVIER- Applied Soft Computing*, vol. 67, pp. 494-504, 13 Mar. 2018, DOI: https://doi.org/10.1016/j.asoc.2018.03.018

[44] Y. Delignon, A. Marzouki and W. Pieczynski, "Estimation of generalized mixtures and its application in image segmentation," *IEEE Transactions on Image Processing*, vol. 6, no. 10, pp. 1364-1375, Oct. 1997

[45] J. Smolka, "Watershed based region growing algorithm," Annales Universitatis Mariae Curie-Skłodowska, Sectio AI, Informatica, vol. 3, pp. 169-178, 2005

[46] J. Zhang, J. Zheng and J. Cai, "A diffusion approach to seeded image segmentation", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 13-18 Jun. 2010, DOI: https://ieeexplore.ieee.org/document/5539891

[47] W.H Elmasry, H.M Mofta, N. EI-Bendary and A.E Hassanien, "Graph Partitioning based Automatic Segmentation Approach for CT Scan Liver Images," *IEEE Federated Conference on Computer Science and Information Systems*, pp.205-208, 2012

[48] M. Borsotti, P. Campadelli and R. Schettini, "Quantitative evaluation of color image segmentation results," vol. 19, no. 8, pp. 741-747, Jun. 1998

[49] T. Kanungo and D.M Mount et.al, "An Efficient k-Means Clustering Algorithm: Analysis and Implementation," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 24, no. 7, pp. 881-892, Jul. 2002

[50] J. Qi, Y. Yu and L. Wang, "An effective and efficient hierarchical k-means clustering algorithm", *International Journal of International Sensor Networks*, 31 Aug. 1017, DOI: https://doi.org/10.1177/1550147717728627

[51] S. S. Khan and A. Ahmed, "Cluster center initialization algorithm for k-means clustering", *ELSEVIER Pattern Recognition Letters*, vol. 25, no. 11, pp.1293-1302, Aug. 2004

[52] R. Jin, A. Goswami and G. Agrawal, "Fast and exact out-of-core and distributed k-means clustering", *Springer Link: Knowledge and Information Systems*, vol. 10, no. 1, pp. 17-40, Jul. 2006

[53] J.C Bezdek, R. Ehrlich and W. Full, "FCM: The fuzzy c-means clustering algorithm," *ELSEVIER: Computers & Geosciences*, vol. 10, no. 2-3, pp. 191-203, 1984

[54] N. R. Pal, K. Pal, J. M. Keller and J. C. Bezdek, "A possibilistic fuzzy c-means clustering algorithm", *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 4, pp.517-530, Aug. 2005

[55] K. S. Chuang, H. L. Tzeng, S. Chen, J. Wu and T. J. Chen, "Fuzzy c-means clustering with spatial information for image segmentation", *ELSEVIER Computerized Medical Imaging and Graphics*, vol. 30, no. 1, pp.9-15, Jan. 2006

[56] D. Comaniciu and P. Meer, "Mean shift analysis and applications," *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 20-27 Sept. 1999, DOI: https://doi.org/10.1109/ICCV.1999.790416

[57] J. Bruce, T. Balch and M. Veloso, “Fast and inexpensive color image segmentation for interactive robots,” *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS 2000) (Cat. No.00CH37113), 31 Oct. 5-Nov. 2000, DOI: https://ieeexplore.ieee.org/document/895274

[58] P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient Graph-Based Image Segmentation,” *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167-181, Sep. 2004

[59] J. Ning, L. Zhang, D. Zhang and C. Wu, “Interactive image segmentation by maximal similarity based region merging”, *ELSEVIER Pattern Recognition*, vol. 43, no. 2, pp. 445-456, Feb. 2010

[60] Y. Boykov and G. Funka-Lea, “Graph Cuts and Efficient N-D Image Segmentation”, *Springer Link: International Journal of Computer Vision*, vol. 70, no. 2, pp.109-131, Nov. 2006

[61] N. Xu, N. Ahuja and R. Bansal, “Objects segmentation using graph cut based active contours”, *ELSEVIER Computer Vision and Image Understanding*, vol. 107, no. 3, pp. 210-224, Sep. 2007

[62] N. Broutin, L. Devroye and E. Mcleish, “Note on the Structure of Kruskal’s Algorithm,” *ELSEVIER: Algorithmica*, vol. 56, no. 141, 2010, DOI: https://doi.org/10.1007/s00453-008-9164-4

[63] A. Mariano and D. Lee et.al, “Hardware and Software Implementations of Prim’s Algorithm for Efficient Minimum Spanning Tree Computation,” *Springer: Embedded Systems: Design, Analysis and Verification*, vol. 403, 2013, DOI: https://doi.org/10.1007/978-3-642-38853-8_14

[64] Peter Corke, ENB339 lecture 9: Image geometry and planar homography, 2012. Accessed on: Dec. 20, 2020. [Streaming Video]. Available: www.Youtube.com

[65] G. G. L Priya and S. Dominic, “Video cut detection using block based histogram differences in RGB color space”, *International Conference on Signal and Image Processing*, 15-17 Dec. 2010, DOI: https://ieeexplore.ieee.org/document/5697436

[66] Z. Xu, X. Liu and N. Ji, “Fog Removal from Color Images using Contrast Limited Adaptive Histogram Equalization”, *IEEE 2009 2nd International Congress on Image and Signal Processing*, 17-19 Oct. 2009, DOI: https://ieeexplore.ieee.org/document/5301485

[67] W. Jia, H. Zhang, X. He and Q. Wu, “Refined Gaussian Weighted Histogram Intersection and Its Application in Number Plate Categorization”, *IEEE International Conference on Computer Graphics, Imaging and Visualisation*, 26-28 Jul. 2006, DOI: https://ieeexplore.ieee.org/document/1663800

[68] Z. Zivkovic and F. V. D. Heijden, “Efficient adaptive density estimation per image pixel for the task of background subtraction”, *ELSEVIER Pattern Recognition Letters*, vol. 27, no. 7, pp.773-780, May 2006

[69] N. A. Mandellos, I. Keramitsoglou and C. T. Kiranoudis, “A background subtraction algorithm for detecting and tracking vehicles”, *ELSEVIER Expert Systems with Applications*, vol. 38, no. 3, pp. 1619-1631, Mar. 2011

[70] K. Kim, T. H. Chalidabhongse, D. Harwood and L. Davis, “Background modeling and subtraction by codebook construction”, *IEEE International Conference on Image Processing*, 2004

[71] A. Graves, A. R. Mohamed and G. Hinton, “Speech recognition with deep recurrent neural networks”, *IEEE International Conference on Acoustic, Speech and Signal Processing*, 26-31 May 2013, DOI: https://ieeexplore.ieee.org/document/6638947

[72] L. A. Gatys, A. S. Ecker and M. Bethge, “Image Style Transfer Using Convolutional Neural Networks”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2414-2423

[73] O. Ronneberger, P. Fischer and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation”, *Springer Link International Conference on Medical Image Computing and Computer Assisted Intervention*, pp. 234-241, 18 NOV. 2015

[74] G. Levi and T. Hassner, “Age and Gender Classification Using Convolutional Neural Networks”, *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 34-42, 2015

[75] S. C. Huang, F. C. Cheng and Y. S. Chiu, “Efficient Contrast Enhancement Using Adaptive Gamma Correction With Weighting Distribution”, *IEEE Transactions on Image Processing*, vol. 22, no. 3, pp. 1032-1041, Mar. 2013

[76] H. Farid, “Blind inverse gamma correction”, *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1428-1433, Oct. 2001

[77] Y. T. Kim, “Contrast enhancement using brightness preserving bi-histogram equalization”, *IEEE Transactions on Consumer Electronics*, vol. 43, no. 1, pp. 1-8, Feb. 1997

[78] S. D. Chen and A. R. Ramli, “Minimum mean brightness error bi-histogram equalization in contrast enhancement”, *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 1310-1319, Nov. 2003

[79] H. Ibrahim and N. S. P. Kong, “Brightness Preserving Dynamic Histogram Equalization for Image Contrast Enhancement”, *IEEE Transactions on Consumer Electronics*, vol. 53, no. 4, pp. 1752-1758, Nov. 2007