CHARACTERIZATION OF ENGINEERED SURFACES

by

Ritwik Verma

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Mechanical Engineering

Charlotte

2010

Approved by:

____
Dr. Jay Raja

____
Dr. Robert Wilhelm

____
Dr. Faramarz Farahi

____
Dr. Brigid Mullany

____
Dr. Rajaram Janardhanam

ABSTRACT

RITWIK VERMA. Characterization of engineered surfaces. (Under the direction of DR. JAY RAJA)

In the recent years there has been an increasing interest in manufacturing products where surface topography plays a functional role. These surfaces are called engineered surfaces and are used in a variety of industries like semi conductor, data storage, micro-optics, MEMS etc. Engineered products are designed, manufactured and inspected to meet a variety of specifications such as size, position, geometry and surface finish to control the physical, chemical, optical and electrical properties of the surface. As the manufacturing industry strive towards shrinking form factor resulting in miniaturization of surface features, measurement of such micro and nanometer scale surfaces is becoming more challenging. Great strides have been made in the area of instrumentation to capture surface data, but the area of algorithms and procedures to determine form, size and orientation information of surface features still lacks the advancement needed to support the characterization requirements of R&D and high volume manufacturing.

This dissertation addresses the development of fast and intelligent surface scanning algorithms and methodologies for engineered surfaces to determine form, size and orientation of significant surface features. Object recognition techniques are used to identify the surface features and CMM type fitting algorithms are applied to calculate the dimensions of the features. Recipes can be created to automate the characterization and process multiple features simultaneously. The developed methodologies are integrated into a surface analysis toolbox developed in MATLAB environment. The deployment of the developed application on the web is demonstrated.

ACKNOWLEDGEMENT

I express my sincere gratitude to Dr. Jay Raja for his support and guidance during my doctoral work. His tireless enthusiasm has been a constant source of encouragement and motivation. I thank Dr. Robert Wilhelm, Dr. Faramarz Farahi, Dr. Brigid Mullany and Dr. Rajaram Janardhanam for agreeing to serve on my committee.

This work was funded by Affiliates to the Center of Precision Metrology, Department of Mechanical Engineering and Engineering Science, UNC Charlotte.

I want to thank Veeco Instruments for providing the datasets used in this work. Additionally, I would like to thank the present and past students of the Center for Precision Metrology Dr. Bala Muralikrishnan, Dr. Son Bui, Dr. Shen Fu, Dr. George Orji, Mohamed Saad and Suresh Ramasamy for enlightening discussions and suggestions.

I would like to thank my wife, Suverna, for all she has done to help me reach my goals.

I will always be thankful to my parents Mrs. Asha & Mr. Jagdish Verma and my sister Richa for their support and patience. I want to express my love for my nephew Shashwat and niece Shrawani for bringing joy and happiness into my life.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 Introduction

In the recent years there has been an increasing interest in manufacturing products where surface topography plays a functional role. These surfaces are called engineered surfaces and are used in a variety of industries like semi conductor, data storage, micro-optics, MEMS etc. Engineered products are designed, manufactured and inspected to meet a variety of specifications such as size, position, geometry and surface finish to control the physical, chemical, optical and electrical properties of the surface. As the manufacturing industry strive towards shrinking form factor resulting in miniaturization of surface features, measurement of such micro and nanometer scale surfaces is becoming more challenging. Great strides have been made in the area of instrumentation to capture surface data, but the area of algorithms and procedures to determine form, size and orientation information of surface features still lacks the advancement needed to support the characterization requirements of R&D and high volume manufacturing.

Evans and Bryan [1] defined engineered surfaces as “surfaces where the manufacturing process is optimized to generate variation in geometry and /or near surface material properties to give a specific function”. All manufactured surfaces are designed to perform specific functions. For example the dimples on the golf ball deliver distance by affecting air flow over the ball’s surface. The surface of the privacy filters for computer screen prevents the information on the computer to be viewed at an angle. Brightness enhancement film helps laptop computers run more efficiently. The textured film on the backlit screen helps direct light toward the user rather than off to the sides. Prismatic retro-reflective films are designed for highway and commercial markets to reduce the scattering of light and reflect three times as much light as traditional sign materials so that safety and direction signs can be viewed from longer distance. Plateau honed surfaces of engine cylinders have better oil retention capability [2]. MEMS devices are being used for several applications like micro-fluidics in which micro channels are made in silicon for precise control of fluids [3], micro-gears have been successfully implemented in a wide variety of miniaturized servo actuators.etc. [4]. Light piping is used to illuminate lamp posts and guard rails. MEMS devices like Gyroscopes, Resonators, Micro-mirrors, Accelerometer, Variable capacitor, Actuator etc. are finding applications in a wide variety of industries.

To achieve the desired functionality, various complex geometric shapes are created on the surface. Some common shapes are micro-lenses used for optical interconnects, wavefront sensors and imaging, solder bumps for electrical connection, posts to retain solid or liquid substance, micro-prisms to control light reflectivity and scattering used in rear-projection LCDs, traffic control and sign sheeting, diffusers in luminaires, solar collector arrays, via and trenches in semiconductor devices to control fluid flow etc.

The creation of these complex micro-features to provide functionality to the surface has become possible with the advent of new manufacturing technologies like MicroSystem Technologies (MST) and MicroEngineering Technologies (MET). MST and MET consists of manufacturing methods like replication, lithography, diamond machining etc. to create various complex geometric shapes (e.g. Micro Electro Mechanical Devices (MEMS)) and patterned feature surfaces (e.g. Structured Surfaces). These technologies allow covering surfaces of large sheets of plastic, rubber, glass and other mouldable materials with microscopic complex shapes raised features. Packed thousands in a square inch, the shapes are arrayed in patterns that improve the performance of a wide range of everyday products.

Due to the increasing nature of complexity of the surface features and the functional role they play in the performance it is required to inspect dimensional characteristics of features on the surface. Contact and non-contact measurement techniques are typically employed to measure microstructures. Commonly available techniques are digital holography and fringe reflection for low accuracy and high speed measurement, confocal microscopy and non contact optical profilometry for medium accuracy and medium speed measurements, and contact profilometer and Twyman-Green Interferometer which are high accuracy and low speed measurement. Scanning Electron Microscope (SEM), and Vision CMM are also used for lateral dimension measurement. High slope of the sidewall and large aspect ratio makes it challenging to get data. Output data is a height map of the measured surface at a given magnification. New interferometric technique that uses index matching liquid for faceted microstructure metrology can measure high slopes at high speed but at low accuracy [35].

After collecting raw data, the next step is to find features of interest and determine texture, form, size and orientation characteristics. These characteristics are compared with the design specification to make pass fail decision about the part. Measurement instruments generally come with a proprietary application and data analysis toolbox containing various techniques and procedures to help the users inspect the quality of their parts. The currently available analysis toolbox contains a gamut of tools like surface texture parameters, histogram analysis, bearing ratio curve, power spectral density etc. These techniques although quite useful for understanding the performance of surface produced using conventional machining techniques but have limited capability to address the shape, size and orientation characteristics of micro-features. Extracting dimensional characteristics with the available techniques is a manual, tedious and time consuming job and lacks automation. Current methods are unsuitable for high volume inspection. There is a need for development of advanced algorithms and procedures that are generic, intelligent and fast in characterizing micro and nano scale features.

1.2 Surface Classification

The number and diversity of technologies to produce micro-components and microproducts is enormous. These could be classified first as top-down manufacturing methods; starting from bigger building blocks and reducing them into smaller pieces. Second classification is bottom-up manufacturing methods, in which small pieces are added for the construction of bigger functional structures and third as either the development of entirely new technologies or combination of existing technologies. Brinksmeier’s classification [5] is currently one of the most widespread. In this classification the machining of precision parts and micro structures is subdivided into two general types of technologies having different origins; Micro System Technologies (MST) and Micro Engineering Technologies (MET). MST is qualified for the manufacture of products of Micro Electro Mechanical Systems (MEMS) and Micro Opto Electro Mechanical Systems (MOEMS). MET comprise the production of highly precise mechanical components, moulds and microstructured surfaces. Other significant surface classifications reported are based on the working principle of micromachining technologies by Mazusawa [6], traditional and non-traditional miniaturization techniques by Madou [7]. Evans and Bryan [1] classified structured surfaces by their manufacturing methods.

Due to the large variety of engineered surfaces and diversity of technologies to produce them we have tried to classify the engineered surfaces on the basis of types of algorithms, procedures and methodologies required to characterize the surface. Based on experimental work done in this dissertation it has been observed that procedure to characterize a surface depends upon several factors like geometry of surface features, quality of data, metrological characteristics required etc. Based on these factors engineered surfaces are classified into three groups as shown in Figure 1. Class I is Textured Surfaces, Class II is Structured Surfaces and Class III is MEMS.

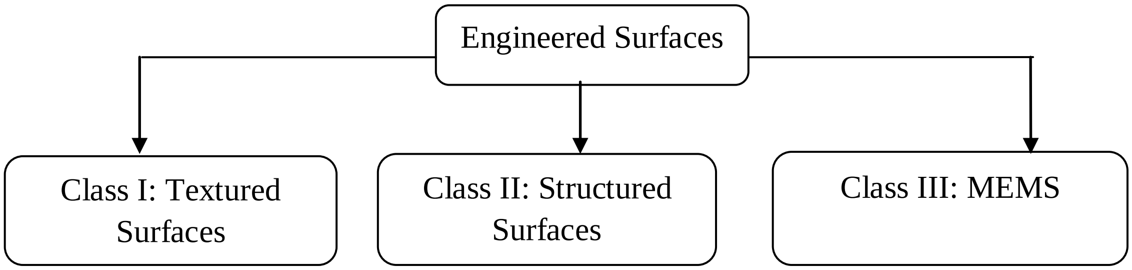


Figure 1: Engineered Surface Classification

The first category in this classification is of surfaces produced using conventional machining techniques where the quality of the surface is often of the utmost importance for the correct functioning of the part and where texture word is restricted to its current standardized meaning of roughness, waviness and lay. Data analysis tools like surface texture parameters, histogram analysis, bearing ratio curve, power spectral density etc.

are suitable to characterize the surface. Class 2 is the structured surface which may be considered as “patterned surface” with some regular array of surface height features. Another characteristic is that they typically have high aspect ratios. Class 3 surfaces are MEMS type of surfaces which have unconventional features geometries, are non-periodic and the features are at different planar levels. For both class 2 and class 3 surfaces, their function cannot be related to traditional finish parameters.

1.2.1 Class I Surfaces

Conventional machining is the broad term used to describe removal of material from a workpiece. It covers several processes like turning, milling, grinding, polishing, honing etc. These surfaces can be well defined by quantifying their texture i.e. roughness, waviness and lay. A lot of work has been done in the past on the relationship between surface texture and function. Measurement of surface finish of conventionally machined surfaces for functional performance as adhesion, paintability, friction, lubrication and wear is done because for the above mentioned applications surface finish is of the utmost important for the correct functioning of the part [10,11]. Data analysis tools like surface texture parameters, histogram analysis, bearing ratio curve, power spectral density etc are suitable to characterize these surfaces. Figure 2 is a plateau honed surface.

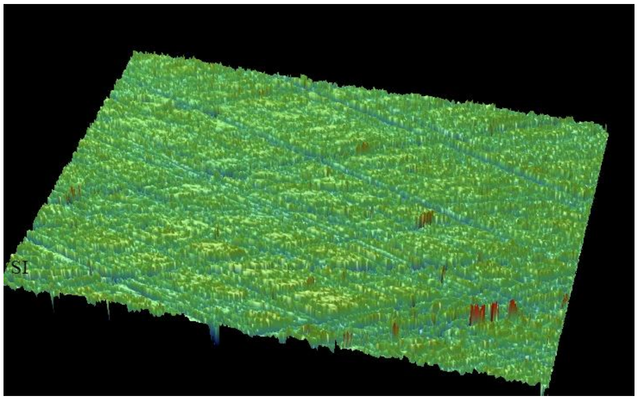


Figure 2: A typical machined surface

For a long time surface metrology has been based upon contact measurement using 2D profilometers [8]. Two dimensional surface roughness parameters were developed in conjunction with the development of the tool to quantify the texture information. Over the past twenty years the appearance of 3D profilometers and non-contact gauges has resulted in the development of three dimensional surface texture parameters initially proposed by Stout et.al [9]. These parameters are extension of their 2D counterparts. The standard was prepared by work group WG16 of the ISO (International Organisation for Standardisation) technical committee TC213. Table 1 shows the list of three dimensional surface texture parameters also called S parameters.

Table 1: S Parameters

Amplitude	Hybrid	Spatial	Functional
Sa - Average Roughness	Sdq – Root Mean Square Surface Slope	ACF – Autocorrelation Function	Sm – Surface Material Volume
Sq – Root Mean Square Roughness	Ssc – Mean Summit Curvature	Sds – Summit Density	Sc – Core Void Volume
Ssk - Skewness	Sdr – Developed Interfacial Area Ratio	Str – Texture Aspect Ratio	Sv – Surface Void Volume
Sku - Kurtosis		Sal – Auto-correlation Length	Sbi – Surface Bearing Index
Sz – Ten Point Height		APSDF – Angular Power Spectral Density Function	Sci – Core Fluid Retention Index
		Std – Texture Direction	Svi – Valley Fluid Retention Index

1.2.2 Class II Surfaces

The last 20 years have seen numerous products emerging with significant improvement in design and performance over their traditional design and are growing in importance. These surfaces have regular array of height features and may also be called patterned surfaces or structured surfaces. Examples include privacy filters for computers, golf balls, road signs etc. These products have specifically designed microstructures to provide the required functionality.

These surfaces have extremely precise microstructures in a continuous, repeating pattern, often in the form of tiny protrusions, depressions, ridges or holes. Manufacturing techniques provide micron-scale precision, high-structure density, high volume and low cost. These surfaces have also been called as structured surfaces and defined as "surfaces with a deterministic pattern of usually high aspect ratio geometric features designed to give a specific function [1]. Surfaces are periodic in at least one dimension.

Figure 3 shows examples of structured surfaces. Prismatic retro-reflective films are designed for specific highway and commercial markets. Microreplicated structures can be used to manage light propagation in displays, optical switches, and other applications. Microlens arrays are often used to increase the light collection efficiency.

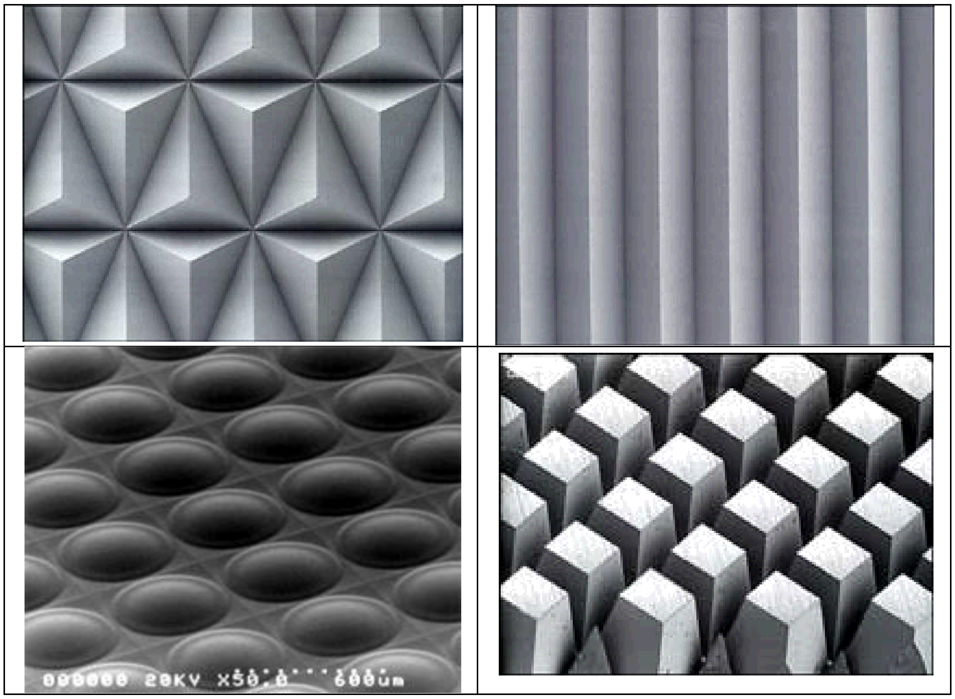


Figure 3: Structured Surfaces [12, 13]

There is a broad range of applications and manufacturing methods for structured surfaces. These surfaces offer new approach for a variety of applications some of which are shown in Table 2.

TABLE 2: Application of Engineered Surfaces [1]

Function	Example
Optical	Gratings, Fresnel Lenses, Diffractive Optics, Reflective road signs, Filters,
Mechanical	Vacuum chucks, Seal surfaces, Diesel injectors, Piston rings/cylinder liners,
Hydrodynamics	Tire treads, Drag reduction film, Deck shoes, Golf balls
Friction & Wear	Undulated surfaces, Abrasives, tools, files
Biological	Breast implants, Bio-MEMS, fluidics

1.2.3 Class III Surfaces

Micro-electro-mechanical systems (MEMS) are the micro-machined miniaturized devices that have the capability to sense the environment. The information thus received is processed to generate a response with a variety of mechanical and electrical actuators. Taking advantage of technology used in the integrated circuit (IC) industry, researchers have developed methods to micro-machine miniaturized devices which could potentially displace or improve upon their macroscopic counterparts [16]. Since the 1980s, micro-electro-mechanical systems (MEMS) have been introduced to the mass market in various applications such as pressure sensors, accelerometers, micro-valves, and ink jet nozzle heads.

Technological advances that are specific to MEMS devices have enabled researchers to fabricate devices with ever increasing aspect ratios (defined as the ratio between the structural height and the minimum lateral dimension of the part). Currently, the LIGA process is seen as the best way to fabricate devices with very tall, parallel sidewalls (up to a few mm) such as those shown in Figure 4 [14].

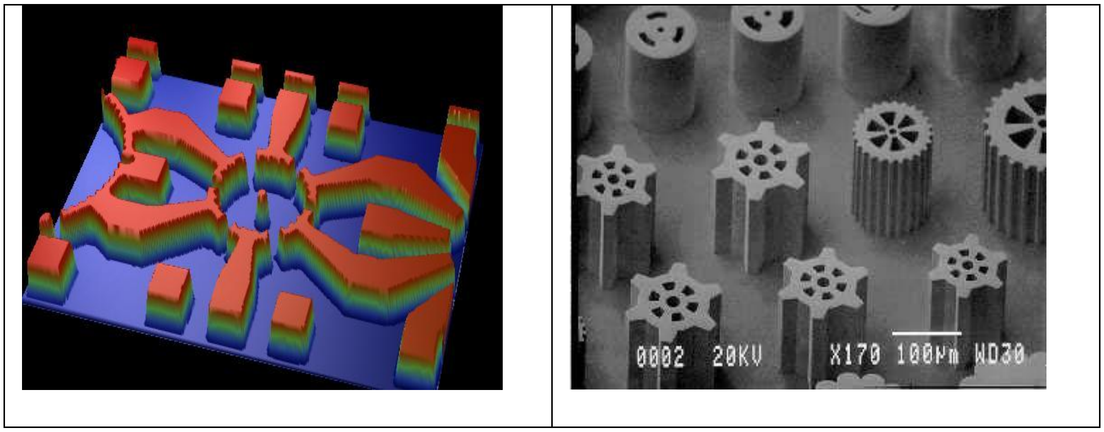


Figure 4: MEMS & LIGA manufactured parts [15, 16]

MEMS find its application in various applications such as pressure sensors, accelerometers, micro-valves, and ink jet nozzle heads [10].

Table 3: Applications of MEMS Devices

Function	Example
Automotive	Airbags, GPS, Vehicle stability control, Tire pressure monitoring
Consumer	iPods, notebooks, Playstations, HDTV
Industrial	Bio-MEMS, Fluidics

1.3 Micro Manufacturing

The demand for efficient and cheaper micro-components and micro-products has played a vital role in the development of diverse manufacturing technologies. These technologies can be classified based on different characteristics of manufacturing process. In principle a micro-product is created either by depositing material called bottom-up approach or by removing material called top-down approach. The third classification involves the development of entirely new technologies or combination of existing technologies [18]. Micromachining techniques used for fabrication of 3D structures on micrometer scale can be grouped as:

a. MEMS processes, like UV-lithography, silicon-micromachining and LIGA.

b. Energy assisted processes like Laser Beam Machining, Focused Ion Beam Machining, Electron Beam Machining and Micro Electro Discharge Machining.

c. Mechanical processes like Diamond Machining (e.g. turning, milling, drilling and polishing), micromilling or microgrinding.

d. Replication techniques like forming, injection moulding or casting.

Most manufacturing techniques can trace their origin to the semi-conductor or IC industry. These techniques were then adopted and modified by other industries like MEMS, micro-optics, data storage etc. to suit their needs. Material layer is deposited on the wafer by putting wafers in a coating chamber and exposing them to metal coating at controlled temperature, pressure and humidity. This is followed by selective removal of material from certain areas of the wafer. Deposition is done using one of the following techniques (i) Physical vapour deposition (PVD), Chemical vapour deposition (CVD), (iii) Electro-chemical deposition etc. Then material is removed from selected locations using one of the two processes (i) wet etching and (ii) dry etching. A photo-mask having desired pattern in combination with photo-resist and UV or X-ray light is used to etch away parts of deposited layer creating a pattern on the substrate. This process is called Lithography. Emerging technologies like stereo lithography are making their way in the industries as a means to build 3D parts. The lateral resolution and accuracy of lithography masks is 0.1um or better.

LIGA is a German acronym for LIthographie Galvanoformung Abformung, which translates to Lithography, Electroforming and Molding in English. Deep X-ray lithography is used to produce high aspect ratio features (1 mm tall and a lateral resolution of 0.2um.). X-ray breaks chemical bonds in the resist. Exposed resist is dissolved using wet-etching process. Either the substrate itself can be etched to generate desired pattern or deposited layers are etched resulting in high aspect ratio feature creation.

True 3D surfaces can be produced by mechanical machining process. Two approaches used to machine micron and sub-micron scale features is by (i) using ultra precision diamond turning machines and (ii) Micro-lathe, micro-mill, micro-EDM techniques. The use of diamonds to generate precision patterns and precision surfaces on a micrometer or nanometer scale has been discussed by Davies et.al [17].

Microreplication technology is used to produce high aspect ratio precision micro-structured geometries in a variety of polymer films. Micro-features are cut on a metal master using a single point diamond turning machine. The pattern generated on the master is then transferred on to a substrate by a mould. This process is called replication.

Producing high aspect ratio precision microstructures is a complex multi-step process that involves masking, tooling, mastering and micro-embossing. There are usually several iterations in the microstructure design and compromises need to be made between desired geometry of the structure and ability of the pattern to be accurately replicated [18]. Energy assisted processes like Laser Beam Machining, Focused Ion Beam Machining, Electron Beam Machining and Micro Electro Discharge Machining are also used in manufacturing of MET surfaces.

Sophisticated micro scale manufacturing technologies mentioned above are enabling production of miniaturized parts for variety of applications. Micro-manufacturing offers a new dimension for manufacturing in general. It fulfills the strong demand from the global market for ever-smaller parts and systems at reasonable cost and with superior performance.

1.4 Micro-feature Metrology

All manufacturing processes require metrology to control part dimensions. The measured dimensions are compared against the design specifications to check the quality of the part and for understanding the performance of micro- fabricated products and micro-fabrication processes. Micro-feature metrology finds its application in a variety of industries like Semiconductor, MEMS, Micro-optics, Aerospace, Biological, Data Storage etc. Dimensional and positional accuracies in these industries are in the sub-micron to nanometer range. Figure 5 shows the techniques widely used to capture surface data.

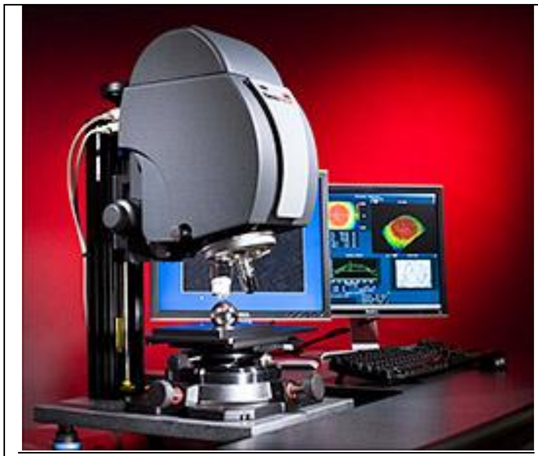
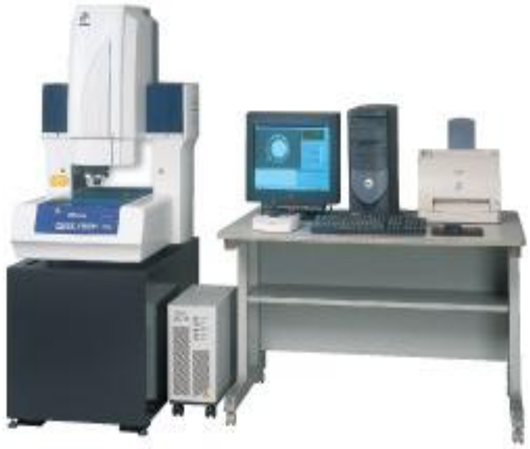
	
White Light Interferometer	Vision CMM

Figure 5: Measurement Equipments

1.4.1 Instrumentation

Commonly used metrology tools are based on contact and non-contact methods. Popular contact tools are surface profilers, AFM and coordinate measuring machines (CMM). Contact CMMs are generally used to measure small to large size, complex shape three dimensional parts. Surface profilers have sub nanometer vertical resolution lateral resolution down to 1 micron. They are used to measure shapes of microfeatures where the slope of the sidewall changes gradually. A diamond tip is used to scan the surface. Tip is specified by the radius of the curvature and cone angle. Smaller the cone angle, higher slopes can be measured. Stylus profiler can measure taller vertical steps than those possible with AFM. AFMs have sub angstrom vertical resolution and sub nanometer lateral resolution. The advantages of contact method are the high range and resolution, high slope measuring capability and insensitivity to the optical properties of the material. On the other hand it is a two dimensional measurement. It is slow and stylus can scratch the surface.

Non-contact methods are preferred choice for micro-feature measurement because they do not deform the surface and are free from mechanical errors. There are several non-contact measurement techniques available like white light interferometry (WLI), confocal microscopy, vision CMM and scanning electron microscopy (SEM) etc.

White light interferometry, or white light profiling is widely used in research and production environment. It is often preferred for its combination of non- contact measurement, repeatable 3D surface measurement, speed, and sub - nanometer resolution. This method is employed for surfaces with average roughness down to 0.1nm and peak to valley heights up to several millimeters, with repeatability of 0.1nm or better

[19]. It can measure parts up to 10mm tall using Vertical Scanning Interferometry (VSI) technique. For smooth surfaces (step less than 160nm) Phase Shifting Interferometry (PSI) can be used. The area measured depends on the magnification used. Higher the magnification, lower the surface area measured. To measure larger surfaces with this method, a number of regions must be acquired and stitched together to form a view of the entire surface [20]. The drawback of WLI technique is that it cannot measure high slopes. With a 50x objective maximum 27 degrees of slope can be measured. Also it is difficult to measure poorly reflective parts.

Confocal microscopy is now getting attention due to its ability to measure large slopes even greater than 60 degrees and also parts that cannot be measured using WLI due to poor reflectivity. Confocal technique is yet to establish itself in a high volume manufacturing environment. All confocal microscopy provides spatial resolution in the z-direction (out of the image plane) by using an aperture to exclude from the image light that does not originate in the object plane of the optical system. To do so, it must acquire data sequentially as the aperture is positioned over all points in the object. Various confocal technologies accomplish this in different ways, but the sequential nature of the acquisition imposes a fundamental limit on speed. Disk-based systems offer a significant speed advantage because they acquire data simultaneously from multiple apertures arranged in a spiral pattern on a rapidly spinning disk [21].

SEM has always been a tool of choice for researchers due to its Angstrom level resolving capability. It is widely used to observe, analyze and measure microstructures at nanometer scale. But the measurements are more qualitative in nature and used more for imaging rather than for metrology. Surfaces that are not conductive are required to be coated with a conductive film to be able to image. It can produce images over a wide range of magnifications and provide morphological, compositional, and physical information. It is one of the primary tools for microstructure metrology. Electrons were first shown to exhibit wave properties proportional to an applied electrostatic potential difference by Davisson and Germer in 1927 [22]. Postek [23] provides a thorough analysis of the capabilities of SEMs. The accuracy of the images captured is highly dependent on machine capability and the specific part being examined. Beam-sample interactions (i.e. charging) are shown to greatly affect the results of any measurement taken with the device. Additionally, despite the high resolutions of the SEM, the output is typically generated from the electron detector and displayed on a cathode ray tube rastered in synchronization with the electron beam. The final result is a two-dimensional image on a screen. Since no coordinate data are directly outputted from the SEM, performing any analysis other than line width measurements directly with the SEM software becomes difficult. SEMs are ideal for visualizing parts, but are inadequate tools for quantitative analysis.

Conventional CMMs can’t be used to measure microstructures because contact probes typically cause enough displacement to prevent accurate results. Vision CMM based micro-metrology is used for lateral dimensional measurements. It can be divided into field of view (FOV) and point-to-point (PTP) measurements. FOV is the area in pixels or micrometers that can be measured or inspected in a single measurement. FOV size depends on the magnification of the objective selected and the numbers of pixels on the image sensor. If the object fits inside the FOV then a single measurement is enough to capture the object. When it is required to measure the distance between two features that do not fit in the field of view then stage is moved. For FOV measurements the measurement precision depends on the lens accuracy while for PTP measurement stage positioning accuracy is more important. Accuracy of a typical high resolution objective is around 1um. Stage positional accuracy of vision CMM can range from 0.8um to 2 um depending upon the stage travel required for the measurement [24]. Ultra precision vision CMM has accuracy of 0.5µm or better. Both FOV and PTP systems require metrology software to perform image analysis to determine points or feature locations with the necessary precision, as well as to control stages and focus mechanisms.

Of primary importance in both FOV and PTP vision metrology is precise focus control to ensure that image features are measured under repeatable focus conditions. Improper focus can inhibit improper edge detection and introduce measurement errors.

After precise focus, repeatable precision lighting is critical in obtaining repeatable measurements. Small fluctuations in lighting might not make it difficult to find an edge, but do make it difficult to measure the edge repeatedly. The same feature measured using different light intensity and colors will affect measurements in the submicron realm.

Measurement tools are available with sub micron lateral resolution and sub nanometer vertical resolution. A survey of commonly used measurement techniques is also discussed by Conor et. al. [25]. Figure 6 shows a survey of available measurement techniques. Different measurement techniques are compared on the basis of accuracy, measurement speed and slope measurement limitation. It is shown that techniques like fringe reflection and digital holography are fast and can measure high slopes but are less accurate. Contact methods are highly accurate and can measure high slopes but are slow and destructive. Non-contact optical profiler provides a good balance of slope measurement capability, accuracy and speed and provides a valuable alternative to these methods.

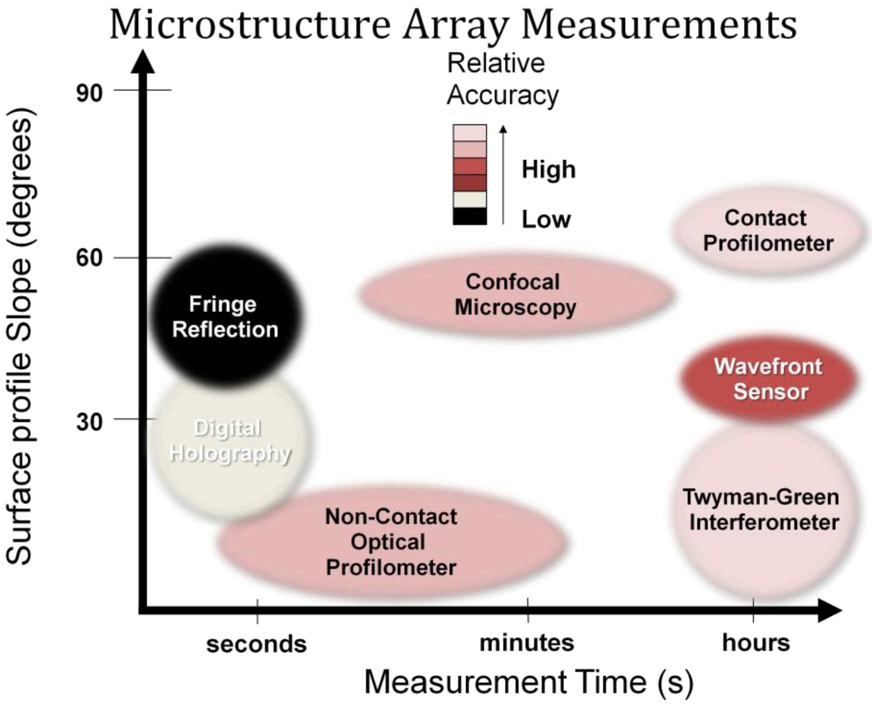


Figure 6: Survey of Measurement Techniques [35]

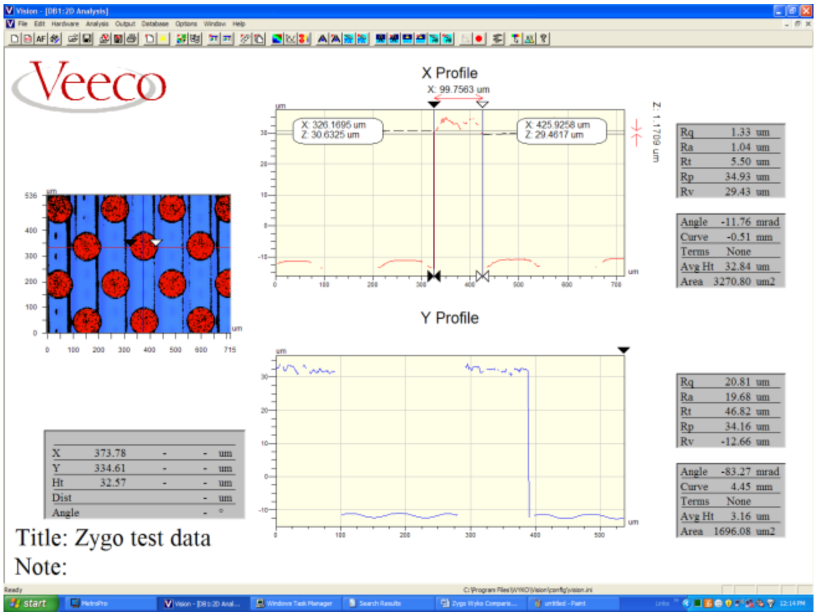
1.4.2 Current surface characterization techniques

Engineered surfaces are also called functional surfaces. Feature shape, size and orientation affect the performance of the surface. To characterize engineered surfaces we need a measurement system to capture data and algorithms to calculate dimensional parameters. For a measurement system the requirement is high lateral and vertical resolution, high accuracy and high throughput. Once the data is taken we need algorithms to calculate metrological characteristics like length, width, depth, angle, radius, distance, position etc. As mentioned in the previous section there are several techniques available to measure parts from sub nanometer level to several microns. The choice of measurement techniques depends upon its suitability for the application. White light interferometry is the tool of choice for three dimensional surface measurements due to its high resolution, speed and accuracy. To validate part geometry, it is crucial that the algorithms and procedure to obtained dimensional parameters are fast, repeatable and reproducible.

Non-contact optical profilers come with analysis software package having functionality to perform various analyses on the raw data. The limitation with using these measurement and analysis software packages is that they produce measurement data in custom file format. This limits the analysis to the techniques available with the measurement system.

Generally the analysis options provided are Autocovariance, Histogram, Bearing Ratio, Power Spectral Density, Surface Texture Parameters etc. which are suitable only for surface texture characterization. Data masking can be done to select the area of interest for analysis. Multiple region analysis features are available which can be used to separate a surface into sub-regions. But the output parameters available are limited to surface texture parameters, height, tilt and other parameters related to texture of the surface.

To calculate the basic dimension like length, width, depth, angle etc. a cross section of the measured surface is used. In Figure 7, an image of solder bumps on a silicon substrate is shown. On the right side a cross section through a row of bumps is displayed. Profile has points at two planar levels like a step. To find out the bump diameter, two vertical lines are drawn contacting a bump on the left and the right side. The horizontal distance between the two lines gives diameter. To determine bump height two vertical lines are drawn, one contacting the substrate and the other contacting the bump. The vertical distance between the two intersection points gives height. Flatness of the bump is the peak to valley distance of the bump data. Pitch is calculated by drawing two vertical lines contacting the edge on the same side of two consecutive bumps. As observed, the process to calculate the dimensional characteristics is very basic, tedious and uses only two dimensional information of a three dimensional measurement.



Rq	1.33	um
Ra	1.04	um
Rt	5.50	um
Rp	34.93	um
Rv	29.43	um

Angle	-11.76	mrad
Curve	-0.51	mm
Terms	None	
Avg Ht	32.84	um
Area	3270.80	um2

Rq	20.81	um
Ra	19.68	um
Rt	46.82	um
Rp	34.16	um
Rv	-12.66	um

Angle	-83.27	mrad
Curve	4.45	mm
Terms	None	
Avg Ht	3.16	um
Area	1696.08	um2

X	373.78	-	-	um
Y	334.61	-	-	um
Ht	32.57	-	-	um
Dist			-	um
Angle			-	°

Title: Zygo test data
Note:

Figure 7: Bump Characterization

It is observed that measurement systems are available with needed resolution to capture data. WLI can generate 3D views of the surface with several details such as height and intensity maps, profile views, image stitching capability, surface texture analysis options etc. but has limited algorithms for calculating feature geometries. The analysis techniques available are created to serve the industries where surface texture is of the utmost importance and surface roughness parameters are sufficient to describe the functioning of the surface. The procedure for obtaining form, size and orientation information is not automated or intelligent and susceptible to operator error resulting in poor repeatability and reproducibility. The methodology for characterization is convoluted and lacks the advancement and flexibility needed to meet the challenges of microstructure metrology in R&D and high volume manufacturing settings. On the other hand CMMs are used for dimensional measurements. Contact and non-contact CMMs are used to capture points on the surface. Then reference geometries are fitted on the point cloud. This reference geometry is used to calculate dimensions, distances etc. The limitation of CMM is its low accuracy and low vertical resolution. It is mainly used to measure macroscopic parts.

There is a need for advanced micro structure characterization methodology that offers algorithms and speed of the CMM for computing size and form at the high resolution of the interferometers.

1.4.3 Recent developments in advanced characterization techniques

Three dimensional surface characterization is used increasingly in industries and has been a focus of intense research in the last few years [26]. The proposed approach is to treat the microstructures on the surface as individual objects and apply the developed characterization methodology on all the objects simultaneously. This problem is similar to object recognition in computer vision. Object recognition is a fundamental concept in computer vision, the goal of which is to develop algorithms and representations to enable a machine to autonomously analyze visual information. It remains a challenging problem due to significant variation exhibited in real world problems. Researchers have been trying to use object recognition techniques in various areas. Kovesi [27] presented a new approach to the reconstruction of surfaces from surface normals using shapelets. Gelfand [28] proposed a method for segmentation of 3D scanned shapes into simple geometric parts by determining the slippable motions by computing eigenvalues of a certain symmetric matrix derived from the points and normals of the shape. Slippable shapes include rotationally and translationally symmetrical shapes such as planes, spheres, and cylinders, which are often found as components of scanned mechanical parts. Object recognition using shape from shading [29] uses curvature and shape-index information to recognize objects based on their surface topography. Nichols J. [16] developed a methodology to qualify MEMS, by implementing a novel computer-aided inspection (CAI) software framework. This software platform uses data acquired from current MEMS inspection hardware, and applies newly developed analysis algorithms to geometrically characterize a part [16]. Paper on surface metrology of MST devices [30] discusses techniques to partition surfaces at different planar levels using image processing techniques. Applications to MEMS devices have become a major focus of new instrument development. Details of instrument function for MEMS metrology appear in several papers, e.g. by Bossebeuf et al. [31], Novak et al. [32] and Grigg et.al. [33]. Shape numbers are also used to describe the boundary of the object [34].

Object recognition and boundary description techniques can help in automation of characterization process to a great extent. It can enable characterization of multiple features at the same time saving time and increasing repeatability of the calculated parameters by avoiding operator variability. The challenge lies in using object recognition techniques from metrology perspective.

1.5 Summary and motivation

The most common and standard technique till date for assessing surface characteristics is doing two dimensional analysis on three dimensional surfaces. The procedure to determine basic geometrical dimensions using the existing techniques is a tedious and time consuming process and is dependent on human interaction resulting in poor measurement repeatability and reproducibility. Lack of advanced algorithms and automation procedures limits the capability of current techniques and hinders their use in mass production industries and adds variability to the results. More significant limitation is the use of profile information rather than surface to describe feature characteristics. ISO has recommended new definitions for calculating feature dimensions that are going to be the standards in the future and will find increased attention.

It is important to have the flexibility to choose appropriate algorithms from several available options for engineered surface characterization. Practical experience and literature review indicates need for a more realistic and effective approach to characterize engineered surfaces. There is an urgent need to respond to the challenges and fast changing requirements in the area of micro feature metrology. This dissertation aims to make basic contribution in the field of micro metrology.

1.6 Objectives

White light interferometry has the resolution to measure micro-features but the characterization techniques are limited, old and time consuming. Vision CMMs have advanced techniques for dimensional characterization but it does not give true 3D representation of the part. Advanced characterization methodology is needed that offers object recognition techniques as in CMM and works on surfaces measured using interferometry. Proposed work will bridge these two domains resulting in increased measurement repeatability and reproducibility.

The objective of the research is to develop fast, repeatable, reproducible and flexible methodologies for measuring micro features at high resolution. This will be accomplished by developing algorithms using image processing and geometric fitting techniques. ISO algorithms for calculation of size will be used as a foundation for calculation and reporting of results. Developed methodologies will be integrated to create a toolbox in MATLAB programming environment. The deployment of the developed toolbox on the web using MATLAB builder for dot net and Visual Studio programming environment will be demonstrated. Software validation will be performed using surface data obtained from measuring NIST certified calibration artifacts. Synthetic data with known dimensions created in commercial software like MATLAB will also be used to validate the results obtained from the toolbox. Developed methodology will be used to analyze engineered surfaces and the results will be compared with other available commercial softwares like Zygo MetroPro and Veeco Vision.

CHAPTER 2: ADVANCED CHARACTERIZATION METHODOLOGY

2.1 Introduction

Significant progress has been made in the field of surface metrology in the last twenty years. Starting with the simple contact stylus instrument, these days a variety of non-contact instruments are available with sub nanometer vertical and sub micron lateral resolution to measure microstructures. These instruments collect height or intensity data and create an image of the surface. The task now is to use techniques to characterize measured microstructures. The process of characterization requires identifying features on the surface and then calculating parameters that describes the features and their relationship with each other. This task can be considered similar to the object recognition problem used in computer vision. For object recognition an image is scanned and partitioned to detect objects of interest. Then characteristics of the identified object are determined using image processing and pattern recognition algorithms. We can use similar approach to advance surface metrology analysis capabilities by providing more qualitative and quantitative description of imaged structural features. The challenge lies in using object recognition techniques from metrology perspective.

Image metrology is the task of determining dimensional information of object features within image data. Edge detection is a sub-task within image metrology in which linear feature of an object called edges are identified. Edge detection is used to calculate dimensions of the feature in vision dimensional metrology tools. The dimensional metrology on these tools is based on image analysis. These systems are typically located in-line, capturing images of the moving part for real time examination of feature of interest and rely on contrast to automatically find edges. Appropriate geometry is fitted to the detected points on the edge to determine dimensions of the feature.

The use of image processing techniques in surface metrology is not new. Segmentation is used in commercial surface analysis toolbox to partition height data. Vecco's Vision software has "Multi-region analysis" and "Surevision" packages that divides an image into islands. Due to a large variety and complexity of applications, the available techniques do not work on all surfaces. There is still no defined methodology that gives user the choice to use techniques suitable for their application. An advanced methodology is proposed in this chapter that consists of a series of algorithms and procedures, applied to the data in a sequence to characterize engineered surfaces. In the next section a methodology based on using image processing and pattern recognition techniques is described and challenges in using it for micro structured surface is discussed in detail.

2.2 Advanced Characterization Methodology

In describing advanced characterization methodology, we distinguish between the three different operations of preprocessing, feature extraction and post processing. Figure 1 shows a more elaborate diagram of the components of a typical surface characterization system. To understand the problem of designing such a system, we must understand the metrology problems that each of these components must solve. Also one fixed prescription is not suitable for all microstructure surfaces. Therefore criteria for selection of tools appropriate for a surface need to be established. Let us consider the operations of each component in turn, and reflect on the kinds of problem that can arise.

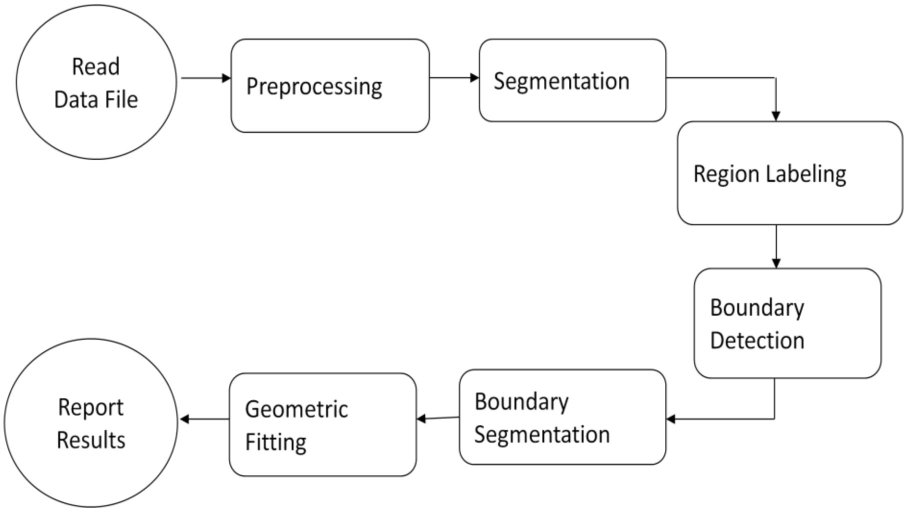


Figure 1: Steps in advanced characterization methodology

The developed methodology is implemented in MATLAB programming environment. All the steps in the methodology are individual functions that are integrated into one characterization system that can be accessed via a Graphic User Interface (GUI). Chapter 3 discusses in detail the software architecture. Henceforth the term characterization toolbox or system will be used interchangeably to refer to the software implementation of the methodology.

2.2.1 Read Data File

The input to the characterization system is often some kind of a camera image created using height or intensity data. The key challenge is that each measurement tool has its own proprietary data file format. Zygo and Veeco White Light Interferometer produce output data in DAT and OPD file format. SEM produces image file e.g. JPG. Vision CMM outputs a CSV file. Generally these files can only be opened and analyzed in the software that created them. Due to this limitation user is restricted to only one kind of tool and file format. There is also the option of reading and saving the data in generic file formats like ASCII, SDF and XYZ but the user can still get unsupported file format errors when trying to read the files in these formats. The user can write their own recipe to read the data files in an external software like MATLAB, which also requires writing analysis routines that can be difficult as it is a time consuming process and difficult for someone who does not have programming background.

The developed characterization toolbox can read data files created in both custom and generic file formats. This gives user the flexibility to take measurements on different tools and analyze the data in one toolbox.

2.2.2 Pre-processing

After the camera captures an image of the part, the image is pre-processed to simplify subsequent operations without losing relevant information. Pre-processing an image could be required due to several reasons like edge effects; data drop out and noise in the data without compromising the integrity of the data. Some of the common components of variation that deviate the metrological characteristics are listed below. These components can cause error when calculating metrological characteristics in the Z axis.

Bad pixels or void pixels: Pixels that cannot be measured caused by lack of sufficient signal quality or strength. Figure 2 shows image of solder bumps in which 20% pixels are void due to either poor reflectance or high modulation threshold setting. There is significant data dropout in and around the perimeter of the bump which could result in inaccurate calculation of height and diameter.

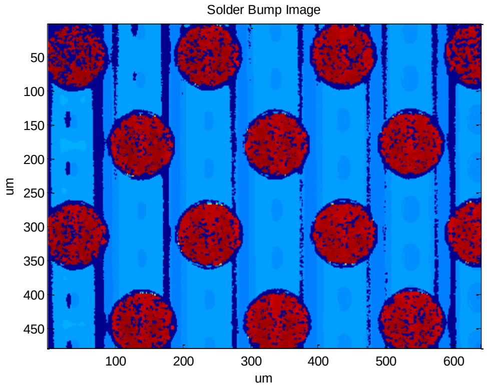


Figure 2: Bad or void pixel image

The characterization system when reads the data file checks for the number of void pixels in the image. If the number of void pixels is greater than 25% then the image is considered as poor dataset and no further analysis is done. If the void pixels are less than 25% then the bad pixels in the dataset are replaced by the string “NaN” which is an acronym for Not a Number in MATLAB. This enables MATLAB to read the file and the void pixels are not used in any calculations thus maintaining the integrity of the data.

Outlier objects in the data: Erroneous measured points related to the optical effects, often related to object surface features that results in weak or distorted signals. Outliers are seen in the form of spikes, negative spikes, particles, pores etc. Filtering of the data is required for recovery of areas covered by contamination and spike. An example of the optical effect is seen in the step height measurement using interferometry is the batwings effect. If the step height is less than the coherence length of the light source then the top portion of the step appears higher and bottom surface appears lower than the actual height. Higher numerical aperture (NA) of the objective produces large batwings. Batwings effect is not seen if phase shifting technique is used [55].

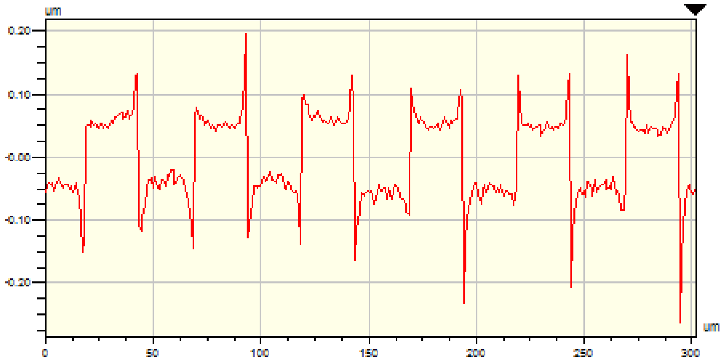


Figure 3: Batwings

Batwings can cause potential errors in calculating metrological characteristics along all the three axes. To calculate width of the trench in the grating shown in Figure 3, the approach published in a paper on measuring line width standard [56] is used. The paper mentions three algorithms for calculating width (i) using the local minima (ii) using the local maxima and (iii) using 50% height threshold. All the three cases are shown in Figure 4. In the characterization toolbox all three algorithms are implemented.

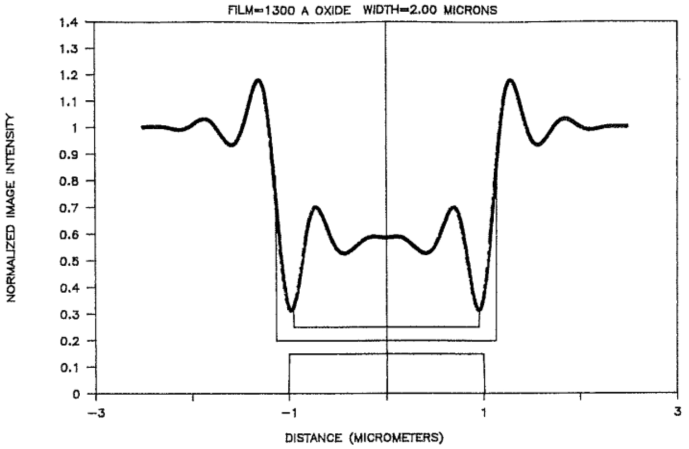
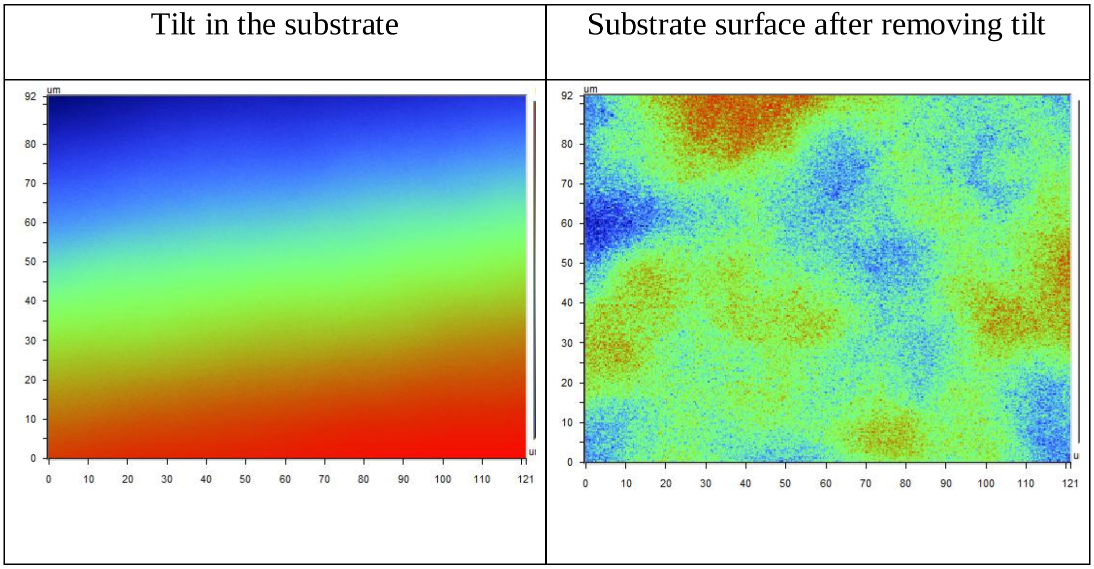


Figure 4: Width Calculation [56]

Tilt: Tilt is the relative angle between the optical axis of the measurement system and the object normal, assuming reference mirror is perpendicular to the optical axis.

Typically tilt is removed from the entire dataset by subtracting a least square plane from the data to level it.

Often times the substrate or the reference surface shows bow or warp also called potato chip to indicate the deformation of a wafer. In that case the substrate data is separated from the feature data and the least square plane fitting is applied only to the substrate data. Figure 5 shows the image of a substrate before and after removing tilt.


Tilt in the substrate

Substrate surface after removing tilt

Figure 5: Tilt Removal

Image Morphology: As mentioned before data collected using optical measurement techniques may contain "void pixels", which are pixels that could not be measured by the instrument. This is often true for interference microscope images, which need a proper reflectance in order to measure the height values. These void pixels may cause the features to be over-segmented.

In Figure 6, image of solder bump obtained using a white light interferometer shown. Histogram of image is bimodal showing two distinct regions. The region with lower average height is the substrate while the other region is the bump.

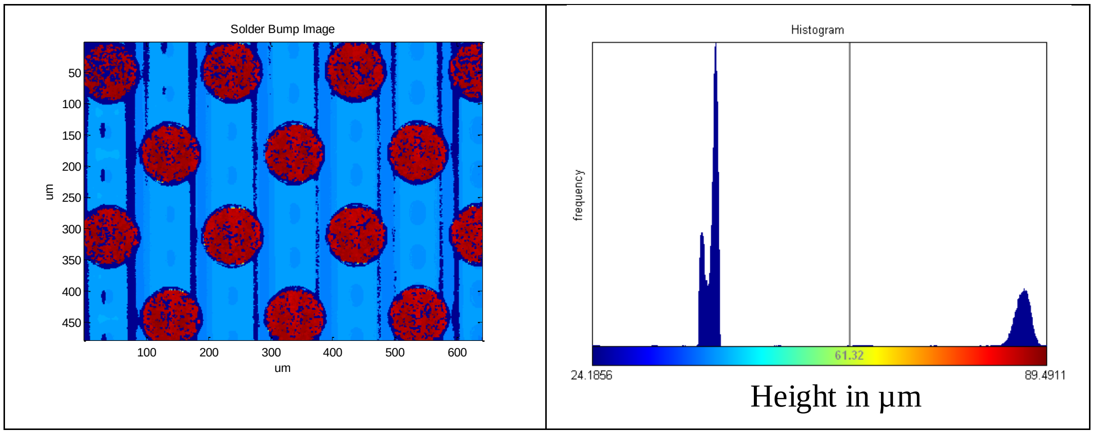


Figure 6: Solder bump data and bimodal histogram

Figure 7 shows two images in the left column show the result of thresholding. The image obtained after thresholding shows perforation in each bump. The desired segmentation should isolate each bump into a distinct region. As there are 14 bumps in the image, we would expect to see the same number of distinct regions after segmentation. But when the segmentation is performed based on the pixel connectivity, due to the perforations in the bumps, over segmentation occurs and we get more than 14 regions, 34 in this case.

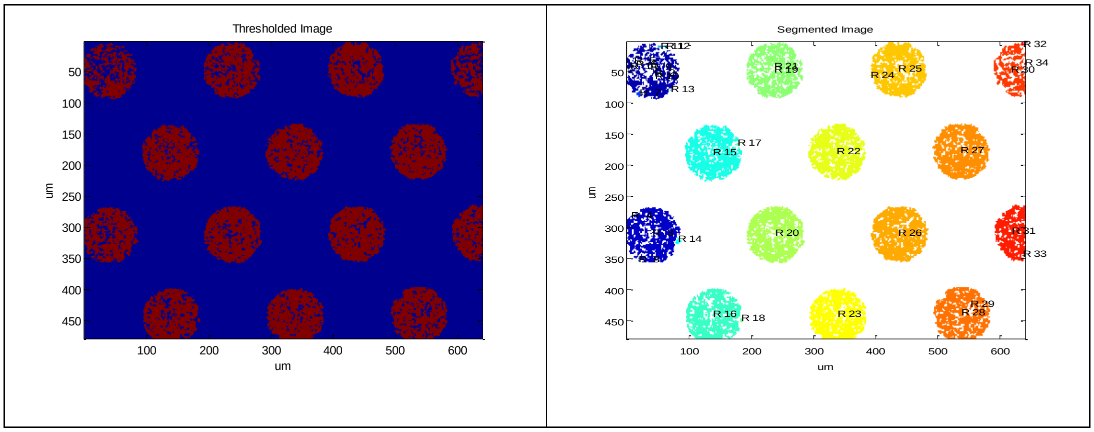


Figure 7: Binary and segmented image

One way to approach the issue of over segmentation is to use image morphology operations like dilation to fill the perforation in the data. This will ensure all the pixels belonging to the same region are well connected and will result in proper segmentation with 14 islands. It is shown in Figure 8. In dilation a structuring element is used to grow the region. In this case a disk shaped structuring element with a diameter of one pixel is used 5 times to fill the holes in the bump data. As a result the holes are filled and segmentation process creates the same number of islands as there are number of bumps. Each region is labeled and shown in a different color to support the characterization of each bump separately or relative to a reference region.

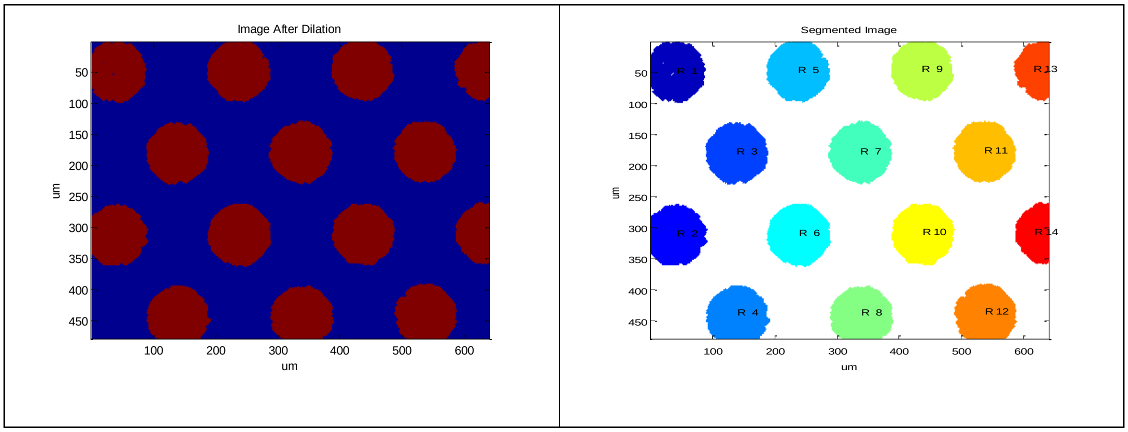


Figure 8: Binary and segmented image after morphology

Morphology changes the size of the object. From metrology perspective it is important to quantify this change. In the example above the diameter of the bump R7 was found after performing three morphological operations. The first operation is dilation, second is a combination of dilation and erosion and third is image filling holes.

Table 1: Result of Morphological Operations

Operation	Radius
None	43.41 µm
Dilation	45.27 µm
Dilation followed by erosion	44.21 µm
Hole fill	43.41 µm

Dilation increases the size of the object. Therefore a dilation operation is followed by erosion. It is also called image closing and defined as

$f.b = (f + b) – b$, where f is the image and b is the structuring element.

Hole filling operation only fills the holes inside the perimeter and does not change the boundary. Therefore the result is the same as obtained when no morphological operation is performed.

Summary

Pre-processing is an optional step where the objective is to deal with measurement artifacts generally present in the data to make it ready for further processing. User can compare the results by choosing different options to decide the most appropriate solution.

2.2.3 Segmentation

Image segmentation refers to partitioning an image into different regions that are homogeneous with respect to some image feature. After preprocessing the next step is performing the segmentation operation on the surface. Segmentation subdivides an image into its constituent regions or objects. The level to which the subdivision is carried out depends on the problem being solved. That is, the segmentation should stop when the objects of interest in an application have been isolated [36,39,40]. Segmentation is a very important part of the characterization. Segmentation accuracy determines the eventual success or failure of computerized analysis procedures. Segmentation can be classified into three categories:

Pixel based techniques: In this technique the intensity (or height) histogram is analyzed to find if the pixels are grouped in some dominant modes. The objects are then extracted by selecting a threshold that separates these modes.

Edge-based techniques: Edge detection is the most common approach for detecting meaningful discontinuities in intensity values. Such discontinuities are detected by using first and second order derivatives. The most commonly used detectors are Sobel, Prewitt, Roberts, Canny etc.

Region based techniques: In this technique regions are found directly. It is like region growing that group pixels or sub-regions into larger regions based on pre-defined criteria for growth. The basic approach is to start with a set of seed points and from these grow regions by appending to each seed those neighboring pixels that have predefined properties similar to the seed (such as specific ranges of gray level or height). Following segmentation algorithms are available in the characterization toolbox.

Thresholding

Typical computer vision applications usually require an image segmentation-preprocessing algorithm as a first procedure. At the output of this stage, each object of the image, represented by a set of pixels, is isolated from the rest of the scene. The purpose of this step is that objects and background are separated into non overlapping sets. Usually, this segmentation process is based on the image gray-level histogram. In that case, the aim is to find a critical value or threshold. Through this threshold, applied to the whole image, pixels whose gray levels exceed this critical value are assigned to one set and the rest to the other. For a well-defined image, its histogram has a deep valley between two peaks. Around these peaks the object and background gray levels are concentrated. Thus, to segment the image using some histogram thresholding technique, the optimum threshold value must be located in the valley region.

For example the height histogram shown in the Figure 9 corresponds to an image, f(x,y), composed of objects on two planar levels, in such a way that object and background pixels have height grouped into two dominant modes. One obvious way to extract the objects from the background is to select a threshold T that separates these modes. Then any point (x,y) for which f(x,y) >= T is called an object point; otherwise, the point is called a background point. In other words, the threshold image g(x,y) is defined as:

$$g(x,y) = \begin{cases} 1, & if\ f(x,y) \geq T \\ 0, & if\ f(x,y) < T \end{cases}$$

$g(x,y)$ is the image composed of features at two planar levels. Pixels labeled as 1 correspond to foreground whereas pixels labeled 0 correspond to the background. T is the threshold that separates these modes. Threshold T is called global threshold calculated using Otsu's method [57]. Otsu's method chooses the threshold to minimize the intra-class variance of the thresholded background and foreground pixels.

$$\sigma_w^2(t) = w_1(t)\sigma_1^2(t) + w_2(t)\sigma_2^2(t)$$

Weights $w_i$are the probabilities of the two classes separated by a threshold t and σ_i^2 variances of these classes.

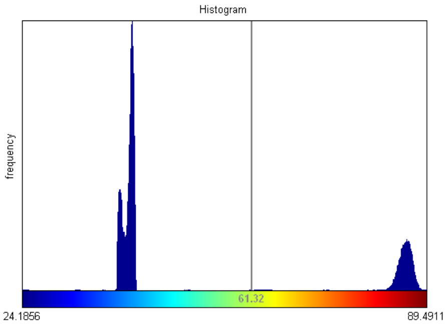


Figure 9: Histogram based thresholding

Binary image shown in Figure 10 is the result of thresholding showing the forground and background in two different colors.

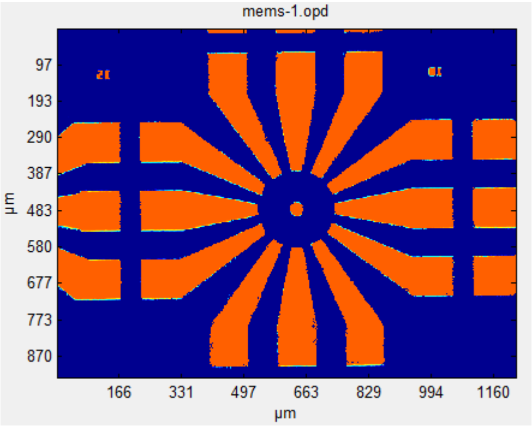


Figure 10: Thresholding example

A myriad of algorithms for histogram thresholding can be found in the literature [41–51]. Some algorithms [45] use an iterative scheme to achieve pixel separation. Entropy based algorithms have been proposed [46,47]. In general, all histogram thresholding techniques work very well when the image gray-level histogram is bimodal or nearly bimodal. On the other hand, a great deal of images are usually ill defined (corrupted by noise and/or irregularly illuminated) leading to a multimodal histogram where, in these cases, the ordinary histogram thresholding techniques perform poorly or even fail. In this class of histograms, unlike the bimodal case, there is no clear separation between object and background pixel occurrences. Thus, to find a reliable threshold, some adequate criterion for splitting the image histogram should be used. A possible one is the use of a measure of similarity or closeness between gray levels. Fuzzy set theory has become a powerful tool to deal with linguistic concepts such as similarity [52]. The advantages and limitations of the thresholding technique are as follows:

Advantages

- Simple, pixel based segmentation technique, easy to implement.
- If an image is composed of distinct region, the histogram of the image usually shows different peaks, each corresponding to one region and adjacent peaks are likely to be separated by a valley.
- Good for surfaces having repeated structure over the surface.

Limitations

- Its use is limited to surface features at the same planar level. If features are present at multiple planar levels and the height difference between each level is small and there is tilt in the surface, then finding the valleys between the modes could be tricky. In Figure 11 the input image has tilt due to which the histogram is not showing two distinct regions.

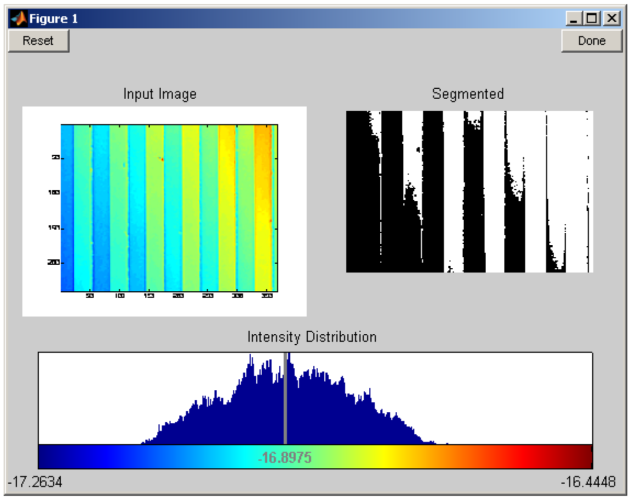


Figure 11: Histogram of a tilted surface

Clustering

Clustering is another pixel based technique that is extensively used for image segmentation. The rationale of the clustering technique is that, the colors in an image tend to form clusters in the histogram, one for each object in the image. In the clustering based technique, a histogram is first obtained by the gray scale values at all pixels and shape of each cluster is found. Then, each pixel in the image is assigned to the cluster that is closest to the pixel color. Among many different clustering algorithms are in existence, K-means has received extensive attention.

The K-means algorithm for clustering partitions n observations into K clusters based on the minimization of sum of the squared distances from all points in a cluster domain to the cluster center. It is an unsupervised clustering algorithm iterative in nature. "*K*" stands for number of clusters, it is typically a user input to the algorithm; some criteria can be used to automatically estimate K. Let $x_1, \ldots, x_N$ are data points or vectors of observations. Each observation (vector x_i) will be assigned to one and only one cluster. C(i) denotes cluster number for the i^{th} observation and Euclidean distance metric is used as dissimilarity measure. K-means minimizes the within-cluster point scatter W(C) and maximizes the between cluster scatter [58].

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(j)=k} \left\| x_i - x_j \right\|^2 = \sum_{k=1}^{K} N_k \sum_{C(i)=k} \left\| x_i - m_k \right\|^2$$

Where, m_k is the mean vector of the k^{th} cluster and N_k is the number of observations in k^{th} cluster.

Fuzzy K-Means is an extension of K-Means to determine soft clusters where a particular point can belong to more than one cluster with certain probability [36].

Figure 12 shows the input images in the left column and corresponding clustered image with four regions in the right column. It is on the discretion of the user to select the number of clusters formed.

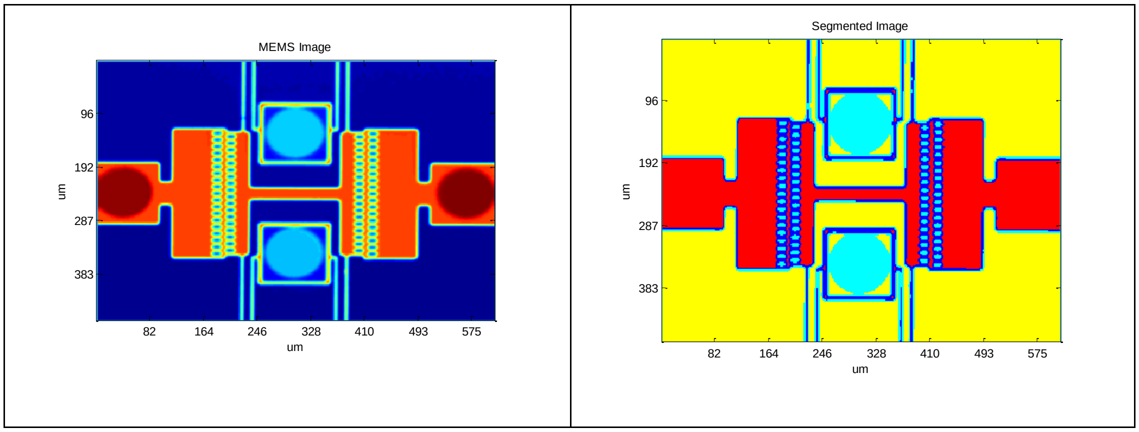


Figure 12: Segmentation using clustering

It is widely recognized that the clustering technique to image segmentation suffers from problems related to: (i) adjacent clusters frequently overlap in the color space, causing incorrect pixel classification, and, (ii) clustering is more difficult when the number of clusters is unknown. The advantages and limitations of the clustering technique are as follows:

Advantages

- It is on the discretion of the user to select the number of clusters formed.

Limitations

- Adjacent clusters frequently overlap in the color space, causing incorrect pixel classification
- Clustering is more difficult when the number of clusters is unknown.

In the clustering operation user has the option to decide the size of the smallest cluster. All cluster smaller than the minimum specified size are omitted from further processing.

Gradient based Separation

Gradient or slope based segmentation technique focus on the discontinuity of a region in the image. Once the edges within an image have been identified, the image can be segmented into different regions based upon these edges. A disadvantage with the edge based techniques is their sensitivity to noise.

In an image g(x,y), the eight nearest neighbor for every pixel are defined as shown below. i is the row number and j is the column number.

i-1, j-1	i-1, j	i-1, j+1
I, j-1	I, j	I, j+1
I+1, j-1	I+1, j	I+1, j+1

Figure 13: Pixel neighborhood

Slope based segmentation is used to detect meaningful discontinuities in intensity or height values. Such discontinuities are detected by using first and second order derivatives. The first order derivative is the gradient.

The gradient function in horizontal direction is defined as:

$$Gx1 = (g(i,j)-g(i,j-1))/dx, \; Gx2 = (g(i,j)-g(i,j+1))/dx$$

The gradient function in vertical direction is defined as:

$$Gy1 = (g(i,j)-g(i-1,j))/dy, \; Gy2= (g(i,j)-g(i+1,j))/dy$$

The gradient function in diagonal direction is defined as:

$$Gxy1 = (g(i,j)-g(i-1,j-1))/dxy, \; Gxy2 = (g(i,j)-g(i-1,j+1))/dxy$$

$$Gxy3 = (g(i,j)-g(i+1,j-1))/dxy, \; Gxy4= (g(i,j)-g(i+1,j+1))/dxy$$

The gradients for each pixel in horizontal, vertical and diagonal directions are compared with the threshold slope. If the gradients are greater than the threshold then pixel value is changed to 1 and 0 if vice versa. The limitation of this method is that a single slope angle is used as threshold for the entire surface.

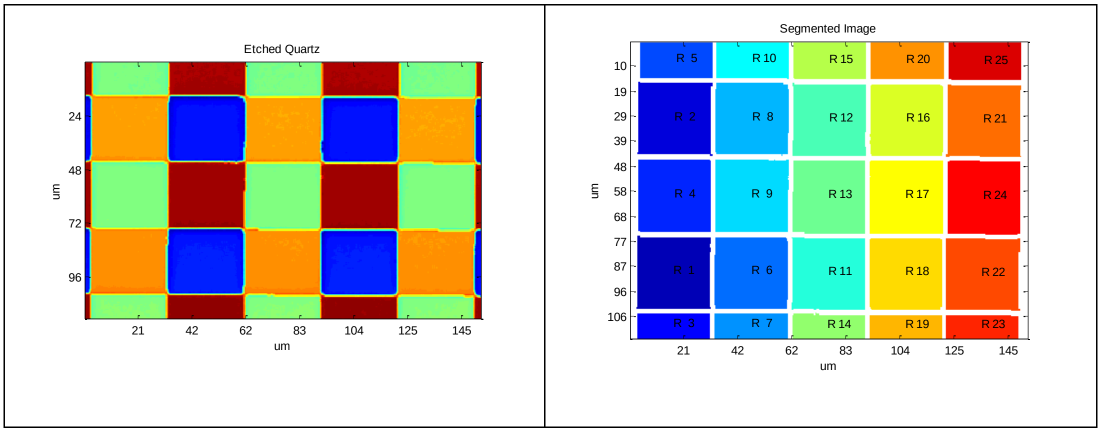


Figure 14: Segmentation using slope detection

The advantages and limitations of the clustering technique are as follows:

Advantages

- Works for the surfaces where features are at different planar levels.

Limitations

- A disadvantage with the edge based techniques is their sensitivity to noise.
- Iterative in nature. One gradient value may not be suitable for all features.

Surface Normals

It is a region growing technique where surface normals are computed at each pixel to capture discontinuity in the surface. These normals are subsequently combined based on similarity of coefficients of normals to the surface to form regions. This technique is useful when microstructure is multi-faceted. Each face will have surface normals pointing in different directions. This is useful when it is required to calculate facet angles.

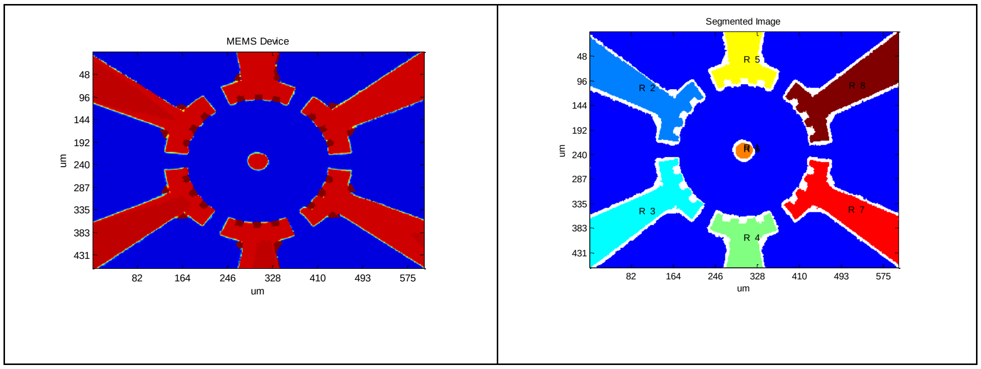


Figure 15: Segmentation using surface normals

Summary

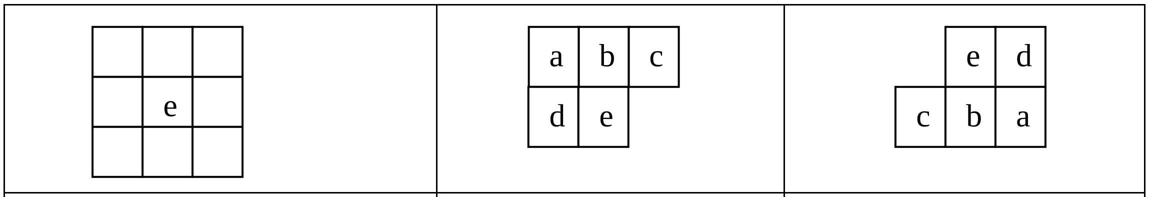
Thresholding is suitable for patterned microstructures where the histogram is bimodal and one threshold value can separate the microstructures from the substrate e.g. solder bump array, gratings etc. Clustering is useful when there are few objects of different size and geometry at different planar levels e.g. MEMS devices. Gradient is used when the number of surface features is large and the features are at multiple planar levels. Surface normals are used when the surface feature is multi-faceted. Clustering is performed after finding the surface normal parameters. Techniques is used for true three dimensional surfaces like cylinder, cone etc. Domain based segmentation is used when it is difficult to do the segmentation using height data. The height data is converted to another domain like FFT, Hough, Radon etc. After performing segmentation in the converted domain, inverse transform is performed to bring the data back in the height domain. This technique is used for micro-mirror and micro-pyramid array, to detect the base and slant edges.

Above classification can be used to choose the right segmentation technique for the surface under consideration.

2.2.4 Region Identification

After an image has been segmented into regions the next step usually is to represent and describe the aggregate of segmented raw pixels in a form suitable for further computer processing. Representing a region involves two basic choices: (1) representing the region in terms of external characteristic (its boundary), or (2) representing it in terms of its internal characteristics (the pixel comprising the region) [39]. Both the representations have its advantages. While the first one helps to calculate the orientation and size of the features the second one helps to find out the flatness, roughness and height of the features.

The result of the segmentation process is the images of the objects in the data with the background removed. Objects are often recognized by their spatial structure. Algorithms are needed to find spatial structures in images and determine their important properties. A basic process in analyzing spatial structures is finding connected components. Connected components are often associated with objects or part of objects. The number of objects depends upon how individual pixels connect. Pixel connection is defined in terms of neighborhoods. The neighborhood of a pixel is the set of pixels in the image that it touches.



(a) 8 connected neighborhood

(b) forward scan mask

(c) backward scan mask

Figure 16: Pixel neighborhood

Wu et. al. [53] described the basic scanning procedure as follows. Let I denote the 2D array representing an image. A pixel is a background pixel if I[i, j] = 0, and an object pixel if I[i, j] = 1. An array L of the same size and shape as I, is used for storing the labels. The problem of connected-component labeling is to fill the array L with (integer) labels so that the neighboring object pixels have the same label. Let the pixel name in the scan mask (illustrated in Fig. 16) as a, b, c, d and e, and also use the same letters in place of their (i, j) coordinates in the following discussion. With this notation, L[e] denotes the label of the current pixel, I[b] denotes the pixel value of the neighbor directly above e in the forward scan mask, and so on. Let i be an integer variable initialized to 1. The assignment of a provisional label for e during the first scan can be expressed as follows:

$$\begin{array}{ll} \mathrm{L[e]} \leftarrow 0, & \text{if } \mathrm{I[e]} = 0, \\ \mathrm{L[e]} \leftarrow \mathrm{l}, (\mathrm{i} \leftarrow \mathrm{l} + 1), & \text{if } \mathrm{I[i]} = 0, \forall \mathrm{i}, \\ \mathrm{L[e]} \leftarrow \min_{i|I(i)=1}(L[i]), & \text{otherwise} \end{array}$$

The above expression states that L[e] is assigned 0 if I[e] = 0. It is assigned a new label l, and i is increased by 1, if its neighbors in the scan mask are all background pixels. Otherwise, it is assigned the minimum of the provisional labels already assigned to a neighbor in the scan mask.

In later scans, labels for object pixels are modified to be the minimum labels of their neighbors, as described by the following expression.

$$\mathrm{L[e]} \leftarrow \min_{i\epsilon(a,b,c,d)|I[i]=1}(L[i]), \quad \text{if } \mathrm{I[e]} = 1, \text{ and } \mathrm{I[i]} = 1, \exists \mathrm{I} \in (\mathrm{a,b,c,d}) \qquad (2)$$

Eventually each pixel will receive the smallest provisional label assigned to any pixel in the connected component, but it may take many scans. Label number gives identification to the region.

2.2.5 Region Boundary

In edge detection the exterior boundaries of objects in the binary image is traced. The number of boundaries detected is equal to the number of objects in the image. Each boundary contains the row and column coordinates of a boundary pixel. Object boundaries are traced in either clockwise or anticlockwise direction.

If a region border is not known but regions have been defined in the image, borders can be uniquely detected. The image with regions is either binary or that regions have been labeled. Figure 17 illustrates the algorithm to cover inner boundary tracing in both 4 and 8 pixel connectivity. The steps are as follows:

Step 1: Search the image from top left until a pixel of a new region is found; this pixel Po then has the minimum column value of all pixels of that region having the minimum row value. Pixel Po is a starting pixel of the region border. Define a variable dir which stores the direction of the previous move along the border from the previous border element to the current border element. Assign

(a) dir = 0 if the border is detected in 4-connectivity

(b) dir = 7 if the border is detected in 8 connectivity.

Step 2: Search the 3x3 neighborhood of the current pixel in an anti-clockwise direction, beginning the neighborhood search in the pixel positioned in the direction:

(a) (dir + 3) mod 4

(b) (dir + 7) mod 8 if dir is even, (dir + 6) if dir is odd

The first pixel found with the same value as the current pixel is a new boundary element Pn. Update the dir value.

Step 3: If the current boundary element Pn is equal to the second border element P1, and if the previous border element Pn-1 is equal to P0, stop. Otherwise repeat step (2).

Step 4: The detected inner border is represented by pixels P0, P1, …Pn-2.

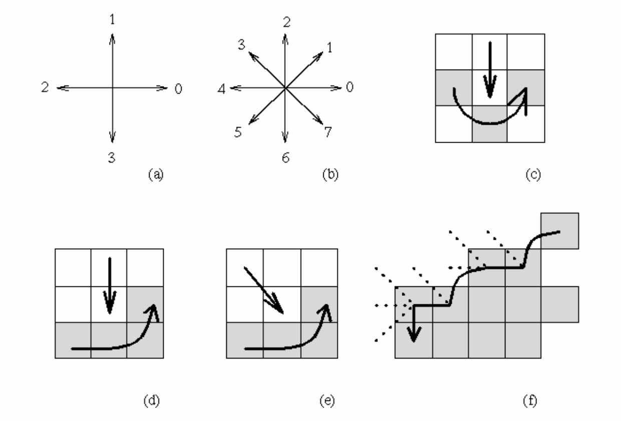


Figure 17: Inner boundary tracing: (a) Direction notation, 4-connectivity, (b) 8-connectivity, (c) pixel neighborhood search sequence in 4-connectivity, (d),(e) search sequence in 8-connectivity, (f) boundary tracing in 8-connectivity (dashed lines show pixels tested during the border tracing) [54]

Figure 18 shows the result of edge detection. A boundary is created along the perimeter of each closed region. Region boundary is a list of X and Y coordinates of the points on the edge. Boundary may consist of straight or curved line segments or a combination of both.

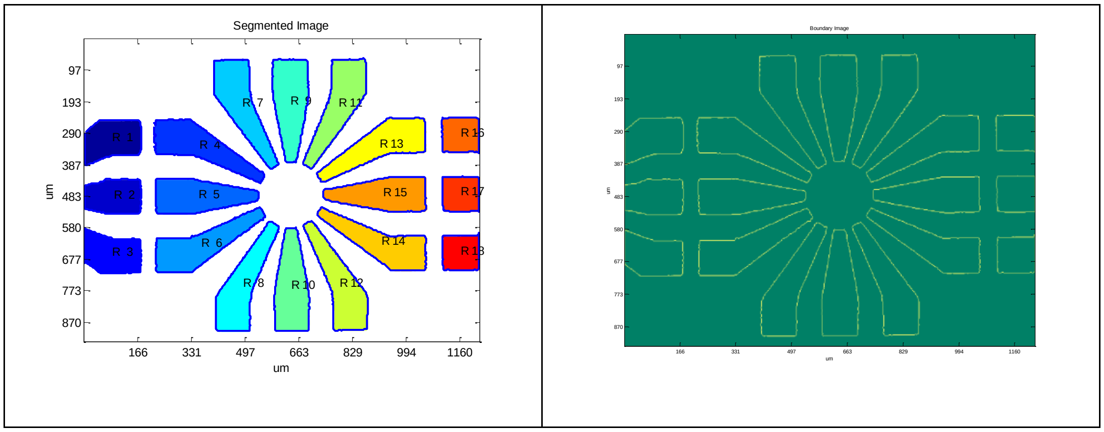


Figure 18: Edge Detection

2.2.6 Edge Segmentation

After region boundary detection the next operation is to divide a feature into line and curve segments. It consists of separation of each object into its constituent geometries like line, arc, and circle. Once objects are splitted into basic geometric elements then dimensional analysis can be performed. There are several techniques to perform boundary segmentation but we are discussing only two techniques that are used in this work.

Signature Method: Signature is a 1-D representation of a boundary and may be generated by plotting the distance from an interior point (e.g. the centroid) to the boundary as a function of angle. The idea is to reduce the boundary representation to a 1-D function.

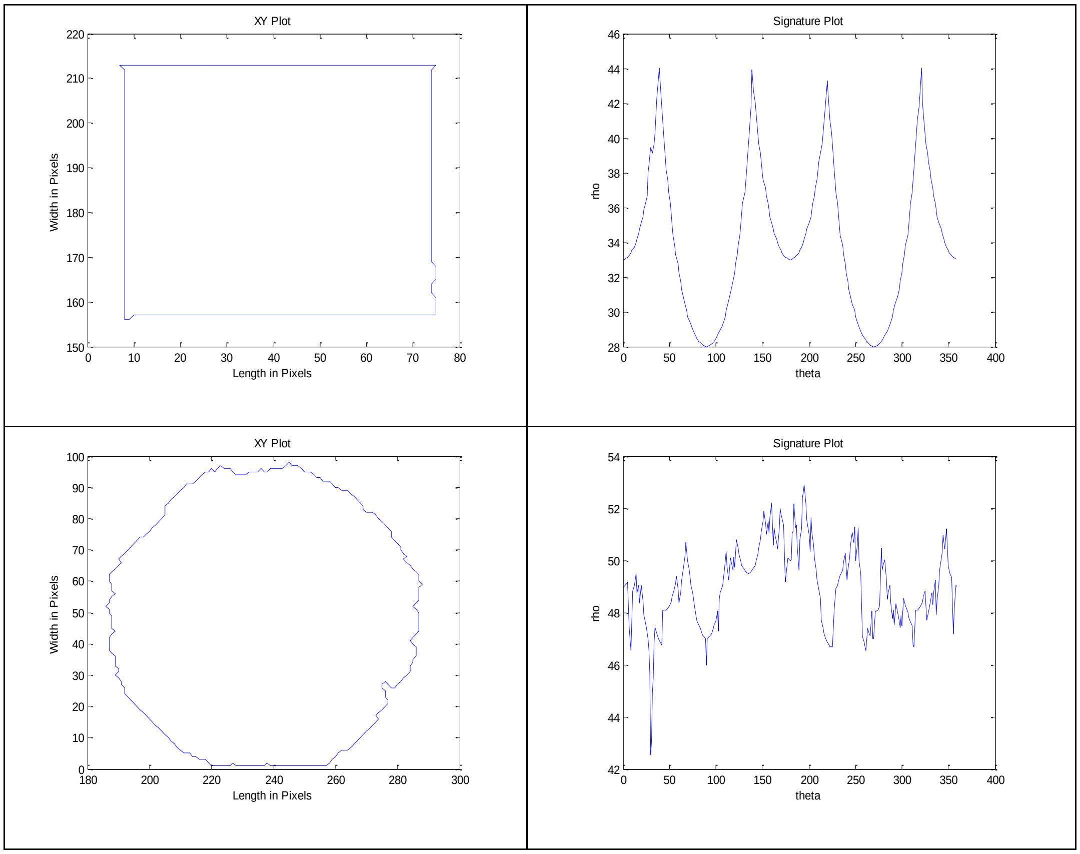


Figure 19: Boundary segmentation using signature method

Sugimoto and Tomita Method: Sugimoto and Tomita [38] proposed a new method for detecting the corner, inflection and transition points of a feature boundary by calculating the curvature and normal vector. They define the feature points as follows:

Corner Point: Corner points are detected as the point where the direction of the normal vector changes abruptly.

Transition Point: Candidates for the transition points are detected at the point where the curvature changes from zero to non zero.

Inflection Point: Inflection points are detected at the point where sign of the curvature changes from positive (negetive) to negative (positive).

Transition Point ◆

Corner Point 🟩

Inflection Point ▲

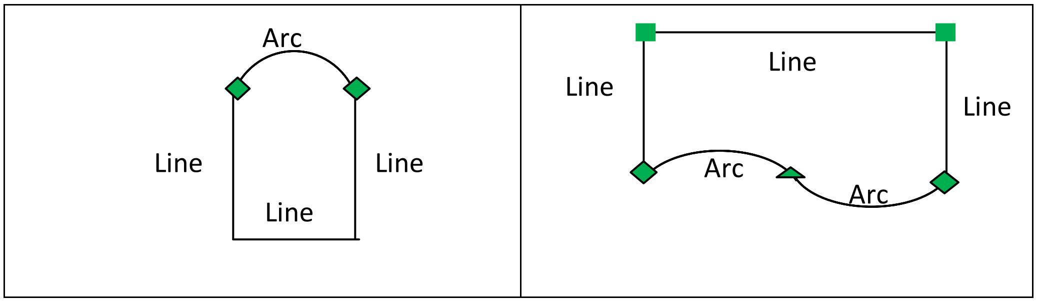


Figure 20: Boundary segmentation showing Transition, Corner and Inflection point

Using the same principle, for every point on boundary of the region, both straight line Lp, and a circle Cp, are approximated using a set of neighbor points within the distance K from the noticed point Pi by the least squares method to calculate the local curvature and the normal vector at each point. Here, K is the number of points used for least square fitting. Depending upon the length of the feature, K can be increased or decreased. The steps in the algorithms are as follows:

Step 1: For each pixel point Pi on the boundary,

K = [Pi-j,……,Pi,…….,Pi+j], where j is less than half the length of the smallest feature of interest.

Step 2: Fit least square line and circle to K.

Step 3: Determine the maximum error in fitting. For the line fit maximum error is the distance of the point that is at the farthest distance from the fitted line. For least square circle fitting, maximum error is the maximum distance between point Pi and the corresponding point on the fitted circle.

Step 4: Point Pi is then assigned to the feature which has minimum error for that point.

Step 5: This procedure is repeated for all the points on the boundary and the points are classified as either line or circle.

In Figure 21, when these pixels are plotted on top of the boundary (red pixels as shown in the images below), they represent all the instances of largest distance from the fitted lines. As observed from the images, the algorithm for line segmentation is not optimized. It is capturing even the insignificant transitions. Boundary segmentation can be improved by applying filters to remove the noise in the edge data and choosing the threshold carefully to select segmentation point.

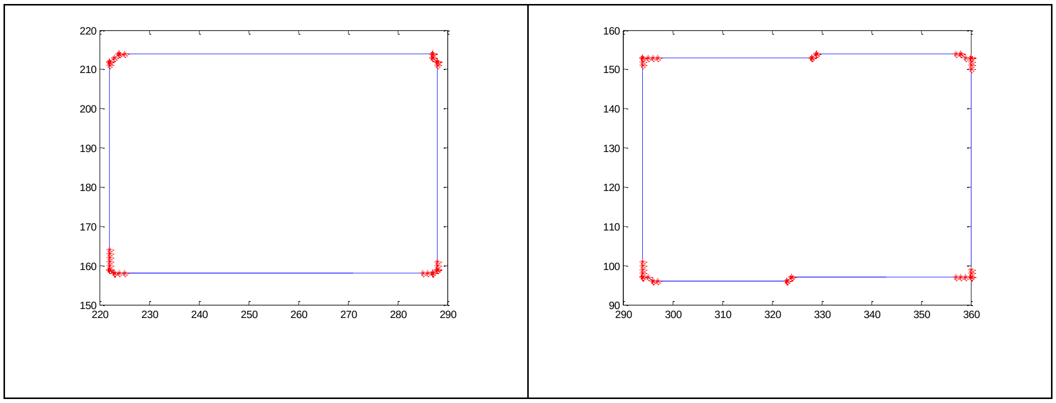


Figure 21: Boundary segmentation applied to rectangular feature

2.2.7 Geometric Fitting

Least square geometric fitting is a mathematical procedure for finding the best-fitting curve to a given set of points by minimizing the sum of the squares of the offsets ("the residuals") of the points from the curve [37]. In practice, the vertical offsets are minimized instead of the perpendicular offsets. This provides a much simpler analytic form for the fitting parameters than would be obtained using a fit based on perpendicular offsets [74].

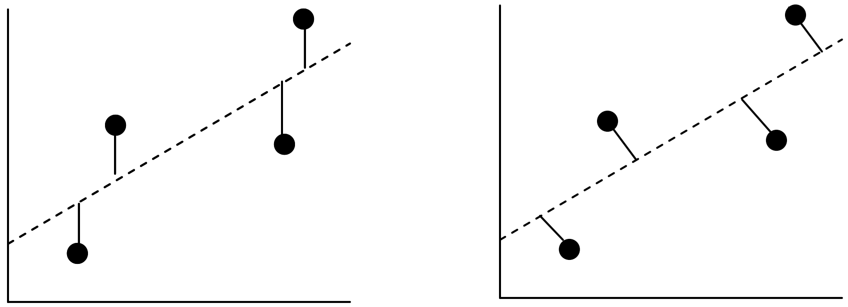


Figure 22: Vertical and perpendicular offsets

Once the boundary of features is segmented into line and curve segments, geometric fitting algorithms are applied to the line and curve segments. The minimum numbers of points required for applying geometric fitting are given in Table 2. Geometrical formulae are used to calculate characteristics of each individual geometric entity and also the relationship between the features.

Table 2: Min. No. of Points for Fitting

Element	Minimum No. of Points
Line	2
Plane	3
Circle	3
Sphere	4

2.2.8 Parameterization

A comprehensive set of dimensional, geometric and textural parameters can be calculated at this stage. A list of all the features, measured or created is shown in the Tables 3, 4 and 5.

Measured features are geometric elements fitted to feature boundary. Constructed features are calculated using the measured features. For example distance is calculated between two points or angle between two lines. Measured features can be created either manually or automatically using the characterization toolbox.

Table 3: Measured Features

Measured Features	Reported Parameters
Point	X, Y, Z
Line	X, Y, Z, Straightness
Plane	X, Y, Z, Flatness
Circle	X, Y, Z, Radius, Diameter, Circularity
Arc	X, Y, Z, Radius, Diameter , Circularity
Sphere	X, Y, Z, Radius, Diameter
Centroid	X, Y, Z

Table 4: Constructed Features

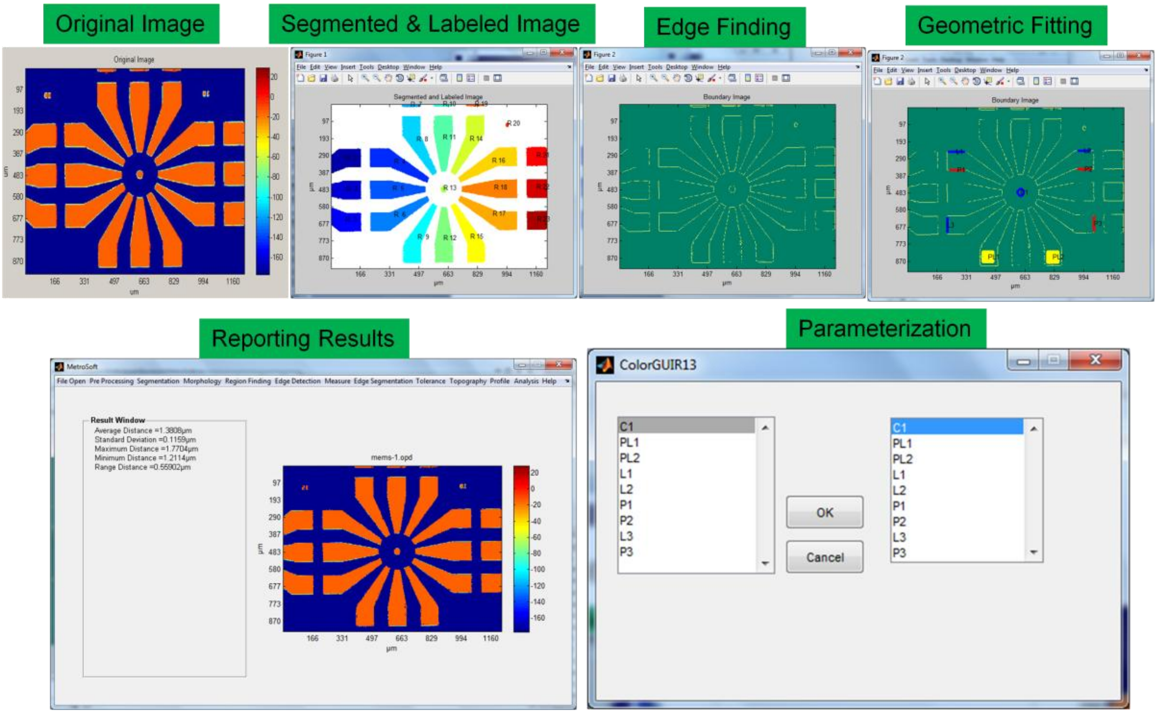
Constructed Features	Reported Parameters
Angle	Line-Line
Distance	2D, 3D
Intersection Point	X, Y, Z
Mid Point	X, Y, Z
Mid Line	X, Y, Z
Corner	X, Y, Z

Table 5: Feature Tolerance

Orientation/Form Tolerance	Applicable Features
Parallelism	Line-Line
Perpendicularity	Line-Line
Angularity	Line-Line

2.3 Summary

This chapter describes the advanced characterization methodology in detail. It shows algorithms for each step in the methodology. Each surface is unique therefore no one set of technique works for all the surfaces. Algorithms for several techniques for each step gives user the choice to be flexible and use the appropriate recipe for their measured surface. At each step in the methodology best practices are discussed to make user aware of the consequences of the chosen procedure on metrological characteristics.


Original Image
Segmented & Labeled Image
Edge Finding
Geometric Fitting
Reporting Results
Parameterization
Original Image
Segmented and Labeled Image
Boundary Image
Boundary Image
MetroSoft
File Open Pre Processing Segmentation Morphology Region Finding Edge Detection Measure Edge Segmentation Tolerance Topography Profile Analysis Help
mems-1.opd
ColorGUIR13
C1
PL1
PL2
L1
L2
P1
P2
L3
P3
OK
Cancel
µm

Result Window
Average Distance =1.3808µm
Standard Deviation =0.1159µm
Maximum Distance =1.7704µm
Minimum Distance =1.2114µm
Range Distance =0.55902µm

Figure 23: Advanced methodology sequence of steps

CHAPTER 3: SOFTWARE ARCHITECTURE AND IMPLEMENTATION

3.1 Introduction

In Chapters 2, the methodology to characterize three dimensional surfaces is described in great detail. The methodology consists of several steps and in each step there are several options available to the user to influence the output result. User can establish procedures to characterize surfaces by selecting the most appropriate option at each step in the methodology. The next step is to present the user with all the algorithms in a flexible yet efficient manner. The objective is to develop a mechanism that integrates all the algorithms in one characterization toolbox and establish a framework which allows user to interact with the system by performing operations like loading data, selecting recipes and reporting the metrology results in a friendly manner. The second objective is to make the developed toolbox available to a larger group of metrology community for dynamic use and continuous improvements.

To achieve the first objective, MATLAB programming environment was chosen to write all the functions. The reason for selecting MATLAB is the widespread use of the software among the research community, easy to learn and understand and availability of various engineering toolboxes to support the development process. Once all the functions were implemented in MATLAB, a Graphic User Interface (GUI) was created to integrate all the functions into one characterization toolbox. GUI acts as an interface between the user and the system. User only needs to understand how the controls on the GUI are used and need not worry about the implementation. Once the interface is created user can access all the functionalities and perform all the selected operations by using the controls on the GUI.

To achieve the second objective an executable version of the toolbox was created. This executable can be used as a standalone toolbox or can be integrated with any other commercially available analysis software. The standalone version can be distributed in the form of a CD and can be installed and run on a machine that does not have MATLAB on it. To reach a larger metrology community a web version of the toolbox was created in Microsoft Visual Studio environment. The advantage of web version is the availability of the toolbox over the internet anywhere around the world. No installation is necessary and feedback of researchers from all parts of the world can be utilized to improve the algorithms and make the toolbox more relevant to the current needs. Users can create their own recipes to suite their specific characterization needs.

This chapter starts with an introduction to Graphic User Interface (GUI) in MATLAB followed by Data Flow Diagram (DFD) for requirement specification and architecture design. Then MATLAB standalone executable is described. The last section talks about deploying MATLAB applications on the web.

3.2. GUI in MATLAB

A graphical user interface (GUI) is a graphical display containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed. GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders etc. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

Before creating a GUI it is useful to create a design that defines all the controls and associated callbacks. The execution of each callback is triggered by a particular user action such as pressing a screen button, clicking a mouse button, selecting a menu item, typing a string or a numeric value, or passing the cursor over a component. The GUI then responds to these events.

3.2.1. Designing a Graphical User Interface

Figure 1 shows flow chart for designing a GUI.

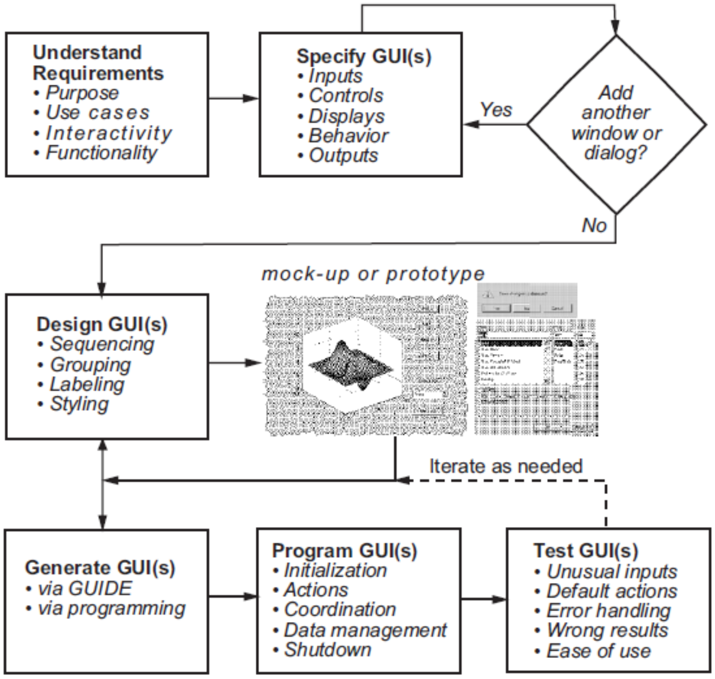


Figure 1: Designing a Graphical User Interface [61]

The designing of new software requires understanding the purposes a new GUI needs to satisfy. User requirements are documented as completely and precisely as possible before starting to build it. This includes specifying the inputs, outputs, displays, and behaviors of the GUI and the application it controls. After designing the GUI, each of its controls needs to be programmed to operate correctly and consistently. Finally, the testing of completed or prototype GUI is done to make sure that it behaves as intended under realistic conditions. If testing reveals design or programming flaws, design is iterated until the GUI works to the satisfaction. Figure 1 illustrates the main aspects of this process.

3.3. Requirement Specification & Architecture Design

Architecture design is used to develop a modular program structure and represent the control relationship between modules. Moreover, it is used to meld program structure and data structure, defining interfaces that enable data to flow through the program.

Data Flow Diagram (DFD) was used to design architecture of the standalone and web based Engineered Surface Characterization System. DFD represents a system at any level of detail with graphic network of symbols shows data flows, data stores, data processes, and data sources/destinations [62]. The purpose of data flow diagrams is to provide a semantic bridge between users and system developers. The diagrams are:

- Graphical, eliminating thousands of words
- Logical representations, modeling WHAT a system does, rather than physical models showing HOW it does it
- Hierarchical, showing systems at any level of detail
- Jargon less, allowing user understanding and reviewing

DFD allows a convenient transition from the analysis model to a design description of program structure. The transition from information flow to structure is accomplished as part of a five-step process:

- The type of information flow is established.
- Flow boundaries are indicated.
- The DFD is mapped into program structure.
- Control hierarchy is defined by factoring.
- The resultant structure is refined using design measures and heuristics.

3.3.1 Requirement Specification

This section emphasizes on the requirement specification of the engineered surface characterization system. The requirement specification shows how the system must interface with other elements such as hardware, people, and database. The following shows how the user interacts with the system and how the system responses back to the user input.

1. The user runs the MATLAB standalone executable to open the 3D Surface Characterization System. For Web based application user enters the URL address into the web browser to enter into the system.
2. The user then chooses the File Open option to select raw data file. This option gives the user choice to open a data file obtained from different measurement tools like White Light Interferometer, SEM etc. For the allowed file formats, once the data is read, a contour plot of the data will be displayed in the figure window. If the file is corrupted or file format is unknown then an error message will be generated.
3. The next step is the pre-processing operation where user can choose to modify the data to improve the quality of the data. Once the raw data is modified then user needs to go back to File Open option for original data. This step is optional.

4. Pre-processing is followed by segmentation process. Segmentation sub-divides an image into its constituent parts. The choice of segmentation technique depends on the type of surface under evaluation. User gets to choose the segmentation technique that is appropriate for the data.
5. Segmentation creates clusters of data points. To identify each region individually, a labeling scheme is generated when user selects the Region Labeling option.
6. User can now interact with the system in two different ways. For Class 1 surfaces user can choose Textured Surface Analysis and for Class 2 and 3 surfaces user can choose the Dimensional Analysis option.
7. After the data is analyzed, it is visualized and displayed in the display panel.
8. User can also choose to create a script file to run custom analysis on a batch of images of the same surface.

3.3.2 Architecture Design

Level 0 DFD

The fundamental system model, which encompasses the level 0 DFD and supports information, is shown in Figure 4 and 5. The user interacts with the system using computer, keyboard and mouse. The user inputs data and information into the system, and system then processes the information and outputs the result on to a display panel.

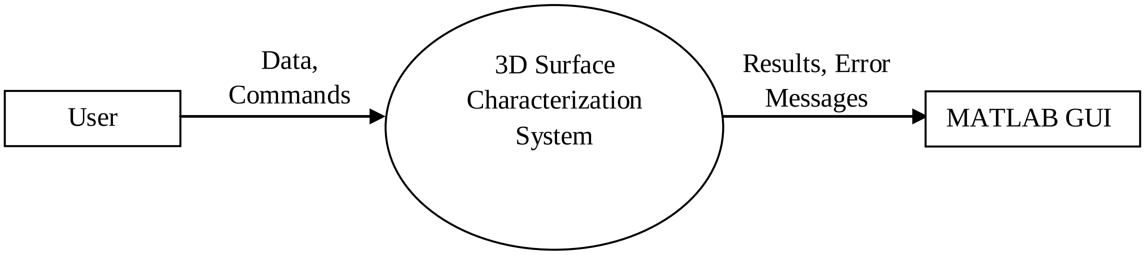


Figure 2: Level 0 DFD - Standalone Executable

In the web based application, the output is displayed on web interface while in case of MATLAB standalone application, GUI on the local machine is the output screen.

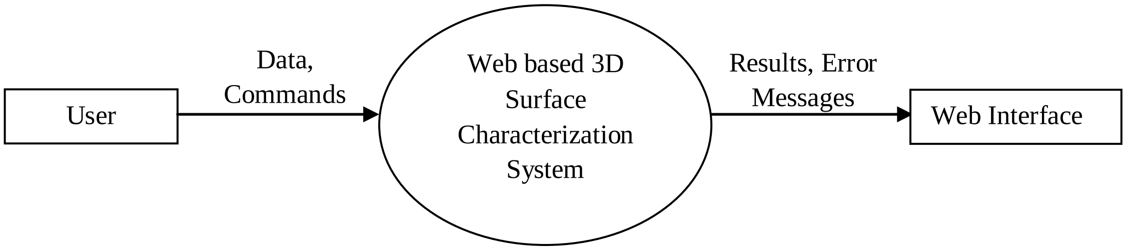


Figure 3: Level 0 DFD – Web Based Application

Level 1 DFD

The 3D characterization system is broken down into three sub systems; Class I, Class II and Class III surface characterization. Class I characterization system is used to characterize surfaces belonging to class I. It provides tools for calculating 2 and 3 dimensional surface roughness parameters, histogram, auto-covariance, bearing ratio, power spectral density, filtering, and perform statistical analysis. Class II and III characterization system is used for dimensional characterization of surfaces. It includes least square geometry fitting for e.g. line, plane, circle and sphere. Other parameters that can be calculated are centroid, midline, midpoint, intersection point etc. Also angle and distance between different geometric entities can be calculated. Performing the analysis generates results which are displayed on the display panel and in some cases can also be saved in comma separated variable file format The level 1 data flow diagram is shown in Figure 4.

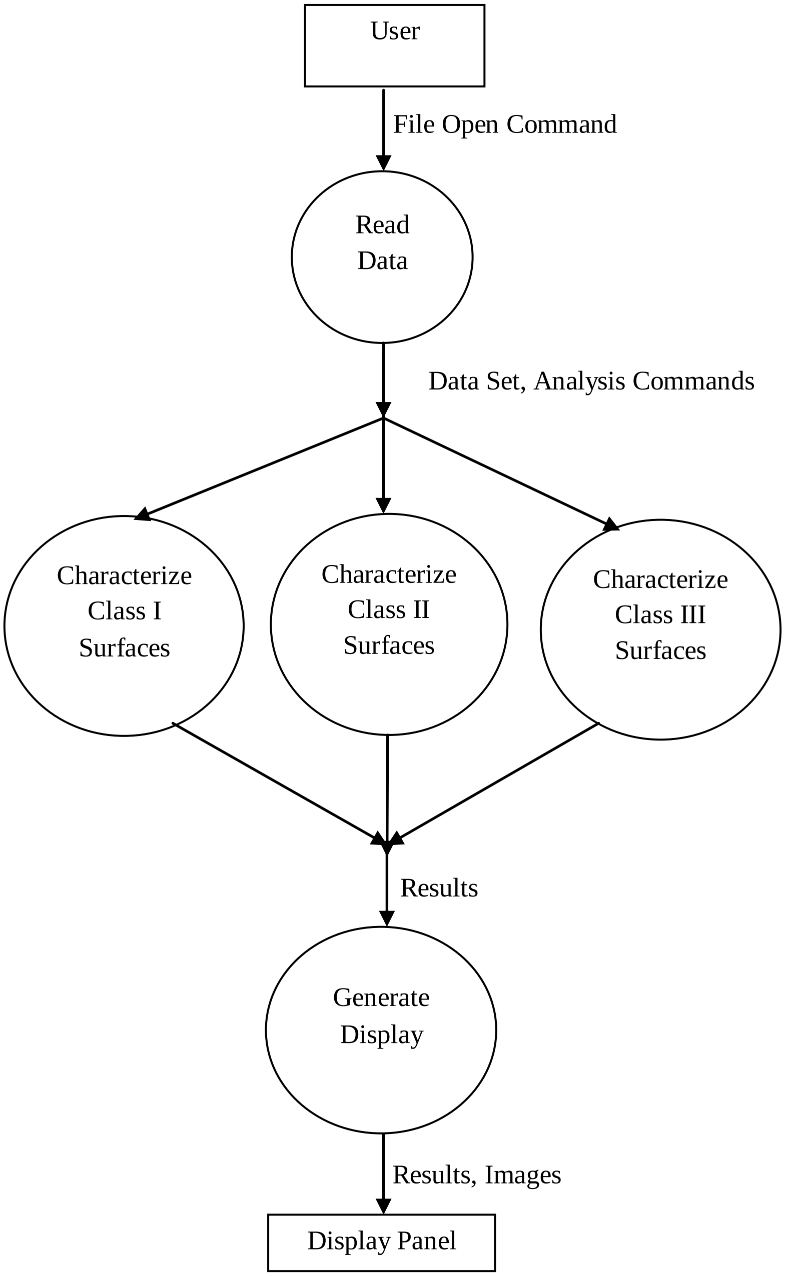

User
File Open Command
Read Data
Data Set, Analysis Commands
Characterize Class I Surfaces
Characterize Class II Surfaces
Characterize Class III Surfaces
Results
Generate Display
Results, Images
Display Panel

Figure 4: Level 1 DFD

Level 2 DFD

Figure 5 shows level 2 data flow diagram. At this level the steps in the characterization methodology in the sequence in which they are used are displayed.

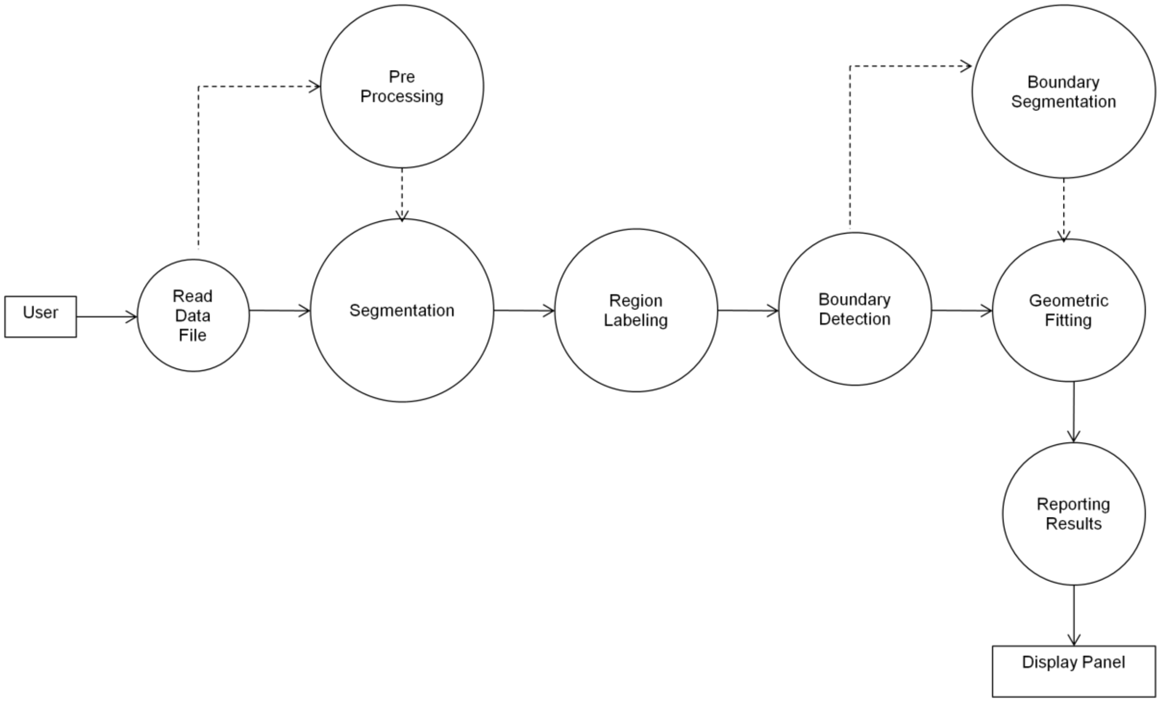


Figure 5: Level 2 DFD

The user runs the MATLAB standalone executable to open the 3D Surface Characterization System. For web based application user enters the URL address into the web browser to enter into the system. The user then chooses the File Open option to select raw data file. This option gives the user choice to open a data file obtained from different measurement tools like White Light Interferometer, SEM etc. From the allowed file formats, once the data is read, a contour plot of the data will be displayed in the figure window. If the file is corrupted or file format is unknown then an error message will be generated.

The next step is the pre-processing operation where user can choose to remove the components that can affect the results by introducing error in one or more axes. Once the raw data is modified then user will need to go back to File Open option to get the raw data back. This step is optional. Pre-processing is followed by segmentation process. Segmentation sub-divides an image into its constituent parts. The choice of segmentation technique depends on the type of surface under evaluation. User gets to choose the segmentation technique that is appropriate for the data. Segmentation is followed by an optional step of image morphology. This is required for images with data dropout. Loss of data can occur due to several reasons like poor surface reflection. Morphology fills the holes created due to data dropout. Segmentation creates clusters of data points. To identify each region individually, a labeling scheme is generated when user selects the Region Labeling option. Each blob or island is assigned a label to give it an identification number and to distinguish it from other regions. User can now interact with the system in two different ways. For Class 1 surfaces user can choose Textured Surface Analysis and for Class 2 and 3 surfaces user can choose the Dimensional Analysis option. In textured surface analysis parameters related to surface texture are calculated that are more suitable for tribology type applications. Dimensional characterization is used to calculate form, size and orientation of surface features.

After the data is analyzed, it is visualized and displayed into a display panel. User can also choose to create a script file to run custom analysis on a batch of images of the same surface. At this time user can save the figures and data in the location of choice.

Level 3 DFD

Level 3 DFD showed in Figure 6 displays the functionality of the system at the lowest level. It includes all the options available at each step in the methodology. User can choose to open data files in one of the acceptable file formats (.opd, .dat and .jpg). Pre-processing is an optional step that gives the user the choice remove tilt from the data or applying filter to remove certain frequencies. Segmentation can be performed using Thresholding, Slope based method or clustering technique to separate the foreground information from background. After segmentation, there is an optional step called morphology, where techniques like dilation and erosion can be used to improve the output of the segmentation process. The data now goes through region labeling and edge detection steps. Edge detection offers algorithms like pixel connectivity, gradient and canny edge detector to detect the boundary of the object. The next step at is Edge Segmentation which detects the corners, transition points and inflection points on the boundary of the object. This can be done automatically or by creating line and curve segments manually. The next step is Measure which can be used to calculate either the surface texture parameters or dimensional parameters. Surface texture parameters can be S Parameters, PSD, bearing Ratio. Dimensional parameters are obtained by dividing the object into geometric entities like line, point, plane, circle etc. and then applying best fit algorithms. Each geometric element like point, line, plane circle etc. when created gets a feature name for identification. When the features are created relationship between features can be determined like calculating angle between two lines or intersection of line etc. The results produced are displayed on the GUI and can also be saved a .CSV file on the local drive.

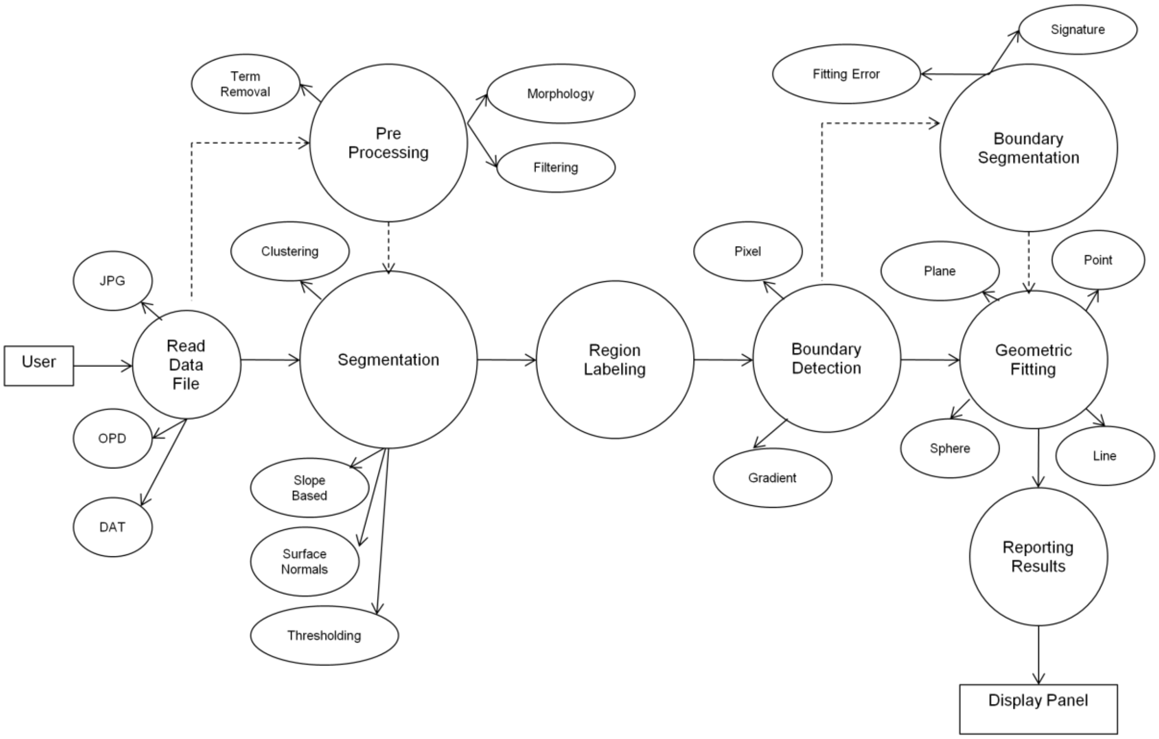


Figure 6: Level 3 DFD

3.4. MATLAB Executables

Standalone applications are a convenient way to package MATLAB recipes and to distribute a customized application to users. The source code for standalone C applications consists either entirely of MATLAB files or some combination of MATLAB files, MEX-files, and C or C++ source code files.

MATLAB Compiler takes MATLAB files and generates C source code functions that allow MATLAB files to be invoked from outside of interactive MATLAB. After compiling this C source code, the resulting object file is linked with the run-time libraries. A similar process is used to create C++ standalone applications. MEX-files can be called from MATLAB Compiler generated standalone applications. The MEX-files will then be loaded and called by the standalone code. Figure 7 shows the various options available with MATLAB compiler. For this project a C standalone executable was created.

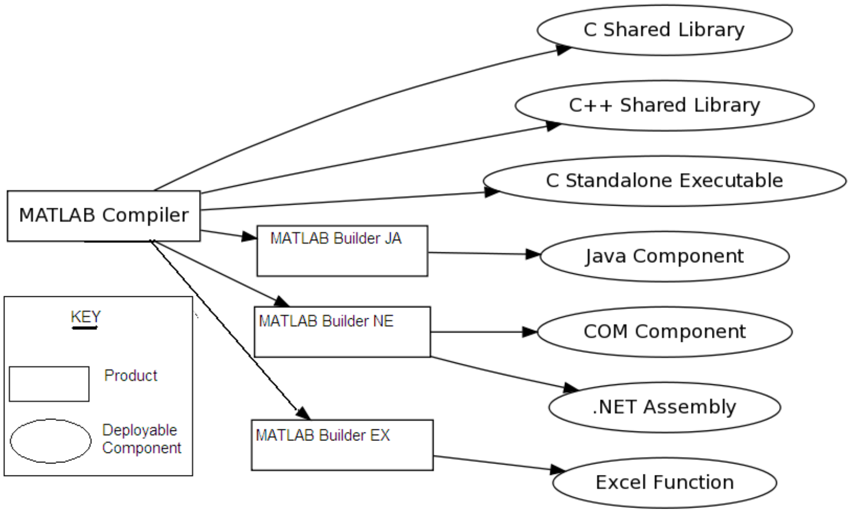


Figure 7: MATLAB Compiler [59]

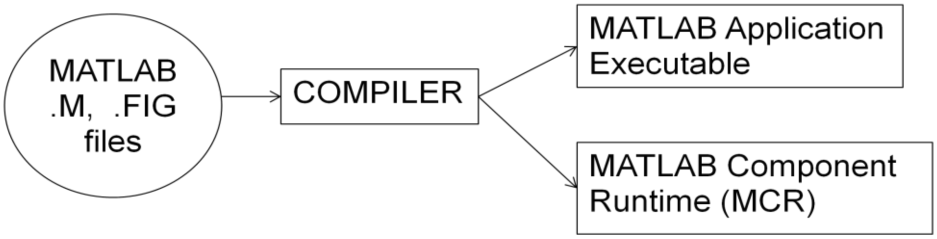


Figure 8: Creating executable using compiler

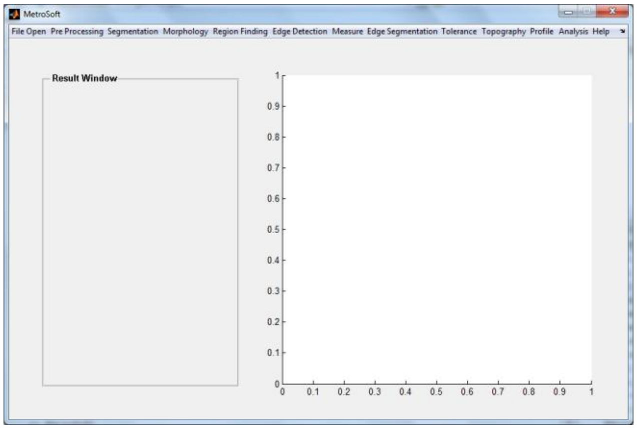


Figure 9: Interface of a Standalone Executable

MATLAB compiler allows sharing MATLAB application as an executable or a shared library. Executables and libraries created with MATLAB compiler product use a runtime engine called the MATLAB Compiler Runtime (MCR). Once the executable is created then it can be installed along with the MCR on any machine and run. Figure 8 shows the interface of the executable created for surface characterization.

3.5. MATLAB Web Application Deployment

The fundamental goal of the application deployment products (MATLAB compiler and the builders) is to enable work that has been accomplished within MATLAB to be deployed outside the MATLAB environment. This is accomplished with the MATLAB Compiler Runtime (MCR), which is a set of libraries that runs encrypted MATLAB code.

3.5.1. MATLAB Web Application Environment

Figure 9 depicts the supported implementation and architectures available when using MATLAB application deployment products. The architecture shown can be divided into different roles which require unique knowledge base and responsibilities. The first phase in a deployed application's life begins when code is written in MATLAB. To accomplish this objective, MATLAB Compiler is used. MATLAB Compiler makes MATLAB code usable by people in vastly different environments who may not have knowledge of MATLAB or the MATLAB language. When MATLAB Builder NE (for Microsoft .NET Framework) is installed along with MATLAB Compiler, MATLAB functions can be encrypted and wrapped in .NET interfaces. At this point in the deployment lifecycle, integration is usually required in order to make the deployed application work with the existing applications. Deployable applications are installed along with the proper version of the MCR, and converts MATLAB data types to native language data types so they can be used without any coupling to MATLAB in other tiers of the installation architecture. When the .NET component is called, it instantiates the MCR to execute the underlying MATLAB code. Once these services are exposed (either as Web services or through an API) Front End Developers can connect to them and use them. Front-end developers are typically responsible for user-visible functionality and know little about under-the-covers implementation. Once the front-end developers create some mechanism for exposing the application functionality to the end user, it is up to the end user to complete the lifecycle by interacting with the application to perform some task or solve some business problem. External users typically achieve this through a Web browser.

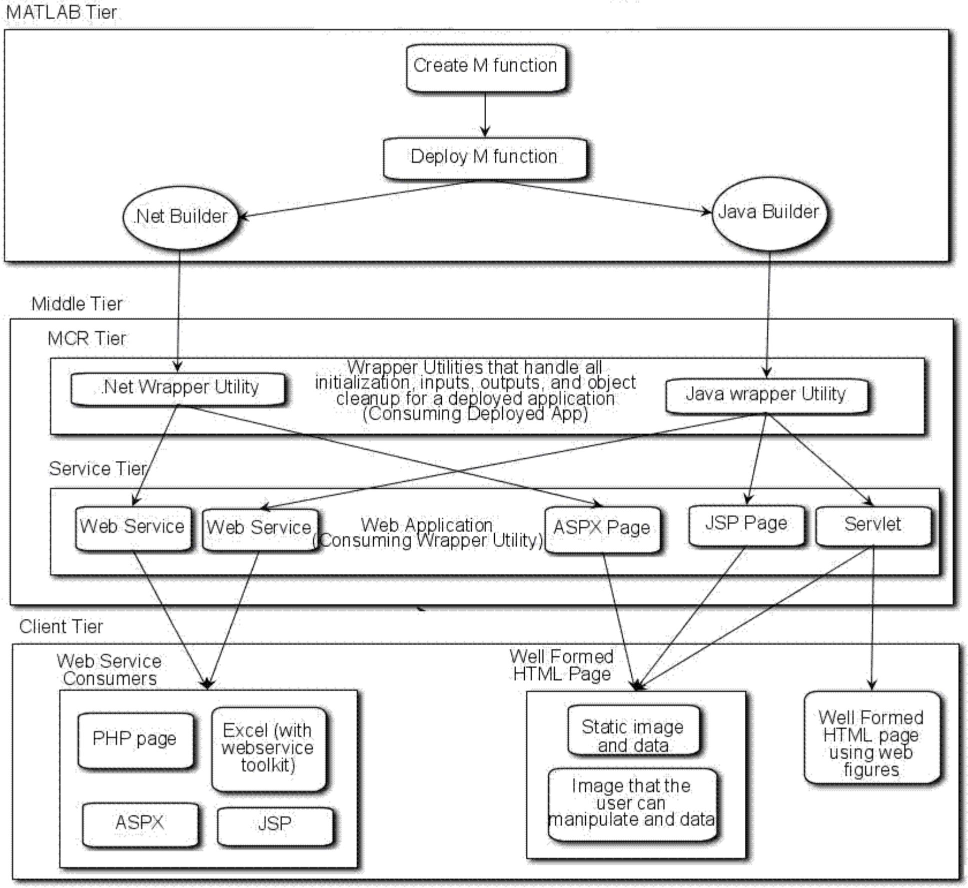

MATLAB Tier
Create M function
Deploy M function
.Net Builder
Java Builder
Middle Tier
MCR Tier
.Net Wrapper Utility
Wrapper Utilities that handle all initialization, inputs, outputs, and object cleanup for a deployed application (Consuming Deployed App)
Java wrapper Utility
Service Tier
Web Service
Web Service
Web Application (Consuming Wrapper Utility)
ASPX Page
JSP Page
Servlet
Client Tier
Web Service Consumers
PHP page
Excel (with webservice toolkit)
ASPX
JSP
Well Formed HTML Page
Static image and data
Image that the user can manipulate and data
Well Formed HTML page using web figures

Figure 10: Web Deployment Architecture [60]

In a Web application, the builder products allow integration of the MCR at the server tier level. This enables end users to execute MATLAB applications over the Web without installing client software. WebFigures is a client and server technology that further extends this capability by enabling end users to interact with a MATLAB figure in much the same way as they use an axis within MATLAB. The WebFigures functionality of MATLAB Builder NE allows users limited to Web access the ability to dynamically interact with MATLAB figures.

3.5.2. Creating a .NET Web Application

To create web application using .NET, the main requirements are the MATLAB Builder NE DLL file that is generated by compiling MATLAB code using MATLAB, MATLAB Compiler, and MATLAB Builder NE and the ASPX page and the code behind it. This page is responsible for taking in user input, displaying the controls, and handling page events.

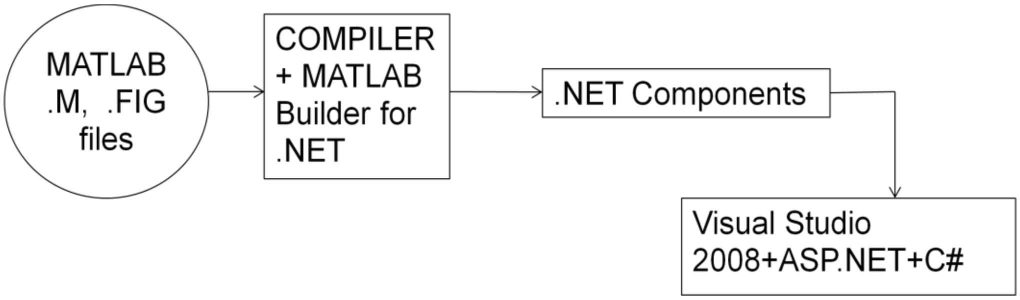


Figure 11: Creating dot net components

Software Requirement: In this project, to build the application in MATLAB, basic MATLAB and several other toolboxes were used. To build the standalone application and .NET web components MATLAB compiler and MATLAB builder for .NET were used. Table 1 and 2 shows version number and revisions of the toolboxes used.

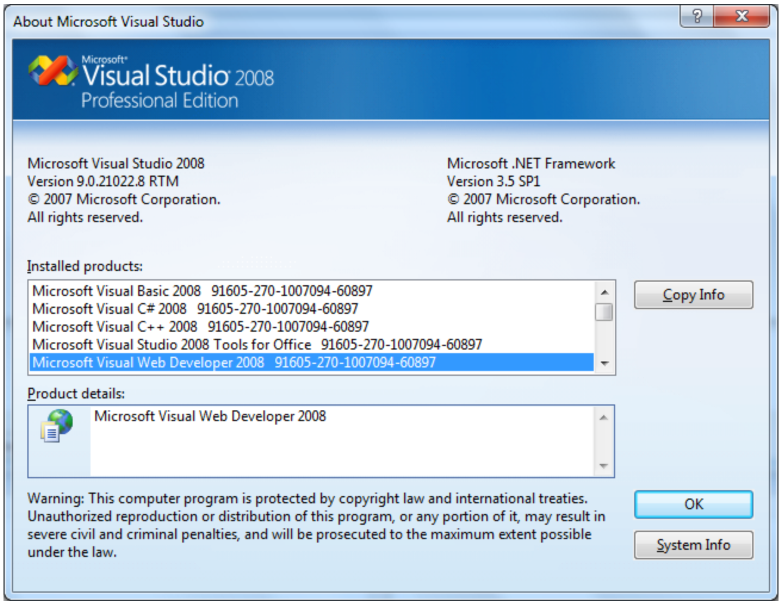
Table 1: MATLAB Version

MATLAB Version	7.9.0.529 (R2009b)
Operating System	Microsoft Windows Vista Version 6.1 (Build 7600)

Table 2: MATLAB Toolboxes

MATLAB	Version 7.9
Image Processing Toolbox	Version 6.4
MATLAB Builder NE	Version 3.0.2
MATLAB Compiler	Version 4.11

To create the Front End and programming the integration with MATLAB, Visual Studio 2008 Professional Edition was used. C# and ASP.NET [63] were used for programming.



Microsoft Visual Studio 2008
Version 9.0.21022.8 RTM
© 2007 Microsoft Corporation.
All rights reserved.

Microsoft .NET Framework
Version 3.5 SP1
© 2007 Microsoft Corporation.
All rights reserved.

Installed products:

Microsoft Visual Basic 2008 91605-270-1007094-60897
Microsoft Visual C# 2008 91605-270-1007094-60897
Microsoft Visual C++ 2008 91605-270-1007094-60897
Microsoft Visual Studio 2008 Tools for Office 91605-270-1007094-60897
Microsoft Visual Web Developer 2008 91605-270-1007094-60897

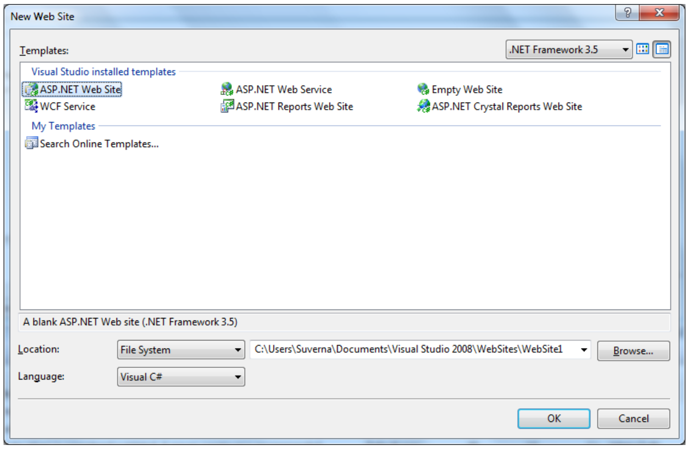
Product details:

Microsoft Visual Web Developer 2008

Warning: This computer program is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

Figure 12: Visual Studio 2008 [64]

Visual Studio .NET 2008 framework was installed on the local machine. In Visual Studio .NET a new ASP.NET Web Site template was created. Visual C# was selected as the programming language.



A blank ASP.NET Web site (.NET Framework 3.5)

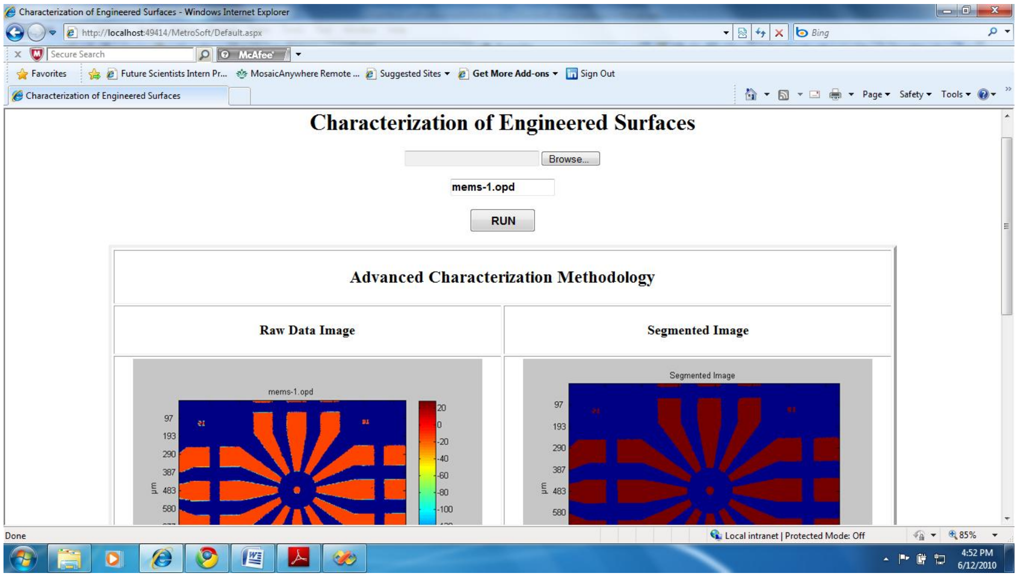
Location: File System C:\Users\Suverna\Documents\Visual Studio 2008\WebSites\WebSite1

Language: Visual C#

Figure 13: Template Window

All the MATLAB .m and .fig files were saved in the directory created above. Then mcc command is used in MATLAB to build the .NET component. This command creates a .dll file in the selected directory. This file was referenced from the Visual Studio project. MCR.exe generated during the build is installed on the local machine. The version of the MCR installed should be the same version as the MCR running with MATLAB when the application was built.

In the current project to demonstrate the web deployment capability File Read, Segmentation and Edge Detection Capability was tested. Code for the ASPX page was created along with the code behind ASPX. The.dll file was added as a reference. When the build is successful, a local IIS server built into Microsoft Visual Studio starts and opens the page inside of it. Figure 12 shows the web page with GUI created in ASP.NET displaying the original and segmented image. User can browse the file and then click on RUN button to start the characterization process.



Characterization of Engineered Surfaces

mems-1.opd

Advanced Characterization Methodology

Raw Data Image	Segmented Image

Characterization of Engineered Surfaces - Windows Internet Explorer
http://localhost:49414/MetroSoft/Default.aspx
Bing
Secure Search
McAfee
Favorites
Future Scientists Intern Pr...
MosaicAnywhere Remote ...
Suggested Sites
Get More Add-ons
Sign Out
Characterization of Engineered Surfaces
Page
Safety
Tools
Browse...
RUN
mems-1.opd
97
193
290
387
483
580
µm
20
0
-20
-40
-60
-80
-100
Segmented Image
97
193
290
387
483
580
µm
Done
Local intranet | Protected Mode: Off
85%
4:52 PM
6/12/2010

Figure 14: Web Based Interface

3.6. Summary

Software architecture design and implementation of the Surface Characterization System was discussed in this chapter. DFD represented a system at any level of detail with graphic network of symbols showing data flows, data processing and data destinations. Architecture design was displayed through different levels of data flow diagram to demonstrate how the all the functions interact. Introduction to GUI in MATLAB was provided which was used to integrate all the functions inside one wrapper function. The procedure to create a standalone executable in MATLAB and also deploying application on the web was demonstrated.

CHAPTER 4: SOFTWARE VALIDATION

4.1 Introduction

Software validation is required to prove that the software is suitable for the purpose it was developed for. It is a confirmation for correct and reliable functioning and of sufficient quality. Validation requires thorough testing and documentation before the software is released. It provided the requested trust. In general, validation means to confirm by providing objective evidence that the requirements for a specific intended use or application have been fulfilled. Validation clearly relates to the application context and includes the requirements as a basic component to understand what evidence shall be provided. Having the requirements defined, it is a matter of appropriateness which validation method is selected. From case to case, different methods may be appropriate to provide the requested evidence. An adequate concept of software validation must take all aspects of validation into account: from the definition of requirements to the provision that the requirements are fulfilled [65]. In this chapter validation of Surface Characterization Toolbox is demonstrated. The toolbox is created to calculate feature dimensions in X, Y and Z direction.

4.2. Requirements

The definition of requirements and their refinement to testable criteria are the fundamental working steps of the validation procedure. For the characterization toolbox validation, the first requirement is to validate the results on a NIST traceable calibration artifact. The second requirement is to compare the results with other comparable or better characterization software and demonstrate that the results are in reasonable agreement.

4.3. Validation

The first technique used for validation was using synthetic data created in MATLAB. A binary image of a square of size 50x50 pixels was created. The pixels comprising the square have value of 1 and the background pixels have value of 0. Edge detection is used to trace the boundary of the square. Figure 1 shows the image of the data used for validation.

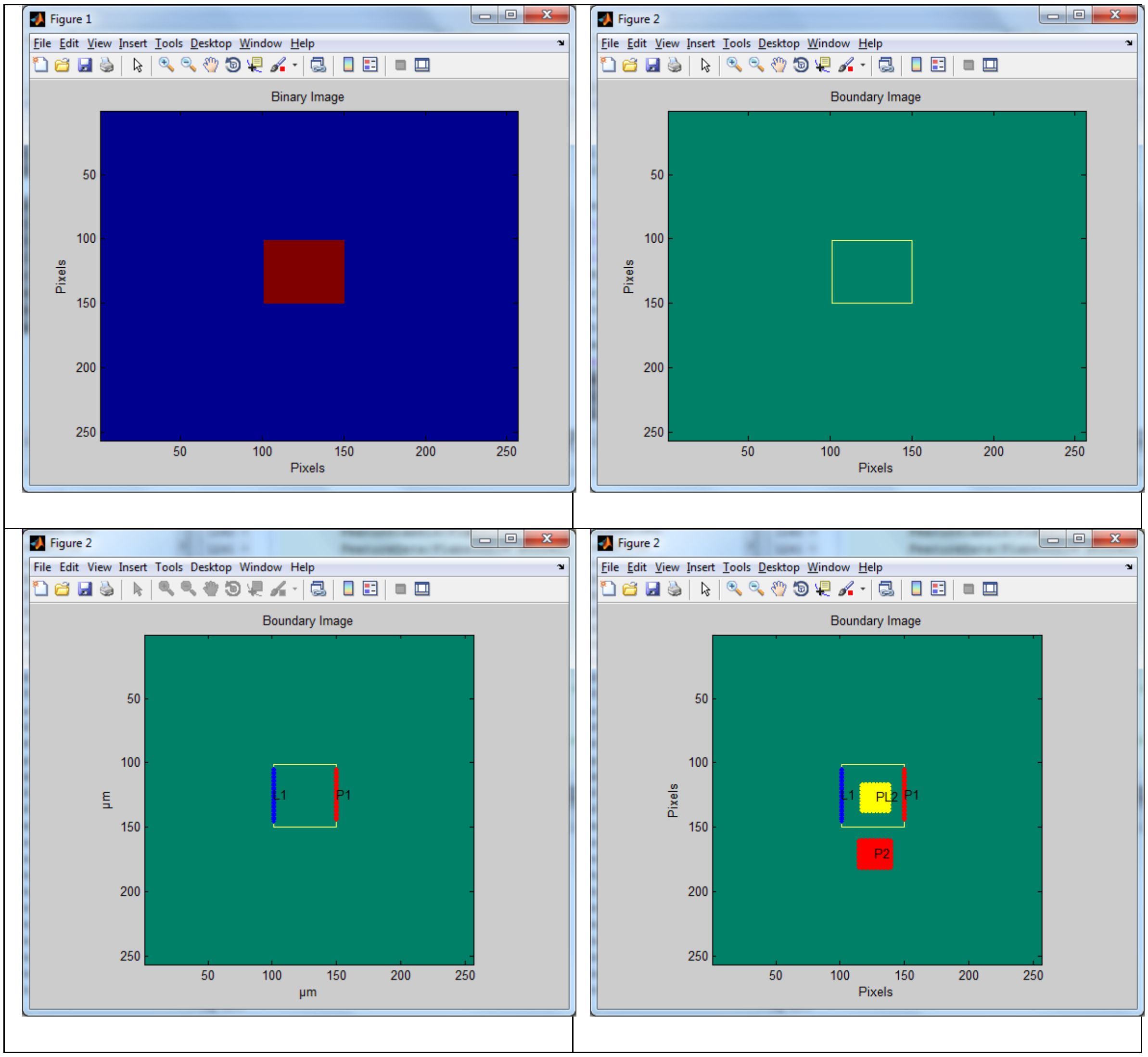


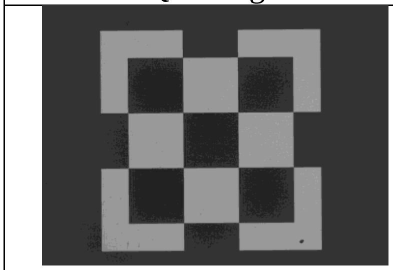
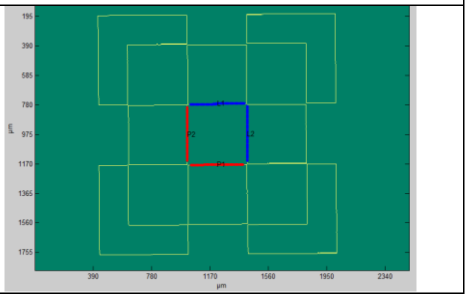
Figure 1: Validation using synthetic data

To validate the lateral dimension a line was fitted on one edge and several points were selected to get an average point on the opposite edge. The distance between point and line was calculated using the mathematical formula. The result is the exact distance in pixels between the line and the point. To validate the vertical dimension a plane was fitted on a sub-region selected on the square and several points were selected to get an average point on the background surface. The distance between point and plane was calculated using the mathematical formula. The result is the exact distance in pixels between the point and the plane.

Table 1: Synthetic data validation results

Parameter	Point – Line Distance in pixels	Point - Plane Distance in pixels
Average	49.00	1
Standard Deviation	0.00	0
Maximum	49.00	1
Minimum	49.00	1
Range	0.00	0

In the second validation method a NIST traceable Mitutoyo Vision CMM lateral calibration standard was used. A 400µm square checker box pattern was measured at two magnifications (2.5x & 5x) and two orientations (0º & 45º) on the vision CMM. X and Y dimensions of the box were recorded and images in JPG format were captured. The images were read in the developed toolbox to calculate feature dimensions. Figure 2 shows the images and Table 2 shows the results.

QV Image	Toolbox
	

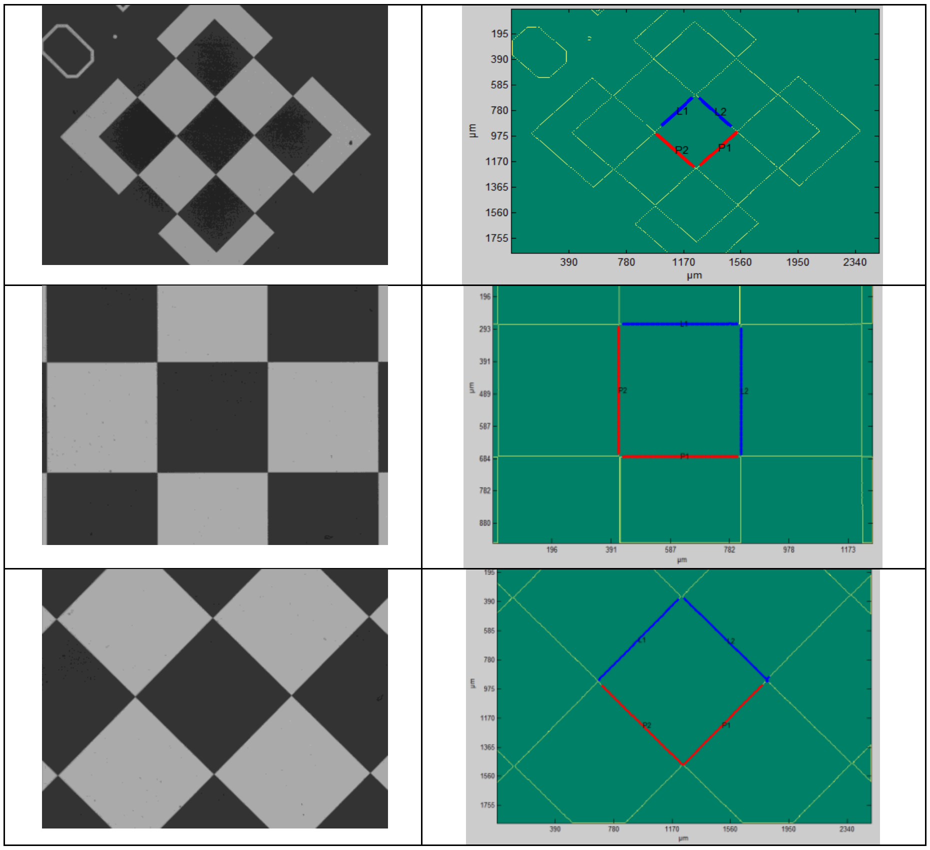


Figure 2: Characterization of Calibration Standard

TABLE 2: Validation

Conditions	QV Results	MATLAB Results	Certificate Results

Size: 400µm
Magnification: 2.5x
Orientation: 0deg
	Y = 400.567 µm
	X = 400.370 µm
		Y = 400.78 µm
		X = 401.70 µm
			Y = 400.16 ±1.5µm
			X = 400.16 ±1.5µm

Size: 400µm
Magnification: 2.5x
Orientation: 45deg
	Y = 400.622 µm
	X = 400.524 µm
		Y = 400.39 µm
		X = 399.68 µm
			Y = 400.16 ±1.5µm
			X = 400.16 ±1.5µm

Size: 400µm
Magnification: 5x
Orientation: 0deg
	Y = 400.589 µm
	X = 400.592 µm
		Y = 400.84 µm
		X = 401.73 µm
			Y = 400.16 ±1.5µm
			X = 400.16 ±1.5µm

Size: 400µm
Magnification: 5x
Orientation: 45deg
	Y = 400.519 µm
	X = 400.565 µm
		Y = 401.50 µm
		X = 401.46 µm
			Y = 400.16 ±1.5µm
			X = 400.16 ±1.5µm

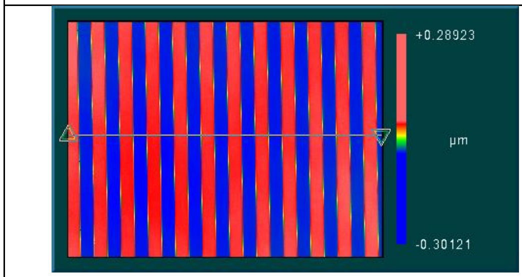
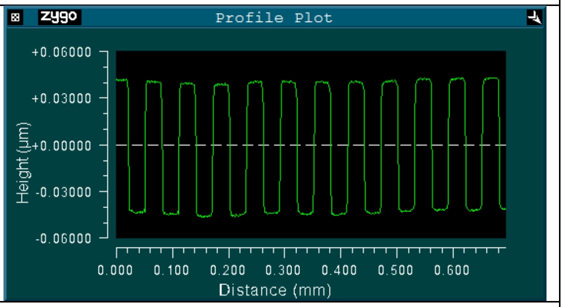
The specification of calibration artifact for 400µm box is: 400.16±1.5µm. It is observed that the X dimension is closer to the upper spec limit in three out of four cases in the MATLAB toolbox calculations. The offset from the nominal value can be attributed to several reasons. The first reason is the incorrect pixel to micron conversion. Pixel size in X and Y is not accurately known. Another reason is the difference in geometric distance calculations. Both QV and MATLAB use Point to Line distance formula for calculation of X and Y dimensions of the box but the actual implementation of the formula used by QV is unknown. In MATLAB calculations a least square line is fitted to one edge and average point on the opposite edge to calculate the distance. Third reason is the edge finding algorithm. In QV edge points are detected by finding the change in contrast or pixel intensity. IN MATLAB toolbox thresholding is used to separate background from foreground and then edge detection is performed. The difference in the edge point detection technique can introduce bias in the result. The bias from the nominal can be reduced by performing the measurement with a properly calibrated lens, controlled and uniform light distribution on the part and use of autofocus tool. Correct pixel to micron conversion and edge detection is required to get results close to the nominal with MATLAB toolbox.

Of primary importance in vision metrology is precise focus control to ensure that image features are measured under repeatable focus conditions. Improper focus can inhibit improper edge detection and introduce measurement errors.

After precise focus, repeatable precision lighting is critical in obtaining repeatable measurements. Small fluctuations in lighting might not make it difficult to find an edge, but do make it difficult to measure the edge repeatedly. The same feature measured using different light intensity and colors will affect measurements in the submicron realm.

4.4. Comparison of Results

Another form of validation is comparing the results with existing established software created for the dimensional analysis. Figure 3 shows a grating surface measured on a white light interferometer. Image size is 640 X 480 pixels and pixel size is 1.09µm. Vision and MetroPro are commercial software available with the purchase of White Light Interferometer from Veeco and Zygo respectively. These data analysis software are well established and popular in the metrology community. Width and Depth of the etched surface are the two metrological characteristics that are compared between Veeco, ZYGO and the developed Characterization Toolbox. In both Vision and MetroPro software surface can be viewed in profile view. Three cross sections were selected from the top, middle and bottom of the surface. Cross hairs were used to determine the depth and width of etched surface. The calculated results are shown in Table 3. In the characterization toolbox to calculate width, least square line was fitted on one edge and average of point is taken on the opposite edge. To calculate depth a mask is created on the top surface and another on reference surface and vertical distance between the two masks is calculated. Figure 4 shows the procedure.

Contour Plot	2D Plot
ZYGO MetroPro	
	
VEECO Vision	

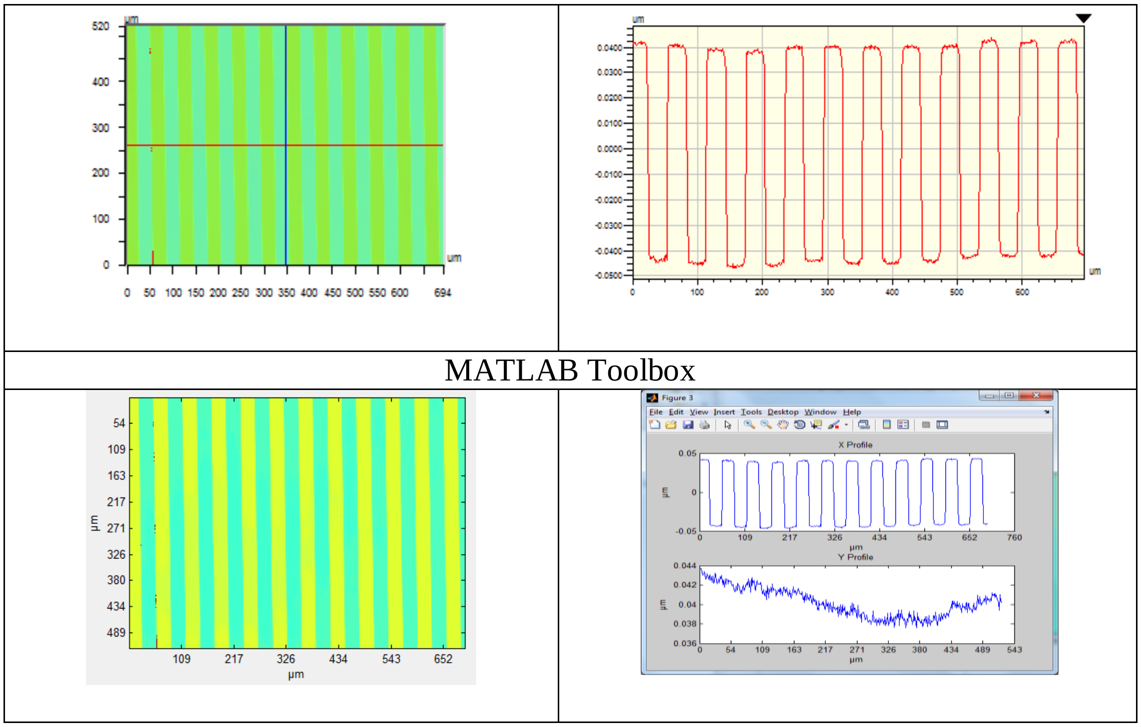


Figure 3: Comparison with Other Softwares

Table 3: Veeco and Zygo Results in

	ZYGO		VEECO	
	Width µm	Depth nm	Width µm	Depth nm
Cross Section Top	30.4	84.34	29.36	84.9
Cross Section Middle	30.4	85.61	30.46	85.6
Cross Section Bottom	30.4	84.9	29.36	85.4

TABLE 4: Characterization Toolbox Results Width µm Depth nm

	TOP		MIDDLE		BOTTOM	
	Width	Depth	Width	Depth	Width	Depth
Average	29.70	84.47	29.52	84.75	29.55	84.19
Std Dev	0.34	0.39	0.31	0.38	0.35	0.47

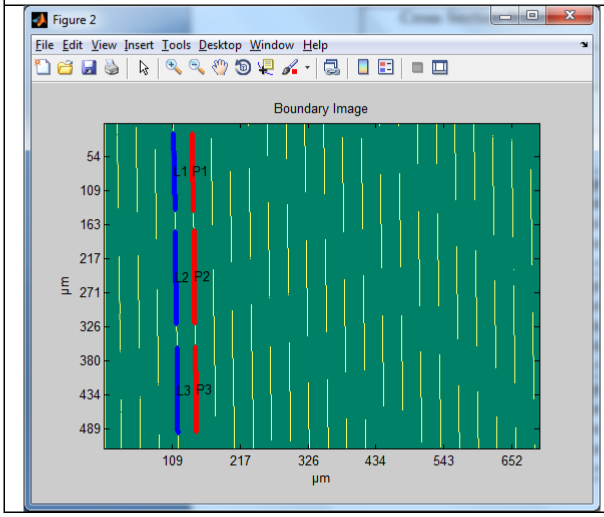
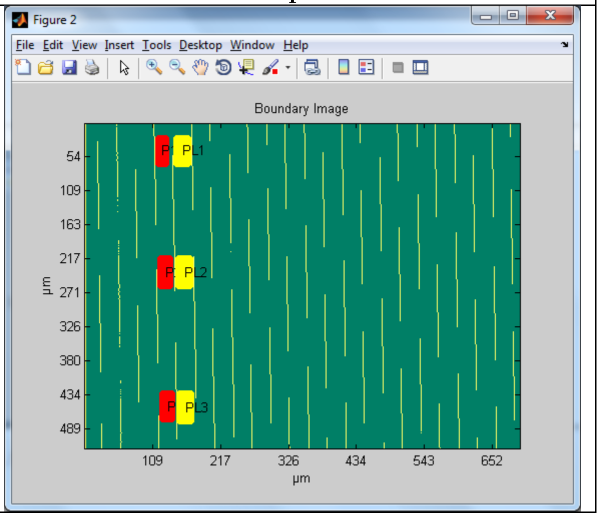
Width	Depth
	

Figure 4: Width & Depth Calculation Using Characterization Toolbox

The results show less than 1µm difference between the three characterization software for width and less than 2 nm for depth parameter. The difference can arise due to the difference in the procedure to calculate the parameters. Characterization toolbox is using surface data instead of profile data to calculate parameters. The second reason is the placement accuracy of cursor on the edge of the step. Due to sidewall slope and roughness on the top surface it is difficult to get a repeatable result. The observed difference is within the repeatability with which the parameters can be calculated.

4.5 Other Tests

Black-box testing of the program was done using the interactive graphical user interface. Several test cases were designed, each of them consisting of input values, expected output values, and the expected program behavior like warnings to the user and protection against abnormal ends. Source code was tested outside the GUI. Each piece of code was tested individually to validate independence of calculated results from interface variables.

4.6 Summary

Software validation is an integral part of the software development cycle. Software should be thoroughly tested before it is released. To validate the developed characterization software first a NIST traceable calibration artifact was used. The results obtained were within the calibration specification of the artifact. A real surface was measured and the results were compared with other comparable characterization software and it was demonstrated that the results are in reasonable agreement.

CHAPTER 5: APPLICATION OF ADVANCED CHARACTERIZATION METHODOLOGY

Previous chapter described the methodology for characterizing the different classes of surfaces in detail. In this chapter we are going to apply the developed methodology on real surfaces. The objective of this chapter is to demonstrate the potential problems in characterization using existing techniques. It will be shown how the new methodology can be used to achieve results. Several examples are discussed hence forth.

5.1. Use Case 1: Etched Grating Surface

Glass and Silicon etching to create deep and shallow trenches is a very important technology in the Semiconductor, MEMS and Optics industry. The creation of deep and high aspect ratio microstructures requires tight control of process. The etching rate is related to the pattern geometry and the process parameters. Any variation in processes appears in feature and will affect the etching micro-uniformity. Etch uniformity is highly dependent on the pattern geometry. The effect is more severe as the feature width becomes smaller. It is important to understand the dominating factor of pattern geometry affecting etching and find a better approach to monitor the pattern uniformity.

5.1.1. Description of the Surface

Figure 1 shows etched pattern in silicon. The important parameters are etched width and depth uniformity. Nominal width and depth are unknown for this data. Pattern was measured using white light interferometer. Image size is 368x240 pixels. Pixel size is

0.82µm and aspect ratio is 1.17. Measured area in microns is 301.9x229.8. Due to the high slope sidewall of the trenches light reflected is outside the capture range of the objective numerical aperture and hence no data. The maximum slope that can be measured with a white light interferometer depends upon the numerical aperture (NA) or magnification of the interferometric objective. The higher the magnification, higher is the NA and larger is the slope that can be measured. With a 50X objective up to 25 degrees of slope can be measured [66].

Typically for etched surfaces the requirement is to characterize trench width and depth uniformity. One possible criterion to determine process capability could be that depth and width should not deviate by more than 1% of the nominal value.

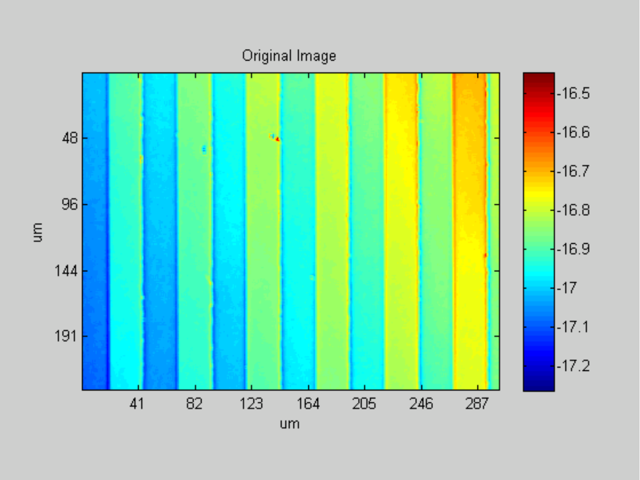


Figure 1: Etched pattern in silicon

Although white light interferometry (WLI) is a good technique for measuring many surfaces, it does not work especially well with step heights that are less than the coherence length of the light source in use. This problem is known as batwings because of the shape of the false information. In this example the depth of the trench is approximately 100nm and light source used in the measurement has a coherence length of 1.2um. Hence batwings appears in the measurement as shown in Figure 2. The top portion close to the edge of step discontinuity, whose height is less than the coherence length, always appears higher and the bottom portion appears lower than it actually is. Higher the magnification (higher NA), more the batwings are. Also if the grating periods are shorter, the height of the batwing increases. Batwings can introduce in error in all three axes. A solution to batwings problem is to use phase shifting technique; in that case false information does not appear. Phase shifting measurement results suggests that the false information has more effect on intensity than on phase [67].

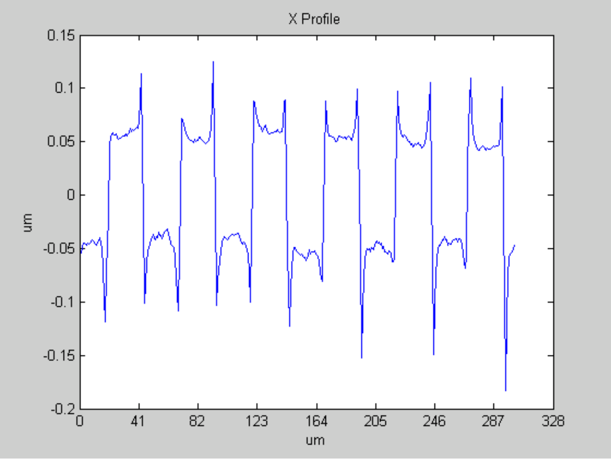


Figure 2: Batwing Effect

Batwings are false information that needs to be treated carefully when calculating metrological characteristics like width and height. In Figure 3, width can be calculated using 3 different algorithms

a) Using the local minima

b) Using the local maxima

c) Using 50% height threshold

In 50% algorithm, a horizontal line is drawn at 50% height between maxima and minima through the optical profile. The points where the threshold crosses the profile are assumed to be the feature edges. The second algorithm chooses the maxima and the third minima as the location of the feature edges.

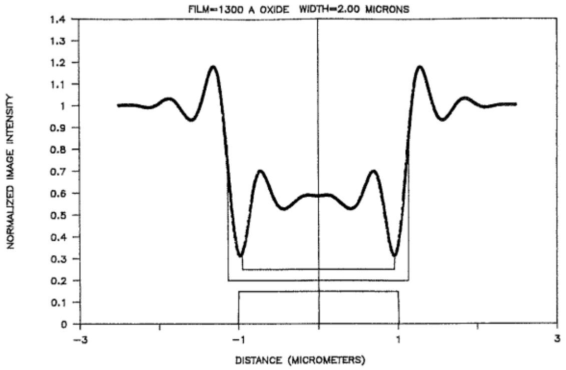


Figure 3: Batwings/ Gibbs Phenomena [72]

5.1.2. Metrological Characteristics

The use of non contact optical measurement technique like WLI has advantage over contact profiling technique due to the fact that it is a surface measurement and not just a cross section scan. Contact techniques like stylus profiling can scratch the surface but can possibly measure larger slopes.

Even though WLI measures surface data but the general practice is to analyze a cross section of the surface to obtain parameters like width and depth. The problem with this approach is that the results are based on two points and the values can change based on how repeatable the two cursors can be placed at the same location. Some data analysis programs allow creating masks. A mask is created on the surface at higher planar level and another mask on surface at lower planar level. Average height of the region inside the mask is calculated for the two surfaces and their difference is reported as the depth. Similar masking technique is used to calculate width also to remove the unwanted edge effects. Although masking is useful but if it is used for averaging the data then the important information about variation in width and depth across the length and width of the feature is lost.

The height image in Figure 1 also shows tilt in the stage as seen by the change in the color scale from green to yellow. Tilt is the relative angle between the optical axis of the system and the object normal, assuming reference mirror is perpendicular to the optical axis. The tilt in the stage can dilute the step. Before any data processing is done, tilt needs to be removed from the surface.

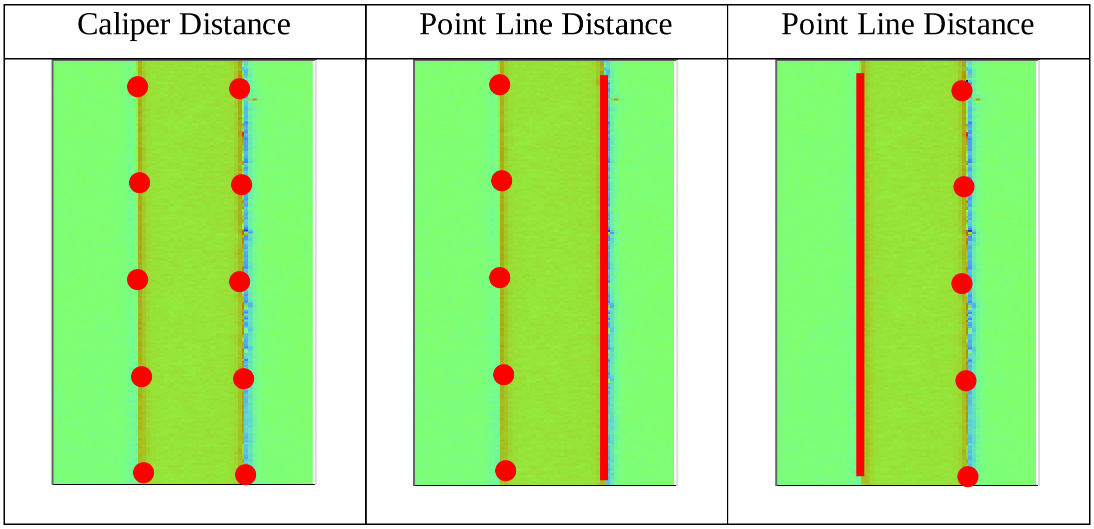
Due to the batwing effect as mentioned in the previous section the edge of the step has false information which is going to adversely affect height and width calculations. It may be required to mask the regions near the edge where the batwings phenomena occur for width and depth calculations.

Typically wafers have fiducial marks on them for aligning the features on the wafer to the measurement axis of the machine. Alignment, sometimes called de-skewing, is essential because it is difficult to put wafer on stage in the same orientation every time.

5.1.3. Approach

The first task is to remove tilt from the measured surface to make the data ready for further processing. A least square plane is fitted to the data to determine the tilt. This least square plane will then be subtracted from the original data to level the surface. This is required for accurate step height calculation. To find grating depth and width uniformity several techniques can be used. For width calculation the possible ways are:

- Caliper Distance (Point-Point): Point to point distance between two points, one on each of the two vertical edges in the same row as shown in figure 4a.

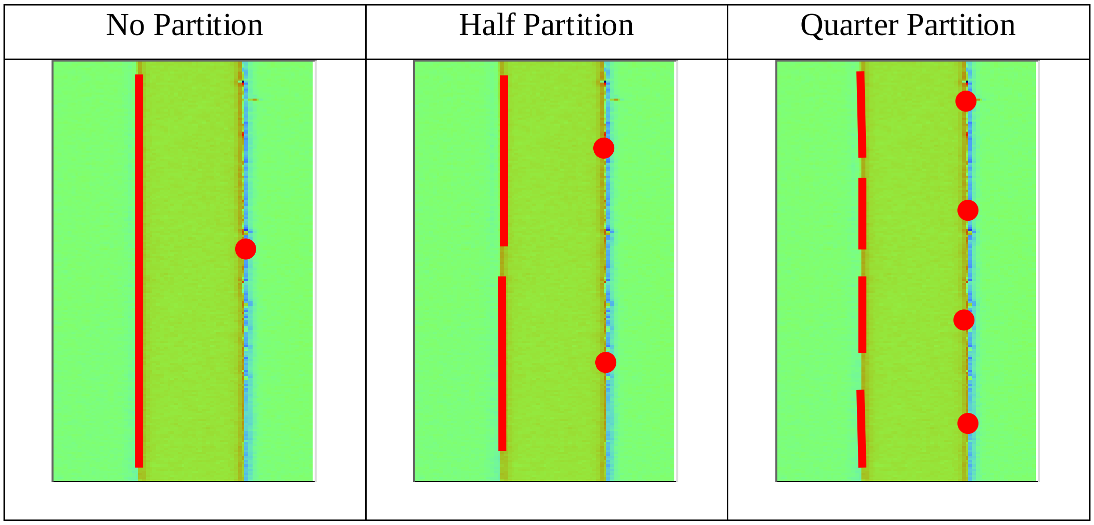

Caliper Distance

Point Line Distance

Point Line Distance

Figure (4a, 4b, 4c): Distance Calculation

- Least Square Distance (Point-Line): Distance is calculated between least square line fitted on one edge and points on the other edge as shown in Figure 4b and 4c.
- Partition Distance: In this case distance is calculated between the two vertical edges using full length, half length and quarter length partitions.


No Partition

Half Partition

Quarter Partition

Figure 5a-no partition, 5b-half partition, 5c-quarter partition

For depth calculation, least square plane can be fitted on one surface and then the distance of all the points on the other plane can be calculated from the least square plane.

Data partitioning is used to determine variation in the depth from the center of the trench towards the edge. Statistics like average width and depth with standard deviation is reported that may help in understanding process variation.

The advantage of calculating the results by partitioning the data provides more local variation by capturing low and high wavelength information to the user to develop a better understanding of the process. It is useful when the surface is measured at high magnification and the high frequency information is visible.

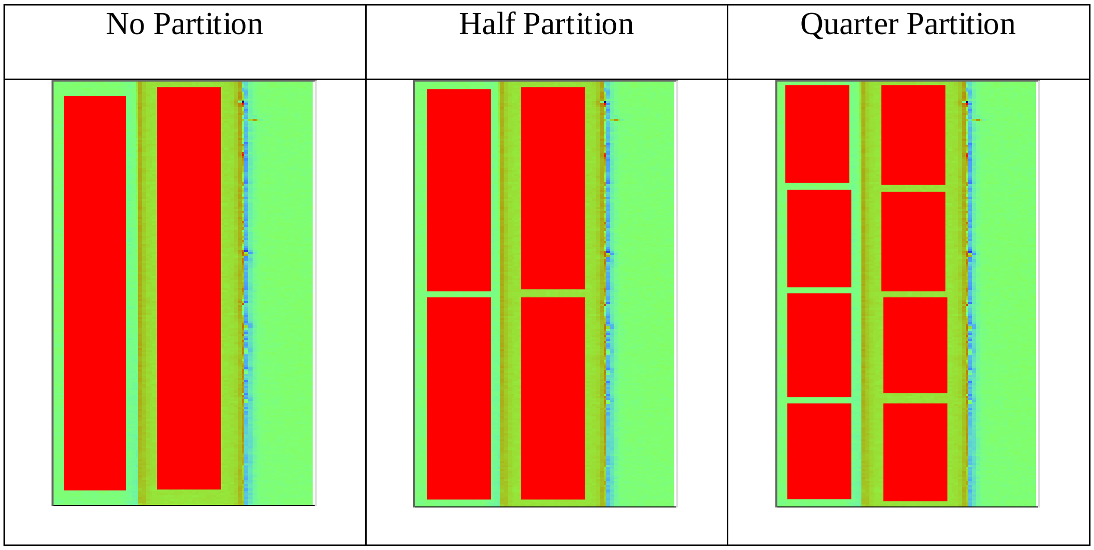


Figure 6a-no partition, 6b-half partition, 6c-quarter partition

5.1.4. Methodology

- Pre-Processing

Tilt in the image is removed by subtracting least square fitted plane from the raw data. In Figure 7 the images in the left column show tilt in the stage which has diluted the step as shown in the histogram. The images in the right column have tilt removed and the histogram clearly shows two clusters present.

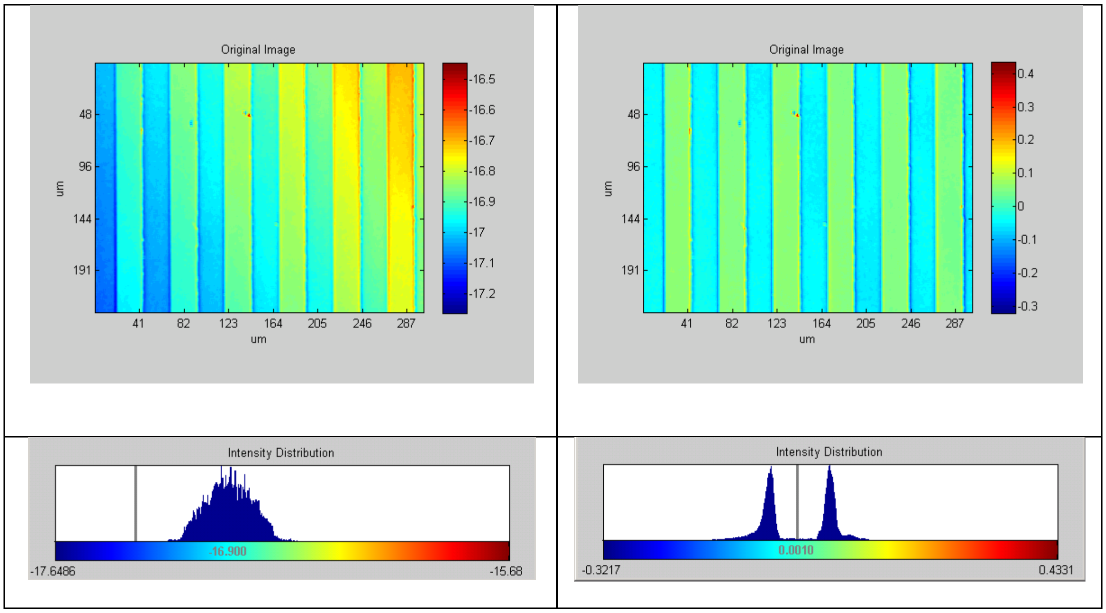


Figure 7: Data Pre-processing

- Segmentation

Segmentation separates the features at two planar levels from each other. As shown in Figure 8, there are two distinct regions in the image histogram. The unetched region is at higher planar level and the second region is the etched region present at lower planar level. As there are only two planar levels in this case, thresholding is used to segment the image into regions.

$$g(x,y) = \begin{cases} 1, & if\ f(x,y) \geq T \\ 0, & if\ f(x,y) < T \end{cases}$$

$g(x,y)$ is the image composed of features at two planar levels. Pixels labeled as 1 correspond to foreground whereas pixels labeled 0 correspond to the background. T is the threshold that separates these modes. Threshold T is called global threshold calculated using Otsu's method. Otsu's method chooses the threshold to minimize the intra-class variance of the thresholded background and foreground pixels.

$$\sigma_w^2(t) = w_1(t)\sigma_1^2(t) + w_2(t)\sigma_2^2(t)$$

Weights $w_i$are the probabilities of the two classes separated by a threshold t and σ_i^2 variances of these classes.

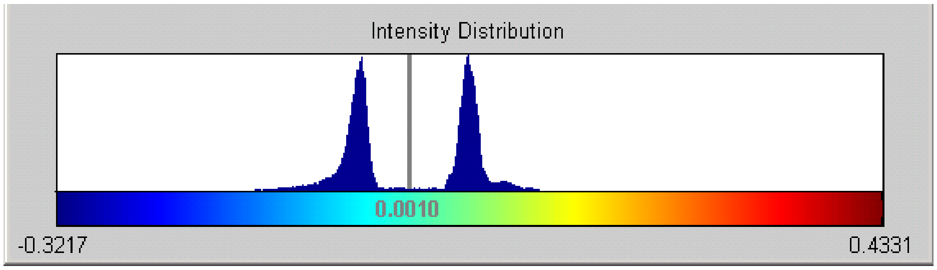


Figure 8: Histogram

.

- Island Detection and Labeling

The next step is to find regions of connected pixels also called islands. Extracting and labeling of various disjoint and connected components in an image is central to many automated image analysis applications Connected component labeling is used in computer vision to detect unconnected regions in binary digital images. For each pixel in the image this algorithm checks all its neighboring pixels to determine if the pixels are connected. If connected pixels are found then they are labeled. All the pixels in one region will have the same label number.

Connected components labeling scans an image pixel-by-pixel (from top to bottom and left to right) and groups its pixels into components based on pixel connectivity, *i.e.* all pixels in a connected component share similar pixel intensity values and are in some way connected with each other. Once all groups have been determined, each pixel is labeled with a gray level or a color (color labeling) according to the component it was assigned to.

Connected component labeling works on binary or gray level images and different measures of connectivity are possible. In this example we have used binary input images and 8-connectivity. The connected components labeling operator scans the image by moving along a row until it comes to a point *p* (where *p* denotes the pixel to be labeled at any stage in the scanning process) for which *V={1}*. When this is true, it examines the four neighbors of *p* which have already been encountered in the scan (*i.e.* the neighbors (i) to the left of *p*, (ii) above it, and (iii and iv) the two upper diagonal terms). Based on this information, the labeling of *p* occurs as follows:

- If all four neighbors are 0, assign a new label to *p*, else
- if only one neighbor has *V={1}*, assign its label to *p*, else
- if one or more of the neighbors have *V={1}*, assign one of the labels to *p* and make a note of the equivalences.

After completing the scan, the equivalent label pairs are sorted into equivalence classes and a unique label is assigned to each class. As a final step, a second scan is made through the image, during which each label is replaced by the label assigned to its equivalence classes. For display, the labels might be different gray levels or colors.

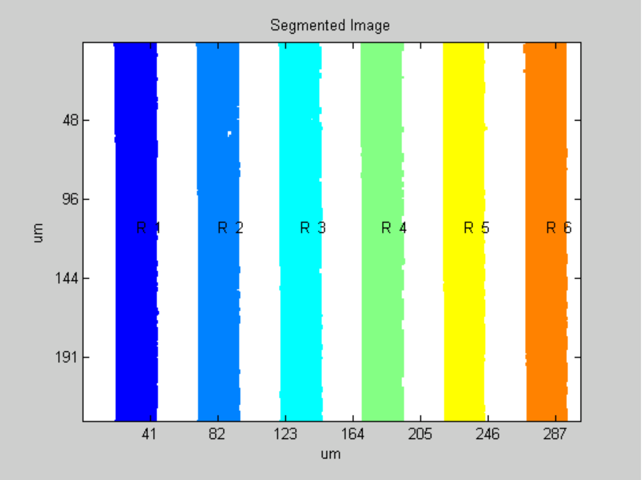


Figure 9: Segmented and Labeled Image

- Edge Detection

Edge detection is used to detect the outer boundary of the features. The detected boundary is used to calculate the dimensions of the feature. To detect the edge segmented image is scanned from left to right. The first non zero pixel in the scan belongs to the first region and is at the boundary pixel. All the eight nearest neighbor of this pixel are checked in an anti-clockwise direction to find another non-zero pixel that is at the boundary. This process is repeated until all the boundary pixels are found (Figure 10).

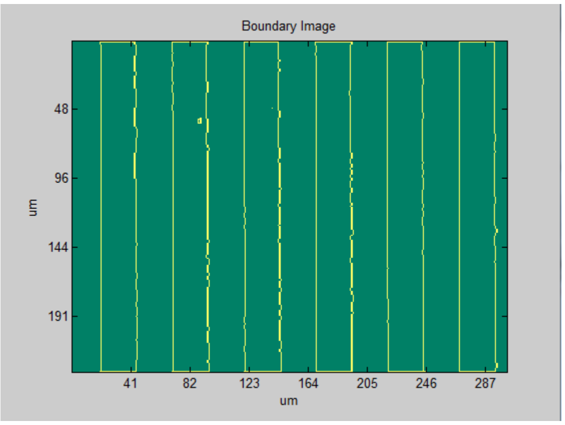


Figure 10: Boundary Image

Boundary Segmentation

In boundary segmentation corner points in the boundary are detected. Corner point is defined as the point where the direction of normal vector changed abruptly.

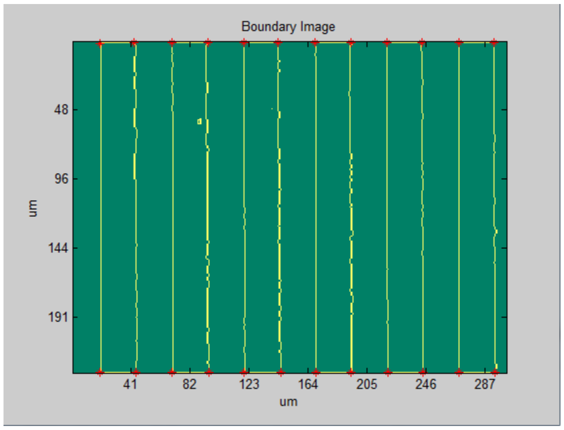


Figure 11: Corner Points

5.1.5. Characterization

Width Calculation

- Caliper Distance (Point-Point): Point to point distance between two points, one on each of the two vertical edges in the same row. Five pairs of points are created labeled as P1-P2, P3-P4, P5-P6, P7-P8, and P9-P10.

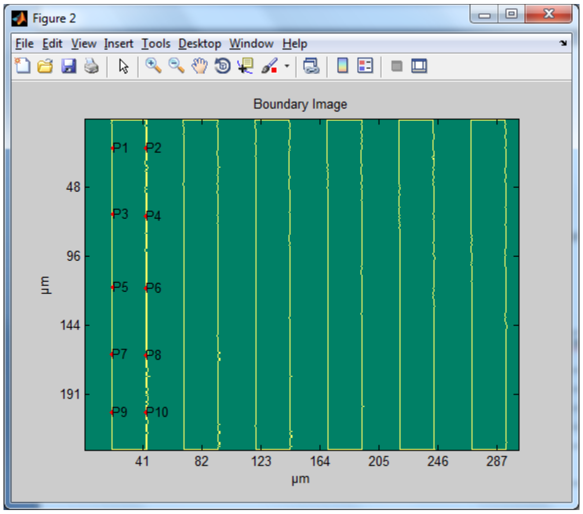


Figure 12: Caliper Distance

Table 1: Caliper Distance Results

Average Distance	MATLAB (µm)	Veeco (µm)
P1P2	23.8	23.35
P3P4	22.98	23.35
P5P6	22.98	23.85
P7P8	22.97	24.18
P9P10	23.79	23.85

- Least Square Distance (Point-Line): Distance is calculated between least square line fitted on one edge and average of points on the other edge.

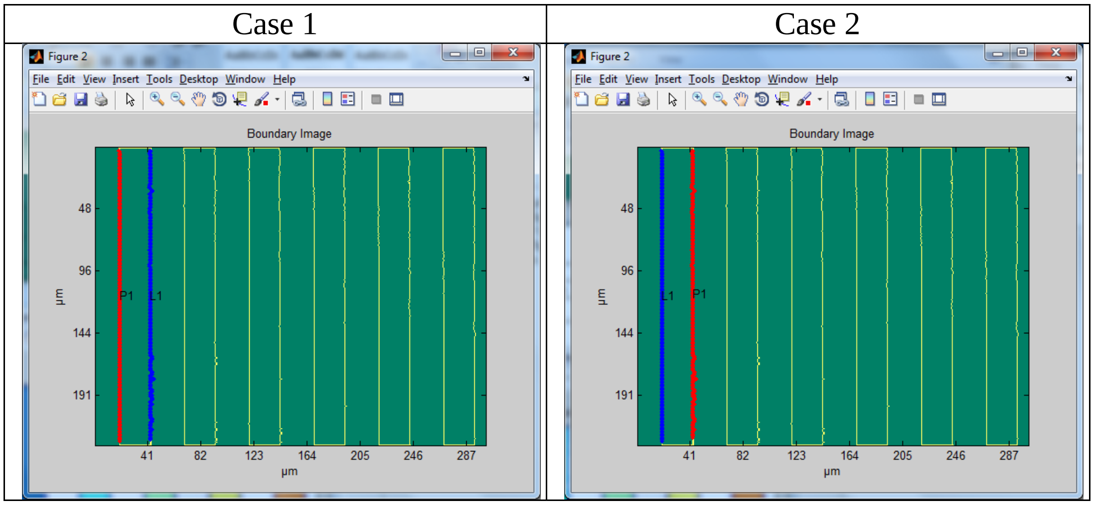

Case 1 | Case 2

Figure 13: Point Line Distance

Table 2: Point Line Distance Results

L1P1	MATLAB (µm)
Average Distance	23.42
Standard Deviation	0.25
Maximum Distance	23.90
Minimum Distance	23.07
Range Distance	0.83

- Partition Distance: Feature is divided into several partitions and distance is calculated. In case 1 line is fitted to the left edge and average point is taken on the right edge and case 2 is vice versa.

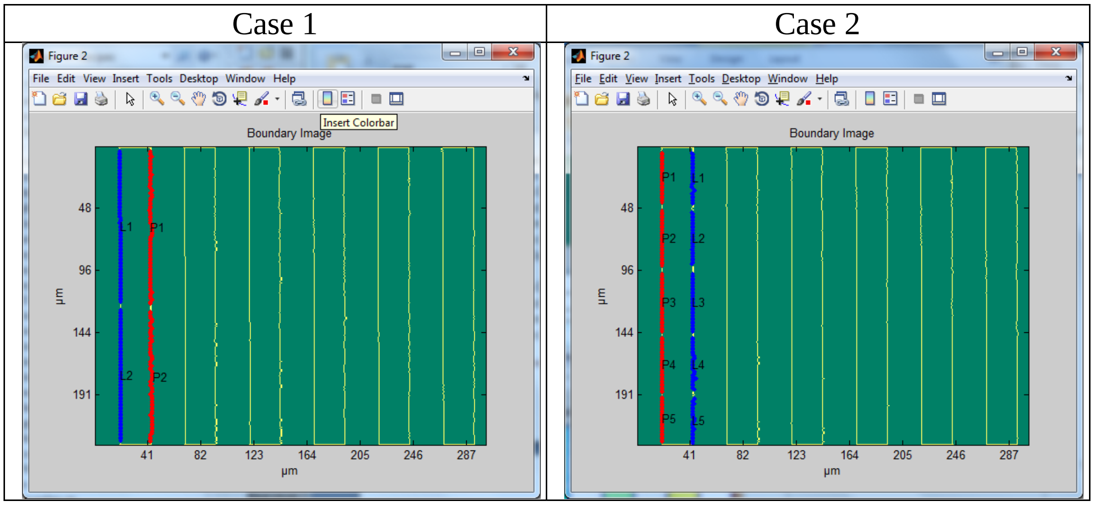

Case 1 | Case 2

Figure 14: Partition Distance

Table 3: Case 1Results

	P1L1 (μm)	P2L2 (μm)
Average Distance	23.45	23.40
Standard Deviation	0.49	0.41
Maximum Distance	24.73	23.79
Minimum Distance	22.65	22.97

Table 4: Case 2 Results

	P1L1 (μm)	P2L2 (μm)	P3L3 (μm)	P4L4 (μm)
Average Distance	23.80	23.03	23.28	23.55
Standard Deviation	0.40	0.21	0.40	0.37
Maximum Distance	24.77	23.79	23.79	23.79
Minimum Distance	22.68	22.97	22.97	22.97

Depth Calculations

Depth is calculated using point to plane distance. In Figure 15 a large area is used to calculate the average trench depth.

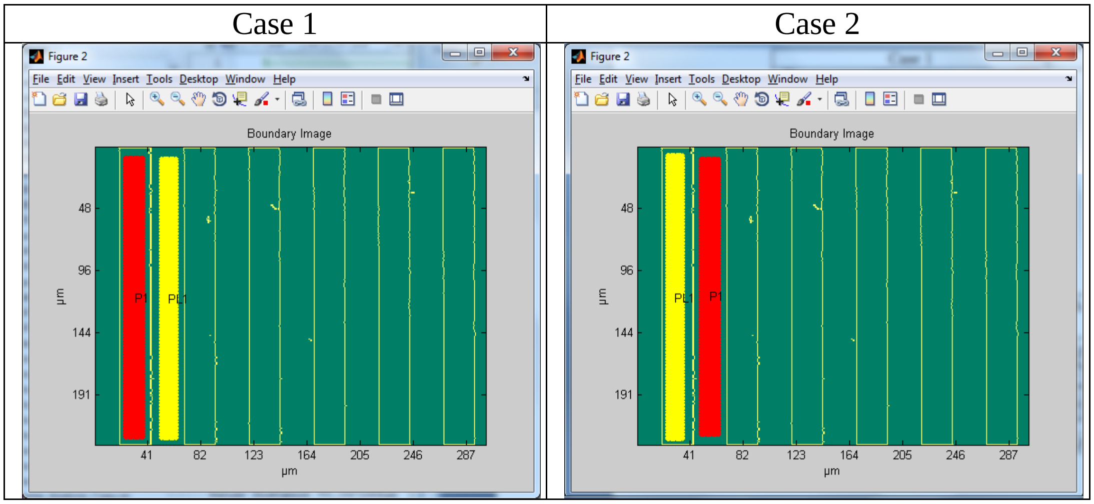


Figure 15: Depth Calculation

Table 5: Trench Depth Results

	Case 1(nm)	Case 2(nm)	Veeco (nm)
Average Distance	84.68	85.27	85.0
Standard Deviation	1.03	0.92	
Maximum Distance	87.6	87.49	86.5
Minimum Distance	82.35	81.81	83.5
Range Distance	5.24	5.68	3.0

Figure 16 shows that the top and bottom of the trench is partitioned

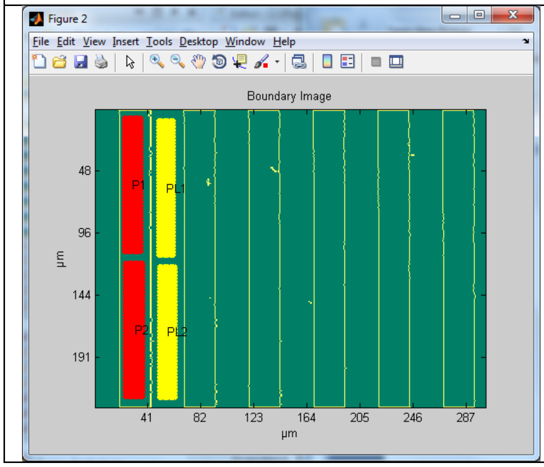
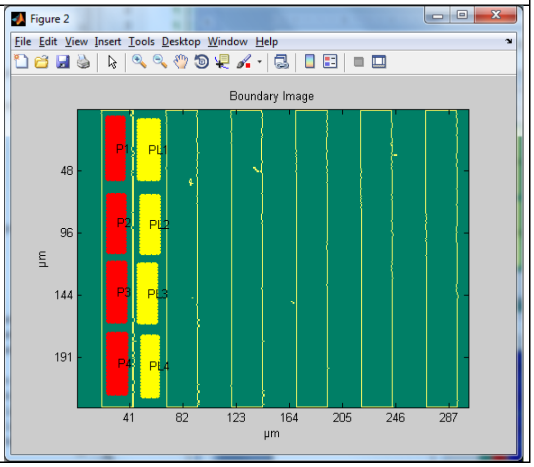
Case 1	Case 2
	

Figure 16: Depth Calculation by Partitioning

Table 6: Results

Case 1 (nm)		Case 2 (nm)	
P1L1	84.76	P1L1	84.14
P2L2	84.87	P2L2	84.81
		P3L3	84.24
		P4L4	84.39

5.1.6 Summary

When calculating the etch uniformity, understanding metrological characteristics using multiple ways can provide deeper insight into the quality of etch. We suggested several choices to determine the width and depth variation. The characterization toolbox gives user the flexibility to average the points, fit least square line and plane, and select regions of varying sizes to calculate the required parameters.

It is recommended to use phase shifting technique to avoid batwings effect. It is possible that the batwings are still present using other techniques also. The choice of three different algorithms is provided to select the option that best suits application needs.

5.2. Use Case 2: Solder Bump Array

Die bumps are used to make electrical contact between chip and substrate. To maintain good electrical contact characteristics like height and flatness of each bump needs to be inspected. A wafer contains hundreds of die and each die has hundreds of bumps. White light interferometer is used to measure these bumps due to its non-contact measurement technique, high speed and repeatability. Vertical scanning interferometry is used to measure due to the high slope and bump height greater than 0.2µm. Phase shifting interferometry although more accurate can only be used to measure height up to 160nm.

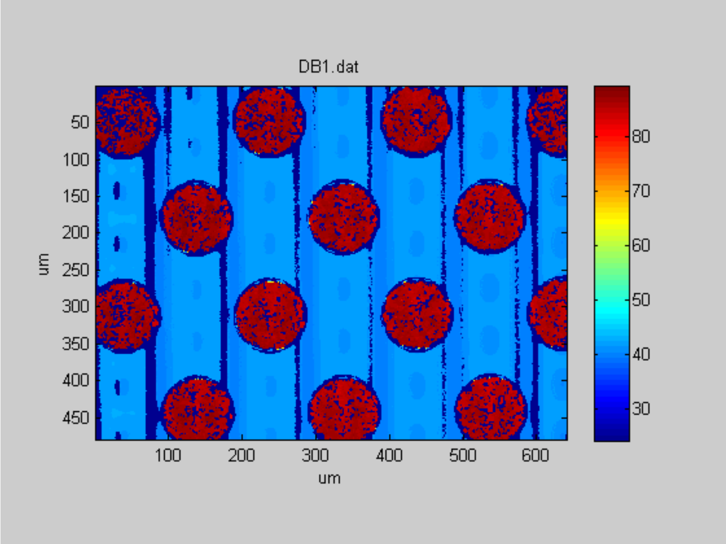


Figure 17: Die bumps

5.2.1. Description of the Surface

Figure 17 shows image of die bump measured using WLI. In this data bump characteristics like diameter, circularity and height are important. Also flatness and coplanarity of the bumps plays an important role in the proper functioning of the part. Nominal dimensions are unknown. Image size is 640X480 pixels. Pixel size is 1.12µm and aspect ratio is 1. Measured area in microns is 640x480. Due to the vertical sidewall of the trenches there is no light reflected back and hence no data. As seen in figure 17, there is significant data dropout in the measured data. Data dropout can occur due to several reasons. One reason is that the surface is poorly reflective and hence not enough light is reflected back from those areas resulting in black spots indicating no data. It is also possible that the modulation threshold is set high which discards pixels those pixels whose intensity is lower than the set threshold. Modulation threshold determines the signal-to-noise level for which a given pixel is considered “valid.” Data points that don’t meet the threshold are marked as invalid and are not considered during analysis. Another thing to check is the scan length in Z for each measurement. If scan length is less than the height of the feature, then there will be no interference outside the scan region and hence no fringes. This will result in data dropout. One more possible reason could be that surface is saturated with light at some locations. This can be easily verified before doing the scan by checking the intensity in the scan region in Z and look for any red spots typically used to indicate saturation points. Intensity should be reduced at all these points.

5.2.2. Metrological Characteristics

Due to significant data dropout around the edge of the bumps, it is difficult to calculate diameter of the bump accurately. Also the calculation to determine the position of center of the bump is dependent on identifying the true edge of the bumps. Another problem may arise in calculating bump height because substrate is warped possibility due to the stresses developed during the dicing process. In this case a better option is to find the local height of the bumps. Flatness and coplanarity calculations require bump data to be separated from the substrate data. Flatness tolerance is defined by two parallel planes within which the surface must lie. No reference surface is required. Coplanarity tolerance is used when it is desired to treat two or more surfaces as a single interrupted or noncontinuous surface. Tolerance applied is similar to flatness tolerance.

5.2.3. Approach

Data Dropout

The holes in the bump can be filled by using image morphology technique like dilation. Dilation causes objects to grow in size as it exchanges every pixel value with the maximum value within a 3x3 window size around the pixel. Dilation connects the disconnected pieces of the edge and creates a continuous boundary. This helps in robust calculation of diameter and position of the bumps.

Diameter & Circularity

Diameter and center of the circle is determined by fitting a least square circle to the edge points. Circularity or roundness tolerance specifies a tolerance zone bounded by two concentric circles within which each circular element of the surface must lie. The radial distance between the two concentric circles is the circularity. Circularity is a form error and desired to be less than 0.5% of the nominal diameter of the circle.

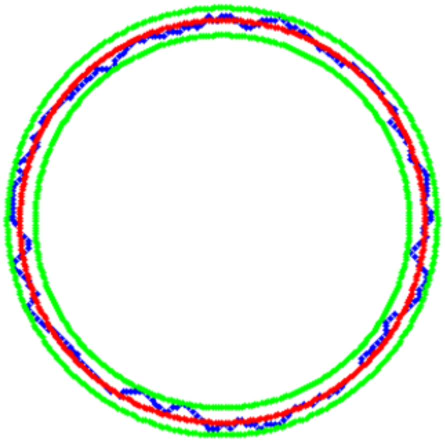


Figure 18: Diameter and Circularity

Position Error

A positional tolerance defines a zone within which the center, axis, or center plane of a feature of size is permitted to vary from a true (theoretically exact) position. Position error can be described in several ways. One way is to find the difference between the measured coordinated and the target coordinated and report difference in X and Y direction. An error vector can be calculated from Δx and Δy which gives the magnitude and direction of error. The measured position can be checked against the allowed tolerance or the magnitude of the error can be checked to see if it is not more than the permissible error.

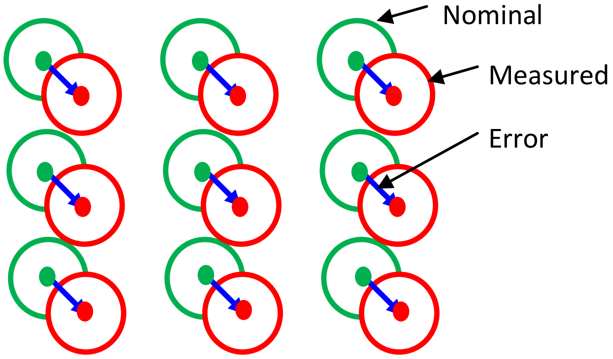


Figure 19: Position Error

Shrinkage Error

Runout error is applicable to materials that either contracts or expands due to materials thermal properties. The error is radial in direction but the magnitude can be different in X and Y direction. An assumption is made that runout is zero at the center. It is different from position error because the magnitude is different for each location. In Figure 20 red circles shows the measured position and green circles the nominal position. In runout the position of red circles gets progressively larger or smaller in the radial direction.

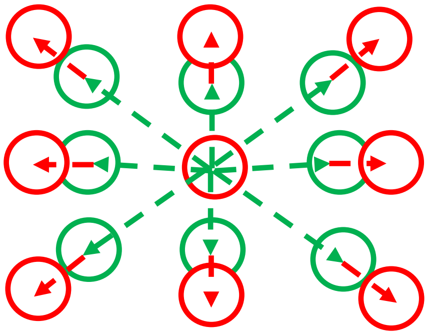


Figure 20: Shrinkage Error

Distance

Calculating distance involves several choices, the three more common ways to report distance are:

(i) Center to center distance

(ii) Shortest distance

(iii) Longest distance

Also the distance can be two dimensional or 3 dimensional.

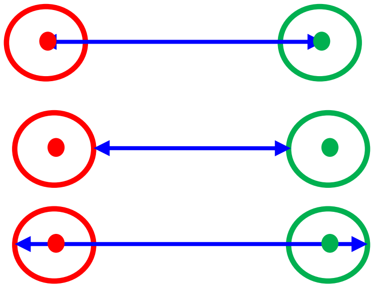


Figure 21: Distance between features

Flatness

To calculate flatness of bumps, image segmentation technique is used to separate features at different planar levels. After separating the bump data from the substrate data, least square plane fitting algorithms is used to calculate the local flatness of each bump.

Figure 22 shows the variation on the surface of a bump. The red line indicates the best fit line.

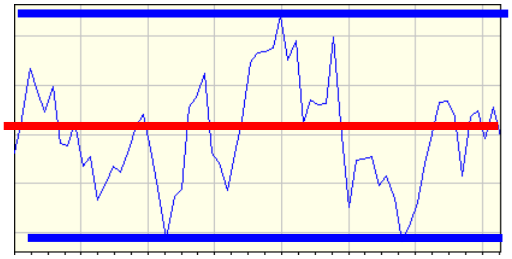


Figure 22: Flatness

Coplanarity

Coplanity is calculated the same way as flatness but instead of using data from one bump at a time, data from all the bumps surfaces is used treating them as one continuous surface. Figure 23 shows three bumps in the field.

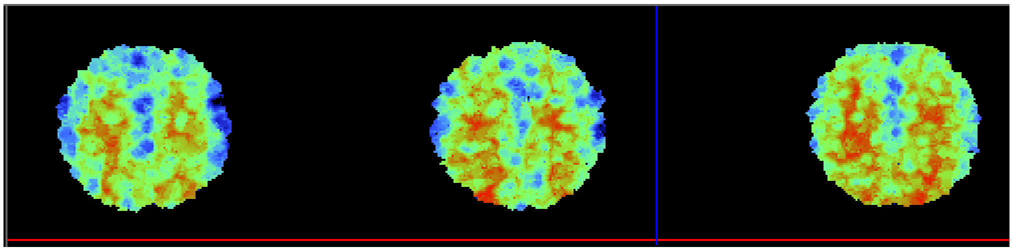


Figure 23: Coplanarity

Height

To calculate the local height of the bump, a bounding box can be created around each bump that fits the bump and surrounding substrate inside the box. This process when repeated can be used to calculate the local height of each bump in the image. Height is calculated using the data in the center of the bump and substrate and using the entire data as shown in the Figure 24. The area selected for calculation can be increased or decreased and the effect on height calculation should be monitored.

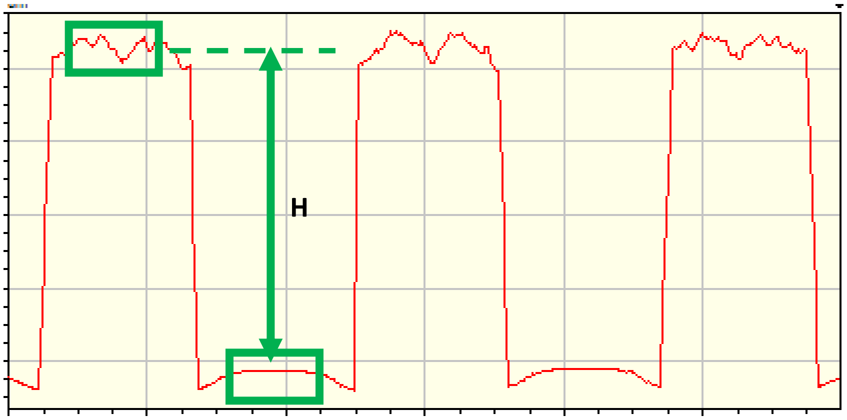


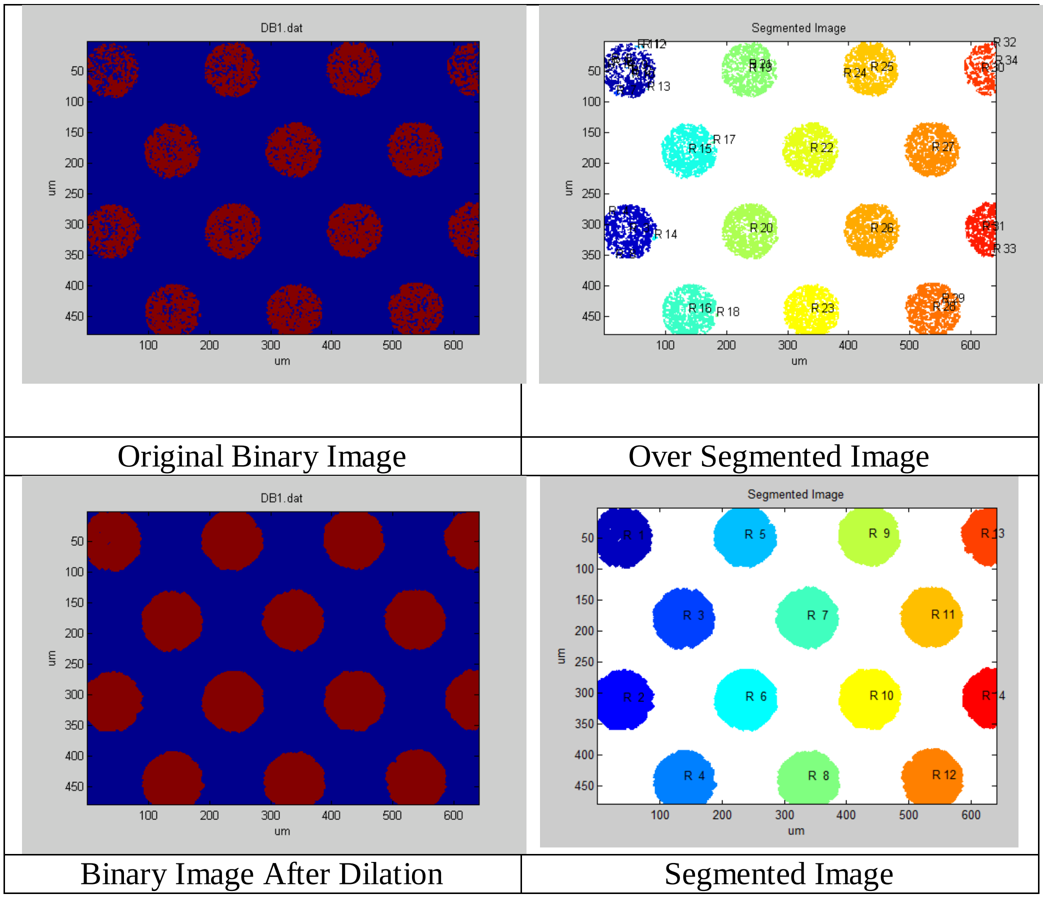
Figure 24: Height

5.2.4 Methodology

Pre-Processing

Figure 25 shows the original binary image of die bump. The image contains perforation in each bump. The desired segmentation should isolate each bump into a distinct region. As there are 14 bumps in the image, we would expect to see the same number of distinct regions after segmentation. But when the segmentation is performed based on the pixel connectivity, due to the perforations in the bumps, over segmentation occurs and we get more than 14 regions, 34 in this case.

One way to approach this issue of over segmentation is to use image morphology operations like dilation to fill the perforation in the data. This will ensure all the pixels belonging to the same region are well connected and will ensure proper segmentation with 14 islands in the current example as desired. In dilation a structuring element is used to grow the region. In this case a disk shaped structuring element with a diameter of one pixel is used 5 times to fill the holes in the bump data. As a result the holes are filled and segmentation created the same number of islands as there are number of bumps as shown in the bottom two images in Figure 25.



Original Binary Image

Over Segmented Image

Binary Image After Dilation

Segmented Image

Figure 25: Binary Image

Segmentation

Thresholding is used to separate bump data from substrate. Thresholding is easy to implement and works well where all the surface features are at the same planar level. In this case a threshold level of 61.32 um is selected. Figure 26 is the histogram of the image showing two distinct regions. The region with lower average height is the substrate while the other region is the bump.

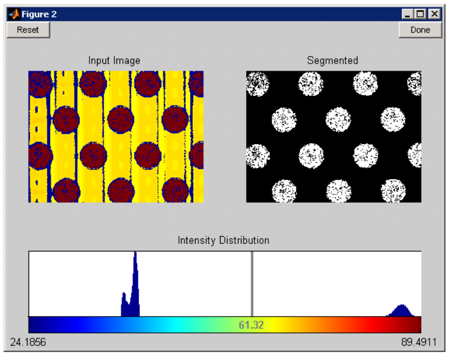


Figure 26: Threshold level selected to separate bump from substrate

Segmentation results in a binary image where the data points above the threshold are converted to 1 and the data points below the threshold are converted to 0 indicated by red and blue colors respectively.

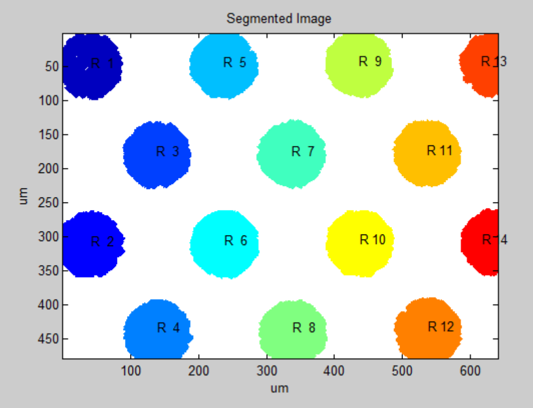


Figure 27: Segmented Image

Island Detection and Labeling

The next step is to find regions of connected pixels also called islands. Connected component labeling is used in computer vision to detect unconnected regions in binary digital images. For each pixel in the image this algorithm checks all its neighboring pixels to determine if the pixels are connected. If connected pixels are found then they are labeled. All the pixels in one region will have the same label number. In this example centroid of each labeled region is found. A rectangular bounding box is created around each bump. Size of bounding box is the minimum size rectangle within which the region can fit. Bounding box is created to calculate the local characteristics like height of the bump and local substrate flatness.

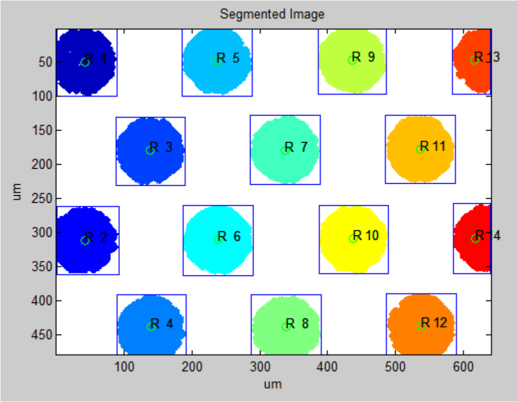


Figure 28: Segmented Image

The data points inside each bounding box are separated into bump and substrate data based on user defined threshold. Average of the bump and substrate data is calculated. Points outside the average +/- 3σ limit are treated as outliers and omitted from the calculations. This process is repeated for all the bumps on the measured surface. For bumps that fit in the field of view partially can be ignored from the calculations.

Figure 29 shows the datapoints belonging to substrate in blue color and bump datapoints in red color. As seen in the image average bump height is 86 µm and average Substrate height is 41um. Bump height relative to the substrate is 45 µm. Standard deviation is around 0.9µm.

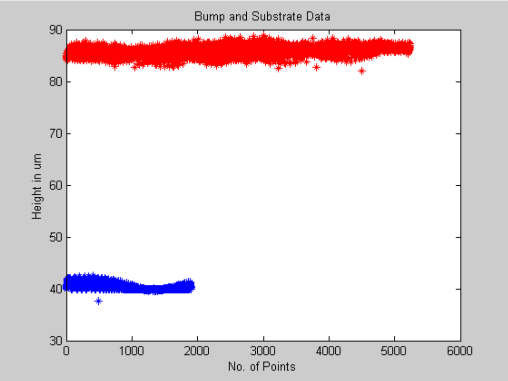


Figure 29: Bump and Substrate Data Points

Edge Detection

Edge detection is used to determine the boundary of bump features. The points on the edge are used to calculate the diameter of bump, circularity and coordinates of the centroid of the bump.

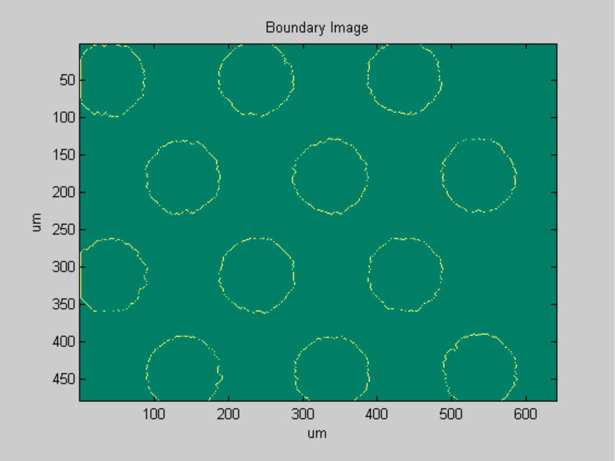


Figure 30: Segmented Image

5.2.5 Characterization

Flatness is calculated by fitting a least square plane to the data and then finding out the maximum and minimum distance points from the plane. Bump height is calculated by fitting a least square plane to the local substrate data for each bump and then finding the distance between point (bump centroid) and a plane (substrate).

Least square circle fitting is done on the bump edge data to get the diameter and center of bump. Circularity is calculated by fitting two circles (inscribed and circumscribed) to the boundary points that are concentric with the least square circle. The radial distance between the inner and outer circle is the circularity.

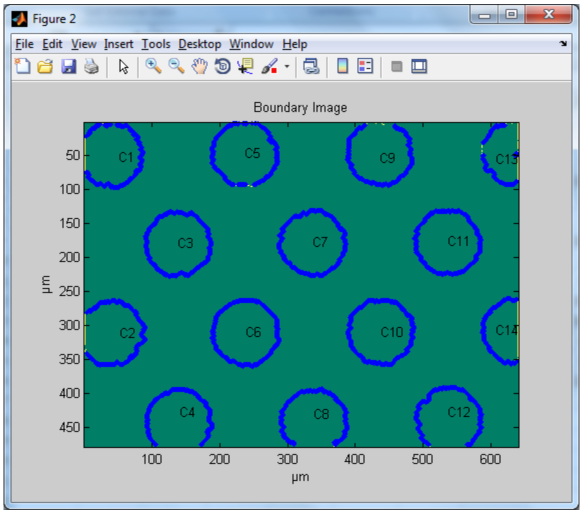


Figure 31: Bump Diameter, Circularity and Position

Table 7: Bump Characteristics in mm

Bump ID	X	Y	Radius	X nominal	Y nominal	dx	dy
1	39.5413	48.945	46.2241	38	50	1.5413	-1.055
5	236.7862	47.6559	47.1237	238	50	-1.2138	-2.3441
9	436.0186	46.1511	47.064	438	50	-1.9814	-3.8489
3	138.5401	179.8376	47.2876	138	180	0.5401	-0.1624
7	337.252	178.7407	47.3451	338	180	-0.748	-1.2593
11	536.1637	177.5645	47.2396	538	180	-1.8363	-2.4355
2	39.9257	311.8026	46.7889	38	310	1.9257	1.8026
6	238.2385	310.8487	48.0063	238	310	0.2385	0.8487
10	437.3889	309.7627	47.4424	438	310	-0.6111	-0.2373
4	139.4913	442.4369	47.3889	138	440	1.4913	2.4369
8	338.8682	442.111	47.3758	338	440	0.8682	2.111
12	538.0648	439.4705	46.969	538	440	0.0648	-0.5295

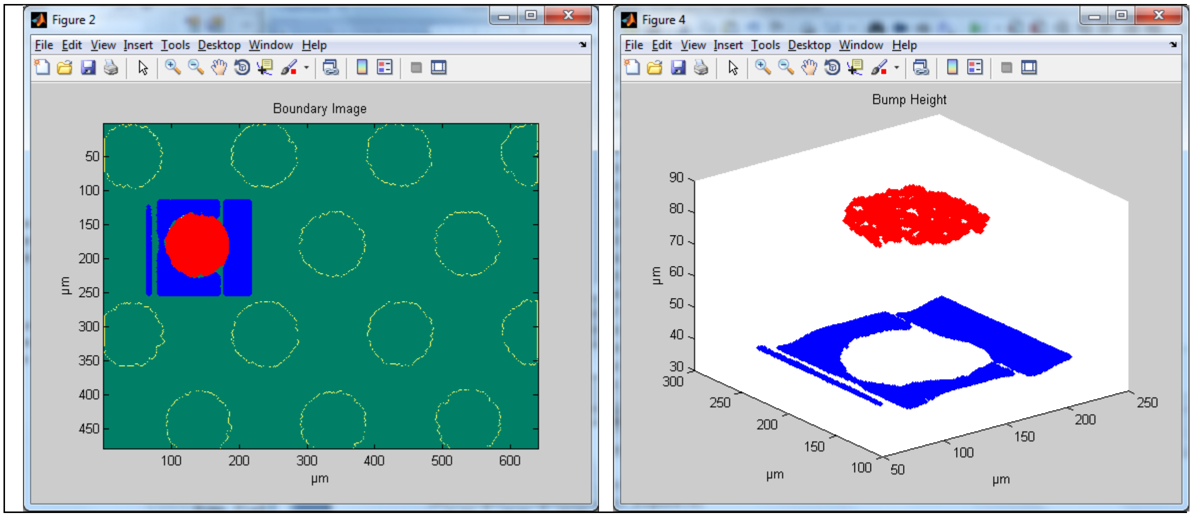


Figure 32: Bump Height & Flatness

Table 8: Height and Flatness Results

Feature	Bump Height (µm)	Bump Flatness (µm)	Substrate Flatness (µm)
C1C2	44.88	5.52	2.84

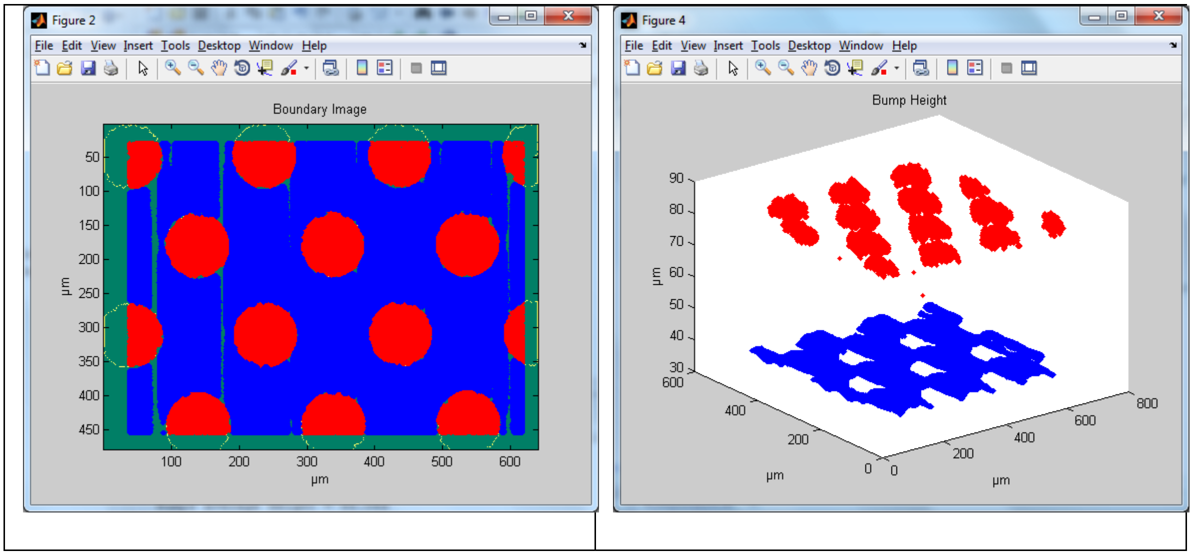


Figure 33: Bump Flatness and Coplanarity

Table 9: Coplanarity Results

Bump Height	Bump Coplanarity	Substrate Coplanarity
44.05µm	16.32µm	3.12µm

5.2.6 Observations

For a patterned surface like solder bumps, several metrological characteristics can be determined to fully characterize it. It was shown how some of the regular parameters like position, height, flatness, diameter, and pitch can be calculated. Some additional parameters like bump position error and shrinkage can give valuable information about the placement accuracy of the bumps. Using the methodology all the measured bumps can be analyzed simultaneously.

5.3. Use Case 3 – Spherical Bumps and Lenses

Discrete or array based spheres are used in a variety of applications. Examples include ball grid array in semiconductor industry, spherical micro-lenses in optics etc. [68]. Size of spheres can vary from few microns to several millimeters. Height, radius, pitch are the chief parameter necessary to be controlled for proper functioning of the part.

Spheres are created by etching or molding glass, molding polymers on substrate etc. The spheres' form can vary across the substrate due to changes or inconsistencies in etching or replication process resulting in deviation from the nominal geometry.

Optical profiling (White Light Interferometry) is used to measure spherical surfaces due to its high speed stage automation, excellent repeatability and high resolution. Stitching option is available to measure large lenses. The only drawback is that the lens slope must be less than $\lambda/4$ pixel to pixel.

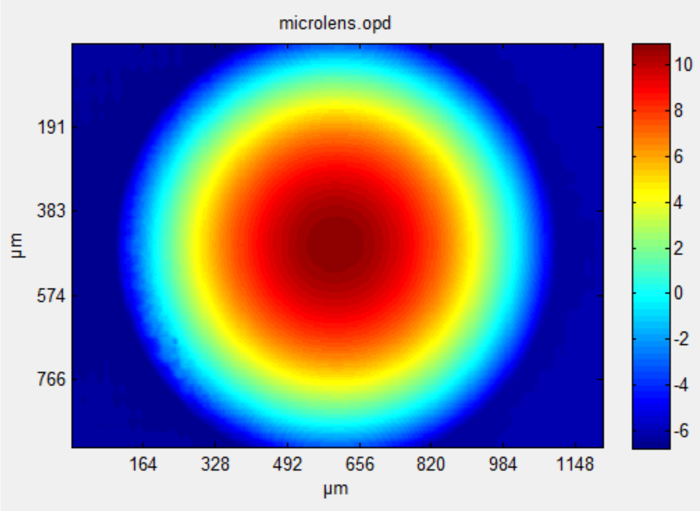


Figure 34: Microlens

5.3.1. Description of the Surface

Field of view is 368x240 pixels. Pixel size is 3.28µm. Measured area is 1.2mmx0.92mm. It is difficult in interferometric metrology to maintain high spatial resolution over a large field of view. Interferometric microscope measurements yield high resolution, but only over a small area. Other conventional interferometric systems can measure large areas, but they fail to provide the necessary spatial resolution. High spatial resolution over a large field-of-view (FOV) can be obtained by stitching together multiple high spatial resolution measurements of adjacent areas of a measured surface. The measurements can be fit together in a global sense, or by matching the piston and tilt over the overlap region. Care must be taken in the stitching process to make sure the measurements are precisely overlapped to minimize errors. The larger the overlap the easier it is to match data sets, but of course more data sets are required to get a given field of view.

5.3.2 Metrological Characteristics

For optical lenses metrology consists of measuring Radius of Curvature (ROC), Sag, power of the lens etc. which is not in the scope of this work. The metrological characteristics that we are discussing here are form error, center of the sphere, rotational symmetry and height. The next section shows the approach to calculate these metrological characteristics.

5.3.3 Approach

Peak to Valley of residual surface

Least square sphere fit is used to determine deviation from the design sphere or asphere. Sphere fit can be absolute or best fit. In absolute sphere fit, least square sphere of the desired radius is fitted and the residual profile is studied. In best fit sphere, sphere is fitted to reduce the sum of squared deviation from the fitted sphere and radius of the sphere that reduces the residuals to minimum is reported. Peak to valley of the residual profile is an important indication used to measure deviation from design.

Radius, Center & Height

Least square sphere can be fitted to the surface to find the radius and center of the sphere. Sphere fitting is done on small to large size of the dataset to see any variation in the center position of the sphere. Figure 35 shows three areas for calculating center.

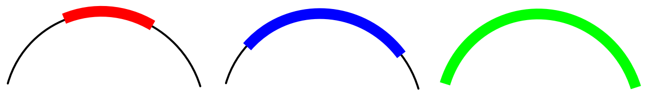


Figure 35: Selecting Area for ROC

Height at the center of the sphere is reported along with the coordinates of the center.

Rotational symmetry

Another parameter is the rotational symmetry of the lenses. Various slices of the sphere in Z can be taken and least square circle can be fitted. The fitted circle should for a ring of concentric circles. The center of all the rings should be inside a defined tolerance zone.

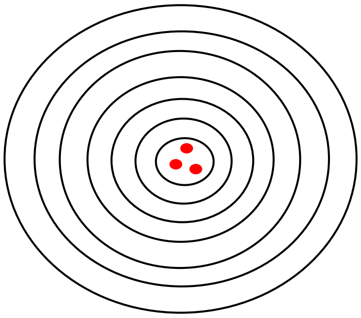


Figure 36: Rotational Symmetry

5.3.4 Methodology

Methodology consists of performing segmentation using thresholding and edge detection.

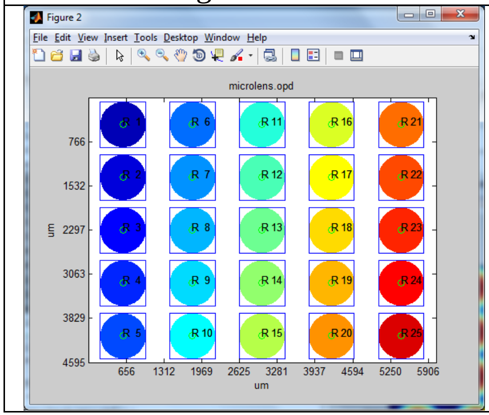
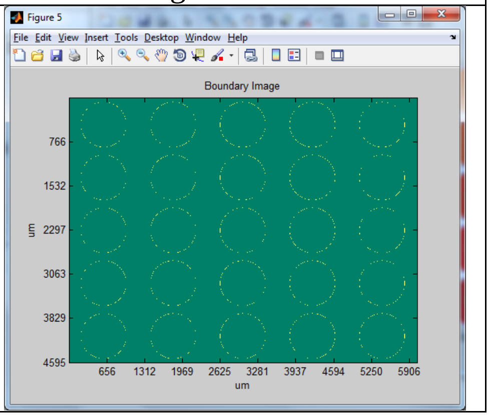
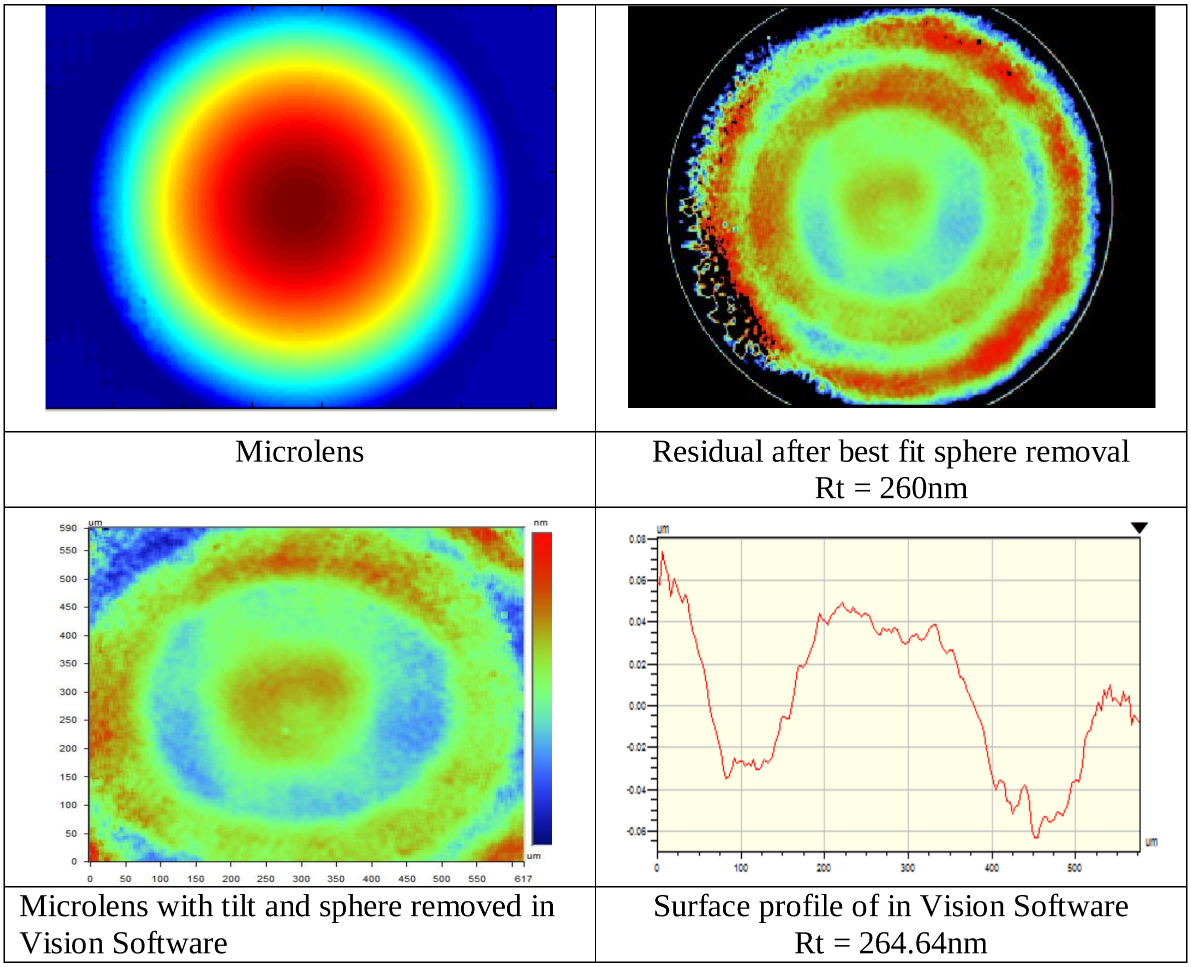
Segmentation	Edge Detection
 Figure 2 File Edit View Insert Tools Desktop Window Help microlens.opd R 1 R 2 R 3 R 4 R 5 R 6 R 7 R 8 R 9 R 10 R 11 R 12 R 13 R 14 R 15 R 16 R 17 R 18 R 19 R 20 R 21 R 22 R 23 R 24 R 25 um 766 1532 2297 3063 3829 4595 656 1312 1969 2625 3281 3937 4594 5250 5906 um	 Figure 5 File Edit View Insert Tools Desktop Window Help Boundary Image um 766 1532 2297 3063 3829 4595 656 1312 1969 2625 3281 3937 4594 5250 5906 um

Figure 37: Segmentation and Edge Detection

5.3.5 Characterization

Figure 38 shows the measured sphere on the left and residual surface on the right. The residual surface is obtained by taking the difference of the measured and best fit sphere. The same dataset was also analyzed in the Vision software. Tilt and sphere were removed from the data. The residual surface showed Rt = 264.64nm. The Rt value obtained from the characterization software is 260nm.



Microlens

Residual after best fit sphere removal
Rt = 260nm

Microlens with tilt and sphere removed in Vision Software

Surface profile of in Vision Software
Rt = 264.64nm

Figure 38: Residual Figure Error

Center & Height

In Vision software the center coordinates and height are obtained in the 2D analysis view. Figure 39 shows the results.

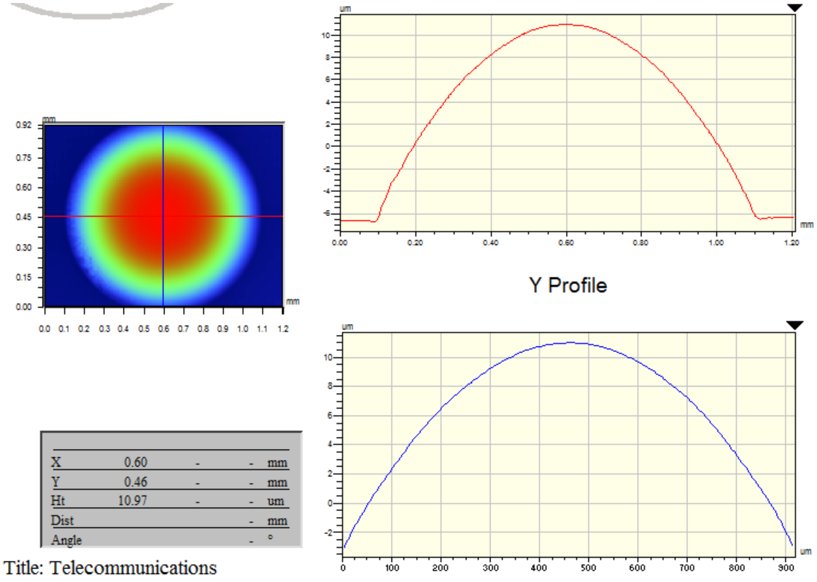


Figure 39: Residual Figure Error

Figure 40 shows analysis done using the characterization toolbox selecting three different sizes area for sphere fitting. The results are shown in Table 10.

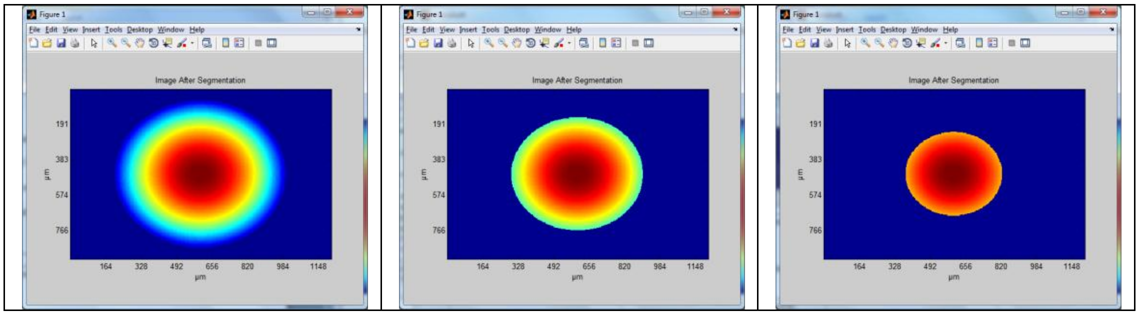


Figure 40: Lens Area for ROC Calculation

Table 10: Sphere Fitting Results

	Vision	Large Area	Medium Area	Small Area
X	600µm	602.17	602.04	601.93
Y	460 µm	458.74	458.48	458.28
Height	10.97 µm	10.965	10.967	10.967

Rotational Symmetry

Boundary data of the lens is used to calculate the center of the lens. Figure 41 shows 3 different sphere sizes chosen for least square circle fitting to calculate center.

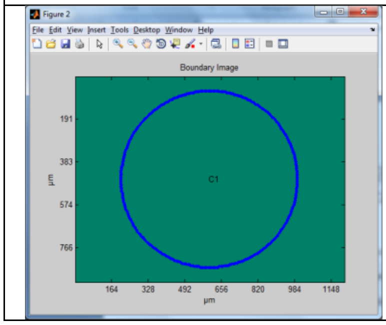
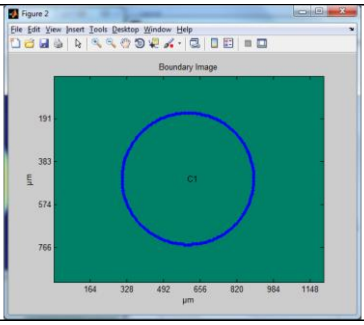
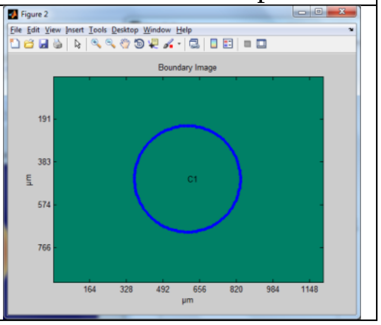
Perimeter Bottom	Perimeter Middle	Perimeter Top
		

Figure 41: Lens Profile

Table 11: Center Coordinates

	Perimeter Bottom	Perimeter Middle	Perimeter Top
X	602.15	602.12	602.24
Y	458.83	458.26	458.37

5.3.6 Observations

The techniques used in this example can be used to characterize lens uniformity and symmetry. It is recommended to use low modulation threshold when measuring steep lenses. If the lens size is big then stitching can be used to capture a large area.

5.4 Micro Prism Array

Prismatic and fluorescent prismatic retro-reflective sheeting offers excellent daytime color and the highest levels of reflectivity. Prismatic sheeting is applicable for all high speed roadways and urban areas where higher ambient light levels can make signs less visible. It is ideal for overhead guide signs and directional signs. Traffic signs manufactured with prismatic retro-reflective pattern provide early detection and an extended range of sign legibility [69]. Sheetings get their color by absorbing some wavelengths and emitting or reflecting other wavelengths. For non-fluorescent sheetings, the absorbed light is lost. For fluorescent sheetings, some of that absorbed light becomes additional emitted light. This is what produces the exceptionally bright colors of fluorescence. The brightness advantage is especially pronounced at twilight and dawn.

Studies have been done on aspects of fluorescence that relate to high-visibility retroreflective materials used for visual signaling and markings [70]. Full colorimetric characterization of these materials requires separation of the fluorescent and non-fluorescent components. Quantification of the individual components allows for accurate prediction of performance under the various conditions of illumination and viewing encountered in practical applications. When measuring retro-reflective materials, it has been found that measurement geometry has a significant influence on the reflected spectral radiance factors. For fluorescent retro-reflective materials, the reflected component is more sensitive than the fluorescent component to geometric restraints. Therefore, geometric specifications and tolerances are critical in the measurement of these materials.

5.4.1 Description of the Surface

High slope angles of the face of the pyramid may result in large data dropout due to the slope measurement limitation of the measurement tool. The data set shown in Figure 42 is measured using index liquid technique. In this technique an index liquid is placed on top of a microlens array which reduces refraction at lens boundary. The advantage of the technique is increased dynamic range that helps in measuring steep structures [73]. The sample was made with high precision molding. The material used is acrylic with an index of refraction of 1.47. The nominal facet angles are 60º. Image size is 736 x 576 and pixel size is 1.15µm. Slant edges are not very pronounced in this image.

It is recommended to use a higher magnification to measure these surfaces to improve the sharpness at the edges.

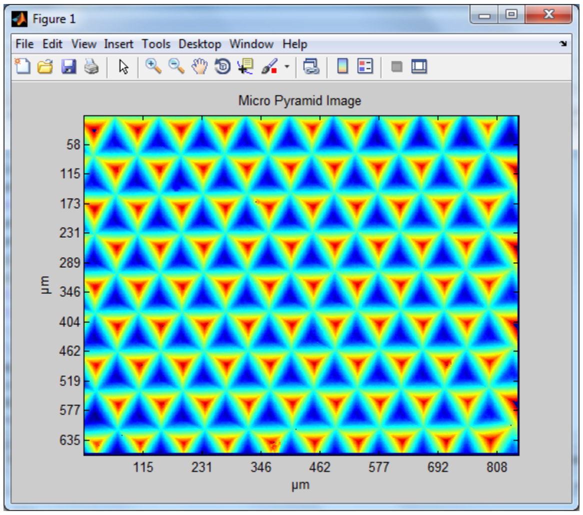


Figure 42: Micro-pyramid surface

Figure 43 shows a square based pyramid generated in MATLAB. Image size is 256x256 pixels. The image consists of 25 pyramids. Each pyramid has a square base and 4 triangular facets.

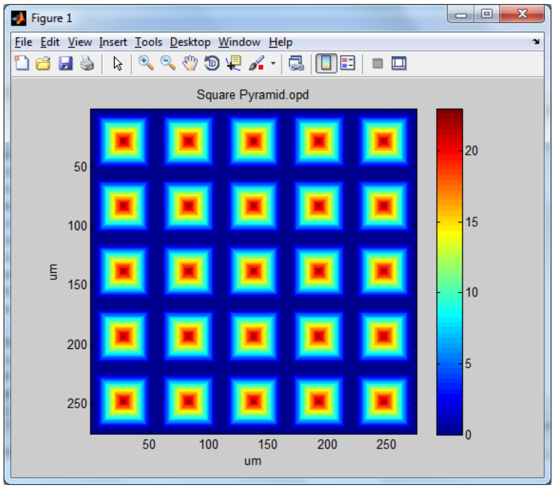


Figure 43: Square based Pyramid

5.4.2 Metrological Characteristics

Pyramids can be of different geometries but in this example we are using square pyramids. The base is a square, and all triangular faces could be equilateral or isosceles triangles. The main geometrical characteristics of a square pyramid are the inclination angle and face angle as shown in Figure 44. Inclination angle also known as dihedral angle is the angle between one of the triangular faces and the base.

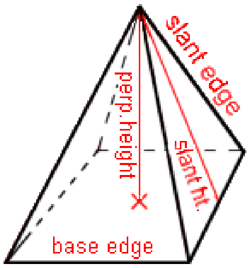


Figure 44: Square Pyramid

Face angle is the angle between two triangular faces of the pyramid. Facet angle affects the reflection efficiency, contrast ratio and enhances brightness. Other geometrical characteristics of importance are slant edge and base edge length, perpendicular and slant heights etc.

5.4.3. Approach

Approach for these types of surfaces requires separating the micro-pyramids from substrate by masking. Term mask is used to subtract tilt from the substrate. Then each individual facet is separated in a micro-pyramid. Normally when surface data is masked, it is done based on height. The proposed concept is to convert the data into another domain, find one or more parameters in the transformed domain, perform masking using some threshold and then map the data back to height domain.

5.4.3 Methodology

In this example segmentation needs to separate all the faces of the pyramid to calculate face-base and face-face angles. To achieve this normals to the surface at each x,y,z point are calculated. The normals calculated are normalized to length 1. The largest magnitude of the normal component is of the flat substrate surface between the pyramids. The magnitude of the normal component of the edges of the triangular faces is lower than the substrate.

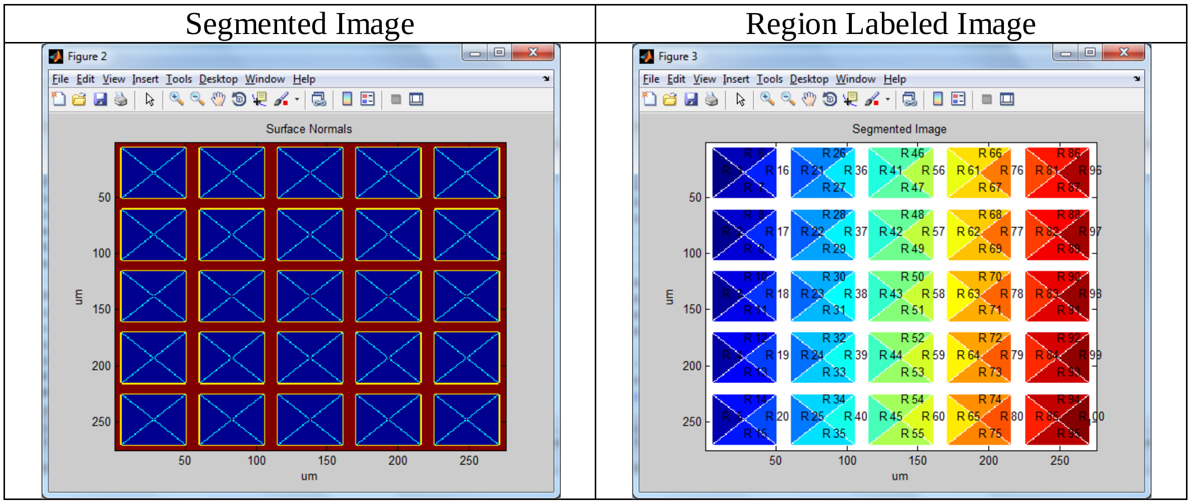


Figure 45: Segmentation

Segmentation is followed by edge detection shown in Figure 46.

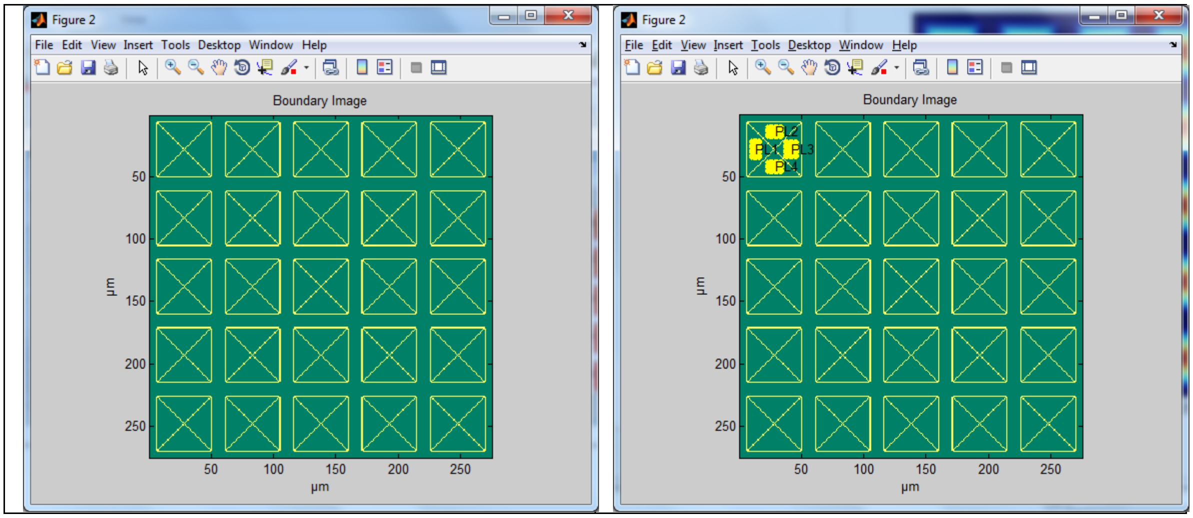


Figure 46: Edge Detection

5.4.5 Characterization

A threshold is used on the normalized image to separate the faces from the substrate data. Segmented image with separated faces with labeled regions are shown in

Figure 45. As obvious each of the four triangular faces of each pyramid is detected. After segmentation x,y,z coordinates of all the data points for each region is used to fit a least square plane. The components of the normal vectors thus obtained for each surface are used to calculate face-base and face-face dihedral angles.

If n1 and n2 are the normal vectors of the plane fitted to two surfaces, then the angle between the two planes is given by the equation

$$\cos\theta = \left| \frac{\langle \boldsymbol{n}_1, \boldsymbol{n}_2 \rangle}{\|\boldsymbol{n}_1\| \, \|\boldsymbol{n}_2\|} \right|,$$

Figure 45 shows planes created on the four faces of square pyramid. The angles between the faces are shown in Table 12.

Table 12: Facet Angles

Face1-Face 2	60^{o}
Face 2-Face 3	60^{o}
Face 3-Face 4	60^{o}
Face 4-Face 1	60^{o}

5.4.6. Observations

Two different techniques are suggested to separate each facet of the pyramid. The first technique is using surface normals and the second technique is masking in a domain other than height.

5.5. Use case 5: MEMS

The fabrication of integrated circuits and MST/MEMS devices is now a mature technology in many senses; however the challenge of the metrology for such devices remains a major technological hurdle. Both optical interferometric and scanning probe instruments have been employed to measure the surfaces and their step heights with varying degrees of success. The difficulty of this metrology lies around the ability to separate and quantify zones of differing planar height and then analyzing the topographies of the zones individually [71]. Complicated surfaces pose particular difficulties in terms of establishing measurement datums and leveling data. These problems are further accentuated by the limited size of data sets available from the current metrology tools.

5.5.1. Description of the surface

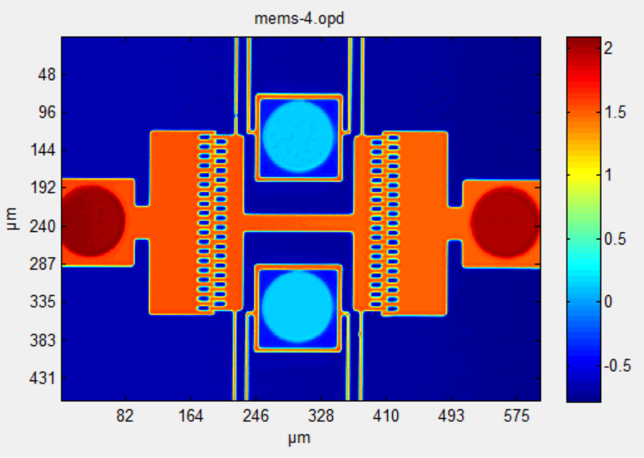


Figure 47: MEMS Device

Figure 47 shows a case of relatively complicated MEMS surface. The size of the image is 736x480 pixels and sampling is 820.96nm. The surface has features at four different planar heights. The task is to find the planar height of features from the substrate.

5.5.2. Metrological Characteristics

The desired metrological characteristic in this case is the height of the features at different planar levels from the substrate. The current leveling techniques remove the tilt from the entire dataset which is not effective in many cases as shown in the bearing ratio curve in Figure 48. Even after removing the tilt from the dataset, each planar surface still has some tilt.

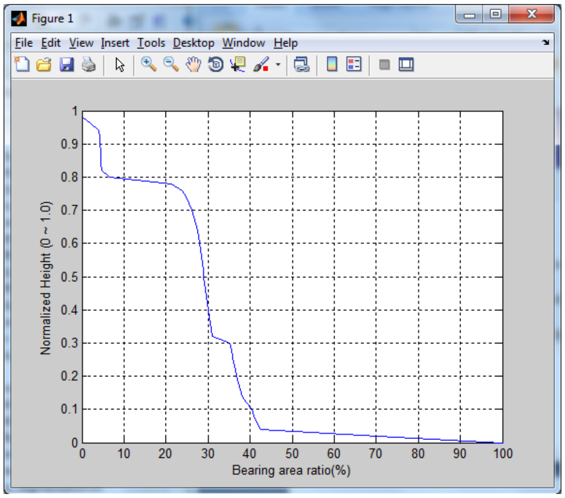


Figure 48: Bearing Ratio Curve

In some worse cases the measurement slope cancel out the height distinction of the different step planes on the bearing ratio curve. Furthermore, the base plane for leveling has been split up into small sub-areas. All of this makes manual and traditional pattern analysis methods difficult and impracticable.

5.5.3. Approach

The approach in this case will be to separate the zones at different planar heights and select a reference surface to level. After that the planar distances between features can be calculated with respect to the reference surface. Gradient method, band pass thresholding and clustering techniques will be used to separate 4 major planar levels. The thresholds are determined using the bearing ratio curve.

5.5.4. Methodology

When using the slope based segmentation, a thick edge occurs for those edges with a wide and sloping height change. As a consequence it is then difficult to determine the actual edge boundary from the thick edge. Clustering methods uses histogram of the image height and segment the image according to the height distribution. Gradually changing slopes can cause inconsistent segmentation with the clustering technique. Band-pass thresholding allows selecting a lower and a higher threshold. All the cases are shown in Figures 49, 50 and 51 respectively.

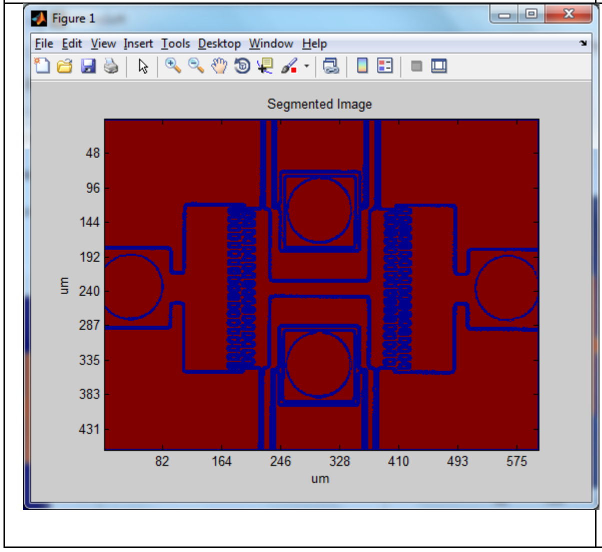
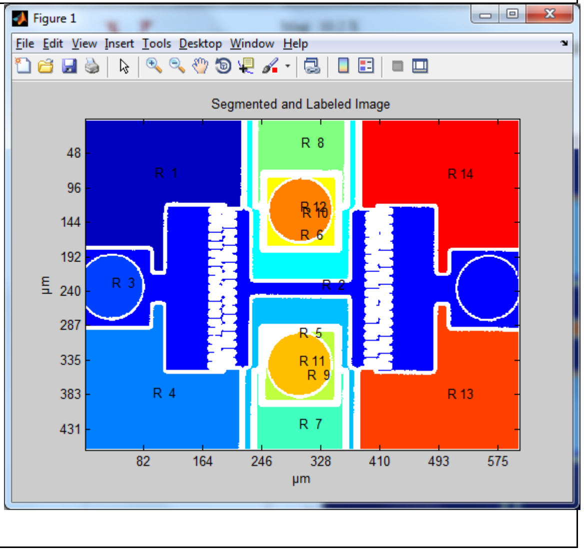
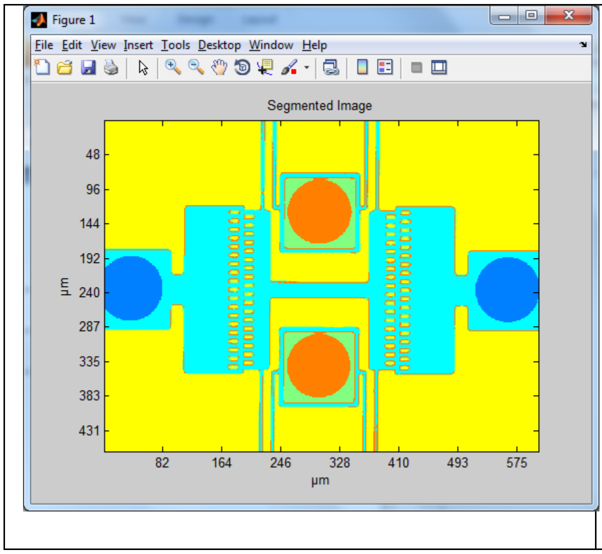
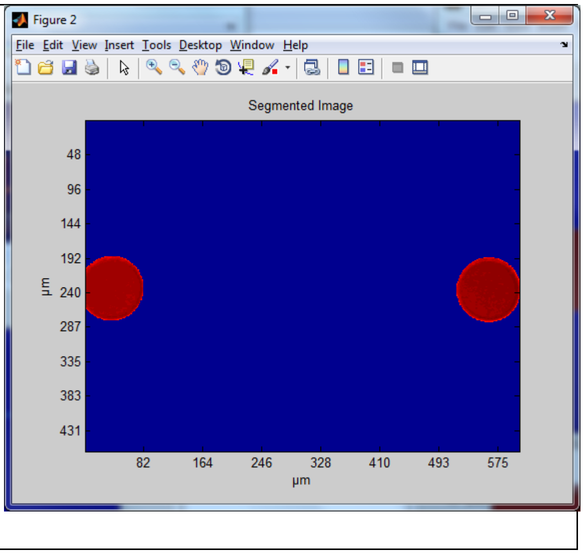
Slope based segmentation	Region Labeled Image
 Figure 1 File Edit View Insert Tools Desktop Window Help Segmented Image 48 96 144 192 240 287 335 383 431 um 82 164 246 328 410 493 575 um	 Figure 1 File Edit View Insert Tools Desktop Window Help Segmented and Labeled Image R 8 R 1 R 14 R 12 R 10 R 6 R 3 R 2 R 5 R 11 R 9 R 4 R 13 R 7 48 96 144 192 240 287 335 383 431 μm 82 164 246 328 410 493 575 μm

Figure 49: Gradient based Segmentation Results

 Figure 1 File Edit View Insert Tools Desktop Window Help Segmented Image 48 96 144 192 240 287 335 383 431 μm 82 164 246 328 410 493 575 μm	 Figure 2 File Edit View Insert Tools Desktop Window Help Segmented Image 48 96 144 192 240 287 335 383 431 μm 82 164 246 328 410 493 575 μm

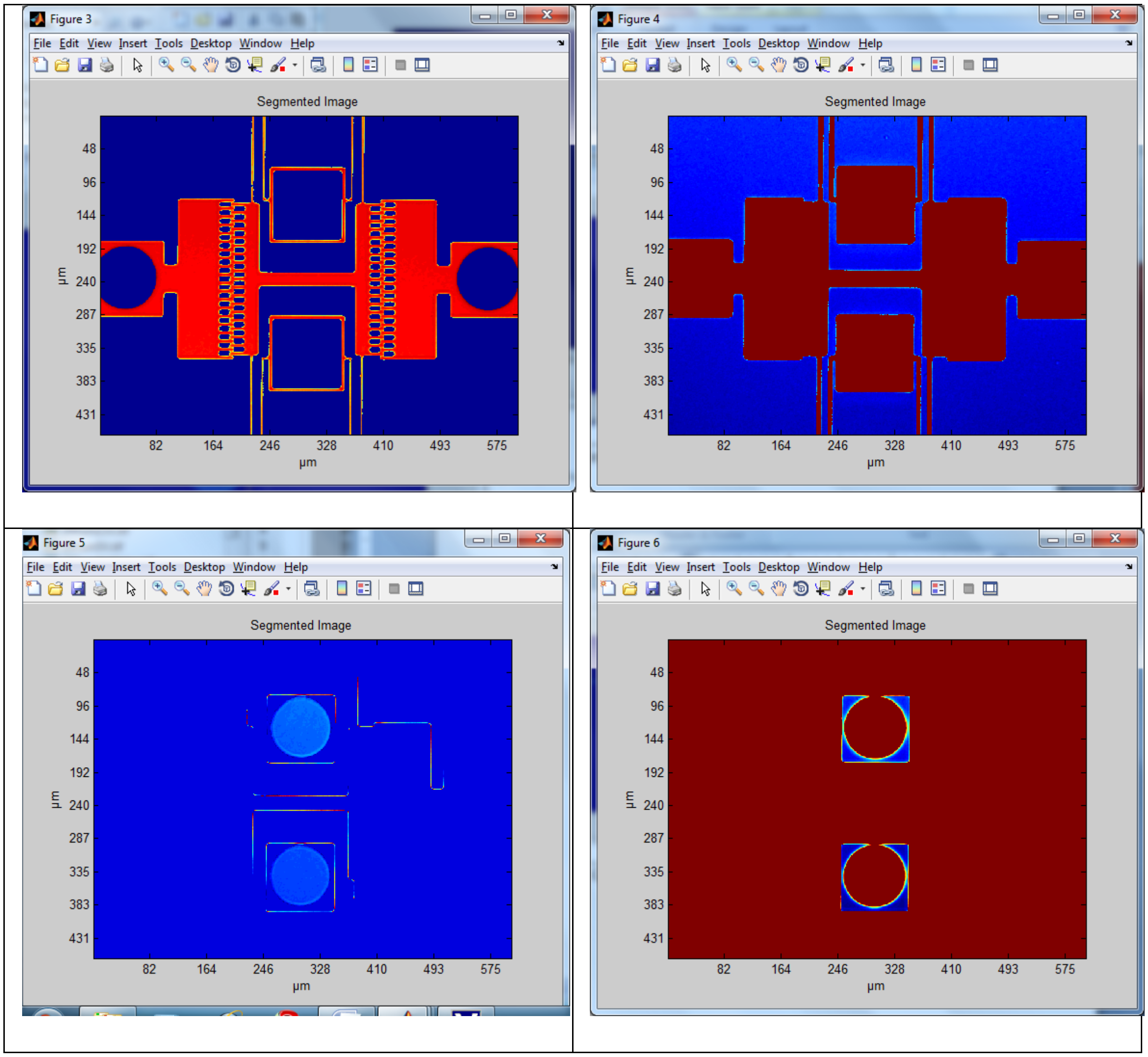
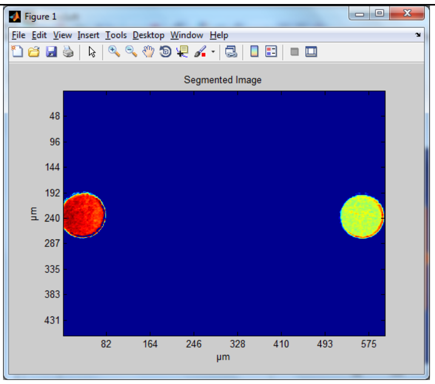
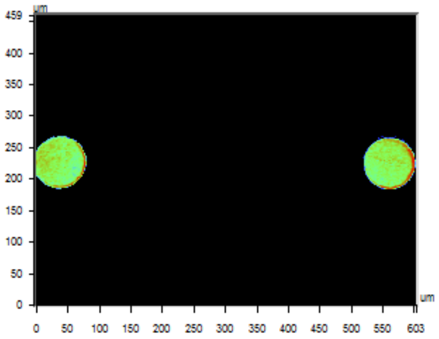


Figure 50: Cluster based Segmentation Results





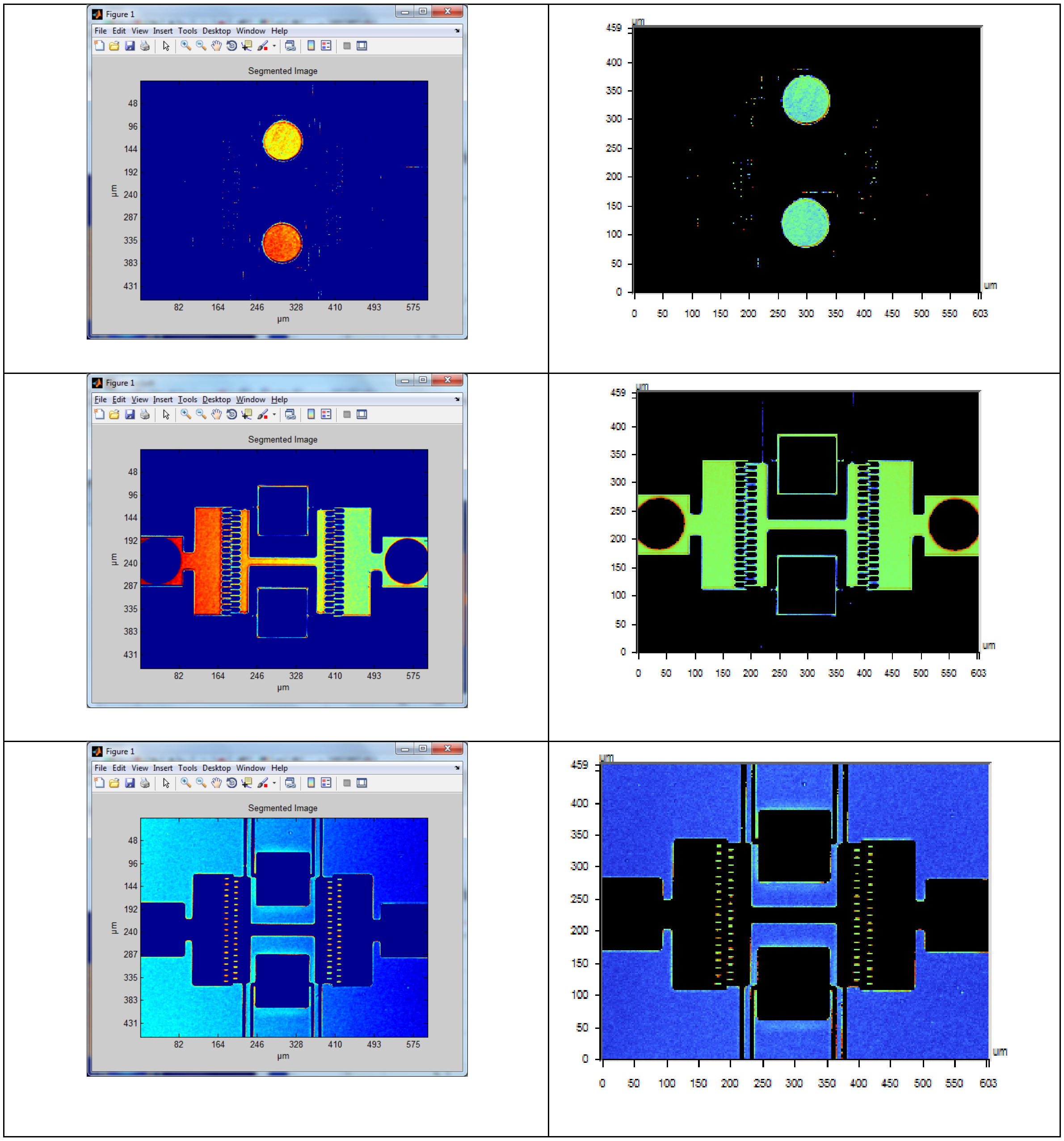


Figure 51: Planar surfaces with tilt removed

5.5.5. Characterization

Table 13: Results

Step Plane	Vision Software	MATLAB Toolbox
Level 3	2.7011µm	2.7279 µm
Level 2	2.1934 µm	2.2265 µm
Level 1	0.8482 µm	0.81255 µm

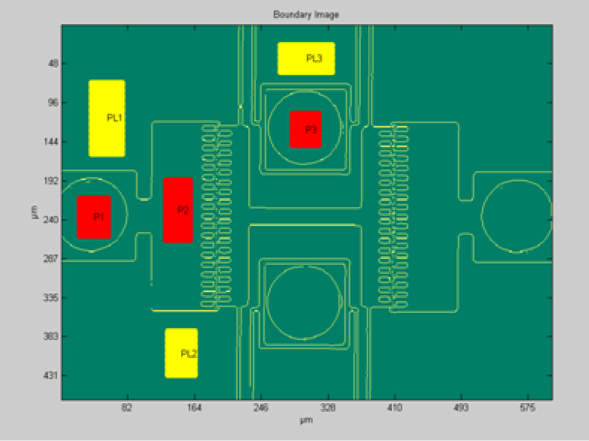


Figure 52: Planar surfaces with tilt removed

5.5.6. Observations

Identification and separation of step planes of the surface is the key for characterizing MST/MEMS surfaces. The procedure was demonstrated using gradient method, clustering and band pass thresholding algorithms. Tilt was removed by subtracting least square plane from each step plane to measure the planar height without the influence of the slope. Edge detection is performed after segmentation and tilt removal to calculate feature dimensions.

Figure 53 is another example in the same class. It has square mirrors at 4 different planar levels. A threshold based on gradient is used to separate the mirrors. Each mirror is treated like a region and is labeled. Edge detection is done to measure the dimensions of the mirror. Point to plane distance is used to calculate the vertical distance from the reference surface. Figure 53 shows the process.

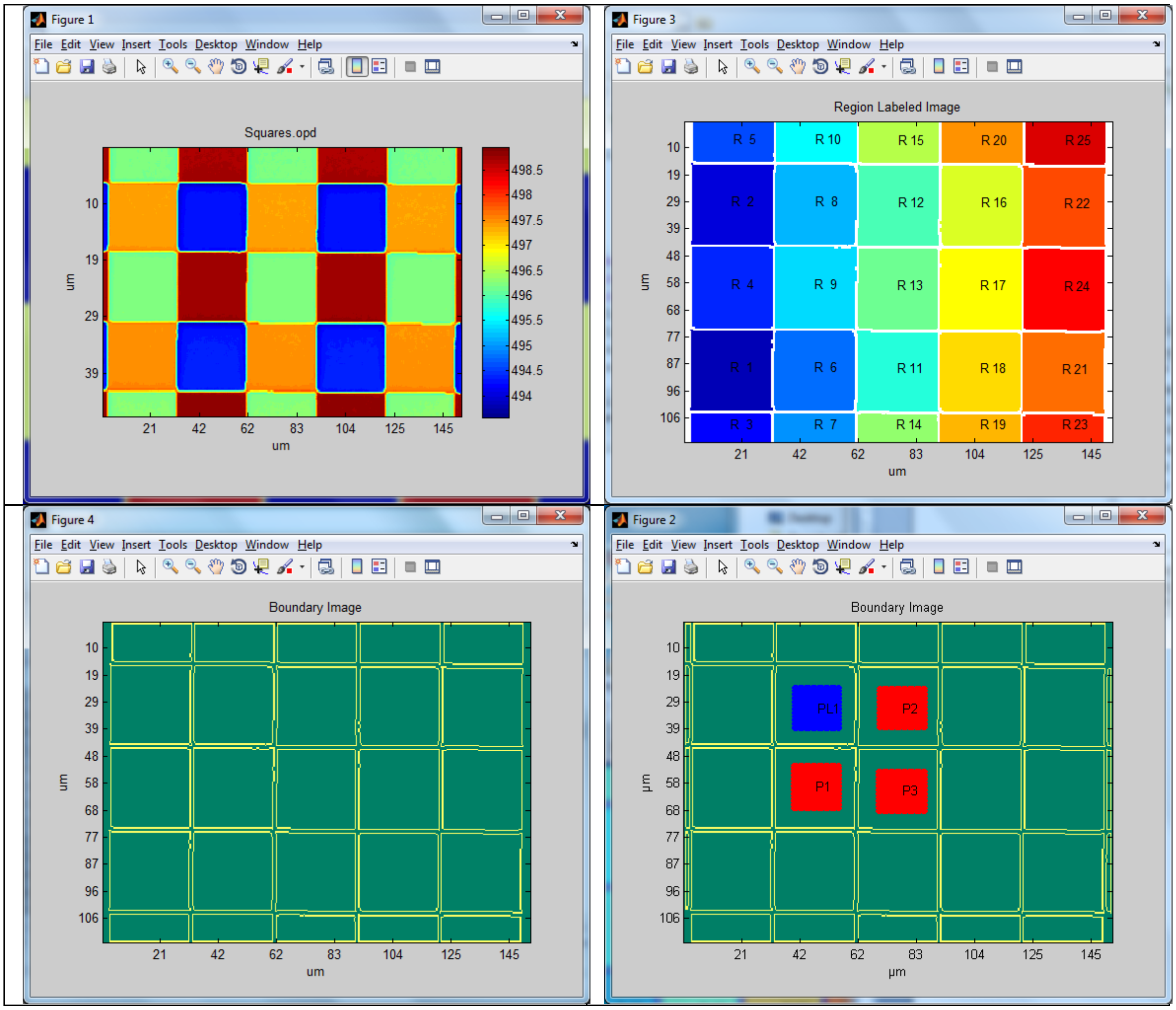


Figure 53: Data Processing

Table 14: Planar Height Results

	MATLAB (µm)	Veeco (µm)
PL1-P1	4.46	4.49
PL1-P2	3.08	3.17
PL1-P3	1.93	1.94

The other example is of a micro-fluidic channel where the width and depth of the channel are the important metrological characteristics. Figure 54 shows the methodology which consists of segmentation and edge detection. Any tilt in the dataset should be removed before performing segmentation. The depth of the channel is calculated at various locations inside the channel.

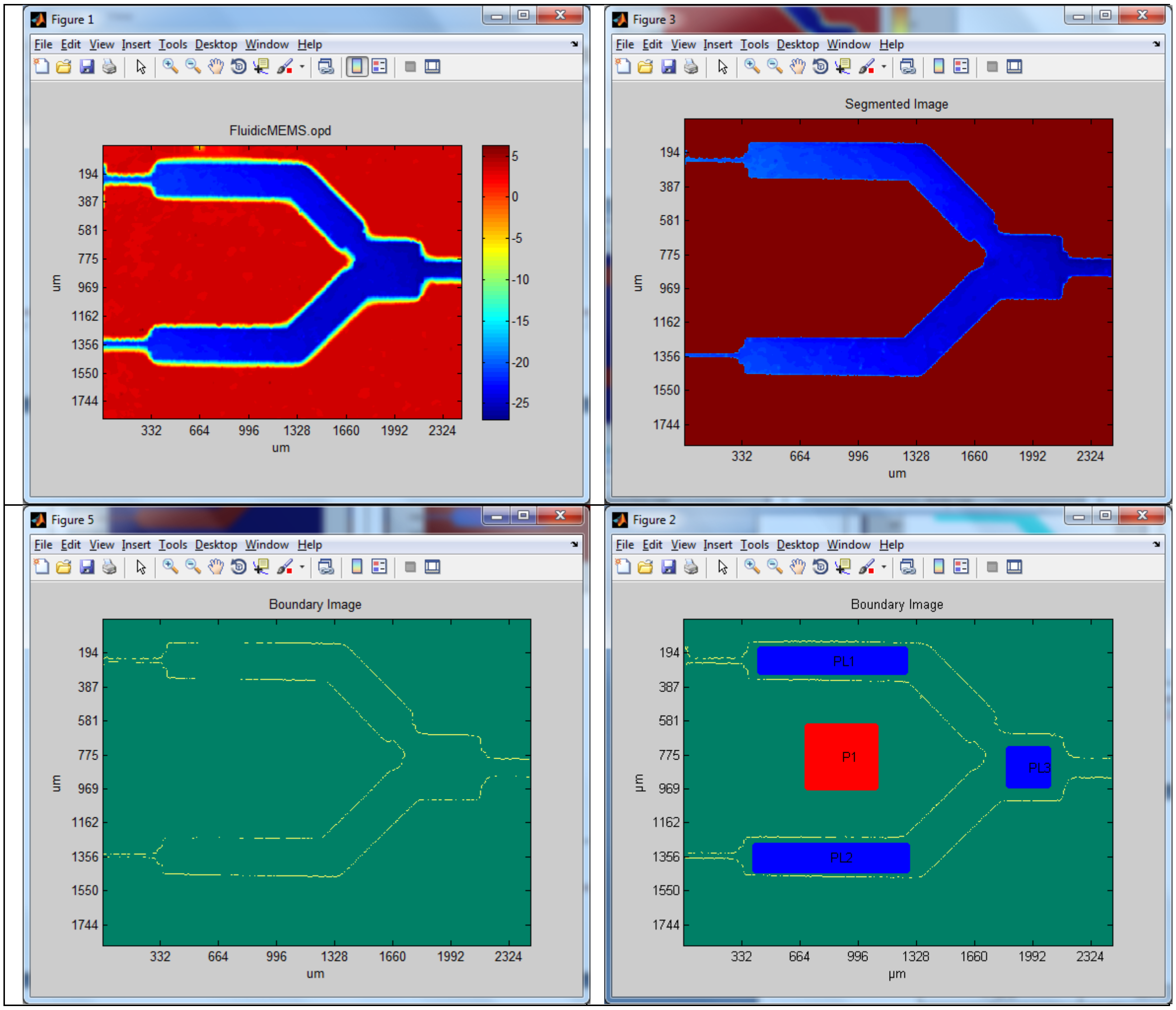


Figure 54: Data Processing

TABLE 15: Channel Depth Results

Distance	MATLAB (µm)	VEECO (µm)
P1-PL1	23.55	25.74
P1-PL2	28.1	26.62
P1-PL3	25.99	28.31

5.6. Use Case 6: Disk drive suspension assembly

The suspension assembly is the mechanism that positions the read/write head above the rapidly spinning disks to maintain a consistent "flying height". The suspension must hold the head at a precise microscopic distance above the disk without allowing unintended movement of the head in any direction. It maintains the position of the head within a safe range during these periods to keep it from hitting the disk's surface.

5.6.1 Description of the Surface

Image size is 3519x916 pixels. Sampling is 1.09µm.

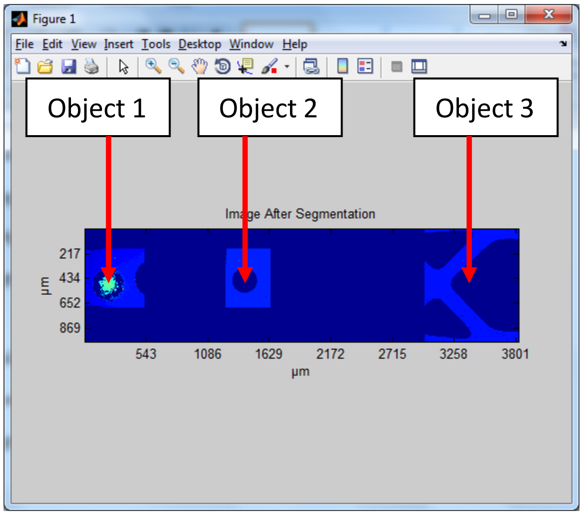


Figure 55: Disk Drive Assembly

5.6.2 Metrological Characteristics

It is required to calculate the misalignment between the three objects 1, 2 and 3 shown in figure 55. Object 1 is a sphere, object 2 is a hole and object 3 is a slot in the assembly.

5.6.3 Approach

A sphere is fitted on object 1 and the coordinates of the center of the sphere are determined. A circle is fitted on object 2 to find out the center of the hole. To find the center of the slot, average points P1 and P2 are created. Then mid-point is determined labeled as P3. The misalignment in Y direction of the sphere and hole can be calculated relative to the center of the slot.

5.6.4 Methodology

Figure 56 shows the segmentation, region labeling and geometric fitting operations.

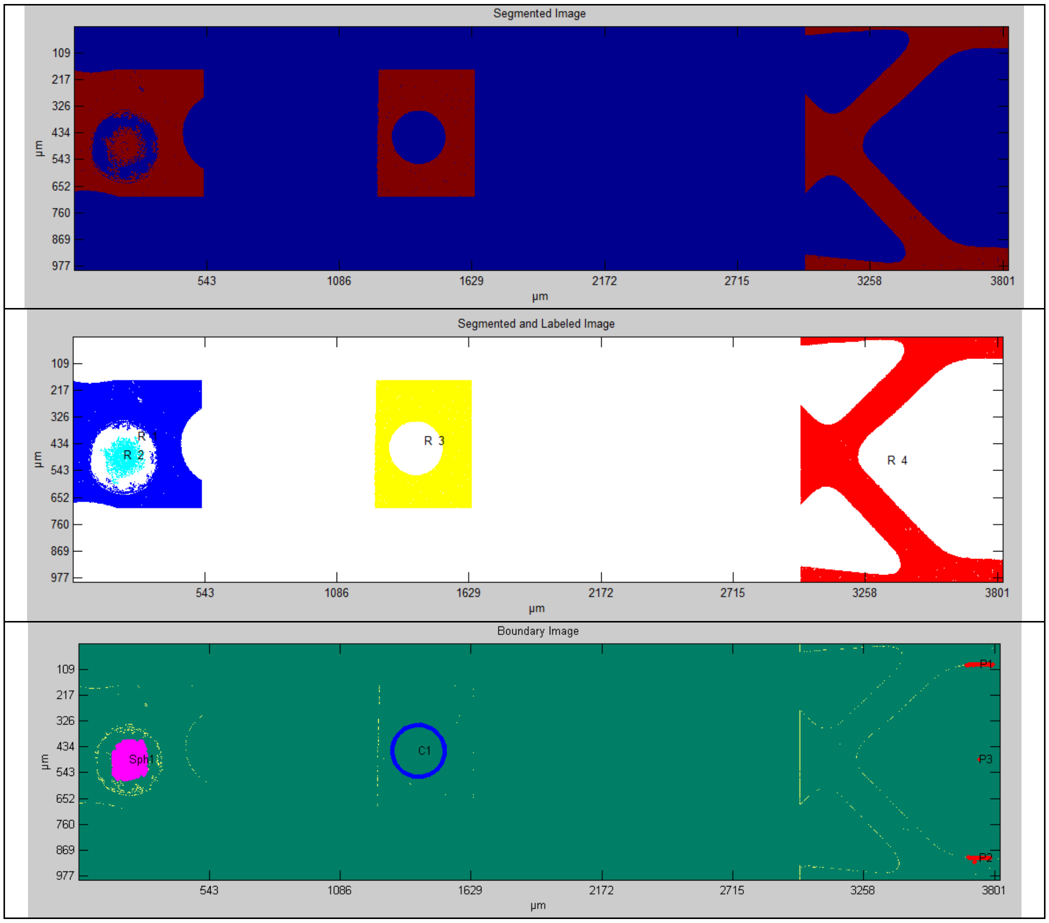


Figure 56: Disk Drive Suspension Assembly

5.6.5 Characterization

The X and Y coordinates of the center of sphere, circle and slot are show in Table 16.

Table 16: Results

Feature	X (µm)	Y (µm)
Center of Mid Line	3751.95	610.4
Center of Circle	1579.41	562.44
Center of Sphere	376.05	598.41

5.6.6 Observations

Fitting algorithms for line, circle and sphere were used to calculate the center. A mid-line was drawn from two parallel lines. The data shows misalignment in Y between the three objects. If the misalignment specification is known then pass fail decision can be made about the part.

CHAPTER 6: CONCLUSIONS

6.1. Summary

The rapid growth of engineered surface in global market is driving the need for advancement in the field of measurement and characterization. Surface topography is playing a very important role in functional performance of products in various industries. Physical, chemical, electrical and optical properties of a product can be controlled by precisely creating microstructures and texture on the surface. Engineered surfaces are surfaces where the manufacturing process is optimized to generate variation in geometry and /or near surface material properties to give a specific function. It is a novel and emerging field of functional surfaces which is finding applications in optics, data storage, pharmaceutical, semiconductor, MEMS and several other industries. In the manufacturing of structured surfaces, measurement and characterization is essential. But because of their special topography, high aspect ratios, high slopes, 3D nature and very fine topographic scale but with very large scaled covering area, barriers are encountered in measurement process using conventional instruments. Besides the conventional surface texture characterization and development of 3D surface texture parameters no work has been done to advance the characterization methods. New measuring and characterization methods are in need. Generic measurement tasks to be performed on engineered surfaces are distance, width, height, geometry, texture and roughness, layer thickness, aspect ratio, slope, angle, edge radius, defect area etc. Characterization methods of structured surfaces and MEMS differ from the conventional machined surfaces and the available techniques are not suitable to determine the shape, size and orientation of micro-features.

This dissertation addressed the development of fast and intelligent surface scanning algorithms and methodologies for engineered surfaces to determine form, size and orientation of significant surface features. Object recognition techniques are used to identify the surface features and CMM type fitting algorithms are applied to calculate the dimensions of the features. Recipes can be created to automate the characterization and process multiple features simultaneously. The developed methodologies are integrated into a surface analysis toolbox developed in MATLAB environment. The deployment of the developed application on the web is demonstrated.

In this dissertation a brief introduction of engineered surfaces is given with their application in real world. A surface classification system is proposed based on the tools required for characterization. A survey of engineered surface manufacturing methods and measurement techniques is presented. Current surface characterization techniques are discussed stressing upon their limitations followed by the recent developments in advanced characterization.

Advanced characterization methodology is a sequence of steps that converts raw data into metrological characteristics of form, size and orientation. A combination of image processing and pattern recognition techniques for object recognition with CMM type fitting algorithms is used to calculate dimensional characteristics of the features. The user is provided with multiple choices at each step in the methodology to carry out the desired analysis. Recipe can be created selecting appropriate techniques for each kind of surface and can be applied to batch of similar datasets.

The algorithms developed are integrated into a surface characterization system implemented in MATLAB programming environment. Software architecture is developed using data flow diagrams. The application created in MATLAB is converted into .NET component to deploy the application on the web. The web version is created in Microsoft Visual Studio programming environment using C# and ASP.NET. A framework is ready to deploy the entire application on the web for access to the characterization toolbox by metrology community all over the world. This will help sharing of data and information and also enable continuous improvement of the techniques and capabilities of the toolbox.

The developed methodology was tested on several real surfaces, measured on White Light Interferometer. A general recommendation is given on the suitability of a technique for a given surface. Numerous parameters derived utilizing the full 3D data were demonstrated emphasizing on the need to use surface data instead of profile data for determining metrological characteristics.

Validation was performed in two ways. Dimensional parameters of industrial surfaces were calculated using the existing data analysis software and the developed toolbox. The results were reasonably similar with slight difference owing to the difference in the way calculation was performed. For second validation, a calibration artifact was measured on White light interferometer. Then the characterization toolbox was used to calculate the dimensional parameters. The results obtained were compared with the certified values and the difference was within the noise of the measurement. A surface was measured at two different magnifications and orientations to demonstrate the uncertainty in the measurement.

6.2. Contributions

The primary focus of the work has been the development of advanced techniques for characterization of engineered surfaces. Image processing and pattern recognition algorithms were refined and integrated with CMM type geometric fitting algorithms to solve metrological issues. Methodologies were created to characterize surfaces with simple and complex geometries. User can create recipes by choosing suitable options for the surface to automate the characterization thus decreasing the time required to obtain the form, size and orientation characteristics of the surface features. Multiple features can be processed simultaneously when the surface is patterned. A MATLAB executable was created that can be used as a standalone application or can be integrated with any data analysis software. The deployment of the developed application on web was demonstrated on local IIS server. Full deployment on the web will serve the purpose of software testing platform for exchange of information.

6.3 Future Research

Future refinements and enhancements of the methodologies are necessary to extend its application further in engineering and manufacturing practices. One of the tasks for research is to expand the concept of segmentation of multi facet geometric surface in domains like Fourier, Hough and Radon Transform. Another area for research is improving intelligence by developing algorithms to segment the boundary of the feature automatically into line and curve segments. Increased availability of real samples and their measured results could enhance the level of intelligence. For the continuous improvement of the characterization it is required to create an open source platform on the web to make it available to a larger metrology community.

REFERENCES

[1] Evans C and Bryan J, 1999, “Structured”, “Textured” or “Engineered” Surfaces, CIRP Annals - Manufacturing Technology, v48, issue 2, 541-556

[2] Malburg MC and Raja J, 1993, Characterization of Surface Texture generated by Plateau Honing Process, Annals of the CIRP, 42(1)

[3] Tabeling P, 2005, Introduction to Microfluidics, Oxford University Press

[4] Hsu T, 2008, MEMS and Microsystems, John Wiley & Sons, New Jersey

[5] Brinksmeier E, Riemer O and Stern R, 2001, Machining of precision parts and microstructures, Proceedings 10th International Conference on Precision Engineering (ICPE), Japan

[6] Masuzawa T, 2000, State of the art of micromachining, Annals of CIRP 49:473–88

[7] Madou MJ, 2001, Fundamentals of Microfabrication, CRC Press, ISBN 0-8493-0826-7

[8] ISO 4287, 1997, Geometrical Prosuct Specification (GPS) – Surface texture: Profile method – Terms, definitions and surface texture parameters

[9] Stout K. et al., 1993, The development of methods for the characterization of roughness in three dimensions, European Report EUR 15178N, ISBN 0704413132

[10] Bryan JB, 1966, Survey of functional characteristics of surfaces, CIRP STC S report

[11] DeVries W, Field M and Kahles J, 1976, Relationship of Surface roughness to functional properties, CIRP Annals, 25/2, 569-73.

[12] http://www.kyodo-inc.co.jp/electronics/nil/img/mold02.gif

[13] http://www.averydennison.com/avy/en_us/Innovation/Technology-Platforms/Microreplication

[14] Ehrfeld W, Hagmann P, Mander A and Münchmeyer D, 1986, Fabrication of microstructures with high aspect ratio and great structural heights by synchrotron radiation lithography, galvanoforming and plastic moulding (LIGA-process), Microelectronic Engineering, vol. 4, 35-56

[15] www.veeco.com

[16] Nichols J, 2004, Metrology of High Aspect Ratio MEMS, PhD Dissertation, Georgia Tech, http://hdl.handle.net/1853/5187

[17] Davies MA, Evans CJ, Patterson SR, Vohra R and Bergner BC, 2003, Application of precision diamond machining to the manufacture of micro-photonics components, Proc. of SPIE Vol. 5183, Lithographic and Micromachining Techniques for Optical Component Fabrication, SPIE, Bellingham, WA

[18] http://www.micromanufacturing.net/didactico/Desarollo/microtechnologies/1-7-micromanufacturing-technology-classification

[19] Olszak AG, Schmit J and Heaton MG, 2001, Interferometry: Technology and Applications, Veeco Instruments, Inc., 2650 E. Elvira Road, Tucson, AZ 85706

[20] Wyant JC and Schmit J, 1997, Large Field of View, High Spatial Resolution, Surface Measurements, In Proc. of the 7th Int. Conference On Metrology and Properties of Engineering Surfaces, 294-30, Chalmers University of Technology, Göteborg, Sweden

[21] Robinson AE, 2007, Advanced Confocal Microscopy: A New Approach to a Mature Technology Brings 3D Imaging to a New Level Microscopy and Microanalysis, 13: 1552-1553 Cambridge University Press

[22] Sheats J, 1998, Microlithography: Science and Technology, Marcel Dekker, Inc., New York

[23] Postek M, 1994, Scanning electron microscope metrology," Proceedings of SPIE – The International Society for Optical Engineering, vol. CR52, 46-90

[24] www.mitutoyo.com

[25] Mahony C, Hill M, Brunet M, Duane R and Mathewson A, 2003, Characterization of micromechanical structures using white-light interferometry, Meas. Sci. Technology, v14, 1807–1814

[26] Mainsah E, Dong WP, Stout KJ, 1996 Procedings of SPIE – The international society for optical engineering 2599, 141-154

[27] Kovesi P, Shapelets Correlated with Surface Normals Produce Surfaces, http://www.csse.uwa.edu.au/~pk/research/pkpapers/shapeletsICCV.pdf

[28] Gelfand N and Guibas LJ, 2004, Shape Segmentation Using Local Slippage Analysis, Eurographics Symposium on Geometry Processing, Computer Graphics Laboratory, Stanford University

[29] Worthington PL and Hancock ER, 2001, Object Recognition Using Shape-from-Shading, Philip L. and Edwin R. IEEE Transactions on pattern analysis and machine intelligence, v23, issue 5

[30] Blunt L, Jiang X, Scott P and Xiao S, 2004, Surface Metrology of MST Devices, Proce. Int. Conf. Colloquium on Surfaces, Chemnitz Germany
[31] Bosseboeuf A and Petigrand S, 2003, Application of Microscopic Interferometry Techniques in the MEMS Field, Proc. SPIE 5145, 1-16

[32] Novak E, Krell MB and Browne T, 2003, Template-based software for accurate MEMS characterization, Proc. SPIE 4980, 75-80

[33] Grigg D, Felkel E, Roth J, Colonna de Lega X, Deck L and de Groot P, 2004, Static and dynamic characterization of MEMS and MOEMS devices using optical interference microscopy,SPIE 5455-55, Photonics Europe

[34] Gonzalez R and Woods R, 2002, Digital Image Processing (New York: Prentice Hall)

[35] Purcell D, Suratkar A, Davies A, Farahi F, Ottevaere H, and Thienpont H, 2010, Interferometric technique for faceted microstructure metrology using an index matching liquid, Appl. Opt.49, 732-738

[36] Duda RO, Hart P and Stork DG, 2001, Pattern Classification (2nd ed.), John Wiley and Sons

[37] http://mathworld.wolfram.com/LeastSquaresFitting.html

[38] Sugimoto K and Tomita F, 1994, Boundary segmentation by detection of corner, inflection and transition points, IEEE Workshop on Visualization and Machine Vision, Proceedings, 13-17

[39] Gonzalez R and Woods R, 2002, Digital Image Processing (New York: Prentice Hall)

[40] Plataniotis KN, 2007, Color Image Processing: Methods and Applications, CRC Press

[41] Glasbey CA, 1993, An analysis of histogram-based thresholding algorithms, Computer Visual Graph Image Processing, v55, 532–537

[42] Otsu N, 1979, A threshold selection method from gray-level histograms, IEEE Trans. Syst., Man, Cybern., vol. SMC-9, no. 1, 62–66

[43] Pal NR and Pal SK, 1993, A review on image segmentation techniques, Pattern Recognition, v 26, no. 9, 1277–1294

[44] Sahoo PK, Soltani S, Wong AKC, and Chen YC, 1988, A survey of thresholding techniques, Comput. Vis. Graph. Image Processing, v 41, 233–260

[45] Ridler TW and Calvard S, 1978, Picture thresholding using an iterative selection method, IEEE Trans. Syst. Man and Cybernetics, v8, 630–632

[46] Brink AD, 1995, Minimum spatial entropy threshold selection, IEE Proc. Vis. Image Signal Process, v142, no. 3, 128–132

[47] Kapur JN, Sahoo PK and Wong AKC, 1985, A new method for graylevel picture thresholding using the entropy of the histogram, Comput. Vision Graphics Image Process, v 29, 273–285

[48] Li X, Zhao Z and Cheng HD, 1995, Fuzzy entropy threshold approach to breast cancer detection, Inform. Sc., v 4, 49–56

[49] Cheng HD and Lui YM, 1997, Automatic bandwidth selection of fuzzy membership function, Inform. Sci., no.103, 1–21

[50] Pal SK, King RA, and Hashim AA, 1983, Automatic gray level thresholding through index of fuzziness and entropy, Pattern Recognition Letters, no.1, 41–146

[51] Huang LK and Wang MJJ, 1995, Image thresholding by minimizing the measures of fuzziness, Pattern Recognition, v28, no.1, pp. 41–51

[52] Tobias OJ and Seara R,2002, Image Segmentation by Histogram Thresholding Using Fuzzy Sets, IEEE Transaction on Image Processing, v11, no.12

[53] Wu K, Otoo E and Suzuki K, 2009, Optimizing Two-Pass Connected-Component Labeling Algorithms, Pattern Analysis & Applications, v12, issue 2, 117-135

[54] Sonka-Hlavac-Boyle*:* Image Processing, Analysis and Machine Vision, CL-Engineering; 2 edition (September 30, 1998)

[55] Harasaki A and Wyant JC, 2000, Fringe modulation skewing effect in white-light vertical scanning interferometry, APPLIED OPTICS, v 39, no.13

[56] Monteverde R., Nyssonen D, 1988, A new line width standard for reflected line inspection, SPIE Symposium on Microlithography

[57] Otsu N, 1979, A Threshold Selection Method from Gray-Level Histograms, IEEE Transactions on Systems, Man, and Cybernetics, v9, no.1, 62-66.

[58] webdocs.cs.ualberta.ca/~nray1/CMPUT466_551/Clustering.ppt

[59] MATLAB Compiler 4, User’s Guide, www.mathworks.com

[60] MATLAB Application Deployment, Web Example Guide, www.mathworks.com

[61] MATLAB 7, Creating Graphical User Interfaces, www.mathworks.com

[62] PhD Dissertation, “A Framework for Internet Based Surface Texture Information and Analysis System”, Son H. Bui, UNC Charlotte, 2002

[63] Beginning ASP.NET 1.1 with Visual C# .NET 2003, ISBN:0-7645-5708-4, Wiley Publishing Inc., Indianapolis, IN

[64]Microsoft Visual Studio 2008 Professional Edition Users Guide, Microsoft Corporation

[65] Norbert Greif, Heike Schrepf, Dieter Richter, Software validation in metrology: A case study for a GUM-supporting software, Measurement 39 (2006) 849–855

[66] www.zygo.com

[67] www.optics.arizona.edu

[68] Optical Profiler Application Notes. www.veeco.com

[69] http://www.reflectives.averydennison.com/prismatic.asp

[70] David M. Burns, Norbert L. Johnson, Metrology of fluorescent retroreflective materials and its relationship to their daytime visibility, Analytica Chimica Acta, Volume 380, Issues 2-3, 2 February 1999, Pages 211-226

[71] L. Blunt, P Scott Surface Metrology of MST devices

[72] A new line width standard for reflected light inspection, Robert Monteverde, Diana Nyyssonen, SPIE Symposium on Microlithography 1988

[73] D.Purcell, A. Suratkar, A. Davies, F. Farahi, H. Ottevaere and H. Thienpont, “Interferometric technique for faceted microstructure metrology using an index matching liquid,” Appl. Opt. 49 (4), 732-738 2010.

[74] Forbes A, 1989, Least-Squares Best-Fit Geometric Elements, Division of Information Technology and Computing, NPL Report DITC 140/89

APPENDIX A: JOURNAL & CONFERENCE PAPERS

[1] Characterization of Engineered Surfaces
Ritwik Verma and Jay Raja, Journal of Physics: Conference Series 13 (2005) 5–8

[2] Measurement of Micro-Features
Ritwik Verma and Jay Raja, Society of Manufacturing Engineers, Micro Manufacturing Conference, Framingham MA USA, 2008

[3] Surface Defects Analysis Using Pattern Recognition Techniques
Verma, R.; Raja, J. (University of North Carolina at Charlotte), ASPE Proceedings, October 9-14, 2005, Norfolk, VA

APPENDIX B: LIST OF MATLAB FUNCTIONS

FILE OPEN
OPD File: *readOpd.m*
DAT File: *readDat.m*
Image File: *readIm.m*
XYZ File: *readXYZ.m*
Save Dataset: *saveData.m*
Save Statistics: *saveStats.m*
Print: *print.m*
Exit: *quit.m*

PRE PROCESSING
Set to minimum: *set2Min.m*
Set to zero: *set2Zero.m*
Set to mean: *set2Mean.m*
Set to median: *set2Median.m*
Set to maximum: *set2Max.m*
Liner Interpolation: *linInterp.m*
Plane: *fitPlane.m*
Averaging: *filterAveraging.m*
Low Pass: *filterLowpass.m*
High Pass: *filterHighpass.m*
High Pass: *fourierHighpass.m*
Low Pass: *fourierLowpass.m*
Band Pass: *fourierBandpass.m*
Image Complement: *imComplement.m*
Extract Profile: *imProfile.m*

SEGMENTATION
Thresholding: *imThresholding.m*
Slope Based: *imGradient.m*
Clustering: *imClustering.m*
Surface Normals: *imSurfnorm.m*
Band Pass: *imBandpass.m*

MORPHOLOGY
Dilation: *imDilate.m*
Erosion: *imErode.m*
Image Fill: *imFill.m*
Image Closing: *imClose.m*
Image Opening: *imOpen.m*

REGION FINDING
Island Detection: *imLabel.m*

Pick Region: *imSelect.m*

EDGE DETECTION
Boundaries: *imBoundaries.m*
Surface Normals: *edgeSurfnorm.m*
Sobel: *edgeSobel.m*
Prewitt: *edgePrewitt.m*
Canny: *edgeCanny.m*

MEASURE
Point: *measPoint.m*
Line: *measLine.m*
Plane: *measPlane.m*
Circle: *measCircle.m*
Arc: *measArc.m*
Sphere: *measSphere.m*
Polygon: *measPolygon.m*
Centroid: *measCentroid.m*
Mid-point: *measMidPoint.m*
Midline: *measMidLine.m*
Intersection Point: *measIntersectPoint.m*
ROC: *measROC.m*
Bump Height: *measBumpHeight.m*
Coplanarity: *measCoplanarity.m*
Angle Line-Line: *angleLineLine.m*
Angle Line-Plane: *angleLinePlane.m*
Angle Plane-Plane: *anglePlanePlane.m*
Distance Point-Point: *distPointPoint.m*
Distance Point-Line: *distPointLine.m*
Distance Point-Plane: *distPointPlane.m*
Distance Point-Circle: *distPointCircle.m*
Distance Line-Circle: *distLineCircle.m*
Distance Plane-Circle: *distPlaneCircle.m*
Distance Circle-Circle: *distCircleCircle.m*
Undo: *undo.m*

GEOMETRIC FITTING
Line Fit: *LS_LineFit.m*
Plane Fit: *LS_PlaneFit.m*
Circle Fit: *LS_Circle.m*
Sphere Fit: *LS_Sphere.m*

EDGE SEGMENTATION
Sugimoto-Tomita Method: *boundaryDetection.m*
Fit Line Segment: *fitLineseg.m*
Corner Detection: *cornerDetection.m*

TOLERANCE
Straightness: *straightness.m*
Flatness: *flatness.m*
Circularity: *circularity.m*
Parallelism: *parallelism.m*
Perpendicularity: *perpendicularity.m*
Angularity: *angularity.m*

TOPOGRAPHY
Amplitude Parameters: *Sa.m, Sq.m, Sku.m, Ssk.m, Sz.m*
Hybrid Parameters: *Sdq.m, Ssc.m, Sdr.m, Sds.m*
Spatial Parameters: *Str.m, Std.m, StdMinor.m, Sal.m*
Functional Parameters: *Sm.m, Sc.m, Sv.m, Sbi.m, Sci.m, Svi.m*
Histogram: *histFit.m*
Autocovariance: *autoCovariance.m*
Bearing Ratio: *bearing Ratio.m*
Power Spectral Density: *PSD.m*
3D View: *3dView.m*

PROFILE
XY Profile: *xyProfile.m*
Peak Valley Detection: *pvFinder.m*