In-situ Electronics Measurement Using X-ray Micro-Computed Tomography System and Data Driven Prognostic Health Management

by

Junchao Wei

A Dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
May 6, 2017

Keywords: Digital Volume Correlation, In-situ Deformation Measurement, Volumetric Data, Electronics Packing, Finite Element Method

Approved by

Pradeep Lall, Chair, John and Anne MacFarlane Professor of Mechanical Engineering and Director, NSF Center for Advanced Vehicle Electronics
Jeffrey C. Suhling, Quina Distinguished Professor and Chair of Mechanical Engineering
David Beale, Professor of Mechanical Engineering
Michael Bozack, Professor of Physics

Abstract

In the 21st century, the electronics is everywhere and covers every aspect of human life. You can find it when watching TV, opening a laptop, or checking emails on a cell phone. Since 2015, 268 million PCs, 455 million tablets, and 253.1 million TVs have been sold globally.

Despite of the power of electronics, the failures of electronics would cause a variety of issues. One of the issues is electronic waste. When an electronics breaks down, it will be more likely to be replaced than repaired, because the cost of repair is usually much higher than buying a new device. There are two approaches to resolve this issue. On the one hand, we increase the reliability of electronics to maximize their lives. On the other hand, we study the Remaining Useful Life (RUL) of electronics, which provides life expectancy information of electronics. Knowing the RUL increases the efficiency of electronics usage in that it helps people avoid unnecessary labor of repairing when electronics approaches its designed life. The In-situ deformation of electronics affects their reliability, which could be used as the failure-leading indicator. The current dissertation focuses on the reliability of electronics, and how to quantitatively assess it during the operation.

The reliability of electronics can be affected by many factors including operational environment, manufacture, and frequency of usage. All of these factors contribute to mechanical failure of electronics, which can be reckoned as one of the major failure of electronics.. The mechanical failures of electronics can result in (a) encapsulate failure, (b) connector failure, (c) open circuit failures and (d) short circuit failures, and (e) component failure. The primary cause of

mechanical failure is deformation. Therefore, it is very import to know how much deformation is generated in the electronics during its operation.

Previously, the topic of deformation has been studied by many scholars. Several techniques are available to investigate the deformation of an object including strain gages, moiré method, and digital image correlation. The strain gages and moiré method require a pre-surface preparation, which creates a surface attachment or surface coating. The Digital Image Correlation(DIC) is an advanced technique that has been developed recently, and it has already been used widely to measure the surface deformation and strain. The current research focuses on the three-dimensional (3D) deformation field, which provides a full field deformation and strain. Specifically, we apply the Digital Volume Correlation (DVC) technique to investigate the In-situ deformation of electronics.

The development of the DVC technique began with Bay, Smith, Fyhrie, and Saad, 1999. The article illustrated the basic concepts and applications on the measurement of 3D displacements and strain in the Trabecular Bone using X-rays 3D reconstructions, under 35 microns voxel sensitivity . Since then, there have been many DVC applications in various fields, using a variety of data acquisition system and under different voxel sensitivities (Lenoir & Bornert, 2007) (Roux, Hild , Viot, & Bernard, 2008) (Haldar, Gheewala, & Grande-Allen, 2011) (Hussein & Barbone, 2012) (Moegeneyer & Helfen, 2013) (Brault, Germaneau, & Dupre, 2013) (Germaneau & Doumalin) (Haase & Noack, 2014) (Leclerc, Roux, & Hild, Projection Savings in CT-based Digital Volume Correlation, 2014). To begin with, the digital volume data can be acquired by the Scatter Light (Germaneau, Doumalin, & Dupre, 2007), and in this application the voxel sensitivity is about 60 micrometers. More recently, the application of micro-MRI (Benoit & Guerard, 2009) processes a larger sized 3D volume of 512-256-256 voxels and 40-20-20 mm in each dimension,

for which the voxel sensitivity is about 78 micrometers. Latest the laser scanning confocal microscopy (LSCM) (Toyjanova, et al., 2014) uses volume with size 512-512-P voxels with P range from the 64 to 256 voxels (i.e. 20 to 77 micrometers). The LSCM is also of the highest voxel sensitivity under the optical light condition, which could process organic tissues and living cells. In summary, all of these methods could generate gray-scale based digital volumes whose sizes vary according to the region of interest (ROI).

Despite of the development of DVC techniques, one of the issues regarding DVC application is its computational cost. The larger the size of the digital volume, the higher the computational cost will be. The paper of Gates, Lambros, and Heath (2012) proposed to increase the computational efficiency of DVC application using Cross-Correlation with the FFT algorithm. The paper implements the parallel computation technique to speed up the processed time. The current study incorporated Gates et al. (2012)'s method in to calculate the deformation of electronics from the X-ray CT scans using the DVC technique, which enables us to compute the full field strains in the package efficiently. In the meanwhile, we also create a 3D data visualization tool to view the results in differential 3D layers.

In Chapter 3, we showed the general 3D image-processing framework. Some applications such as defect and failure detection (Lee & Nguyena, 2000) were established using this visualization technique. Then, we introduced the DVC algorithm and its development. A detailed description of DVC algorithm would be presented in the Chapter 4.

In Chapter 5, several applications using the DVC were presented. First of all, we demonstrated the proof-of-concept with three points bending test using aluminum beam. Secondly, we applied the theorem along with the DVC technique to measure the In-situ LED chip deformation caused by the CTE mismatch. Then, we showed that the deformation could be

measured in the operational BGA package. In the latest development, we introduced this technique to measure the soft material deformation, in which we solved several issues that adapting the DVC algorithm to the soft-matter and low-density materials.

In Chapter 6, we introduced an advance Finite Element Method(FEM), which is based on the mesh generated from the X-ray CT scan. Using the 3D volumetric image, the geometry of object can be reconstructed from the ISO-surfaces, and we can create the FEM discretization within those surfaces. The FEM discretization utilizes the four vertices and four faces tetrahedron elements. Then, this mesh work can be used with the FEM simulation. Moreover, in chapter 6, we demonstrate a thermo-mechanical simulation on the BGA package, in which the defects such as solder void and crack has been presented. In the end, the cumulative plastic strain energy has been computed, and the stress-strain hysteresis loop has also been computed.

In Chapter 7, we used another technique to track and to predict the life of electronics--the Data Driven Prognostication technique, which finds and tracks leading failure indicators of a system. In electronics systems, the electronic resistance is usually picked as leading indicator for the mechanical failures because by Ohm's Law, resistance will increase with the presence of a solder crack. Similarly, in LED system we use the lumen maintenance as a leading indicator for mechanical failures, because the lumen maintenance output will decrease as the system degrades. Using the data driven methods (i.e., Kalman Filter, Extended Kalman Filter, and Bayesian Method), we iteratively update the system state vectors to find the best predictor of RUL. Then, the RUL can be predicted and calculated from the extrapolations using characteristic function. Overall, in these two systems, the PHM framework has been established, and we will cover this in details in the Chapter 7.

Acknowledgments

I wholeheartedly thank my research advisor and mentor, Dr. Pradeep Lall, for all his kind guidance during my study and research at Auburn University. I sincerely thank my committee members: Dr. Jeff Suhling, Dr. David Beale and Dr. Michael Bozack for supporting me in the process of dissertation construction and completion. I want to thank my dissertation reader Dr.Prorok for helping me check the dissertation. My special thanks to my research fellows and friends for their kind helps in my research and life. In the meantime, I thank the staffs in the Mechanical Engineering for their assistance in the department. Many thanks to my parents for their unselfish love and encouragement in my life. Also, I want to thank my special one in my life: Yuan Yuan for her supports in this work. Lastly, I would like to thank anybody who ever provides me a help in the past, and thank you all very much.

Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

1.1 Research Motivation

Due to the development of semiconductors, the high-speed processors (i.e., CPUs and GPUs) have become much faster in last two decades. The applications of small-scale PC and flexible electronic became more popular in many areas. Today, electronics are commonly used in contemporary lives. We can find them from every aspect of daily lives, from home to office, from personal devices to automobiles, from aircrafts to satellites, etc.

Although electronics are widely used, environmental and mechanical factors (e.g., temperature, moisture, mechanic shock, and vibration, etc.) could potentially threaten their performances. In order to manage the survivability and reliability of electronics, a monitoring framework system needs to be established, and several techniques are required to accommodate the system. The Prognostication Health Management (PHM) framework is an online and offline electronic-monitoring framework. The PHM framework can be adapted to any existent system of electronics to keep track of failure of a leading indicator, and assess the current damage of the system.

One important function of PHM system is to estimate the Remaining Useful Life (RUL) of electronics. The RUL predicts the expected failure time of electronics. With the quantitative prediction that RUL provides, the system maintenance can be scheduled and new components can be booked and shipped in advance, so that the total labor and extra cost of maintenance will be reduced.

The current dissertation proposed a data-driven method to calculating RUL, which can be used to track and predict the Time-to-Failure (ToF) in the target system.

1.2 Review on the Electronic Package History

The earliest record in the history of electronic packaging dates back to 1884, when the American Institute of Electrical Engineers (AIEE) society was founded. The initial development of electronics was contributed by the radio wireless system development. In 1895, Guglielmo Marconi (1974-1937) made first radio transmission. After ten years, Greenleaf Whittier Pickard (1877-1956) creates a solid-state point-contact diode based on silicon material. The first TV demonstration comes out in the year of 1925. Due to the military application during World War II, the Radar system was invented in 1940. In 1947, John Bardeen (1908-1991), Walter H. Brattain (1902-1987), and William B. Shockley (1910-1989) invented the bipolar transistor at the Bell Laboratories. Only three years after, the first color TV was designed.

In 1952, Shockley described the unipolar field-effect transistor. In the same year, a commercial production of silicon bipolar transistors was started at the Texas Instruments (TI). In the year of 1956, Bardeen, Brattain, and Shockley received Nobel Prize for the invention of bipolar transistors. Soon in 1958, the first integrated circuit (IC) was invented by TI. Three years later, the First commercial digital ICs became available by the Fairchild Semiconductor.

A very import event in the electrical engineering society occurred in 1963. The AIEE and the IRE merged to become the Institute of Electrical and Electronic Engineers (IEEE) that is the professional society today. In 1967, the first semiconductor RAM (64–bit) was discussed at the IEEE International Solid-State Circuits Conference (ISSCC). In 1968, the first commercial ICs operational amplifier was published by the Fairchild Semiconductor. In 1970, the Dennard at the IBM invented the one-transistor dynamic memory cell. In 1971, the Intel proposed the 4004 microprocessor. The next year, in 1972, Intel published 8008—the first 8-bit microprocessor. In 1974, the first commercial 1-kilobit memory chip was developed. In 1974, the 8080

microprocessors were introduced. In 1978, the first 16-bit microprocessor was developed. In 1984, megabit memory chip was introduced. The experimental gigabit memory chip was presented at the IEEE ISSCC only 20 years ago in 1995.

Along with the development of semiconductors, the size of leading edge CMOS node has been reduced. In 1985, the leading edge CMOS node was of 0.25 um, which reduced to 0.13 um in 1995, 90 nm in 2000, and 65 nm in 2010. In 2015, the size is 28 nm. This trend conforms to the Moore's law, which claims that the density of transistors on an IC chip would double every year.

As IC chips become more condense, the surface mount technology also improves. Figure 3 shows the migration of IC Pin Counts with time. The number of IC pins increases rapidly as the year forwards.

As IC chips become more sophistication and diversification, the industry faces with greater challenges to maintain their reliabilities. The current dissertation focuses on the reliability of electronics, and how to quantitatively assess it during the operation. In response to this demand, the current dissertation proposes a new possible technique to measure and predict the reliability of electronics

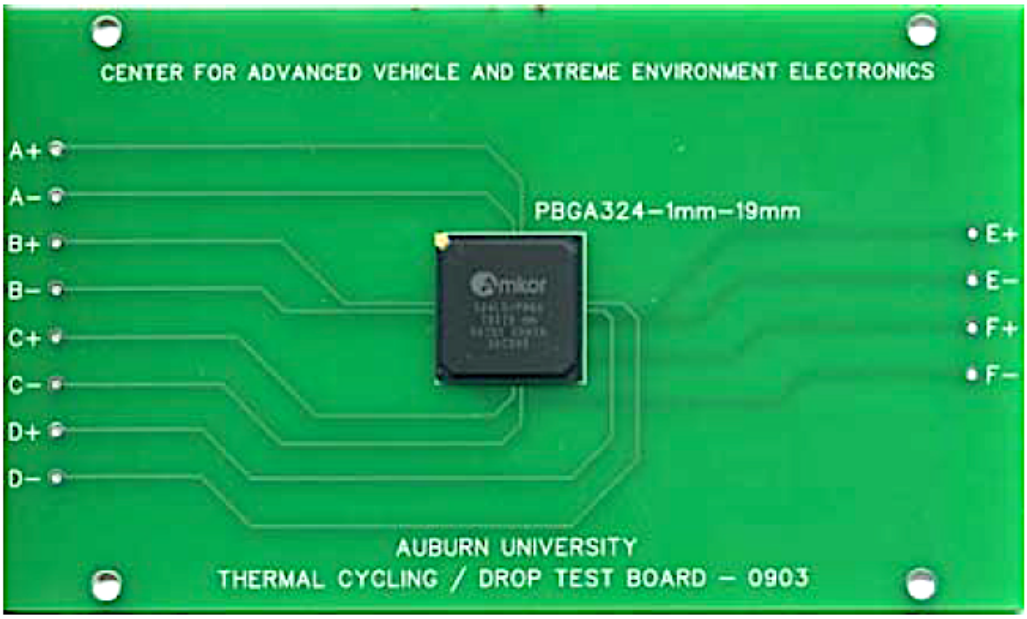


Figure 1 Surface Mount Electronic Components

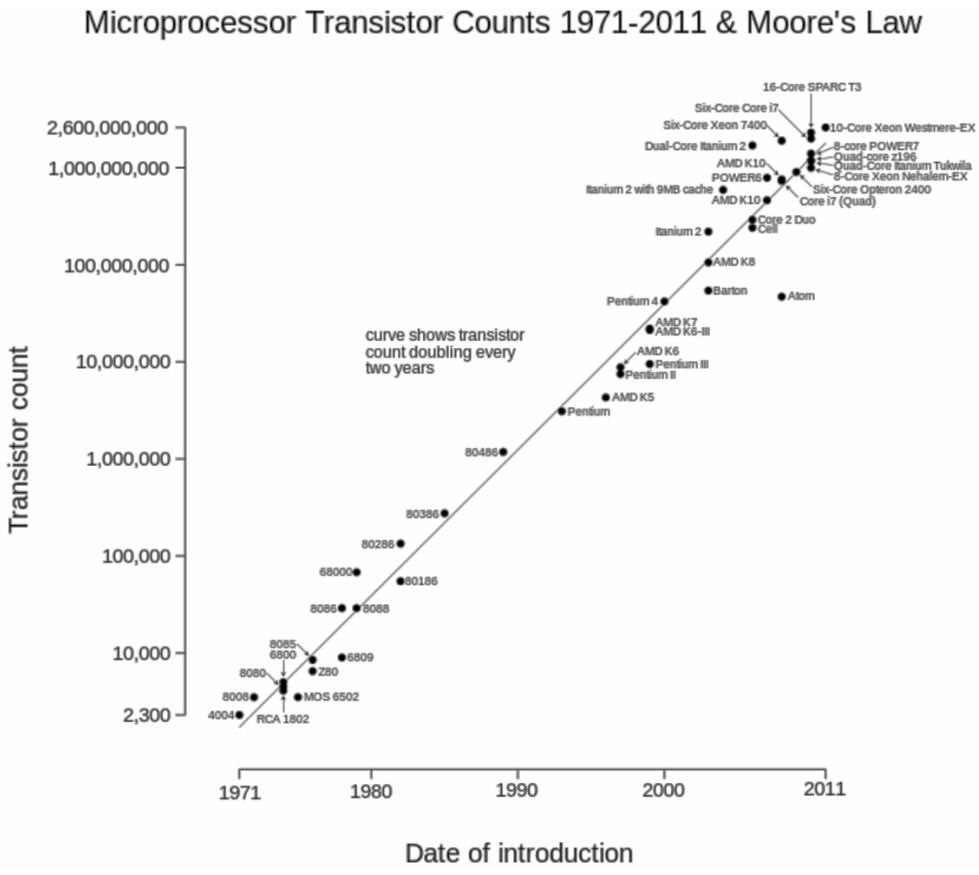


Figure 2 Moore's Law on the transistor counts of the CPU (Wgsimon, 2011)

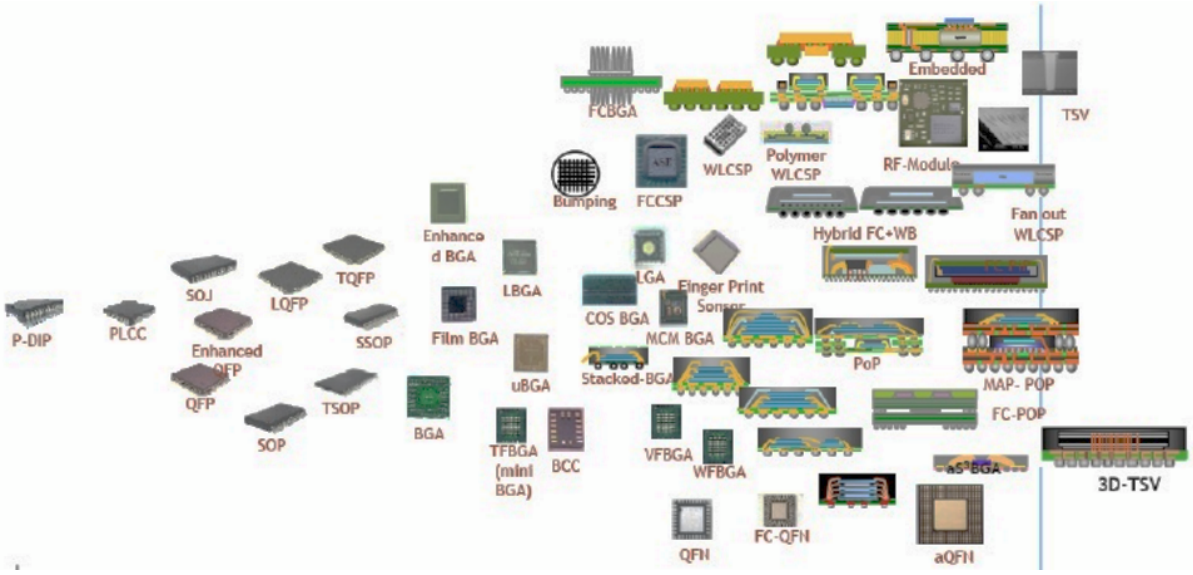


Figure 3 The evolution of electronics (Terlizzi, 2011)

1.3 Overview X-ray Micro CT Development

The first theory of the X-ray dated back to 1917, when Johan Rodan (i.e., a mathematician from Austria) proved that an n-dimensional object could be reconstructed from (n-1) dimensional projections (Radon, 1917) (Huang & Peng, 2011). The first CT scanner was then created by Godfrey N. Hounsfield in 1927. The first EMI-Scanner was installed in the Atkinson Morley Hospital in Wimbledon, England. The invention was of such importance that Cormack and Hounsfield were awarded the Nobel Prize in 1979 in the field of Physiology or Medicine.

The early application of X-ray CT technology lies in medical field. The first Micro-CT was built by Jim Elliott at the beginning of the 1980s, with voxel sensitivity of about 50 micrometers. The first commercial trabecular bone scanner appeared in 1994, which is standard bone research equipment. Nowadays, there are several manufactures all around the world producing X-ray Micro-CT scanners that provide in-vivo scans.

The current study calculated the deformation of electronics from the X-ray CT scans using the DVC technique, which enables us to compute the full field strains in the package efficiently. In the meanwhile, we also create a 3D data visualization tool to view the results in differential 3D layers.

1.4 Overview Three Dimensional Image Reconstruction

The three-dimensional (3D) image is an extension of the two-dimensional (2D) image. In the 2D image, each pixel is assigned to gray scale value ranging from 0 to 255 (8-bit). In 3D image, each voxel—which is similar concept to the pixel in 2D image—is also assigned with a greyscale value, ranging from 0 to 65535 (16-bit). The modalities of the 3D image can be reconstructed from several sources including X-ray CT,, Magnetic Resonance Imaging (MRI), Ultrasonic Image, and the Optical Microscopy.

The 3D scene can be projected onto the 2D plane. If two projections from a 3D image are available, the intersection of the two projections can be used to recover the points of the original 3D image. The intersections of the projections can be found by solving the linear system, which strives to find the appropriate view of projects that contains some vital points for the construction of a 3D object.

The current research focuses on the three-dimensional (3D) deformation field, which provides a full field deformation and strain. Specifically, we apply the Digital Volume Correlation (DVC) technique to investigate the In-situ deformation of electronics.

1.5 Algorithms for Prognostics and Monitoring Reliability of Electronics

Algorithm is an important component to the framework of PHM, which tracks a feature of degradation—the leading indicator—in the system. With sophisticate algorithm, the state of a system can be predicted by studying the migration of system state vectors.

The algorithm of PHM system usually incorporates advanced machine learning algorithms. These algorithms are data-driven method, and they make PHM framework adaptable to any information system, both online or offline. The high performance of modern computing networks (e.g., a cloud or parallel computing network) benefit the PHM framework. Thus, some computational expensive algorithms, including the Particle Filter (PF) and the Monte Carlo Markov Chain (MCMC), can be achieved and actualized in the system.

The current dissertation integrated and incorporated several important PHM algorithms. Firstly, the Kalman Filter (KF) has been used to track the resistance change of electronics during the vibration test. The Kalman Filter has been proved very effectively to track the linear system. Secondly, the Extended Kalman Filter (EKF) can adapt to a nonlinear system, in which a pseudo distribution can be chosen from the Gaussian, or Non-linear distributions.

The current study quantified the accuracy of prediction in the PHM system using statistical criterions. Following the criterions, the performance of the algorithm can be analyzed and compared to other algorithms. We presented several plots to illustrate the precision of algorithm. The Alpha-Lambda plot describes the accurate of prediction at any given time. We also presented the Beta plot, describes the Confidence Interval (CI) of integral prediction. The relative accuracy plot describes an absolute difference between the prediction and actual data. The details of the algorithm will be covered later in Chapter 7.

Chapter 2 Literature Review

2.1. Dimension and Spatial Measurement System

2.1.1 Computer Aided Vision Measurement

The Computer Aided Vision (CAV) is a new technique that comes out in last decade. CAV is the method extracting the information in the digital images. The goal of the CAV is to assist engineers in modeling and to measure the environment by utilizing a visual sensing (Ballard & Brown, 1982). Recently, the Computer Aided Vision has been widely used in the mechanical engineering field to measure the object dimension, which dramatically reduces an amount of labor of engineer to elaborately.

The earliest application with the CAV starts from the Artificial Intelligent in the late 1960s, in which many algorithms try to extract the information from the digital images (Szeliski, 1989). At the end of the 1970s, a significant number of advanced Computer Vision Algorithms have been invented, and some of them are still used today. Those algorithms include the extraction of edges from an image that is known as an edge detector, segmentation an object from a background of the picture, optical flow and motion estimation (Burton & Radford, 1978) (H.S, 1999).

In the 21st century, there are many new developments in CAV techniques such as projective 3D reconstructions and camera calibration. The 3D reconstruction can combine several 2D images to form a three-dimensional view, which includes the deep information that has lost in the single 2D image. This rich research benefits a lot of engineering field. It is believed that the Computer Aided Vision can be integrated into the electronic packaging area in the mechanical engineering such as failures detection, defects detection (Koch & Georgieva, 2015), and deformations computation (Lall & Wei, 2015).

2.1.2 Strain Gages Device Measurement

A strain gage is a device to measure strain on an object. It was invented by Edward Simmons and Arthur Ruge in 1938. The strain gage contains a metallic foil trace, and it is flexible due to the elongation and bending (Hannah & Reed, 1992). The object is deformed as the foil is deformed, which produces the electrical resistance change due to the Ohm's law. Using the Wheatstone bridge, the electrical signal can be measured precisely (Larry, 1992). Therefore, the amount of strain produced can be acquired.

The primary advantage of strain gages is that their sizes are small, which enables the measurement of strain to be very precise. Also, the implementation cost is low, which is affordable to many companies and research institution. In the meantime, the strain gage can be pasted and applied to the surface of variety metallic materials such as aluminum, copper, and some alloys.

The disadvantage of strain gage is that they are tiny and hard to install on a target; the surface has to be cleaned before the installation, and the environmental temperature has to be maintained constant. Moreover, the strain gage cannot apply to soft material in general, due to its stiffness. The stiffness of gage is larger than soft material, which requires a calibration and a correction factor.

All in all, the strain gage has a wide application in the experimental mechanics as well as industrial projects such as smart bridges, railways, wind tunnels, vehicle classifications, etc. The strain gage is still the most widely used electronic sensor in the mechanical engineering field.

2.1.3 Moiré Based Method

A Moiré method is an optical method measuring length, rotation, and shape evaluation (Gustafsson, 2000). The Moiré patterns are extremely useful to help understand basic interferometry and interferometry test results. The moiré pattern is produced by two identical straight-line gratings rotated by a small angle about each other. The Moiré pattern is used to study the movement of two fringes on the sample surface (Houxun & Alireza, 2016).

The Moiré methods can be categorized into two groups; the first one is Geometric Moiré. Its fringes patterns can be interfered based strictly on "geometry." The displacement can be measured both in-plane and out-plane. The in-plane displacement can be measured using in-plane Moiré method, and the out-plane displacement can be measured using shadow Moiré method. The second group is Moiré interferometry, which needs consider the diffraction effects (C.A. Walker, 1994).

The major advantages of this Moiré method are that it is relatively easy to be implemented in deformation measurement, and it is also less expensive to create the fringe patterns. The analysis is quick and secure, and the computation is not elaborate. It a is the fully advanced method that can be used broadly the in engineering field to find the deformation, and it also can be applied to deal with a dynamic problem.

There are also several drawbacks of Moiré method. One is that it is very sensitive to shake and oscillation, which requires stabilized table and vibration-free condition when conduct the testing. Also, the accuracy is not high enough for a large spaced grating. It requires the pre-knowledge of the displacement field. Moreover, it requires the surface preparation for the experiment, which can affect the original property of the surface, and it also requires a high-quality digital camera to record the image.

2.1.4 2D/3D Digital Image Correlation Method

Digital Image Correlation (DIC) is a method of measurement based on the optical images (Anuta, 1970). The DIC measurements include the displacement and strain measurement, which are on the surface (Pan, Qian, Xie & Asundi, 2009). The displacements and strains can be either 2D or 3D, and it depends on the different scenario. Currently, it widely used in a variety of science and engineering. Digital Image Correlation has been growing in popularity (Wattrisse & Chrysochoos, 2000) (Zhang, Eggleton, & Arola, 2002) (Barthelat, Wu, Prorok, & Espinosa, 2003) (McGinnis, Pessiki, & Turker, 2004) (Subhash, Liu, Moore, Ifju, & Haile, 2011). The DIC method has been widely used in micro and nanoscale mechanical testing (Keating & Wolf, 1975). The advanced camera and computer system enable a rapid development in the DIC applications.

In the Digital Image Correlation, we create an exclusive surface random pattern. Normally, the smaller and finer pattern is, the more precision DIC calculation will be. In many circumstances, the pattern has been created by the reflection of light received by the camera sensor. As a result, the light cannot change in the room. Otherwise, an artificial variation is introduced, and it will produce an error in the subset correlation.

The basic concept of the DIC is to correlate subset between the reference and deformed image (Sutton & Orteu, 2009). Firstly, we have a subset in the reference image, called a template, then we will search all the possible subset in the deformed image to find the best match with this template. Once the best subset is determined, we can compute the displacement. In general, the DIC algorithm is categorized as one import algorithm in the image processing. In the early 1980s, the USC researchers have brought the DIC in the field of mechanics. The stress and strain can thereby be measured non-contact on the surface of the object. There are two types of Digital Image Correlation (DIC): one is the 2D-DIC, and another is 3D-DIC. In general, the 2D-DIC is used to

measure the in-plane displacement and strain (Bing, Hui-min, 2006), and the 3D-DIC is used to measure the out-plane displacement and strain.

For the 2D-DIC method, the initial papers were published by the Peters and Ranson in 1982. The proposed approach indicates that the deformation could be calculated by the correlation of the digital images for numerous small subsets between the uniform images and deformed images. Using this method, in 1983 Sutton developed numerical algorithms to demonstrate the 2D-DIC method.

In the 1980s, researchers have begun to use the 2D-DIC to measure the crack growth in the fracture mechanics. Early work by Steven McNeill demonstrates the use of 2D-DIC to compute stress intensity factor estimation. Amstutz measured tensile and shear deformations in Arcan specimens experiencing mixed model I/II am loading. Han et al. exhibited a high magnification optical system to measure deformations nearby stationary and growing crack tips under nominal mode-I loading. Later, the researchers started to use the 2D-DIC to understand material properties including of metals, plastics, wood, ceramics, and paper. At the end of the 1990s and 2000, there were lots of studies on the damage in composites and concrete (McNeill, 1997)(Amstutz, 1997).

In the past several years the 2D-DIC has experienced rapid growth worldwide in mechanical engineering. Schreier (1997) shows the importance of the image reconstruction process when attempting to increase the accuracy of the matching process. His study reveals that the image pixel accuracy could be dramatically improved by using a higher order spline interpolation. Later, Lu and Cary (1999) have continued his work, and they show that quadratic shape functions yield some advantages in the subset matching algorithm in the non-uniform strain field. Other researchers have demonstrated some extensions to various aspects of the 2D-DIC improvement including the search algorithm, matching approach, and registration process. Then,

the researchers have started to develop a full-field DIC method that combines analytical models with the measurements. Recently, in the applications, the similar methods have been actively utilized for mechanical property evaluation such as elastic material, heterogeneous material, and hyper-elastic material.

Later, the 3D-DIC has been developed, in which two cameras have been applied to the out-of-plane measurement. The two cameras have to be calibrated and synchronized. In the calibration, the focal length, the magnification, and the radial and tangential distortion are required to be determined. During the matching process, the translation vector and rotation matrix are calculated. In the CAVE-3 lab, we have used the 3D-DIC intensively for capturing the high-speed strain of object during the impact. The object has been coated with black and white sputtering random pattern. During the experiment, two constant light sources have been placed on the top of the object to illuminate the surface of the object. Then, we use a two synchronized high-speed camera to capture the impact video, in which the frames could be employed for the 3D-DIC analysis (Rethore & Tinnes, 2008) (Rae, Williamson, & Addiss, 2010). By correlating each paired frames, we can compute the deformation on the surface of the object. We have successfully used this technique to study the failure of electronic assembly under a high-G shock system. Moreover, we also adapt this 3D-DIC with vibration test, in which high-frequency strain component are computed and studied.

The DIC method has many advantages containing: (1) the preparation of the specimen is relatively straightforward as the speckle patterns accepted as unique targets are produced by spraying coating its surface, (2) universal and ordinary white light are used to illuminate the surface of the specimen, which is comparatively cheap and feasible for varied application, (3) the size of the specimen could be varied. The tested specimen could be large, as long as the surface pattern

makes an adaption to it, (4) the experiment is insensitive to the vibration and shock. However, it requires that the cameras to be stable and the light source stay constant.

The DIC method also has some limitations and disadvantages: (1) the DIC is highly sensitive to the light fluctuation and variation, and this is because the high variation could modify the contrast of images and the mathematic correlation function can not distinguish the shifting, (2) the 3D-DIC requires more equipment to be implemented, and additional light sources and high-resolution cameras are needed, (3) the capability of the computer during the post-processing must be high, and the mathematic algorithm is required to resolve the computational efficiency issue, (4) The calibration of cameras before the experiment can not be neglected, and calibration must be redone once the equipment and environment change.

2.1.5 Digital Volume Correlation Method

Digital Volume Correlation is a refreshing technique for measuring the three-dimensional displacement and strain, which we can have an idea bout the shape of deformation in the space (Leclerc, Perie, Roux, & Hild, 2010) (Leplay & Rethore, 2013) (Lecomte-Grosbras, Rethore, & Limodin, 2014) (Hu & Luo, 2014). Also, the Digital Volume Correlation can be used to find the movement or deformation by mapping uniform and deform images. It has already been established and well developed through the recently 15 years, and many scholars and engineers have put an effort to develop and apply it.

Digital Volume Correlation has been rapidly developed in the recent decades. Volumetric imaging and image analysis have been an active area of research and development within the medical community for decades, benefited by sophisticated imaging capabilities. Micro and Macro Computer Aided Tomography (CT), Magnetic Resonance Imaging (MRI), Confocal Imaging

Microscopy (CIM) and Positron Emission Tomography (PET) are examples of technology that are currently available for full-volume imaging of biomaterial systems.

The Digital Volume Correlation has been firstly proposed by the (Bay, Smith, Fyhrie, & Saad, 1999), it illustrates the basic concepts and fundamental applications on how to measure the three-dimensional full field strain in the Trabecular Bone. The application voxel sensitivity is a 35-um cube with a gray-scale depth of 16 bit. After it, there are many DVC application papers in (Rueckert & Sonoda, 1999) the various fields, which can be characterized by the method of data acquisition system. In the Bay's paper, it shows the fundaments of Digital Volume Correlation and the raw volume size is 520-520-580, and voxel sensitivity is 35 micrometer. In the Scatter Light application (Germaneau, Doumalin, & Dupre, 2007) and Germaneau's paper voxel sensitivity is 60 micrometer, and the subset is 31-31-31 voxels, and in the micro-MRI (Benoit & Guerard, 2009) and it processes the volume size 512-256-256 with 40mm-20mm-20mm that voxel sensitivity is about 78 micrometer. In the laser scanning confocal microscopy (LSCM) (Toyjanova, et al., 2014), it uses the 512-512-P voxels with P range from the 64-256 voxels (20-77micrometer). The laser scanning confocal microscopy is the highest voxel sensitivity so far in the optical light scan so that it can deal with various organic tissues and cells (Frank, Hong, Maskarinec, 2007). All of those methods are targeting to generate the gray-scale analyzed digital volumes. The volume size can be varied according to the analysis region. For example in the X-ray CT system, the analysis digital volume can be as large as 20483 voxels. In the paper (Gates, Lambros, & Heath, 1999), it explains well how to efficiently correlate the two digital volumes using the Cross-Correlation Coefficients and FFT algorithm. Also, it implements the parallel processing technique to speed up the processing efficiency.

Industrial Micro-Computed Tomography (uCT) has found applications in defect detection and failure analysis of electronics components. Using the same basic principles as used in medical computed tomography techniques, it offers very high resolution, magnification, and data processing. Parts tested in the industrial usually have no side-effect of prolonged x-ray exposure, which enables the user to perform quality scans and take thousands of projections of the sample. These images have been combined and transformed into 3D rendered volume providing interior details about package architecture and the ability to inspect the volume on a slice-by-slice basis at a user specified angle to detect defects. The quality of X-ray imaging and reconstructed volume depends on lots of parameters, (e.g. acceleration voltage of electron beam, filament current, monochromaticity, and the distance between the sample and the X-ray tube or the detector). For analysis purpose, X-rays are made a pass through the specimen as the sample is rotated in the X-ray area. When X-rays penetrate the sample, some amount of energy is absorbed, (i.e.,X-rays are attenuated). This attenuation depends on density, thickness, wavelength of X-ray and an atomic number of the material medium through which X-rays are passing. The Detector detects this attenuation and represents on a computer screen in the form of different gray scale values. The object is rotating in the X-ray field. The resulting projections are then joined using complex reconstruction algorithms to create a 3D rendered version of the object. The YXLON Cougar system located at CAVE3 facility at Auburn University, which has been utilized for the examination of samples. The YXLON Cougar Machine is a state-of-art machine with a detail detectability of less than 1 micron and a measurement resolution of 0.1. A server-grade workstation with step-up transformer is providing constant acceleration voltage in the range of 70KV to 160KV to accelerate and direct the electron beam. Tungsten filament has been used to produce an electron beam. X-rays are generated from electron beam in an X-ray tube. Vacuum is maintained in the

tube by the rotary vacuum pump. Coherent and monochromatic X-rays are generated in the X-ray tube. Focus lenses are provided to have the sharp focus and to avoid blurry images. Volume Graphics software has been used for the 3D reconstruction process. The software takes the still images of the object acquired as the object is rotated in the X-ray field and uses the data to reconstruct the three-dimensional version of the object under examination. High-resolution measurements of the geometric dimensions can be made non-destructively.

2.2 Prognostic Health Management of Electronics

2.2.1 PHM on PCB Assembly

The electronic chips can be subjected to a variety of extreme environments including wide temperature extremes, very high temperature, very low temperature, mechanical shocks, and vibration. Damage Diagnosis and Failure monitoring can provide insight into the proximity of the system to catastrophic failure. Currently, the methodologies, used in damage diagnosis, include the offline detections like the microscopy, X-ray, as well as the diagnostic techniques such as built-in self-test, fuses, and canaries. Previously, feature vectors and joint time-frequency characteristics have been successfully used for estimating the system state and predicting RUL in plastic ball-grid array packages (Lall, 2004). Also, resistance spectroscopy with phase-sensitive detection has been used to develop leading indicators for the failure of interconnects and prognosticate Remaining Useful Life (Lall, 2005) (Constable 1992).

The Prognostic Health Management (PHM) has been used widely in many fields. First of all, the Avionics Systems, autopilot system and remote control system, always ask for high consistency and reliability, which timely replacing the near-failure electronic chip is critical for achieving such requirement and maintaining high reliability. Therefore, the PHM, a very useful

and pertinent method, has been introduced and applied to the detection and quantification of the Remaining Useful Life (RUL) in such high-reliability systems. Moreover, other fields that prevalent using powerful semiconductor and electronic devices also require systems to be built on trustable and reliable electronics. Therefore, the system's life must be visualized and controllable, which can be fulfilled through the ways of PHM. Therefore, it is developing lots of methods and tools to realize such PHM in the electronic systems over many years.

In the past several years, the surface exams and appearance detections are very limited and provide no insight into the system. However, the expense to thoroughly check the system offline for one time is too high to bear. Thus, a new method came into being that the time-based prevent maintenance and early RUL prediction relying on the system's state vector. Therefore, not only can we check the system online or offline with relevant time and life, but also we could save labors and time for getting the predicted the RUL. The current RUL detection mainly comprises two methods, (1) Data-Driven Prognostic, (2) Model Based Prognostic (M. Daigle, 2010). The first part focuses on establishing a model for cumulative damage, which learns directly from the data. The second part incorporates a physical understanding of the system into the estimation of RUL. The author primarily focuses on the Data-Driven Prognostic, which concentrates on the data of Resistance Spectroscopy and Phase Shifting during the experiments. Compared with other approaches, it could be displayed quicker and cheaper way in the world of PHM, and its current application includes, fatigue damage in the aircraft (Munns, 2000), electronic chips (Baldwin, 2002), railway (Barke, 2005), and power plants (Jarrell, 2002).

Kalman Filtering (Kalman, 1960)(Zarchan,2000), the very powerful recursive algorithm, has been previously used for the GPS (Kim, 2003), Ground Navigation (Bevly, 2007) and Economic Prediction (Solomou, 1998). Kalman Filter tool has also been imbedded in PHM with

the prediction of RUL (Ryan, 2010). The Kalman Filter (KF) only deals with the linear problem while the Extended Kalman Filter (EKF) could solve the non-linear problem in realistic. However, the performances between Kalman Filter and Extended Kalman Filter in such PHM fields have not been thoroughly compared yet. So, in this paper, the performances between those two are compared in different ways with various prognostic metrics such as the Prognostic Horizon, $\alpha+\lambda$ metrics, relative accuracy, convergence (Saxena & Celaya, 2009).

2.2.2 PHM on Solid State Lighting System

The Light Emitting Diodes (LEDs) have been widely used since the last decade, and there is a tendency that LEDs will dominate the lighting market because of their advantages in light efficiency, energy saving, improved physical robustness and extended operating hours compared to the traditional fluorescent light. The industrial utilization of LEDs in some extreme environments requires the LEDs have a significant long life and high reliability. It is also necessary that these LEDs must withstand high relative temperature (RH) and high relative humidity (RH) environment without much lumen maintenance degradation after long term wearing. The most critical value for the failure of LEDs is 'L70' (the Lumen Maintenance reaches 70% of the original and pristine started from 100%). There is extremely long-term degradation for the LEDs to reach its L70, which is usually over 30,000 hours for the Philips LUXEON Rebel lamps. The former methodology to quantify the LED's L70 is TM-21 (Technical Memorandum) wrote by the Philips, Osram, Nichia, Illumitex, GE, and Cree. The TM-21 uses the Arrhenius Equation to calculate the Activated Energy of System (AES) and to combine with the Curve-fitting methodology. However, to correctly apply the TM-21, we have to obtain decay rate and pre-decay factor with the upper and lower temperatures, which is always subjected to multi-temperature situations. Therefore, the method tends to be impractical for the single temperature case, and it is often difficult to calculate

the AES precisely with large temperature Intervals. Thus, we have introduced KF/EKF algorithm to make an extension and a complement for the LEDs' L70 life extrapolation and prediction.

The KF/EKF algorithm has been previously used for monitoring and predicting the electronic system failures, which is a part of the Prognostic Health Management (PHM) of Electronic systems (Lall, 2004). The KF/EKF could watch the tendency of resistance change and monitor it when crossing the failure threshold of identified system. Similarly, in this paper, we use the KF/EKF to track the Lumen Maintenance degradation lines and make an extrapolation into the future space to project current system model and to calculate the LED Remaining Useful Life (RUL). In the KF/EKF prognostic system, the KF is used for Linear-system tracking while the EKF can be utilized for the case in the Non-linear system. Both algorithms can generate dynamic and update estimations at each data points of interest, and then we can get the distribution of pseudo L70 life from those estimations.

In the Chapter 7, we firstly introduce the general theories and correlations between TM-21 calculator and KF/EKF algorithm. Then we will discuss the several identified LM degradation model of LEDs' system, and combine those models to find pseudo L70 life distribution. In the end, the F (t) failure cumulative function has been obtained, and the reliability function has also been shown with 95% confidence interval.

Chapter 3 X-ray Micro CT System

3.1 X-ray Micro CT Equipment

In the CAVE3 lab, there is a 'YXLON' Industrial X-ray Computed Tomography Machine, which made by the German Company. From the top to the bottom of interior, there are (1) overall cameras; (2) the X-ray Insulator; (3) the X-ray detector; (4) Tilt Rotation Arm; (5) CT tray; (6) X-ray Generating Tube; (7) Vacuum Generator; (8) High Voltage Power Supply, and (9)Vacuum Pump. The concept is very simple, which the X-ray tube generates the X-rays penetrating the target and reaching to the X-ray detector. There are lots of X-ray sensors on the detector, from which the signal will be transmitted to the scan control computer. Then, the scan images will be sent to the reconstruction computer. There are two computers in the CT system, one is responsible for controlling the CT scan in real time, and another controls the reconstructing the projection images into 3D analyzed digital volumes. Then, we can use the Digital Volume Correlation(DVC) technique to correlate the two digital volumes from two different scans. The CT computer controls Graphic User Interfaces (GUI) to use the specifications of the scan and move the CT hardware (e.g. CT tray inside of the CT machine). The CT reconstruction computer has large PC's Random Access Memory(RAM) and high capability graphics cards as well as fast CPUs because of it needs a fast computation unit to process the volume reconstruction and image processing. Those two computers can exchange the data simultaneously during the CT scan, which can save the space of the computer and take advantage of the reconstruction computer.

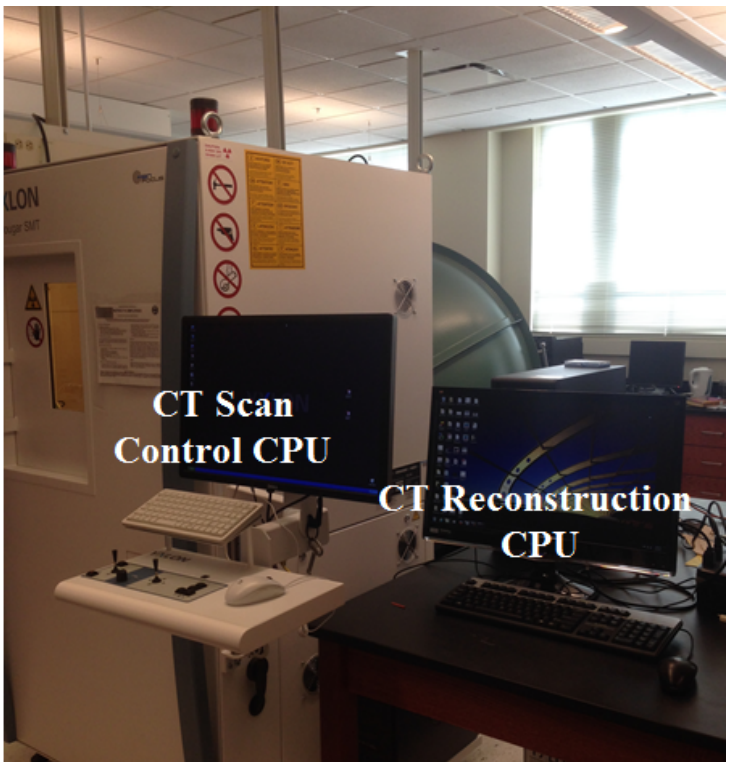
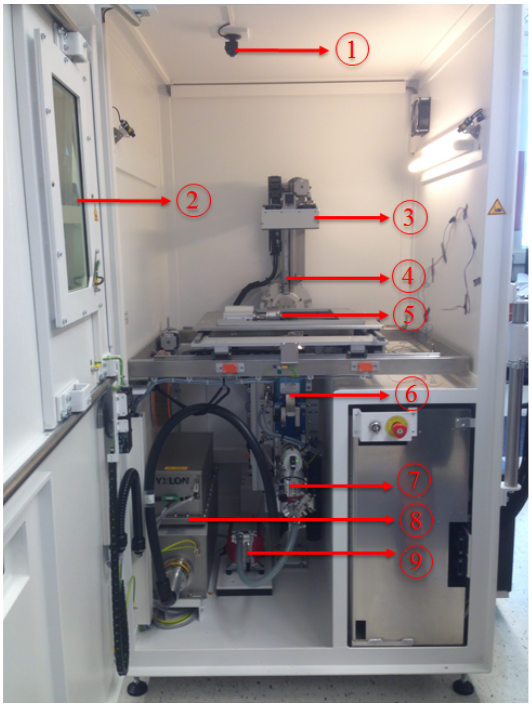


Figure 4 CT Machine and Two types of Computers



1) Overall Camera
2) X-ray Insulator
3) X-ray Detector
4) Tilt Rotation Arm
5) CT Tray
6) X-ray Generating Tube
7) Vacuum Generator
8) High Voltage Power Supply
9) Vacuum Pump

Figure 5 The YXLON X-ray Industrial CT Configuration

3.2 Grayscale Level Image:

The Cross-Correlation Coefficien(CCC) computation is based on the gray level value of image voxels, which can be loaded as a 3D matrix into the MATLAB. The 3D matrix is formed slice by slice, in which each slice contains the cross-section planar information of the 3D object. For example, we can look at the one cross section of solder in the following Figure 6. The white part is the solder, and each voxel has its unique value, which presents the density of the object. The high-density part has a large gray scale value. So, the different part of the components in the object has a variation of grayscale value, which generates a histogram. Typically, the voxel value applies the unsigned 16-bit image (i.e. grayscale value is ranged from 0 to 65536), which saves the space of the machine. On the other hand, more precise and detailed images can be generated using an unsigned 32-bit image (i.e. grayscale value is from 0 to 4294967269) in the computational process of the Digital Volume Correlation.

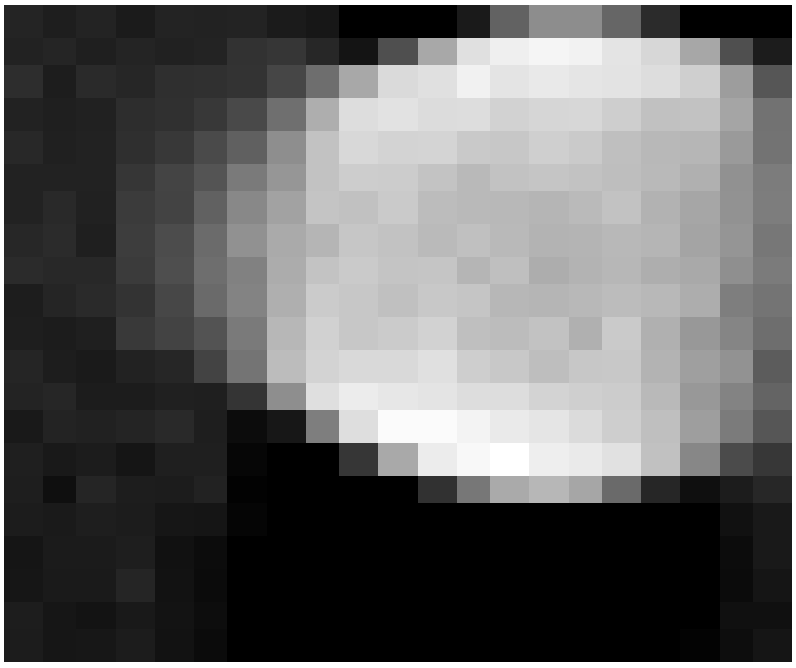


Figure 6 Grayscale Image of Solder Reconstruction

5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0.1070	0.0393	0.1063	0.0584	0.0798	0	0	0	0	0	0	0	0	0	0	0
0.1016	0.0790	0.1391	0.1076	0.1038	0.0059	0	0.1175	0.4204	0.6897	0.7739	0.8108	0.7033	0.4655	0.1125	0
0.1204	0.1349	0.1406	0.2293	0.1770	0.1918	0.4377	0.7774	0.9034	0.9463	0.9248	0.9649	0.9237	0.8617	0.6680	0.4073
0.1344	0.1166	0.1801	0.2533	0.4477	0.7285	0.8634	0.9383	0.9357	0.9191	0.8828	0.8552	0.8300	0.8658	0.7972	0.5884
0.1179	0.1608	0.2478	0.4305	0.7151	0.8741	0.9176	0.9222	0.8807	0.8837	0.8275	0.8284	0.8222	0.8095	0.7601	0.5852
0.1347	0.2066	0.3583	0.5430	0.7736	0.8682	0.9071	0.8539	0.8523	0.8690	0.8222	0.7905	0.8126	0.7631	0.7686	0.6257
0.1745	0.2660	0.3795	0.5521	0.7956	0.8381	0.8541	0.8346	0.7865	0.7579	0.7883	0.7784	0.7824	0.7667	0.7411	0.5884
0.2110	0.3154	0.4412	0.6088	0.8094	0.8672	0.8745	0.7789	0.7926	0.7637	0.7633	0.7547	0.7774	0.7302	0.6867	0.5889
0.1885	0.2586	0.4008	0.6049	0.8484	0.8592	0.8719	0.8075	0.8125	0.7601	0.8284	0.7550	0.7626	0.7893	0.6690	0.5601
0.1805	0.2812	0.3944	0.6532	0.7961	0.8771	0.8779	0.8177	0.7899	0.7707	0.7569	0.7904	0.7369	0.7515	0.6585	0.5531
0.1327	0.2167	0.3630	0.6353	0.8534	0.8876	0.8635	0.8779	0.8506	0.7563	0.7862	0.8020	0.7660	0.7246	0.6168	0.5439
0.1287	0.1310	0.2268	0.4779	0.8069	0.9706	0.9228	0.8904	0.8677	0.8580	0.8231	0.8154	0.8189	0.7641	0.6288	0.5374
0.1021	0.0974	0.1649	0.1827	0.4588	0.8588	0.9364	0.9508	0.9342	0.9157	0.9240	0.9136	0.8411	0.7936	0.6752	0.4777
0.0767	0.1309	0.1258	0	0	0.3791	0.8322	0.9564	1	0.9845	0.9504	0.9141	0.8690	0.8474	0.6623	0.3624
0.1135	0.1199	0.0483	0	0	0	0.2255	0.6325	0.8942	0.9502	0.9730	0.8861	0.8360	0.6185	0.3401	0.1574
0.0840	0.0954	0.0325	0	0	0	0	0	0.0960	0.2554	0.3393	0.2944	0.0885	0	0	0.0598
0.1009	0.0805	0.0186	0	0	0	0	0	0	0	0	0	0	0	0	0.0478
0.0565	0.0614	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0441
0.0497	0.0515	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0217
0.0452	0.0493	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0393
0.0351	0.0598	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0578

Table 1 Normalized Gray Scale Values

The physical length and precision of each grayscale image could be distinct in each CT scan. In the CAVE3 lab, the micro-CT can achieve to a micrometer per-voxel level in the 3D reconstruction object. Normally, there are 512-512-512 and 1024-1024-1024 as well as 2048-2048-2048 volumetric sizes. The voxel size is depended on the focus length of each CT scan. Namely, the higher volume size is, the more details can be presented in the volume. However, if we were only interested in a small portion of the object, we can zoom to that region to get a higher resolution. During the experiment, since there is a limitation on the object's size and memory, Therefore, it recommends using the isotropic reconstruction original volume such as 1024-1024-1024 or higher 2048-2048-2048 volume in the 3D reconstruction. For the 512-512-512 volume, truncation is applied, and for the 2048-2048-2048volume, extrapolation is applied.

3.3 Greyscale Image Histogram:

The histogram of gray scale values in the volume, which can be checked in the following Figure 7. The x-axis shows the grayscale value for the different portion of the object. For example, the higher value is, the denser part of the object is. In the graph, the wire bond region has the highest density so that the X-ray can barely penetrate and the gray scale value is the highest for that part. It can be viewed in the right part of the histogram. On the middle part, there is a silicon chip region, which has the second highest gray scale values. The air and fixture are the part that X-ray can easily pass through, therefore the gray value for this region is small. There is an significant portion of air and fixture in the volume. Thus, we can see the voxel counts and numbers for the air and fixture is huge.

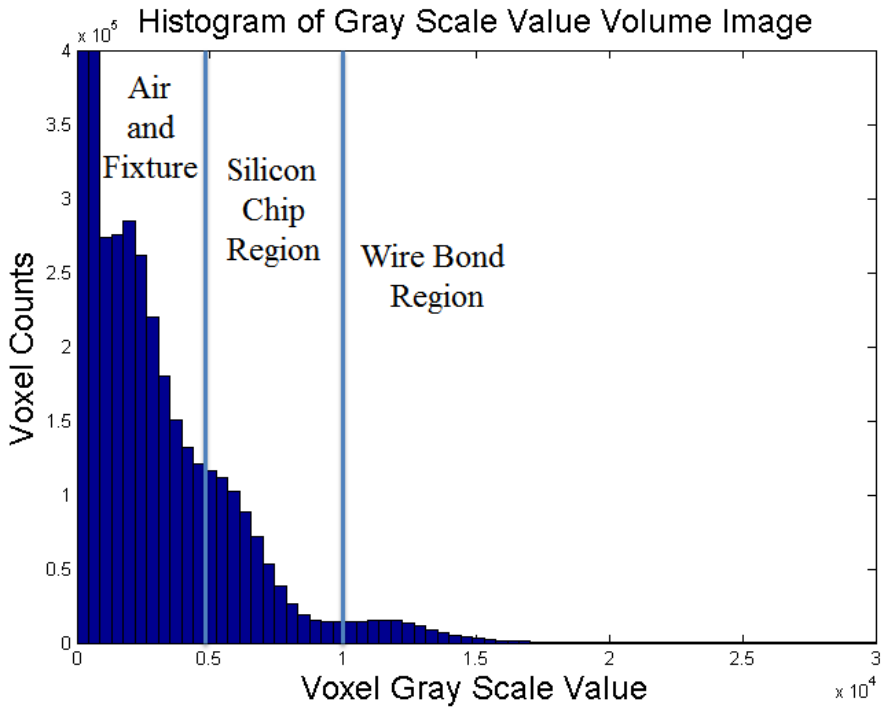


Figure 7 Histogram of Volumetric Image

The data structure for the Digital Volume contains several parts: (1) Original CT Scan Tiff Images. (2) Volume Graphics VGL 3D Reconstruction File. (3) DICOM exchangeable files. (4) MATLAB MAT data file. (5) Stereo-Lithography (STL) file. The original CT scan tiff images are the raw data generating by the CT Data Acquisition (DAQ) system. The CT detector has varied resolutions. There are 512 by 512 pixel tiff images, and 1024 by 1024 pixels Tiff images as well as 2048 by 2048 pixel tiff images. However, the standard and default resolution is 1024 by 1024. The Volume Graphics (VG) can load a two-dimensional tiff slice images to form a three-dimensional digital volume. Then, the Volume Graphics(VG) can export the 'VGL' file to the DICOM slice files, which is the medical standard exchange files. The MATLAB can read those DICOM slice files, and it can load the data into a 3D Matrix that is '[m, m, m]'. Therefore, we can perform the correlation directly using matrix manipulations and computations. Lastly, the MATLAB can write and output the STL files, which is a surface mesh file with numbers of facet. The general flowchart can be seen from the following Figure 8.

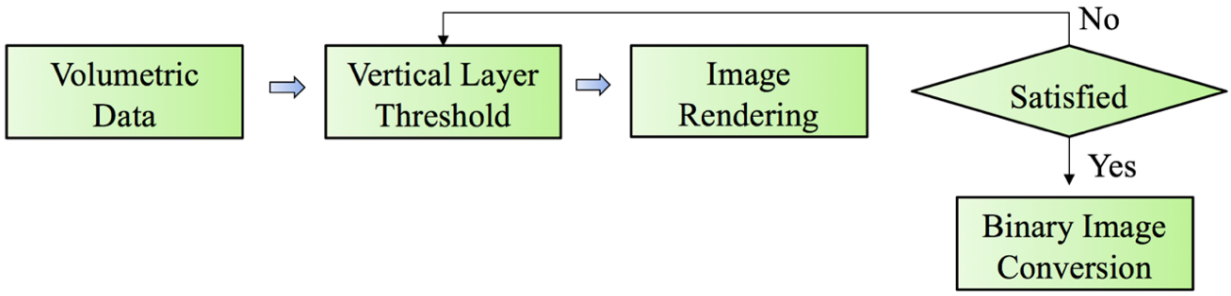
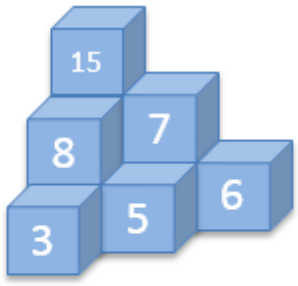


Figure 8 Volume Voxel Image

The voxel represents a numerical value in a space grid coordinate system, and each voxel should have its value representing the X-ray density property in the CT scan and the X-ray will decay as it passes through the solid object. The air is the lowest density in the volume, and usually,

the metal part has the highest density in the volume. Due to this property from acquisition system, the volume data shows that the interior part of the solid object has higher voxel value than the surface of the solid object. The following Figure 10 demonstrates the voxel values from the reconstruction volumes. From the figure, we can see that idea of voxel is a combination of "volume" and "pixel," and the pixel is a combination of "picture" and "element". The voxel shows both the density value and location in the volumes. In our research, the image data type is an unsigned 16 bit.



Voxels

$$unsigned\ 16\ bit: 0\ to\ 2^{16} - 1$$

Figure 9 the Voxel Representation in the 3D Image

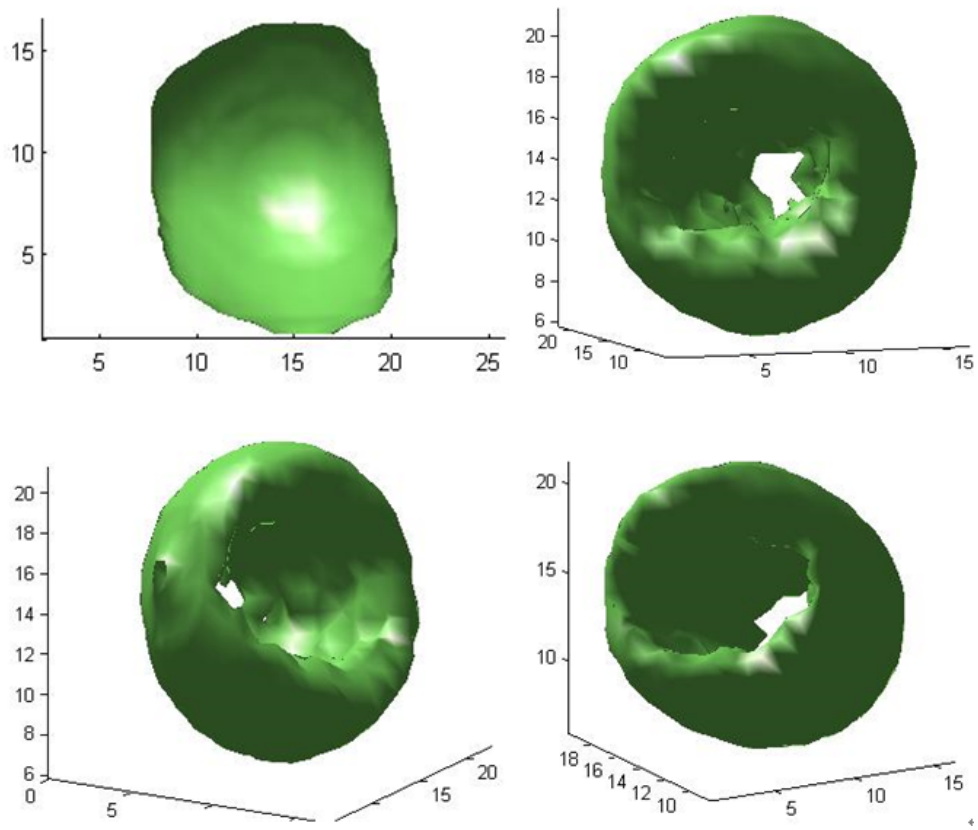


Figure 10 the Representations of Different ISO Layers in the Solder Balls

3.4 Global CT Coordinator System and Reconstruction Coordinator System:

The global CT Coordinate system shows in the following Figure 11. In the horizontal panel, there is XM axis and YM axis for X coordinator and Y coordinator respectively, and the CT tray is rotating along the X axis. Also, there is an offset degree to the vertical position. The ZD is the distance from the horizontal panel to the detector, and the ZT is the distance from the horizontal panel to the high-power target. The standard specification for one CT scan shows below in Table 2:

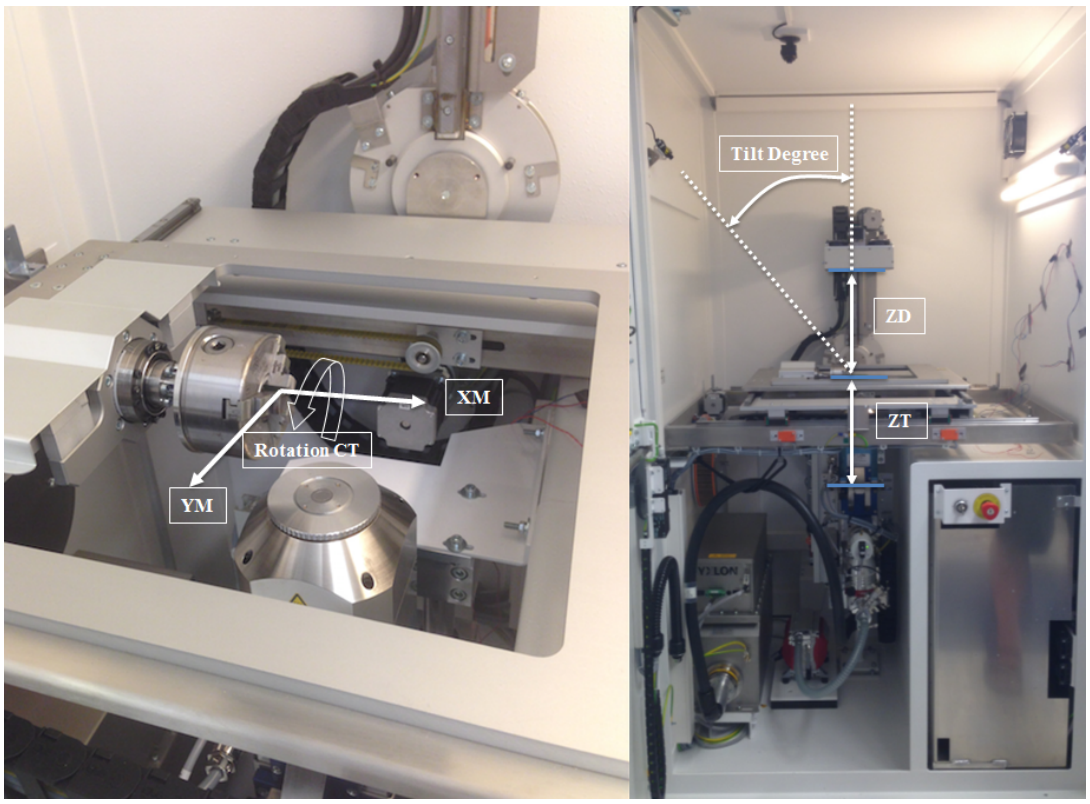


Figure 11 CT Axis and Scan Panel Configuration

Names	Values
XM	120300 um
YM	167288 um
ZT	-62490 um
ZD	295000 um
TiltD	0 m°
RotCT	358558 um°

Table 2 the X-ray CT scan Variables Setting

The Volume Graphics will read the scan specifications from the Global CT coordinate system for reconstructing the 3D volume, and the VG loads the projection images shown below Figure 12:

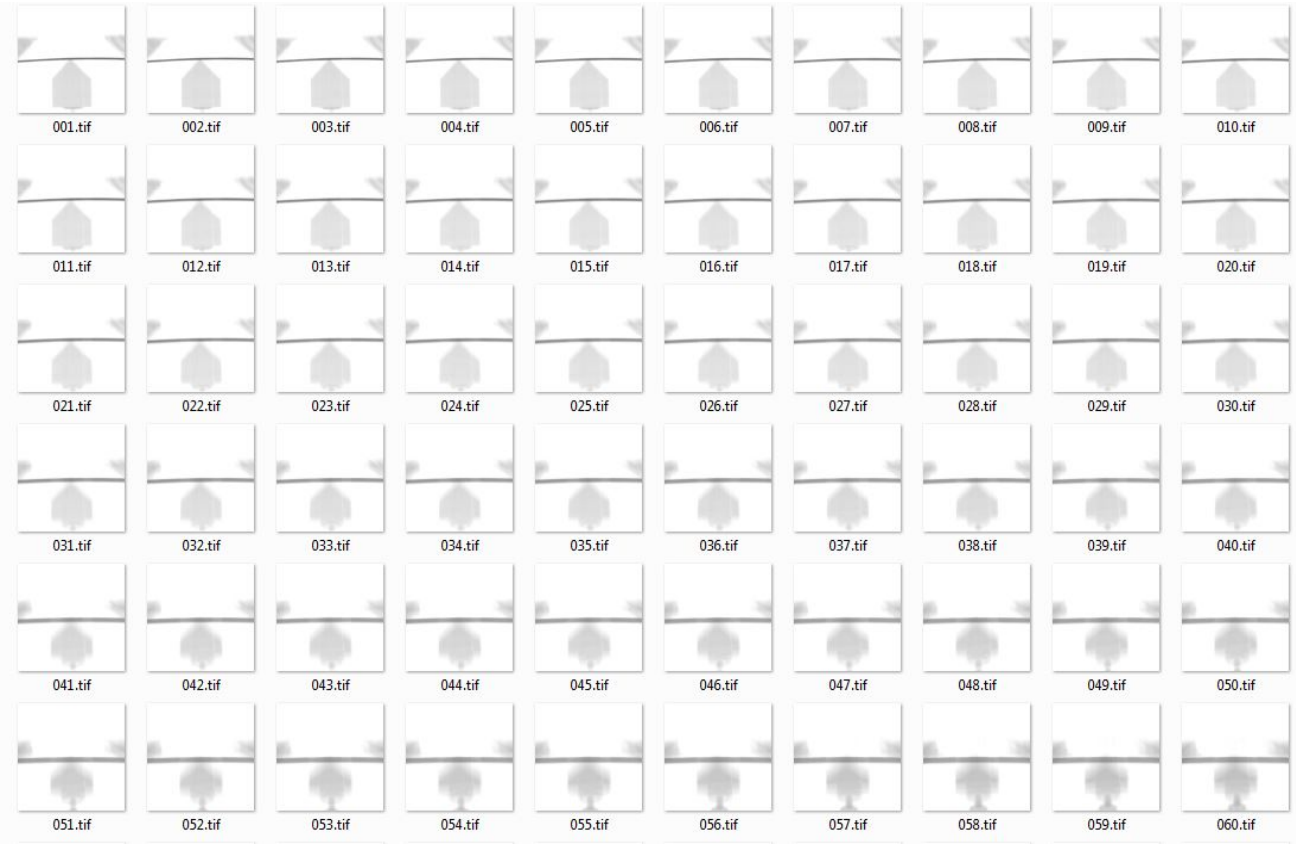


Figure 12 Example of CT scan Projections

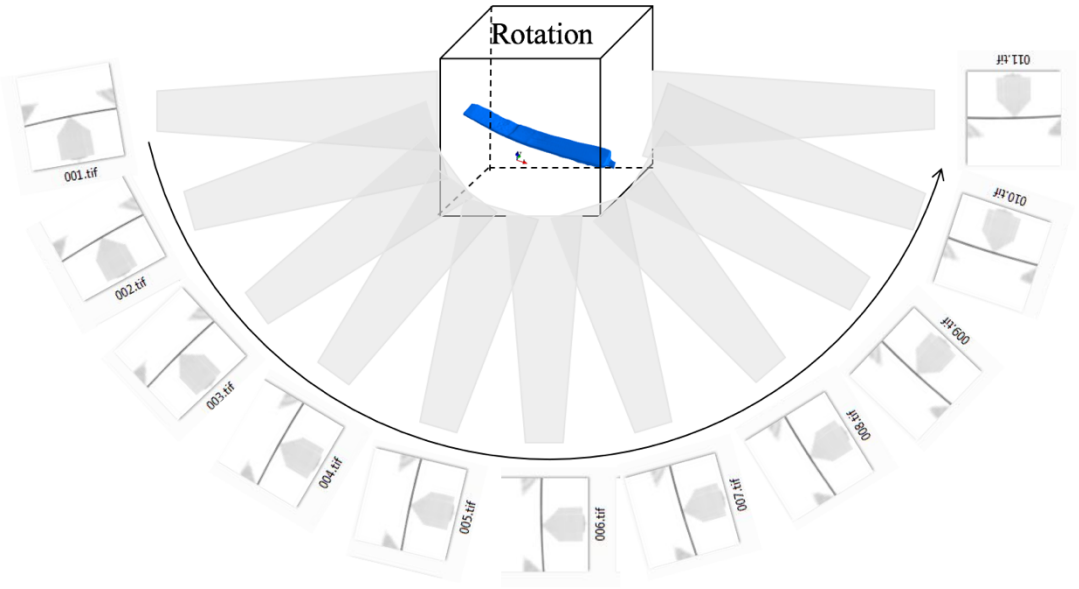


Figure 13 X-ray CT Projections and Rotations

The Volume Graphics reconstruction global coordinate shows in the following Figure 14. The big red box shows the global system, which has covered the whole object in the CT scan. The small yellow box shows the local system, which the region has been reduced from the global system. The DVC computation starts at left corner (i.e. (0,0,0)) that is the origin of the system.

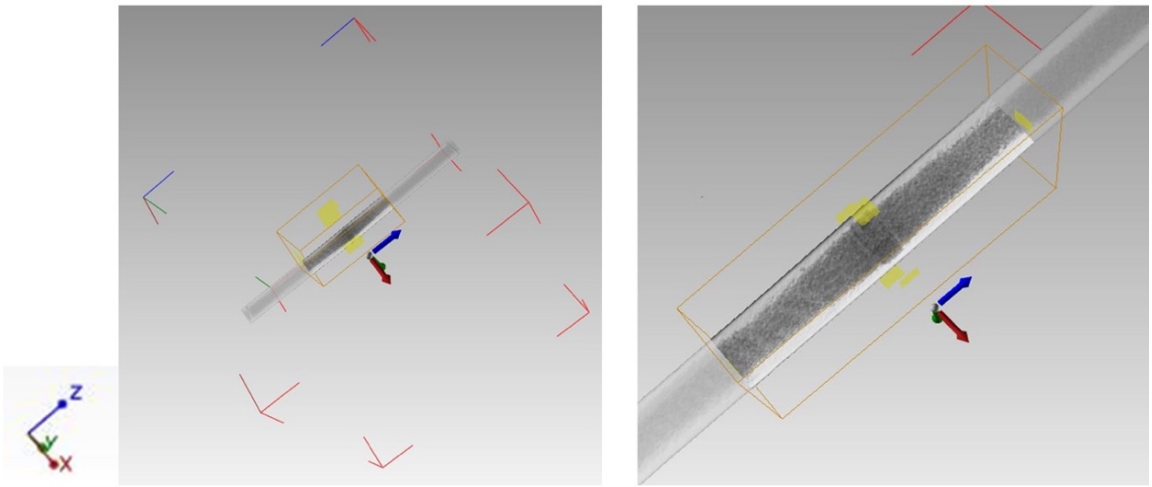


Figure 14 The Global Coordinate System and Location Coordinate System

The voxel sensitivity is defined as unit physical length per voxel. The voxel sensitivity is mutable according to each CT scan set up. Typically, if the X-ray Target distance (ZT) is small, a high voxel sensitivity could be achieved. However, if X-ray Detector distance (ZD) is small, so the voxel sensitivity is low in the scan. The following Table 3 lists the relationship of physical voxel size according to the CT target and detector distance:

Variables	**Scan 1**	**Scan 2**	**Scan 3**	**Scan 4**	**Scan 5**
XM (um)	80169	52913	103215	120300	131384
YM (um)	166257	165212	164717	167288	16757
ZT (um)	-200000	-131496	-126679	-62490	-33424
ZD (um)	102234	294999	294999	295000	294999
TiltD (m°)	0	0	0	0	0
RotCT (um°)	31443	358558	355525	358558	7199
Volume Size	1024	1024	1024	1024	1024
One Voxel Length(mm)	0.163	0.0383	0.0374	0.0224	0.0131

Table 3 Scan Detector Distance and Magnification

From above Table 3, we can see that when the X-ray target moving closer to the scanned object, the voxel sensitivity is low. Therefore, the smaller object the less distance between the X-ray tube and the scanned object. We have also quantitatively shown the relationship between the scan distance and voxel sensitivity according to various scans. In following Figure 15, we can design a suitable scan fixture according to the desired voxel sensitivity value.

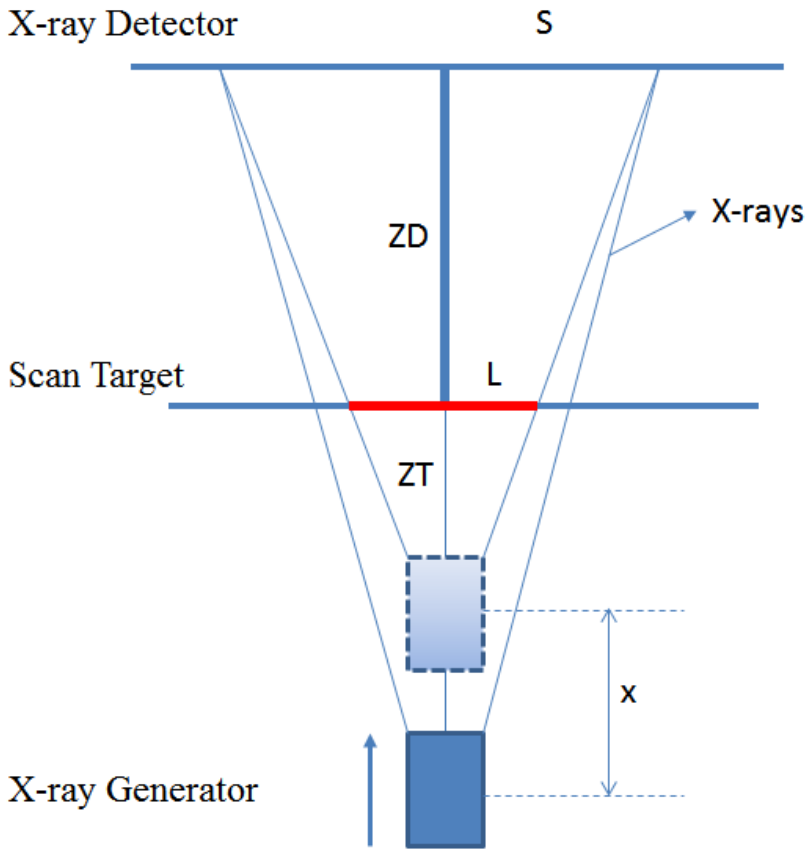


Figure 15 X-ray CT Scan Geometry and Magnification

$$\frac{\mathrm{ZT}}{(\mathrm{ZT}+\mathrm{ZD})}=\frac{\mathrm{L}}{\mathrm{S}}=\text{Voxel Sensitivity} \tag{1}$$

Where: ZT is the distance between the target and X-ray tube

ZD is the distance between the target and X-ray detector

The following Figure 16 shows the theoretical relationship between target distance and voxel sensitivity, and the red line is a linear fit, the green dot is from experiment data, and the blue line is a theoretical line.

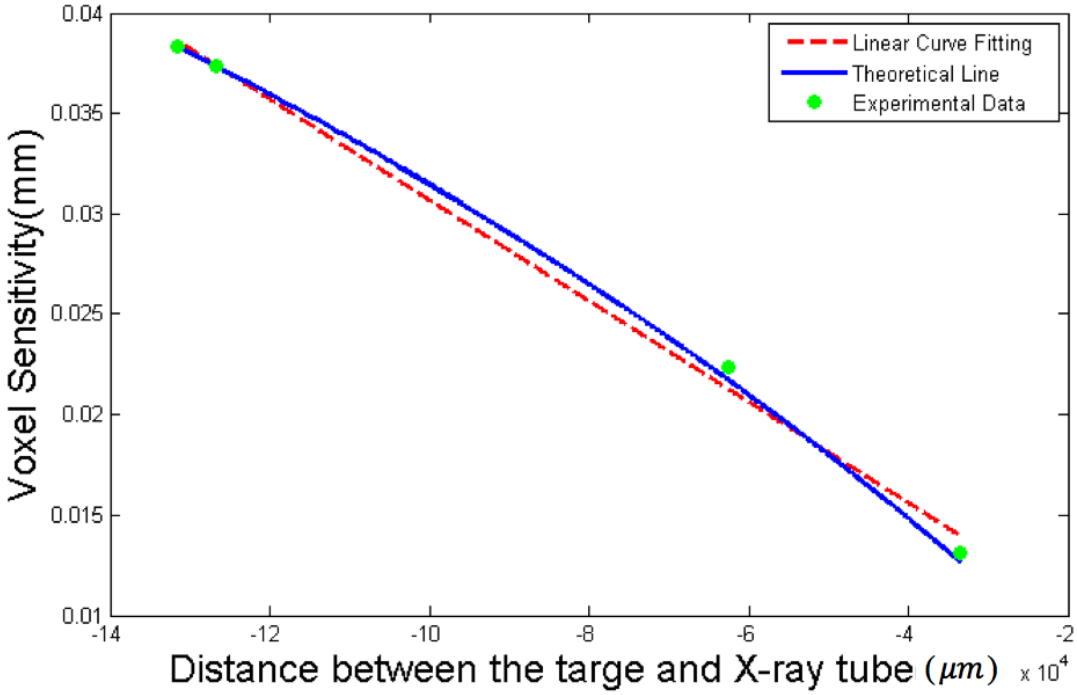


Figure 16 Voxel Sensitivity and Scan Distance

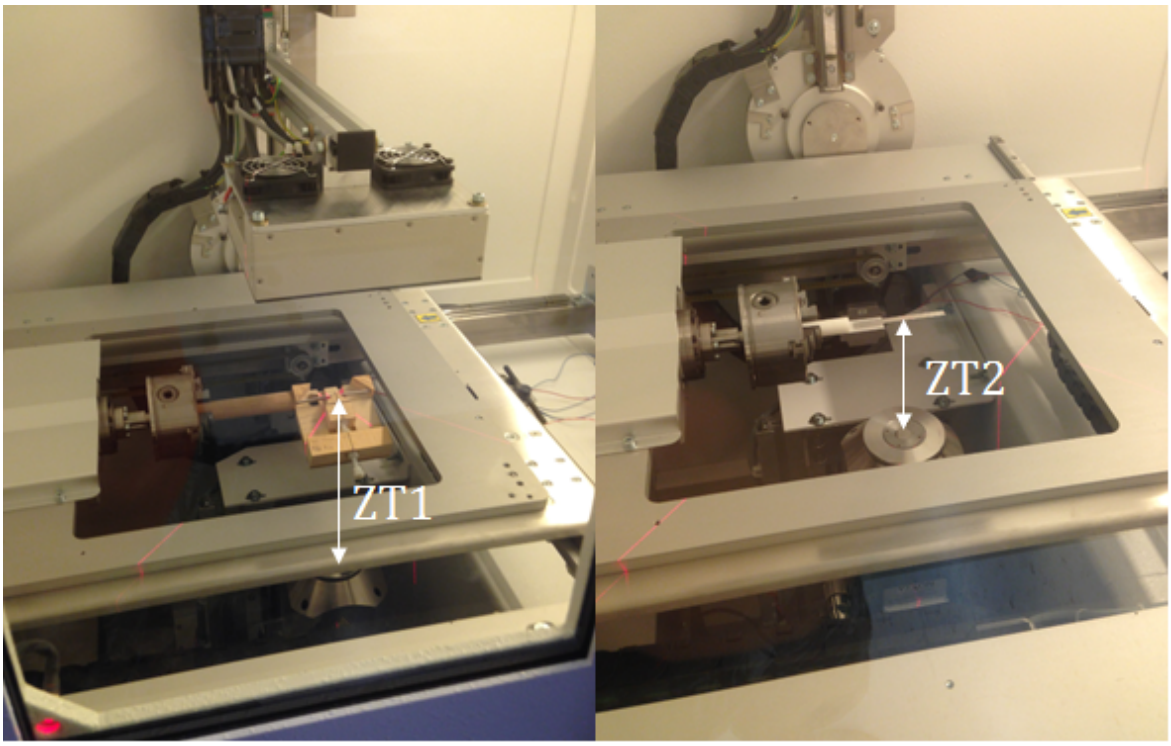


Figure 17 A Demonstration on Two Different Distances between the Target and X-ray Tube

3.5 Image Reconstruction and Failure Detection

3.5.1 Image 3D Reconstruction:

The industrial world is undergoing a transition of materials for the second level interconnects including the usage of lead-free solder. The complicated system may have a mix of components with different solder alloys in the printed circuit assemblies. In this chapter, an imaging-based methodology using the X-ray micro-CT has been proposed to exam the failure of electronics devices. Several failure mechanisms: (1) the solder ball lifting, (2) pillow defect, (3) solder cracker, and (4) solder voiding, have been shown in the 3D reconstruction image. The 3D reconstruction image provides an intuition from the geometry representation of electronics failure modes. The 3D reconstruction and 3D rendering method have been developed by non-commercialized MATLAB and PYTHON-FORTRAN code. The 3D visualization is based on the ISO-surface layer view, and the same density of the material has been grouped as a identical layer in the contour.

The concept of 3D reconstruction can be briefly expressed as following Figure 18: First of all, we convert each projection image into a frequency domain image using Fast Fourier Transform(FFT) (Francesco & Silva, 2003). Slice by slice, we can populate the 3D FFT frequency volume. Finally, we perform an inverse FFT to get back the spatial domain volumetric image.

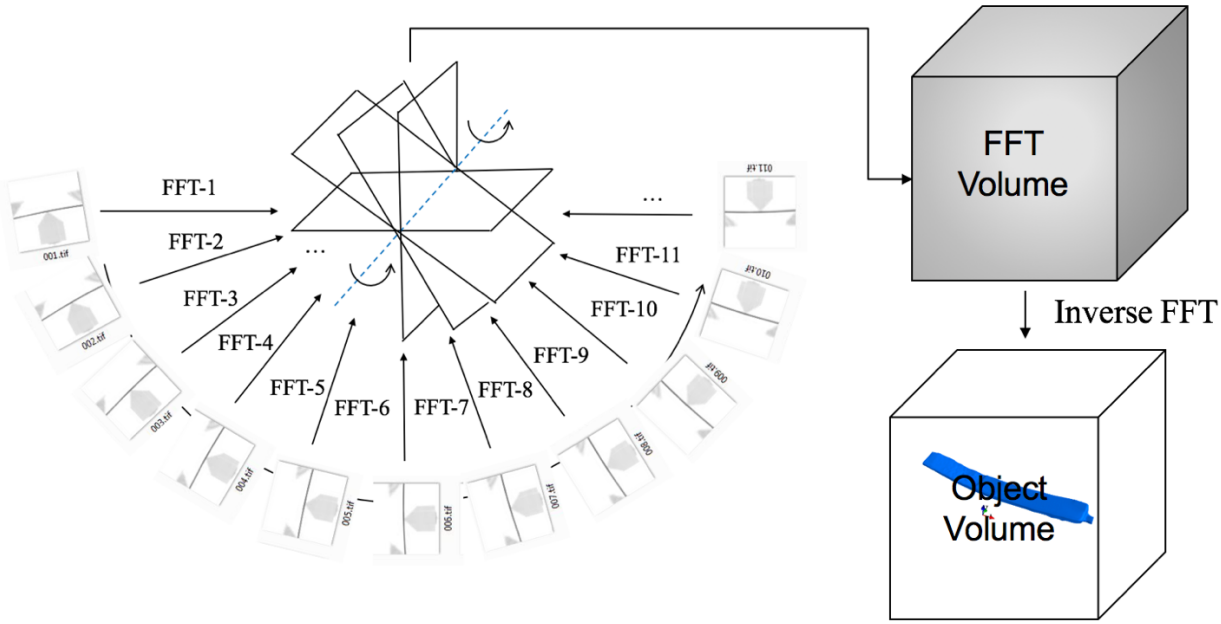


Figure 18 Illustration of X-ray CT 3D Reconstruction

Figure 19 3D Reconstruction of Solder Crack

The following 3D reconstruction Figure 20 shows a full solder crack created by mechanical bending, and the crack width is about 20-30 microns. This solder neck region has been separated by the crack. The Figure 21 below also shows the corner model view. We can clearly see the distribution of crack geometry and the crack size.

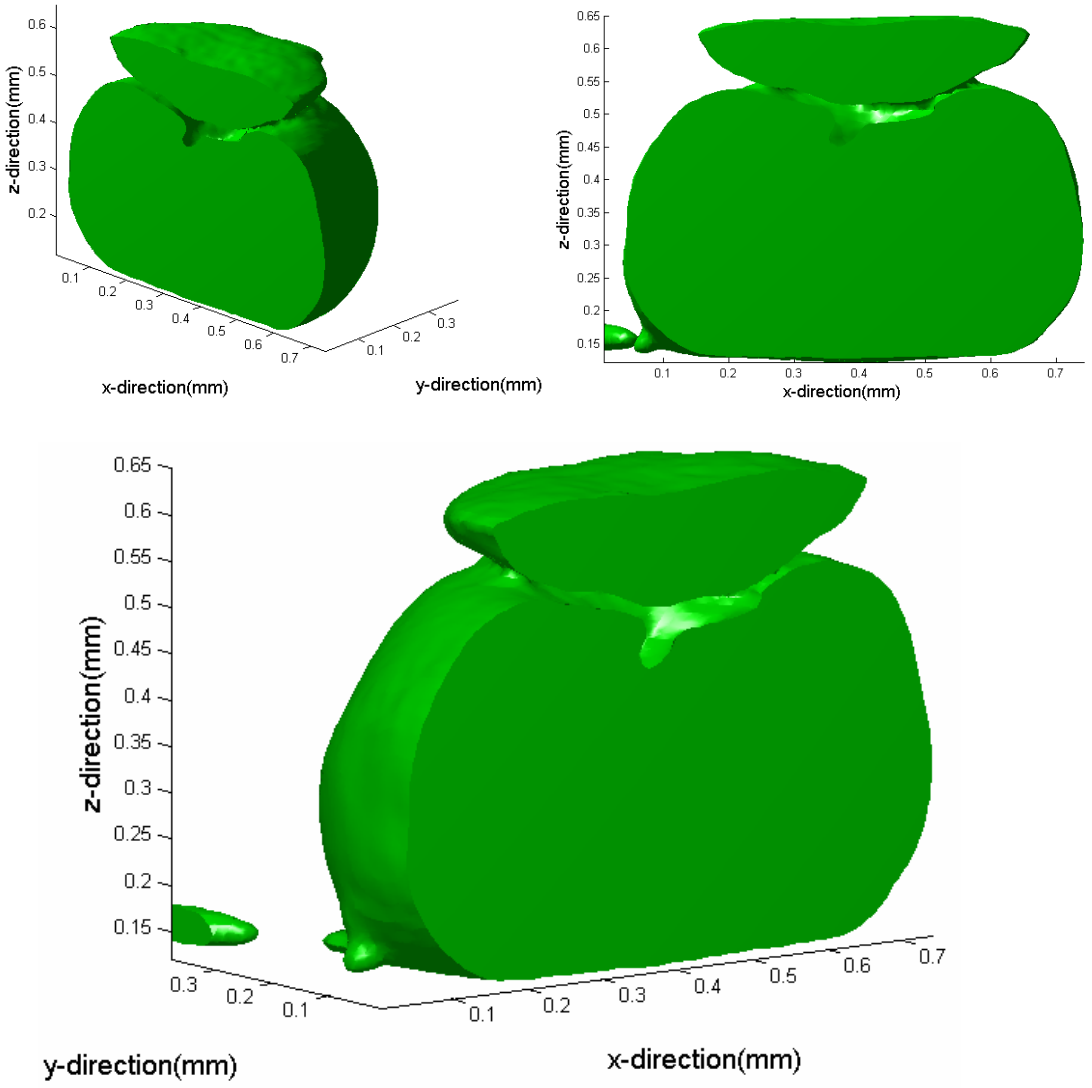


Figure 20 Mechanical Bending Crack on Solder Joint

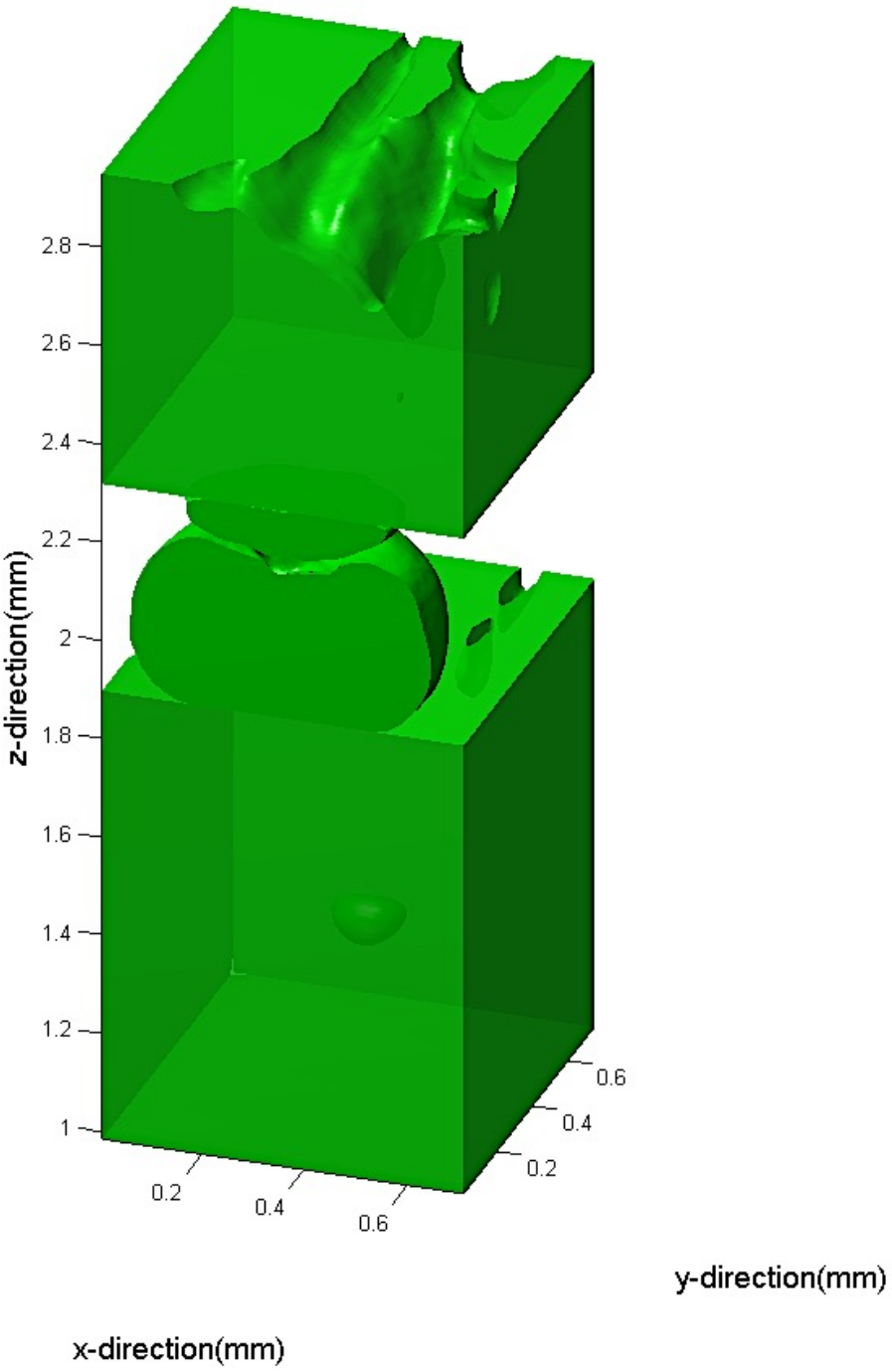


Figure 21 Mechanical Bending Crack on Corner Model View

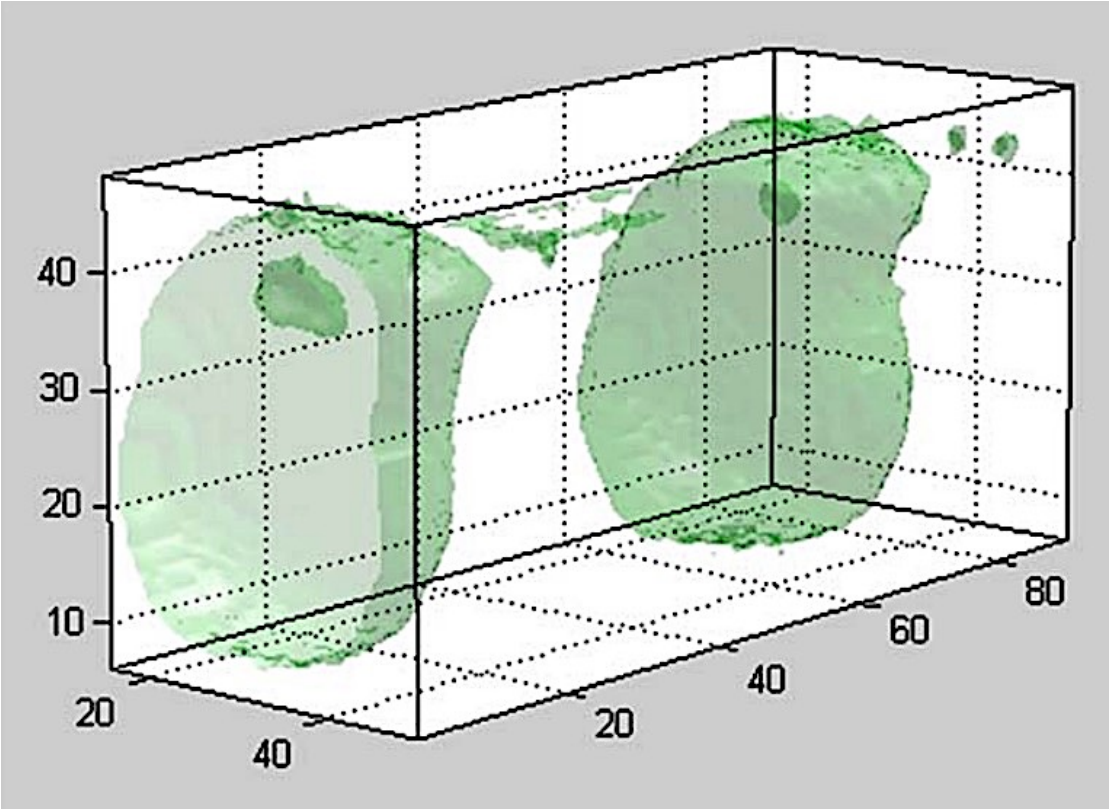


Figure 22 Solder Voids

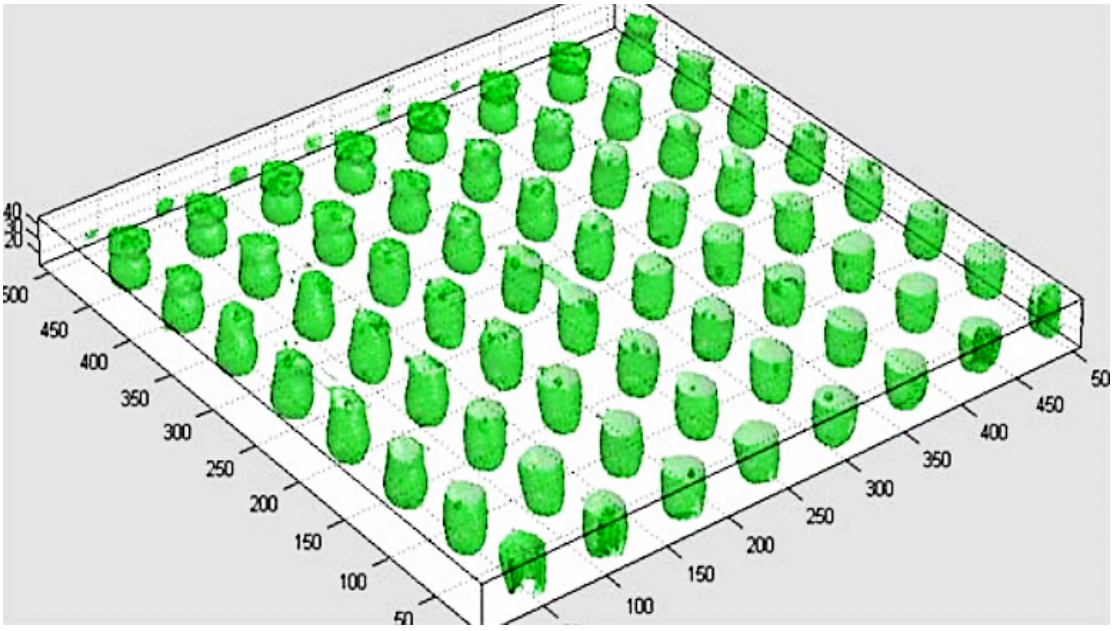


Figure 23 Voids in the Interconnects

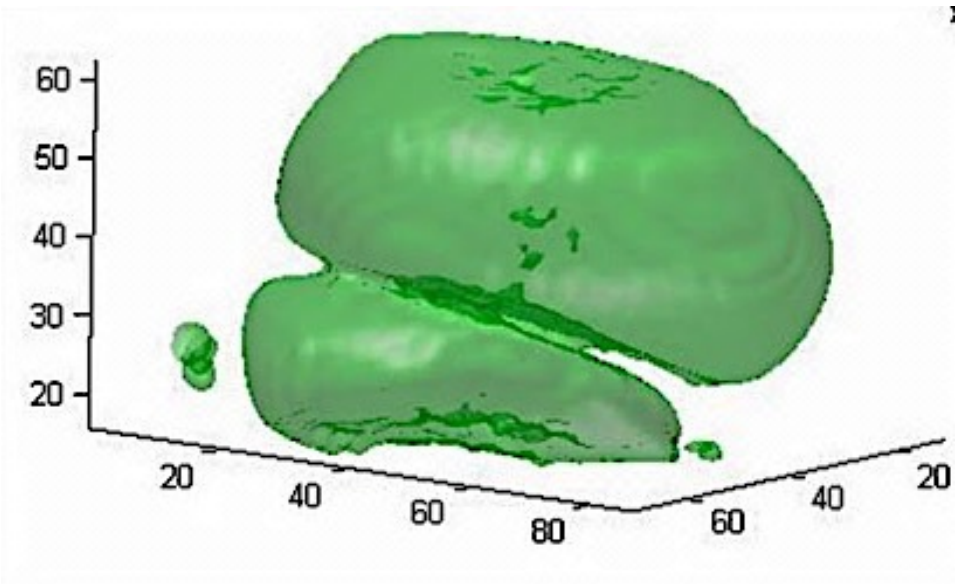


Figure 24 Solder Pillow Crack

Figure 25 3D Solder Ball Lifting

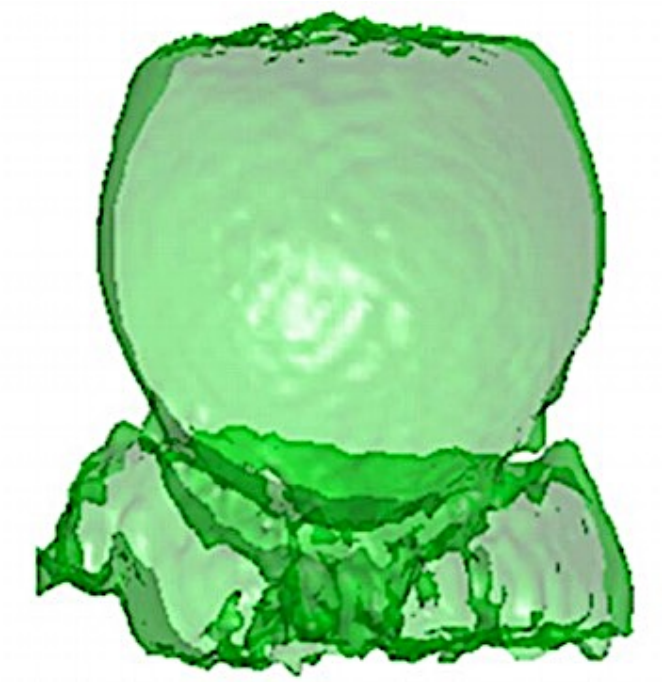


Figure 26 2D Solder Ball Lifting

3.5.2 Scan Setup and Specifications:

The following Table 4 shows a standard scan for silicone chip with a high magnification of X-ray CT scan, and the resolution reaches about 1.5 microns. There are 2800 projects, from which 512-512-512 volume has been reconstructed.

Specifications	Value
Scan Projections	2800 Projects
Reconstructed Volume Size	512^3/1024^3/2048^3 Voxels
Region Rendering Size	512^3 Voxels
Voxel Size (X,Y,Z)	(2,2,2)/(1,1,1) um
Scan Property	Quick/Quality Scan
Result Physical Size(X-Y-Z)	1400-600-600 um
Digital Data Type	Unsigned 16 bit

Table 4 Reconstruction Table

In the scan, the small difference of scan configuration has large effects on the quality of reconstruction. The following Figure 27 shows the detail of CT fixture and the position sample has been placed.

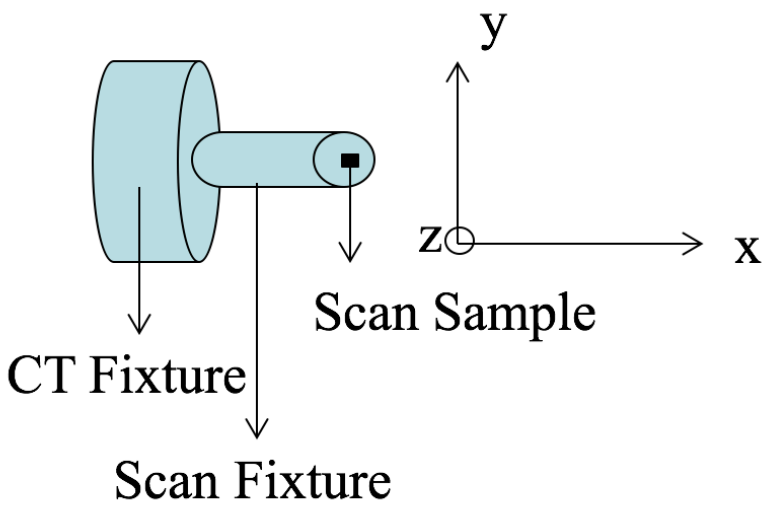


Figure 27 X-ray Micro CT scan system

The following Figure 28 shows standard three configurations for tiny scan object:(1) horizontal X setup, (2) horizontal Y setup, (3) vertical Z setup. Generally speaking, the horizontal X setup and horizontal Y setup will generate a clear and high-resolution view of the solder joints. However, the quality of this configuration is not always confirmed for the silicon chip scan. From our experiment and experience, the vertical Z setup has a better view of the silicone chip and any square-like regions.

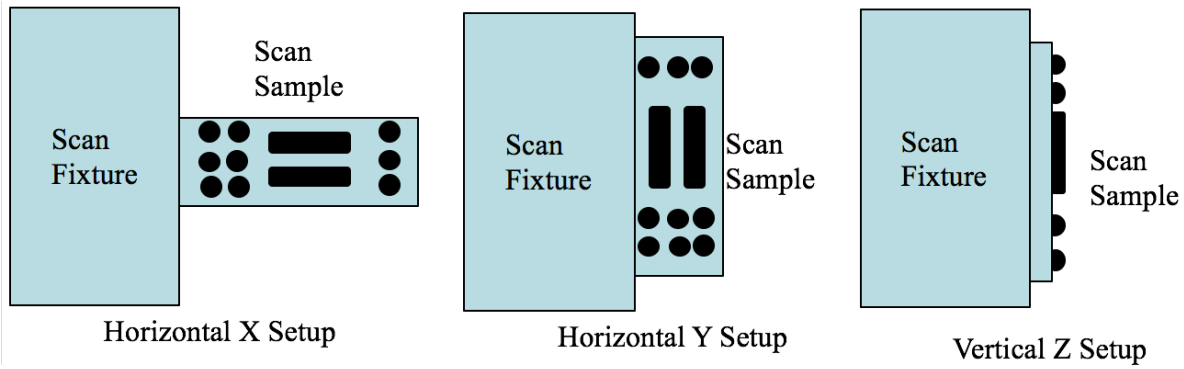


Figure 28 Three Types of Configurations for CT scan

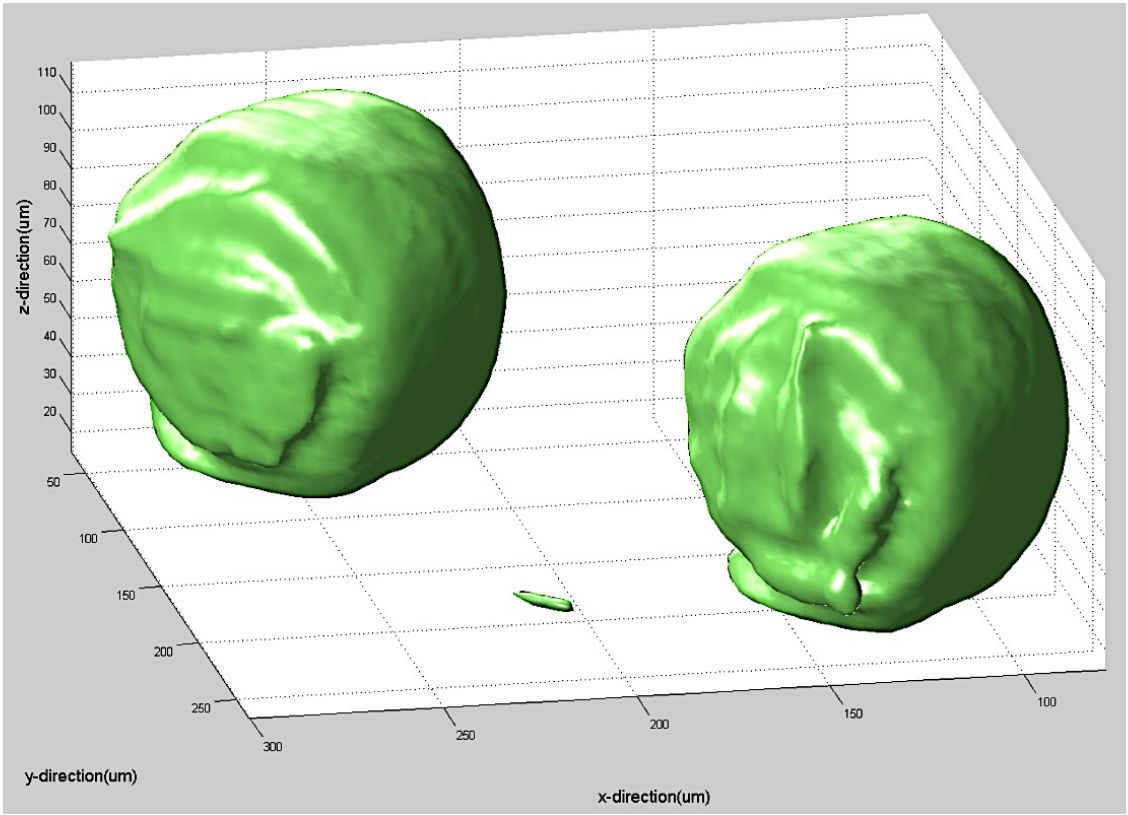


Figure 29 Reconstruction Tiny Solder Bumper on the Silicone Chip

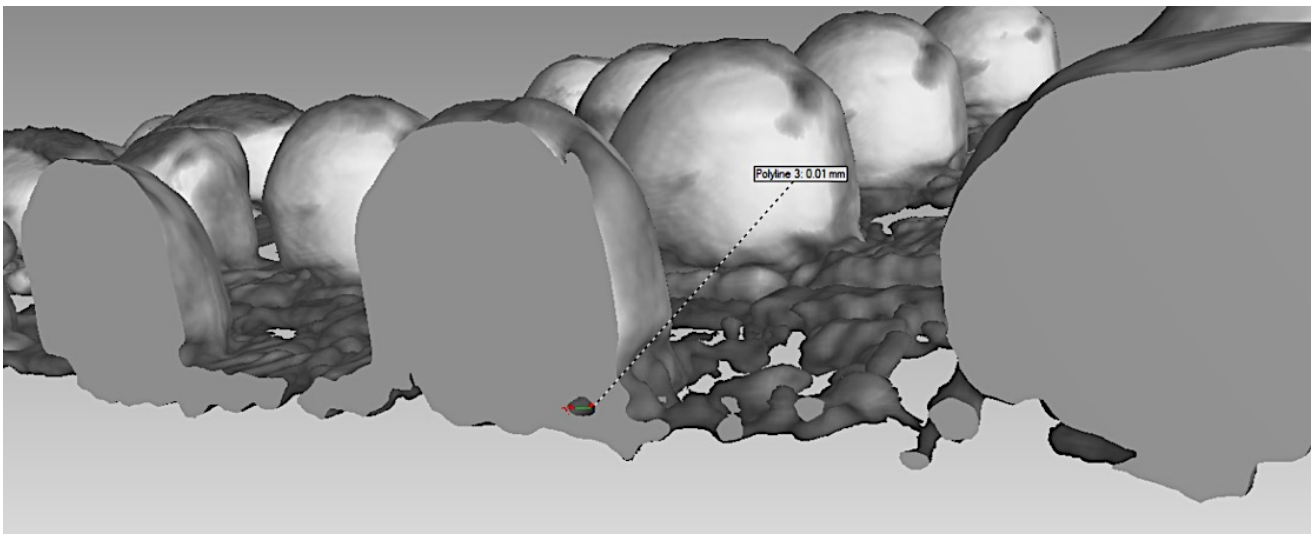


Figure 30 Voids in the Solder Bumper

Chapter 4 Image Correlation Technique and Algorithm Development

4.1 3D Digital Image Denoising

The 3D image denoising has a long history, and it plays a very import role in the field of digital image processing (Motwani & Gadiya, 2012). The evolution of image processing starts from a spatial image processing to a frequency image processing, using the wavelet basis and Fourier transform to a complex domain. Recently, a standard statistical method has merged into the existing methods such as Hidden Markov Chain(HMC) and Bayesian method. In the spatial domain method, the noise reduction process applies a linear convolution, in which each voxel would be managed individually and used repeatedly. The wavelet method utilizes the image transformation in the frequency domain to obtain a frequency representation of the image, and the arithmetic convolution operator can be simplified by using vector product in the frequency domain. From the convolution theorem (Yitzhak, 1976) (Eric, 1976), under a suitable circumstance, the Fourier Transform of a convolution is a point-wise production in the Fourier domain. The high-frequency noise will be eliminated using the low pass filter with selected cut-off frequency. The following Figure 31 shows an example of a denoised image for the LED chip in 3D reconstruction representation.

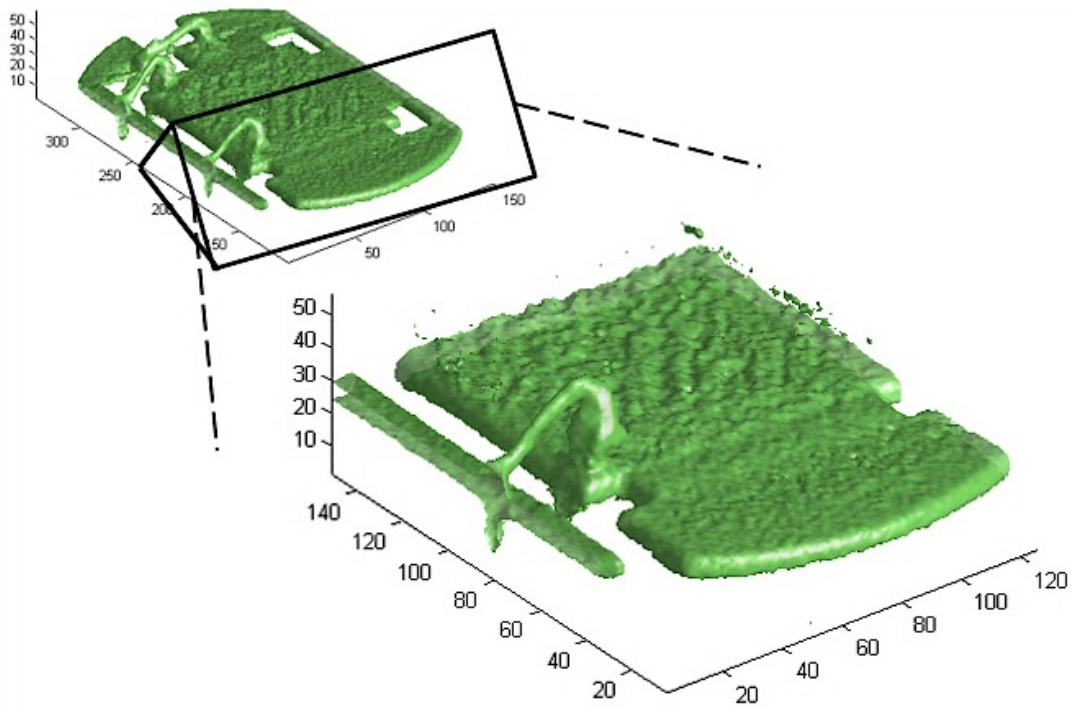


Figure 31 Noise Image of LED Silicon Chip Pad

The framework of image denoising shows in the following flowchart Figure 32: the kernel has to be initialized in the beginning. Then, we have to choose a suitable parameter for the kernel, and normally the selection of parameter can affect the result of image reduction. For a large size of the image, we often adopt the frequency domain method, which the algorithm complexity is reduced to the N∗log(N). However, after processed a image in the frequency domain, we have to transfer the frequency domain representation back to a spatial domain, this transform could cause the losing information if we do not follow Nyquist Theorem properly (Harry, 1928).

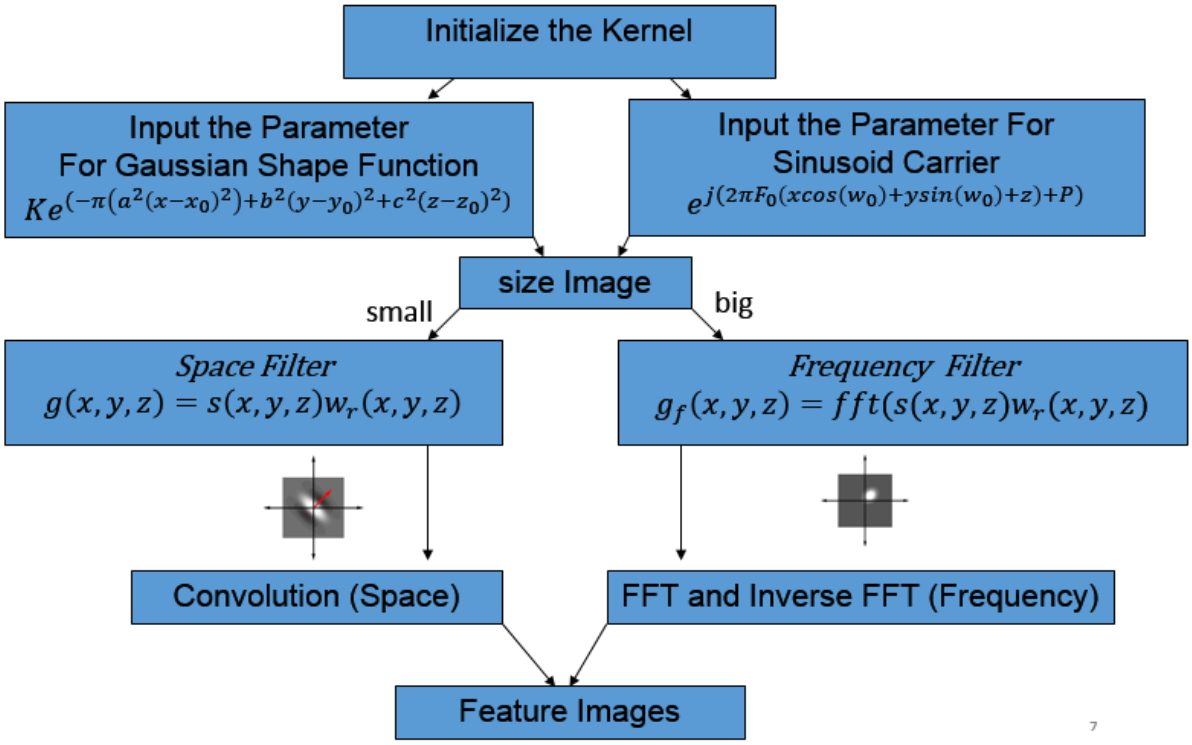


Figure 32 The Digital Image Processing Framework

The following Figure 33 shows the frequency representation of an image, the red rings in the frequency image show the cut-off frequencies. The high frequency outside of the ring should be eliminated.

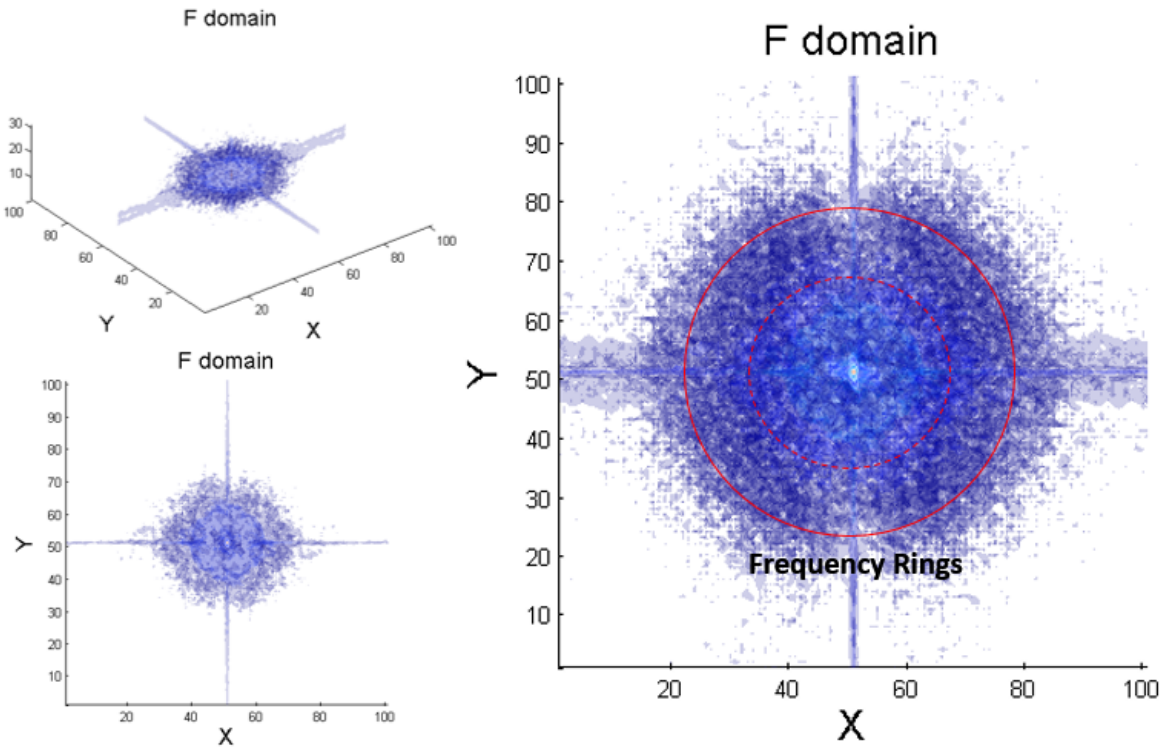


Figure 33 FFT Transformation of the Image

4.2 Gaussian Kernel in Image Denoising

In the image processing algorithm, we introduce the Gaussian Kernel for noise reduction is necessary. The Gaussian Low Pass Filter can be written as following equation: there are three parameters in the equation, a, b and c. Each parameter controls a variance in each direction of the coordinator. The large variance gives the more smoothing in that direction.

$$F(x, y, z) = e^{(-a(x-u)^2-b(y-v)^2-c(z-w)^2)} \qquad (2)$$

(a, b, c): inverse variances in each direction
(u, v, w): the Gaussian mean and location of kernal

The following Figure 34 shows the example of 'Gaussian Kernel' for different dimensions: the Gaussian Kernel in the first dimension is a bell curve, which is exact one-dimensional Gaussian

distribution. Similarly, the Gaussian Kernel in the second dimension is a multivariate Gaussian distribution. In the three-dimensional Gaussian Kernel is the ellipsoid-like shape.

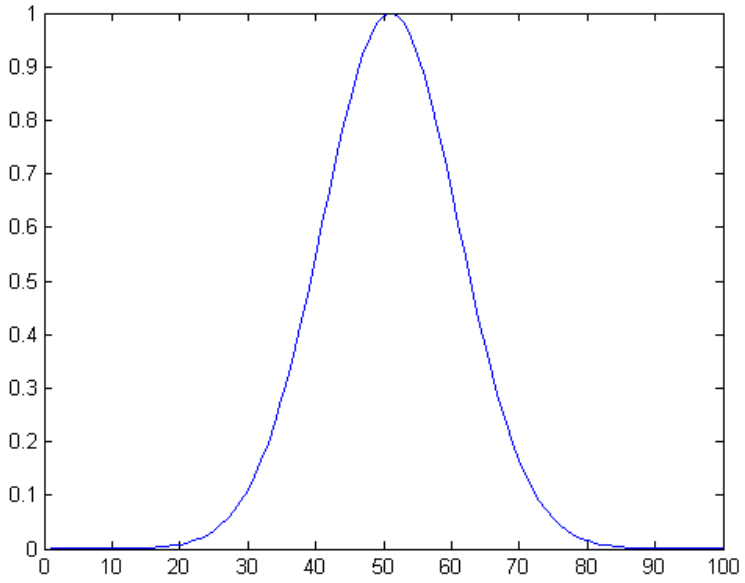


Figure 34 One Dimensional Gaussian Kernel

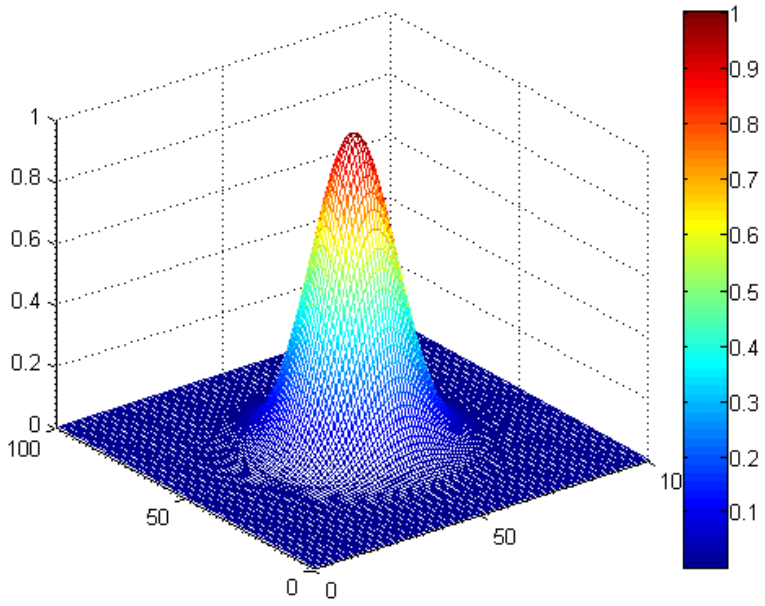


Figure 35 Two Dimensional Gaussian Kernel

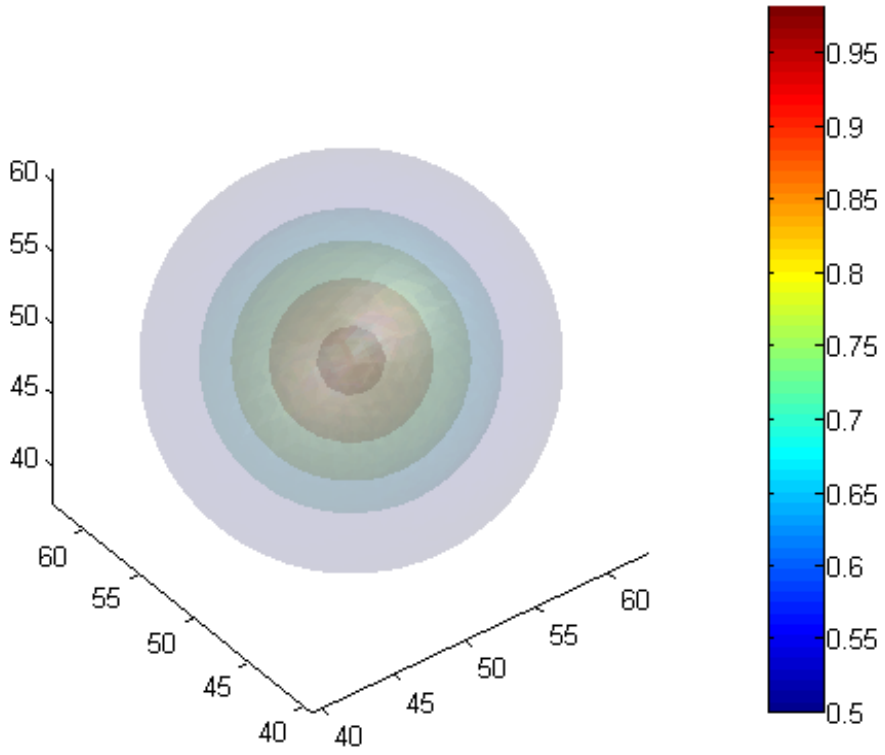


Figure 36 Three Dimensional Gaussian Kernel

The following Figure 37 shows the image of noise reduction using a different parameter of variance. We can see that the large variance in the Gaussian distribution leads to a significant smoothing in the 3D image. So, in the image smoothing and image denoising, we need to find an optimal value of variance to avoid over-smoothing in the experiment.

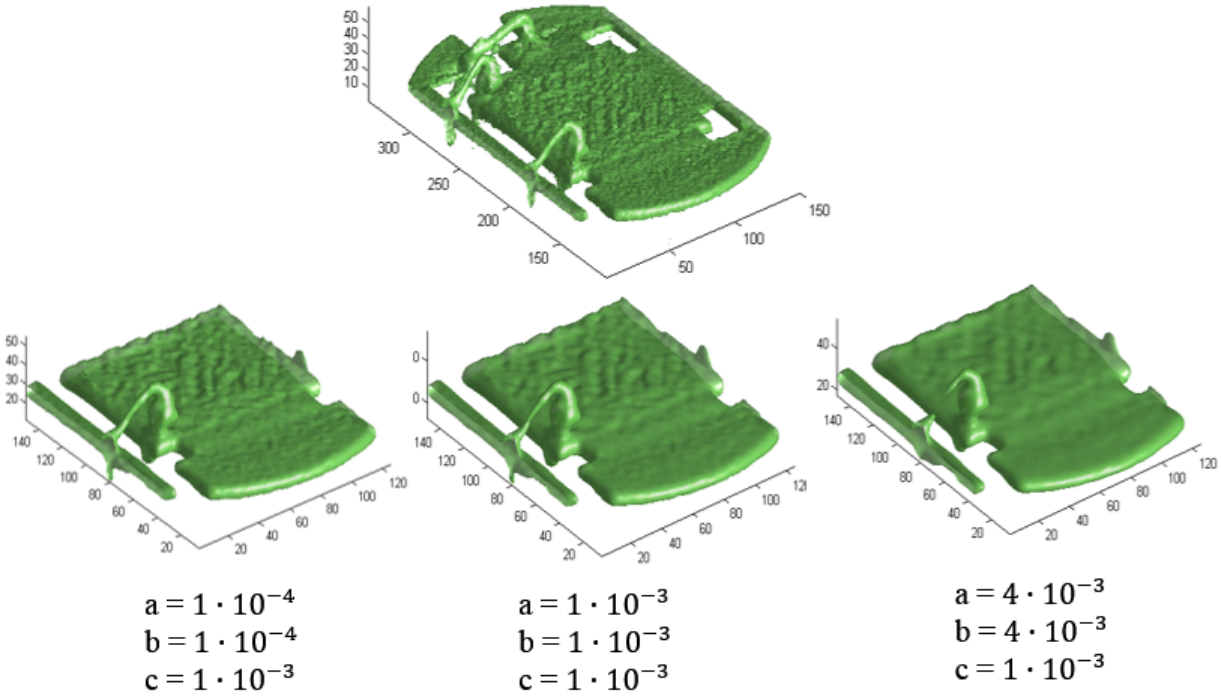


Figure 37 Gaussian Low Pass Filter Smoothing with Different Parameters

4.3 3D Digital Image Segmentation

An image segmentation algorithm is required to get the target from a raw volume of the image. There are many sophisticated algorithms to do the segmentation (Liu & Soran, 2012) (Grady, 2006). In general, there are two categories of the algorithm. The first is a human-aid interactive method, and the second is an automatic method. In this research, we applied an automatic method using statistical analysis of voxel distribution. The goal is to retrieve the threshold for separating the target in foreground from noise background. One the of famous method is called Otsu's method (Otsu, 1979), which automatically finds a threshold, and transforms the grey-scale level image into a binary level image. The basic concept of Otsu's method is to maximize the variance between the groups or to minimize the variance within the group in the histogram equivalently. It intensively searches for the optimal numerical value by

looping the histogram index from the minimum to the maximum, and at each index we calculate the group variance, then we compare those values to find the maximum with respect to the index. The optimal index will be utilized as the threshold value in the histogram for the segmentation. The mathematic foundation has been shown as following equation:

$$\sigma_b^2 = \sigma_t^2 - \sigma_w^2(t) = w_1(t)w_2(t))[\mu_1(t) - \mu_2(t)]^2 \quad (3)$$

$$w_i(t) = \sum_0^t p(i) \quad (4)$$

$$u_i(t) = \left[\sum_0^t p(i)x(i)\right]/w_i \quad (5)$$

σ_b : between group variance

σ_t : within group variance at index t

w_i: cumulative probability at index t

$u_i(t)$: group mean at index t

After finding the optimal greyscale threshold, we need to extract the ISO-surface using Marching Cubes (MC) algorithm in a three-dimensional discrete scalar field and to convert the image into a binary 'zero and one' image (LEWINER & LOPES, 2003). Lastly, we use image morphological operation to calculate the Euclidean Distance (ED) of binary image, and this will generate a distance form of an image for the mesh generator. The distance transformation of image can be viewed as iterative morphological erosion operation on the boundary of input binary image, and in each iteration of voxel erosion, the distance increases by one unit.

4.4 Digital Image Registration

The pre-registration is required when there is a significant body translation and rotation (Silva, 2014). Several popular image registration methods are introduced in this study to compensate the rigid body motion: (1) Principal Component Analysis (PCA) registration. (2) Singular Value Decomposition (SVD) registration. (3) Iterative Closest Points (ICP) registration. The Principal Component Analysis registration is the fundamental method for rough alignment in the 3D image, and it is unsupervised learning method, which needs less information input. The Singular Value Decomposition registration requires pre-computation of paired points between two groups. The Iterative Closest Point registration is a recursive form of SVD method, and the paired feature points are calculated and updated at each iteration. We will look at them individually in the rest of this chapter.

The geometrical transformation contains translation and rotation (Smith & Bay, 2002). In the 2D world, the translation can be written as the following matrix form:

$$\begin{bmatrix} P_x \\ P_y \end{bmatrix} = \begin{bmatrix} V_x \\ V_y \end{bmatrix} + \begin{bmatrix} P_{x1} \\ P_{y1} \end{bmatrix} \tag{6}$$

P_i: the voxel position at i

V_i: the voxel position at i

P_{x1}: the translation vector component in the x

P_{y1}: the translation vector component in the y

In the 3D world, each point has depth information:

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} + \begin{bmatrix} P_{x1} \\ P_{y1} \\ P_{z1} \end{bmatrix} \tag{7}$$

The rotation can be represented as matrix form:

$$\begin{bmatrix} P_x \\ P_y \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} P_{x1} \\ P_{y1} \end{bmatrix} \quad (8)$$

α: rotation angle in the coordinator

In the 3D world, the rotation vector is based on several axes: (1) the rotation around x-axis. (2) the rotation around y-axis. (3) the rotation around z-axis.

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \quad (9)$$

$$R_y = \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \quad (10)$$

$$R_z = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (11)$$

$R_x(\alpha)$: rotation along x axis

$R_y(\beta)$: rotation along y axis

$R_z(\gamma)$: rotation along z axis

The total rotation R can contain yaw, pitch and roll angles respectively (FAA, 2004):

$$R = R_z(\gamma)R_y(\beta)R_x(\alpha) \quad (12)$$

If we combine the rotation and translation, the rigid body motion equation can be written as the following equation:

$$P' = R(P - Q) + Q \quad (13)$$

P' is new location for the point

R is total rotation vector

Q is translation vector

4.4.1 Principal Component Analysis

The Principal Component Analysis(PCA) is one of fundamental statistic method to find the principal information between the multivariate data using the correlation matrix or covariance matrix. In the PCA registration, the major axis of an image could be aligned using the statistical PCA analysis. In the data, each point is three-dimensional data that includes x, y and z coordinates. Compared with traditional data, the PCA uses all the dimensional information rather than reduced dimension information checked by the scree plot. The limitation of the PCA is that the axes can be unreliable for circular objects, which eigenvalues become similar and eigenvectors become unstable.

In the PCA analysis, first of all, the whole dataset must contain 3-dimensional samples, and the mean vector for each dimension of data. Secondly, the scatter matrix, the correlation matrix or covariance matrix will be computed. Generally speaking, the accuracy of applying covariance matrix will be higher than the covariance matrix. This is because the large difference in the dimensional data will generate a bias in analysis. Then, the eigenvectors and eigenvalues will be computed. In the end, to align the two groups, the reference points will be transformed using n-k eigenvector matrix. The following vector shows three-dimensional data vector:

$$P_i = [x_i \quad y_i \quad z_i] \tag{14}$$

$$Q_i = [x_i \quad y_i \quad z_i] \tag{15}$$

x_i: column vector of points′ x coordinator

y_i: column vector of points′ y coordinator

z_i: column vector of points′ z coordinator

P_i: reference point column major vector

Q_i: target point column major vector

The covariance can be expressed as a dot product matrix computation:

$$\sigma_{PQ}^2 = \frac{1}{n} PQ^T \tag{16}$$

Applying the Singular Value Decomposition (SVD), the orthogonal vector and principal values can be computed:

$$X = U\Sigma V^T \tag{17}$$

U, V: orthogonal basis matrix

Σ: singular values diagonal matrix

The the PCA transformation is showing as the following equations, and the 'm' is the dimensionality of the dataset, and 'n' is the number of points (e.g. In the 3D, m is 3). A left side is a matrix form combining aspects of m dimensions. The right side shows transformed each point representation in the given vector space:

$$\begin{bmatrix} - & m & - \\ n & & \\ - & & \end{bmatrix} \begin{bmatrix} | \\ m \\ | \end{bmatrix} = (\text{positive number}) \begin{bmatrix} | \\ n \\ | \end{bmatrix} \tag{18}$$

The complete form can be written as the following:

$$\begin{bmatrix} - & m & - \\ n & & \\ - & & \end{bmatrix} \begin{bmatrix} - & m & - \\ m & & \\ - & & \end{bmatrix} = \begin{bmatrix} - & n & - \\ n & & \\ - & & \end{bmatrix} \begin{bmatrix} - & m & - \\ n & \text{diag} & \\ - & & \end{bmatrix} \tag{19}$$

The transformation between two sets applies the following matrix calculations:

$$x_A = E_A E_B^T (x_B - u_B) + u_A \tag{20}$$

x_A is the point in the A set

x_B is the point in the B set

u_B is the center of B set

u_A is the center of A set

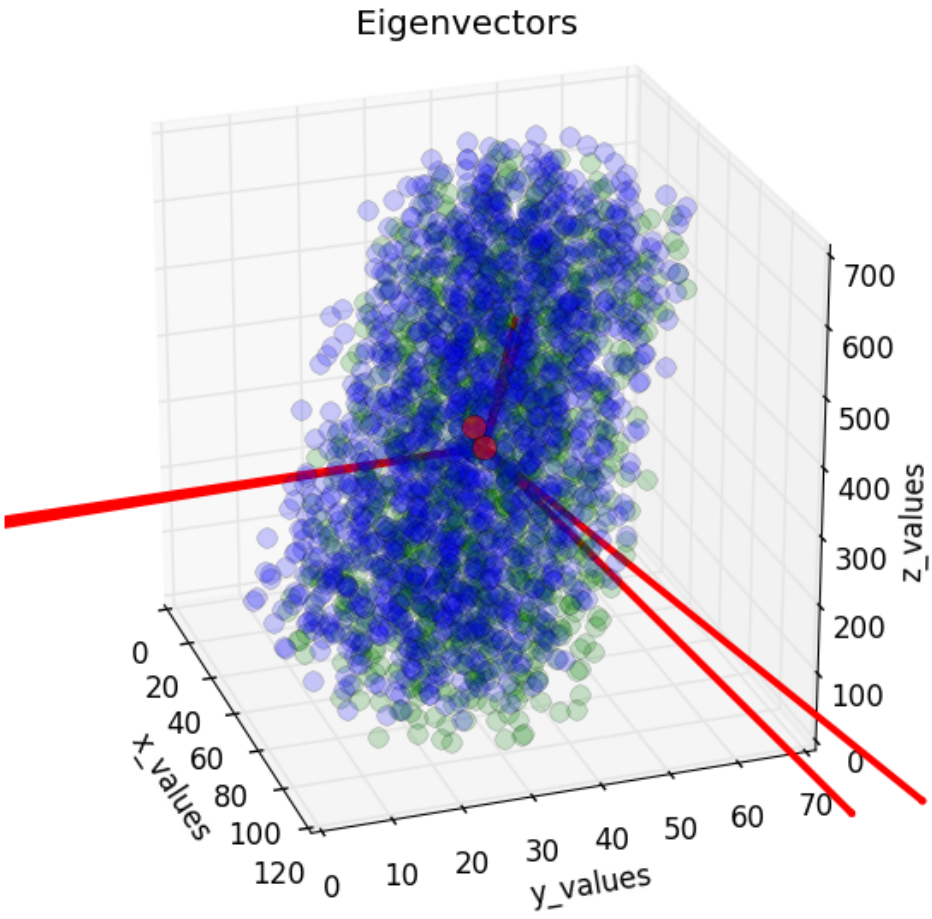


Figure 38 The Principal Axis and Point Clouds

To test the effects of PCA alignment, the following case has been created, and it shows a thin plate metal part, on which an artificial dent has been set up after the reference scan using the CT. Before the registration, we cannot get and locate the dent location from the image correlation. However, when we used compensated PCA algorithm for the rigid body motion and rotation, the dent location can be identified by the computation and correlation. The result can be viewed from the following Figure 42.

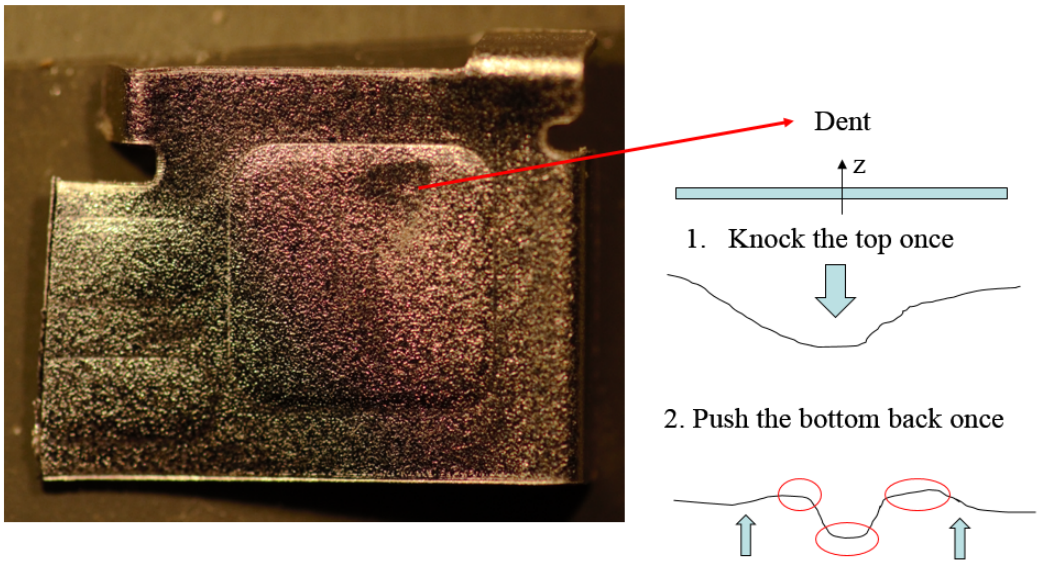


Figure 39 The PCA Registration for Identifying Surface Texture

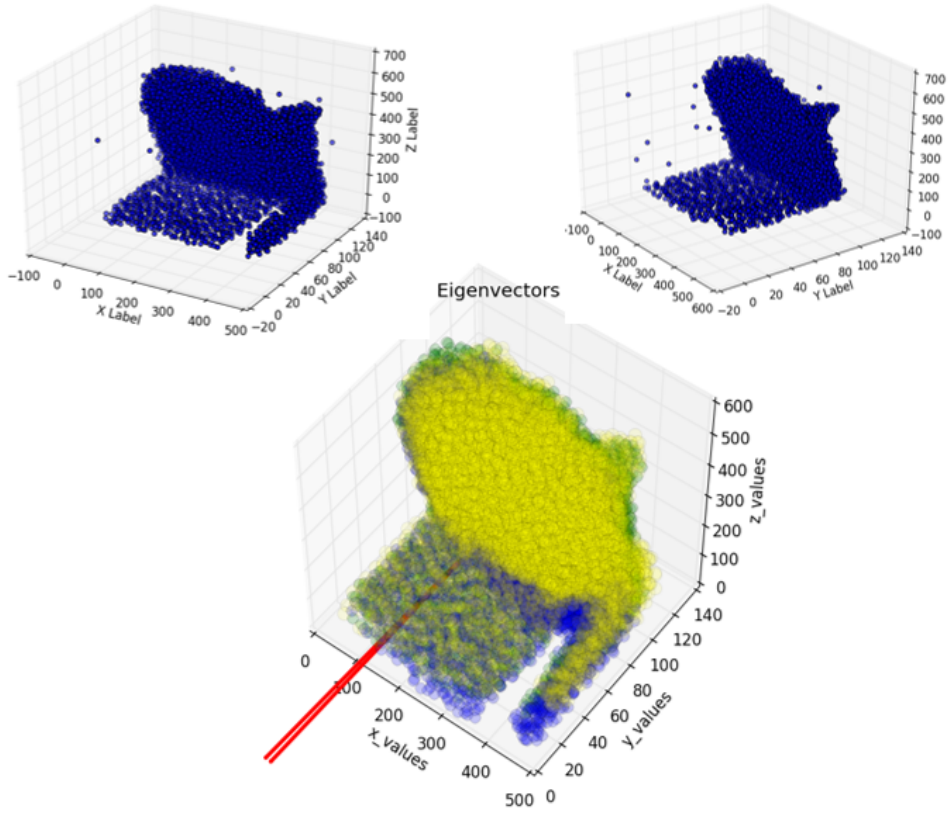


Figure 40 The PCA Alignment of Object

The following Figure 41 shows the initial computation without the registration. We can see due to the rigid body motion, in the vertical direction, there is no significant sign for identifying the dent. However, when we use the PCA registration, the location marked with a circle is obvious and clear in the Figure 42.

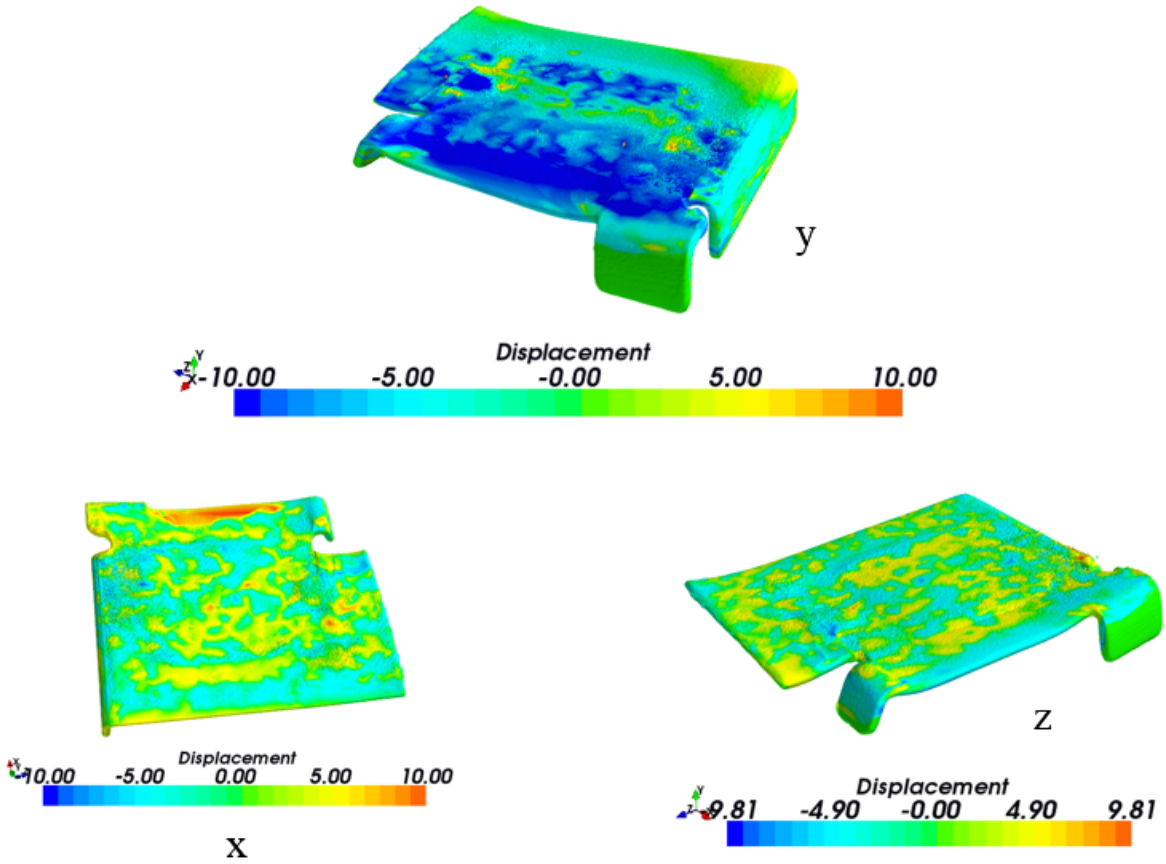


Figure 41 Deformation Estimation Without Pre-registration

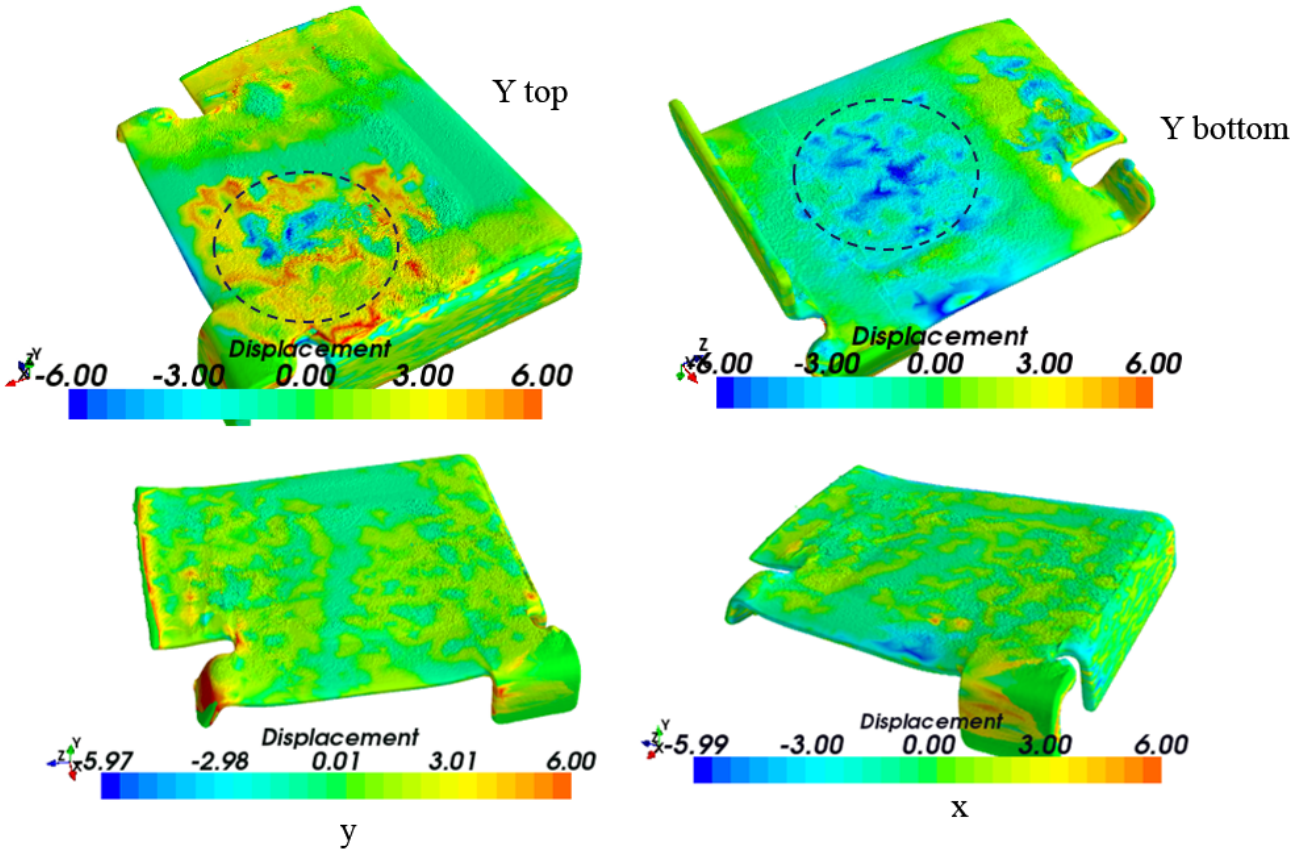


Figure 42 Deformation Estimation Without Pre-registration

4.4.2 Singular Value Decomposition

Using the Singular Value Decomposition (SVD), the optimal alignment between corresponding points can be computed. In the SVD mathematic problem, let P be a matrix of the target point, and Q be a matrix of source point. For each element in the target point, there is the corresponding source point. The covariance matrix M can be calculated by the following form:

$$M = PQ^T \tag{21}$$

The Singular Value Decomposition (SVD) of an m by m matrix M:

$$M = U \cdot W \cdot V^T \tag{22}$$

U, V are ‘m’ by ‘m’ orthonormal matrices (i.e. rotation matrix):

$$R = V \cdot U^T \tag{23}$$

W is a diagonal matrix with non-negative entries.

$$P_i' = C_T + R \cdot (P_i - C_s) \tag{24}$$

P_i': new point

C_T: translation vector

R: rotation vector

C_s: rotation centroid

Overall, the SVD is much more reliable than the PCA registration, but the SVD needs to compute the corresponding features between two domains. So, it requires the human input for finding the corresponded pair of points. Generally, the SVD registration can reach nearly perfect alignment of two groups.

4.4.3 Iterative Closest Points

In the alignment technique, the PCA rigid body motion and rotation can introduce a bias and variation for the result of computation, which is very sensitive to the outliers. Also, the large body translation would require big subset search area to find the target displacements. Moreover, manually calculating paired points could be expensive in the pre-process. Therefore, we apply the Iterative Closet Point (ICP) method to align two volumes, and it is based on the point cloud representation. The Iterative Closest Point (ICP) extends the SVD algorithm, which the corresponding features are computed iteratively. After each iteration, for each point in the source group, we find the closest point using the Euclidean Distance in the target group. This process will

be repeated at each step until the minimum error has been reached. The minimum error calculator shows below:

$$E = \sum_{i=1}^{n} ||q_i - p_i'||^2 \tag{25}$$

q_i: target points

p_i': source points

The termination criteria can be a maximum number of iteration or a Root Mean Squared Distance (RMSD):

$$RMSD = \sqrt{\frac{\sum_{i=1}^{n} ||q_i - p_i'||^2}{n}} \tag{26}$$

n: number of points

The Iterative Closest Point (ICP) method is the best automatic method for rigid registration body. Therefore, in the real application, for the reason of computational cost, first of all, we use the PCA to initially and roughly align the object, and then apply the ICP registration to get a fine registration between two objects. In the ICP implementation, we can use the Octree (Donald, 1980) (Zhou & Gong, 2010) algorithm to speed up the computation.

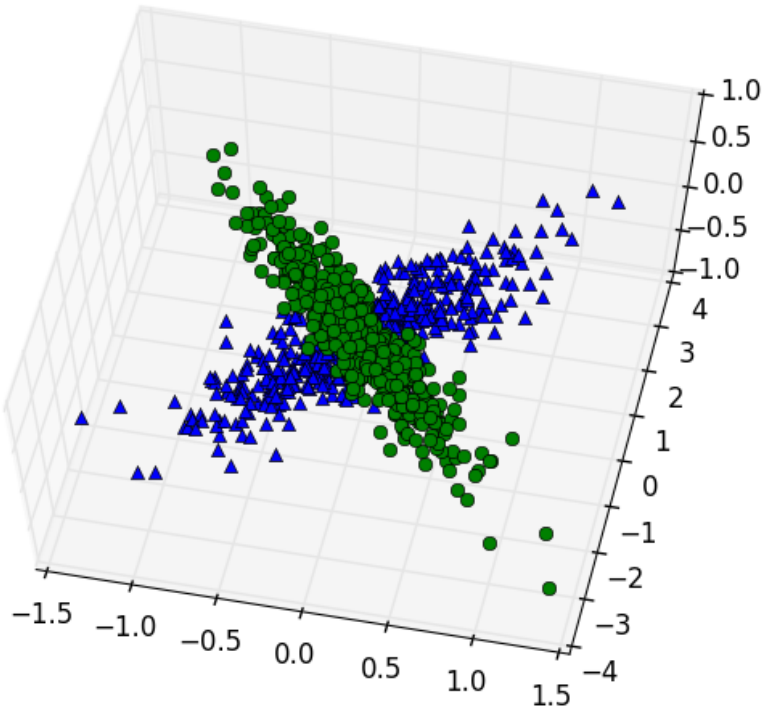


Figure 43 Example of Two Distributions

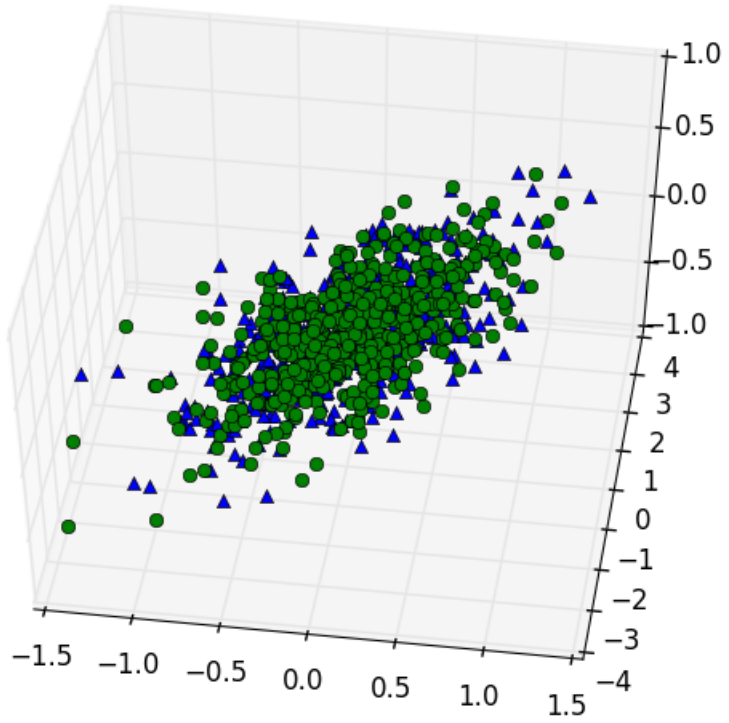


Figure 44 Example of ICP Registration

In the real world alignment algorithm, the SVD often works together with ICP. At each ICP calculation, the paired points have been computed, then the SVD of two paired group has also been computed. Therefore, the movement of translation and rotation can be known at each iteration. The following Figure 45 shows the two point clouds with a corresponding closest point at each iteration:

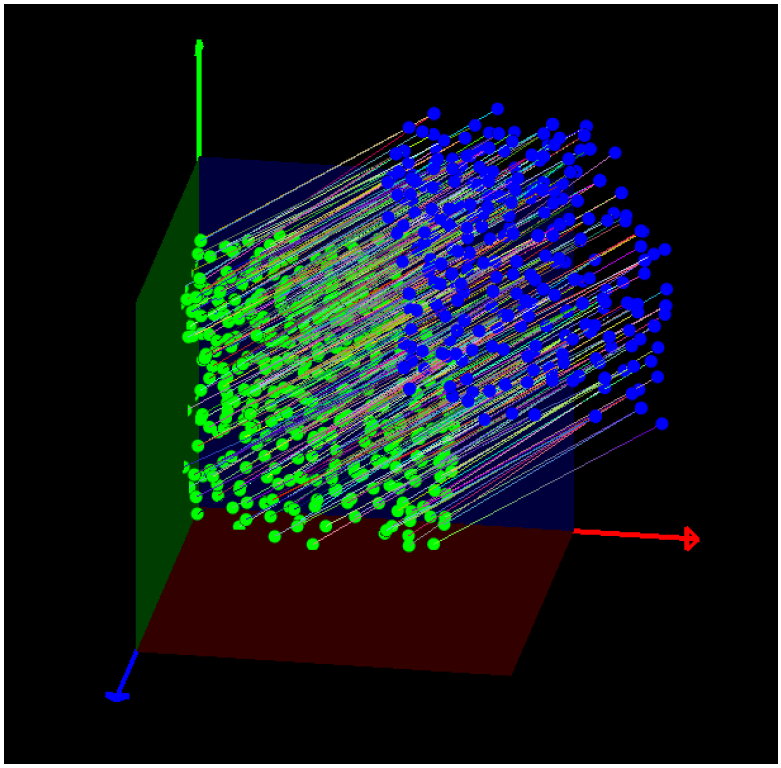


Figure 45 3D View of Two Clouds Using ICP and SVD

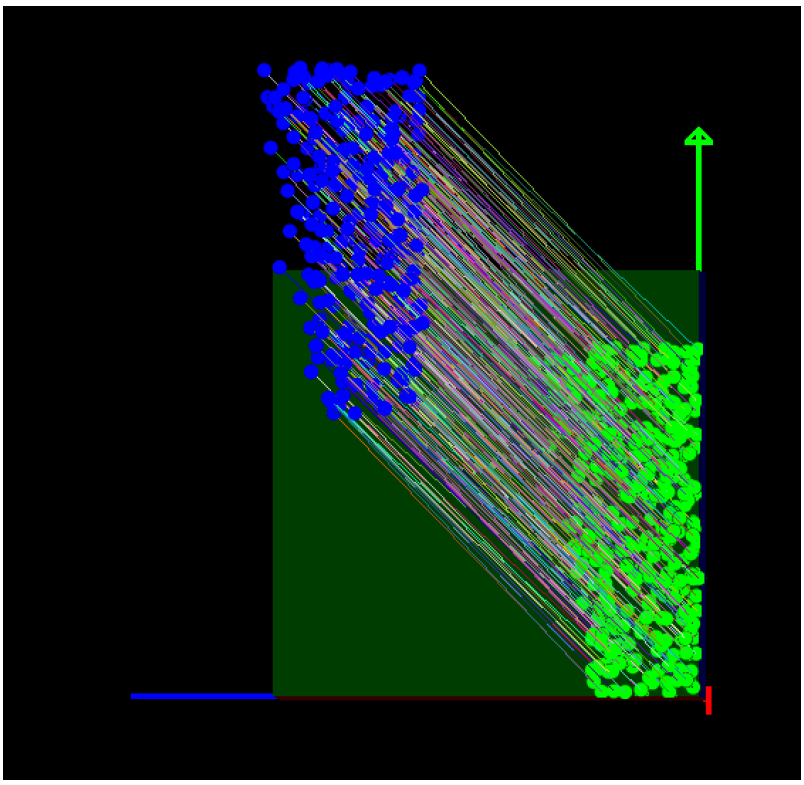


Figure 46 Z and Y Plane View

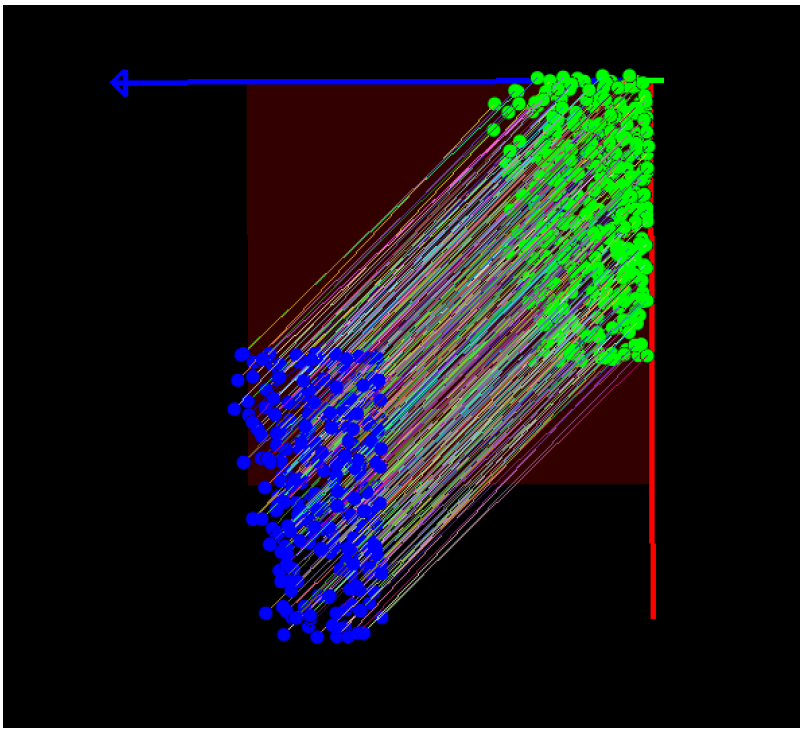


Figure 47 X and Z Plane View

4.5 Image Transformation

After the rotation and translation matrix have been found, we need to deform and transfer the image into a new image. This is called Image Resampling. There are two existing methods: (1) Forward Image Registration (2) Backward Image Registration. The relationship between the reference volume and deformed volume has been shown in the following Figure 48. This Image Resampling technique will be used to verify the DVC deformation later, and we will discuss it later in next chapter.

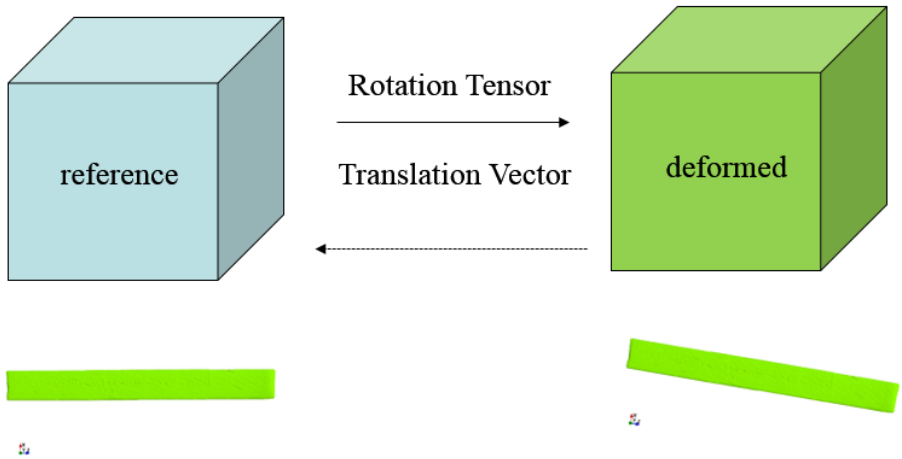


Figure 48 Image Deformation Illustration between the Reference and the Deformed

The following Figure 49 shows the transformation between the original coordinator and the new coordinator. The coordinator is in the form of homogeneous coordinator form, in which a unit has been appended at the end of a column vector. We combine the affine matrix with translation vector to perform a matrix product as the following, and this could save lots of computations.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} r11 & r12 & r13 & T1 \\ r21 & r22 & r23 & T2 \\ r31 & r32 & r33 & T3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (27)$$

rij: affine matrix component

Ti: translation vector

x': new coordinator

x : old coordinator

The interpolation used in the image transformation includes:(1) Nearest Interpolation (2) Trilinear Interpolation (3) C2/C3 Cubic Interpolation (4) Spline Interpolation.

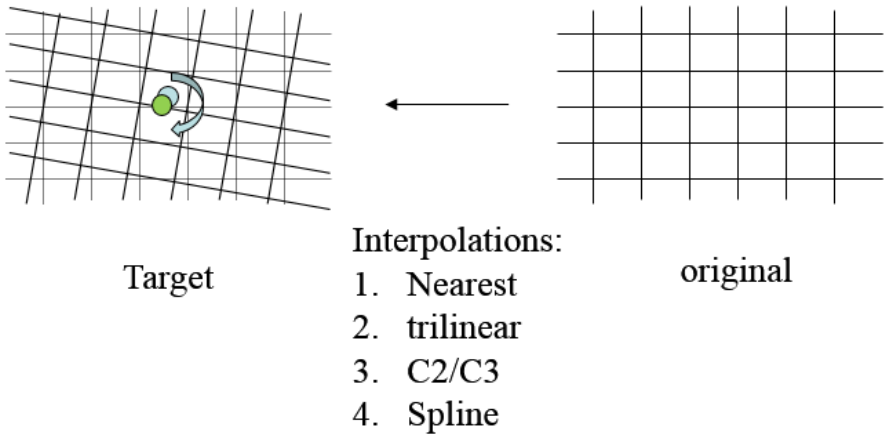


Figure 49 Image Transformation and Interpolation

In the forward image registration, the value of each pixel in the deformed image is directly calculated to use the transformation. Therefore, it can be viewed that each moved pixel in the deformed image is not the integer value after the transformation, and we will use its closest grid point value to approximate it. However, this would create a gap or staircase in the image. It can be seen in the following Figure 50, in the forward registration plot, the staircase has a large influence on the result of pseudo-zero displacement calibration.

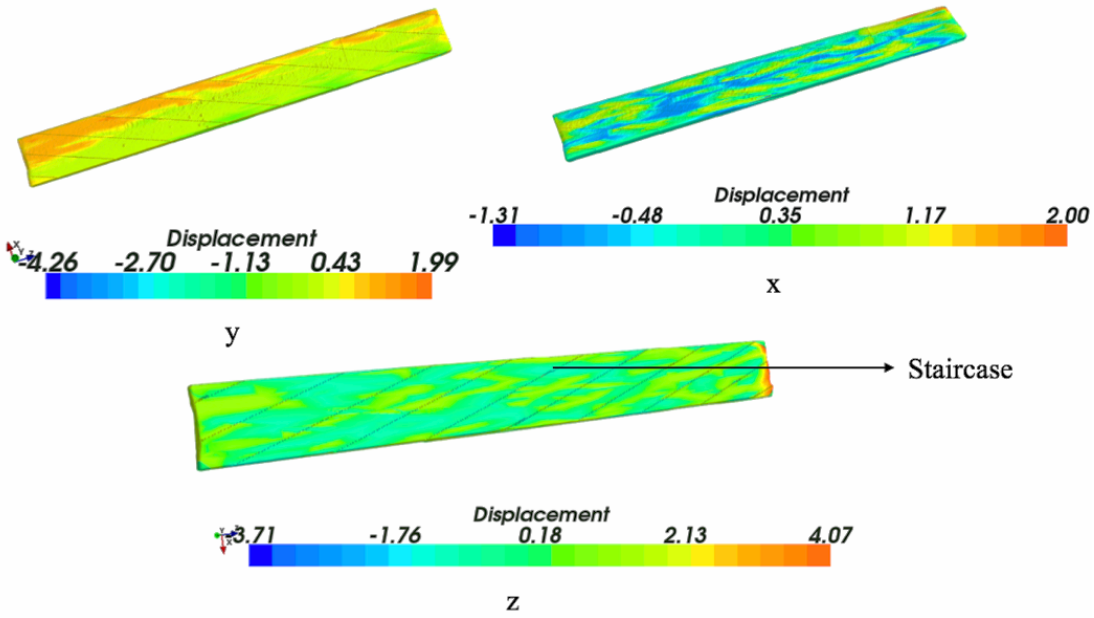


Figure 50 Image Aliasing in the Forward Registration

The following Figure 51 shows the backward registration; the staircase has been removed by the interpolation between the grid points. After the computation, the error displacement in the vertical y-direction has been minimized. Therefore, the image aliasing issue could be handled using the backward registration method. This backward image registration can be also integrated into the iterative image registration algorithm. We will discuss those two methods in details for the image transformation and validation in next chapter.

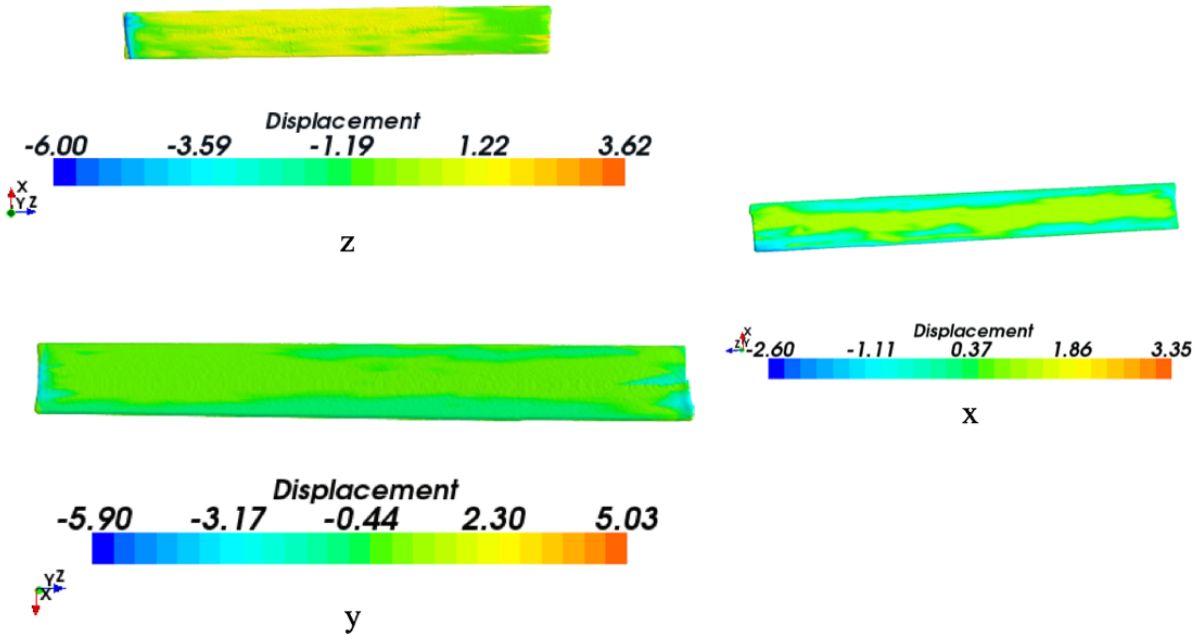


Figure 51 Image Resampling with Backward Registration

4.6 Introduction to The Digital Image Correlation

The Digital Image Correlation (DIC) has a long history in the experimental mechanics. The basic idea is to correlate two images using the subsets matching algorithm. From the intensity of digital images, the window pattern can be identified and used to match the subsets. The following Figure 52 shows two subsets, which the intensity can be viewed as the height value in the z-direction.

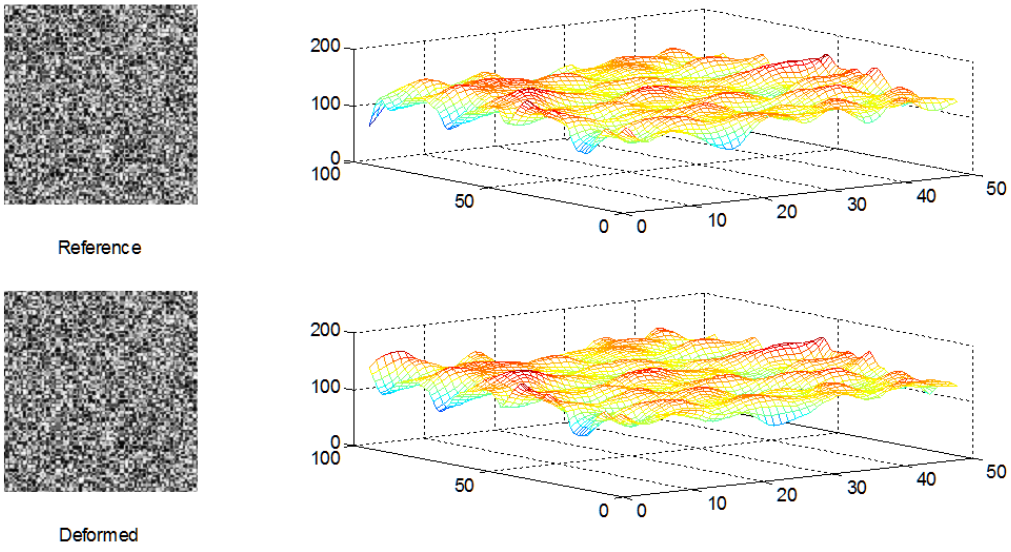


Figure 52 Example of Two DIC images and its Intensity

There are many correlation criterions: (1) Least Squares (2) Normalized Least Squares (3) Unbiased Normalized Least Squares (4) Scalar Product (5) Correlation Coefficient (6) Normalized Cross Correlation (7) Zeroed Normalized Cross Correlation (8) Fast Fourier Transform. Each criterion has a mathematic representation, which is listed as the following sections:

The Least Squares:

$$\mathrm{LSQ}(u,v) = \sum_{x,y} [I_1(x,y) - I_2(x+u, y+v)]^2 \qquad (28)$$

x, y: location in the reference image

u, v: deformation vector component

I_1: reference image

I_2: deformed image

The Normalized Least Squares:

$$\mathrm{NLSQ}(u,v) = \frac{\sum_{x,y}[I_1(x,y) - I_2(x+u,y+v)]^2}{\sqrt{\sum_{x,y} I_1(x,y)^2 \sum_{x,y} I_2(x+u,y+v)^2}} \tag{29}$$

Unbiased Normalized Least Squares:

$$\mathrm{UNLSQ}(u,v) = \sum_{x,y}\left[\frac{I_1(x,y) - \overline{I_1}}{\sqrt{\sum_{x,y}[I_1(x,y) - \overline{I_1}]^2}} - \frac{I_2(x,y) - \overline{I_2}}{\sqrt{\sum_{x,y}[I_2(x,y) - \overline{I_2}]^2}}\right]^2 \tag{30}$$

$$\overline{I_1} = \frac{1}{2M+1}\sum_{i=-M}^{M}\sum_{i=-M}^{M} I_1(x,y)$$

$$\overline{I_2} = \frac{1}{2M+1}\sum_{i=-M}^{M}\sum_{i=-M}^{M} I_2(x',y')$$

The Scalar Product:

$$\mathrm{CC}(u,v) = \sum_{x,y} I_1(x,y) \cdot I_2(x+u,y+v) \tag{31}$$

Normalized Cross Correlation:

$$\mathrm{NCC}(u,v) = \frac{\sum_{x,y} I_1(x,y) I_2(x+u,y+v)}{\sqrt{\sum_{x,y} I_1(x,y)^2 \sum_{x,y} I_2(x+u,y+v)^2}} \tag{32}$$

The Zeroed Normalized Cross Correlation:

$$C(u,v) = \frac{\sum\sum[I_1(x,y) - \overline{I_1}][I_2(x+u,y+v) - \overline{I_2}]}{\sqrt{\sum\sum[I_1(x,y) - \overline{I_1}]^2 \sum\sum[I_2(x+u,v+v) - \overline{I_2}]^2}} \tag{33}$$

$$\overline{I_1} = \frac{1}{(2M+1)^2}\sum_{i=-M}^{M}\sum_{j=-M}^{M} I_1(x+i,y+j)$$

$$\overline{I_2} = \frac{1}{(2M+1)^2}\sum_{i=-M}^{M}\sum_{j=-M}^{M} I_2(x'+i,y'+j)$$

The Fast Fourier Transformation:

$$< f,g > = \mathrm{IFFT}(\overline{\mathrm{FFT}(f_{pad})} \circ \mathrm{FFT}(g_{pad})) \tag{34}$$

$\circ$ is vectorized production

$\overline{\mathrm{FFT}}$ is the conjugate of FFT Transform

The criterion is to find the best match between two subsets, which the local minimum should be avoided. The following Figure 53 shows the peak of correlation coefficients in the two-dimensional domain. The peak is the best match point between the subsets. The ideal correlation criterion should differentiate the cross-correlation coefficient value around the best match point.

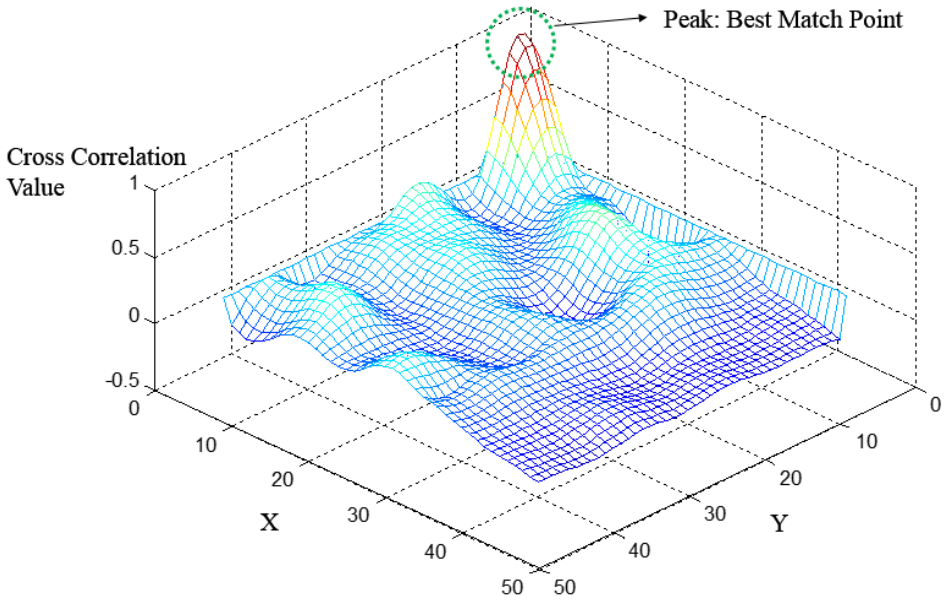


Figure 53 DIC Cross Correlation Field

In a deformation field, there is a relation between the rotation and strain, in the 2D, if we only consider the rotation, every point should remain the same on the plane. Therefore, the normal stains are zero in the following equation:

$$\frac{\partial \mathrm{u}}{\partial \mathrm{x}} = 0.0 \tag{35}$$

$$\frac{\partial \mathrm{v}}{\partial \mathrm{y}} = 0.0 \tag{36}$$

u is the displacement in x direction

v is the displacement in y direction

The gradient of displacement field is the following:

$$\begin{bmatrix} 0.0 & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & 0.0 \end{bmatrix} \tag{37}$$

If we consider the full gradient field, the deformation matrix can be decomposed into a normal strain, shear strain matrix and rotation matrix:

$$\begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial x}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial x}) & \frac{1}{2}(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}) \\ \frac{1}{2}(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}) & \frac{1}{2}(\frac{\partial v}{\partial y} + \frac{\partial v}{\partial y}) \end{bmatrix} + \begin{bmatrix} 0 & \frac{1}{2}(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}) \\ \frac{1}{2}(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}) & 0 \end{bmatrix} \tag{38}$$

The first term, the symmetric component, is strain tensor, and the second term, the skew-symmetric component, is rotation matrix. In general, the deformation matrix can be written as the following:

$$\frac{\partial u_i}{\partial x_j} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right) = \epsilon_{ij} + \omega_{ij} \tag{39}$$

ϵ_{ij}: strain components

ω_{ij}: skew rotation components

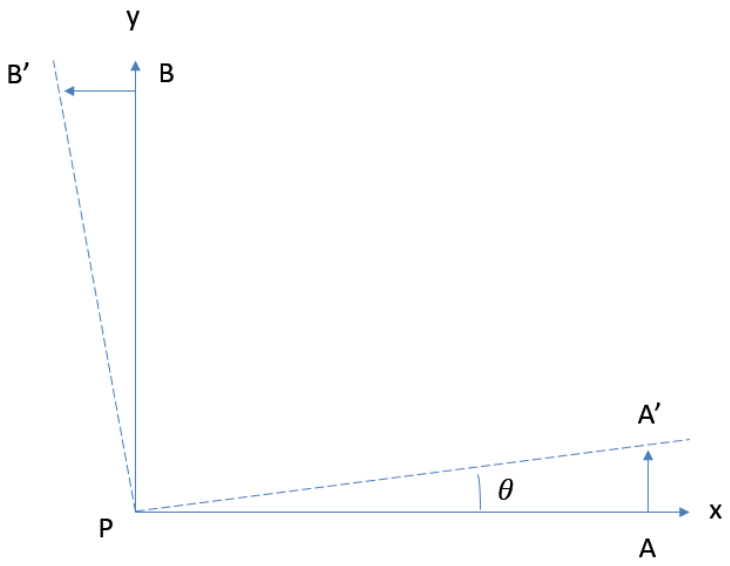


Figure 54 Coordinator Rotation and Strain

The first-order mapping function can be written as the following:

$$x' = x + \bar{u} = x + u + u_x \Delta x + u_y \Delta y \quad (40)$$

$$y' = y + \bar{v} = y + v + v_x \Delta x + v_y \Delta y \quad (41)$$

u_x: gradient of u displacement in x direction

u_y: gradient of u displacement in y direction

v_x: gradient of v displacement in x direction

v_y: gradient of v displacement in y direction

To improve the subset correlation accuracy, the reference subset is mapped to the target subset by a second-order mapping function (Lu & Cary, 1999):

$$x' = x + \bar{u} = x + u + u_x \Delta x + u_y \Delta y + u_{xy} \Delta x \Delta y + u_{xx} \Delta x^2 + u_{yy} \Delta y^2 \quad (42)$$

$$y' = y + \bar{v} = y + v + v_x \Delta x + v_y \Delta y + v_{xy} \Delta x \Delta y + v_{xx} \Delta x^2 + v_{yy} \Delta y^2 \quad (43)$$

In the shape function above, we need to compute full 12 parameters for matching the deformable body. To transform the subset, the interpolation function needs to be applied. At each iterative step, the Cross Correlation Coefficient (CCC) value and its gradient on the strain components have to be calculated. Therefore, the image interpolation algorithm is required to improve the accuracy of non-integer values between the grid pixel points, the nearest neighbors. As we mentioned before, there are several interpolation methods: (1) Nearest-neighbor (2) Bilinear (3) Bicubic (4) C-spline. The simplest way to get the value of non-integer pixel value is to use the nearest-neighbor, which the value of nearest neighbor will be used to cover the non-integer point.

In the bilinear interpolation, four coefficients must be determined from the four surrounding pixels' grey-level values. Therefore, at the boundary, the value is continuous and the first derivative is not continuous.

$$g(x,y) \approx g_{ij} + a_{ij}\Delta x + b_{ij}\Delta y + c_{ij}\Delta x\Delta y \tag{44}$$

$$g_{ij}, a_{ij}, b_{ij}, c_{ij}: \text{bilinear interpolation coefficients}$$

To get the continuous derivatives at a critical point, the Bicubic interpolation is applied to reach a higher accuracy of the deformed image. The Bicubic is smoother than the surface obtained by bilinear interpolation and nearest neighbor interpolation. There are at least nine constants to be determined from the definition.

$$p(x,y) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij}x^i y^j \tag{45}$$

$$a_{ij}: \text{bicubic interpolation coefficients}$$

The spline interpolation is the most advanced technique to achieve a smooth surface and continuity of the first derivative and second derivative. The linear spline has defined as following equation:

$$f = Af_k + Bf_{k+1} \tag{46}$$

$$A = \frac{x_{k+1} - x}{x_{k+1} - x_k} \tag{47}$$

$$B = 1 - A = \frac{x - x_k}{x_{k+1} - x_k} \tag{48}$$

The cubic spline adds two more terms in the linear spline:

$$f = Af_k + Bf_{k+1} + Cf_k'' + Df_{k+1}'' \tag{49}$$

$$C = \frac{1}{6}(A^3 - A)(x_{k+1} - x_k)^2 \tag{50}$$

$$D = \frac{1}{6}(B^3 - B)(x_{k+1} - x_k)^2 \tag{51}$$

Another advanced spline is B-spline, in the B-spline, it utilizes a set of piecewise cubic polynomials defined on some sub-intervals, in which the first and the second derivatives are continuous at the boundaries of intervals. Those polynomials would form a set of Basis functions, and the linear combinations could satisfy continuity between adjacent intervals. If the interpolation covers the region, then it is defined by the following equations:

$$B_0 = \begin{cases} 0\ ;\ (x \le x_0 - 2h) \\ \frac{1}{6}\left(2h + (x - x_0)\right)^3 ; (x_0 - 2h \le x \le x_0 - h) \\ \frac{2h^3}{3} - \frac{1}{2}(x - x_0)^2\left(2h + (x - x_0)\right); (x_0 - h \le x \le x_0) \\ \frac{2h^3}{3} - \frac{1}{2}(x - x_0)^2\left(2h - (x - x_0)\right); (x_0 \le x \le x_0 + h) \\ \frac{1}{6}(2h - (x - x_0))^3 ; (x_0 + h \le x \le x_0 + 2h) \\ 0\ ; (x \ge x_0 + 2h) \end{cases} \quad (52)$$

where: $h = x_{k+1} - x_k = \frac{x_N - x_0}{N}$

The B_0, B_0' and B_0'' are continuous at -2h, -h, 0, h and 2h in the defined spline above. The shifted B_0 function is defined as the following: B_0 functions shifted to the right by k nodes $(B_{k-1}, B_k, B_{k+1}, B_{k+2})$:

$$B_k(x) = B_0(x - kh + x_0) \quad (53)$$

The cubic spline function, $S_3(x)$ in $[x_0, x_N]$ is the linear combination of B_k :

$$S_3(x) = \sum_{k=-1}^{N+1} a_k B_k(x) \quad (54)$$

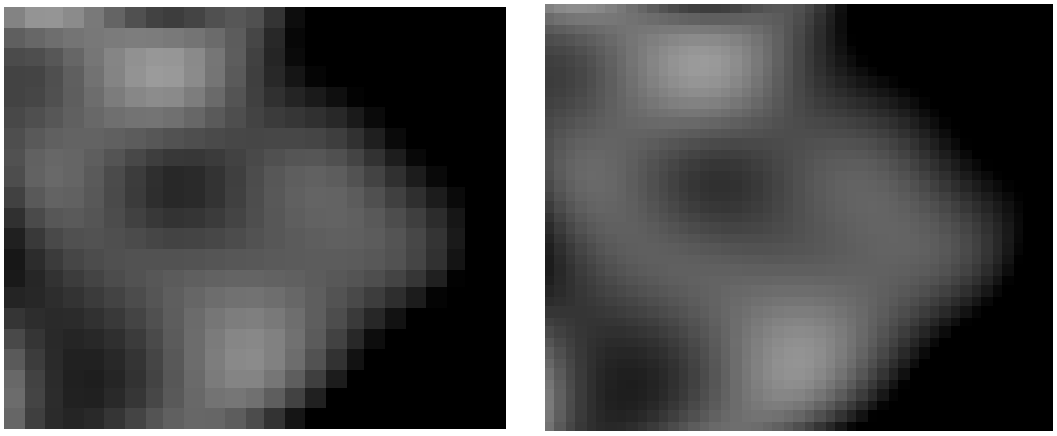


Figure 55 An Example of B-spline Interpolation: Left (Cubic) Right (B-spline)

In the optimization part, there are many gradient-based optimization methods to find the deformation vectors between two subsets. The recent methods are gradient based methods including the Gaussian Newton method, Forward Additive method, and Inverse Compositional method. The Gaussian Newton method is the fundamental of all optimization methods (Shao, Dai, & He, 2014), and in the Gaussian Newton method, the Taylor expansion has been used to approximate the cross-correlation coefficient field given a small step:

$$\mathrm{CC}(\mathrm{p}_0 + \Delta\mathrm{p}) = \mathrm{CC}(\mathrm{P}_0) + \nabla\mathrm{CC}(\mathrm{p}_0)^{\mathrm{T}} \cdot \Delta\mathrm{p} + \frac{1}{2}\Delta\mathrm{p}^{\mathrm{T}} \cdot \nabla\nabla\mathrm{CC}(\mathrm{p}_0) \cdot \Delta\mathrm{p} \tag{55}$$

$$\mathrm{CC}: \text{cross correlation value}$$

The gradient of Cross Correlation(CC) is the following:

$$\frac{\mathrm{dCC}(\mathrm{p}_0 + \Delta\mathrm{p})}{\mathrm{d}\Delta\mathrm{p}} \approx \nabla\mathrm{CC}(\mathrm{P}_0) + \nabla\nabla\mathrm{CC}(\mathrm{P}_0) \cdot \Delta\mathrm{p} \tag{56}$$

If we set this gradient equals to zero, we can get the solution for choosing the deformation vector $\Delta\mathrm{p}$:

$$\nabla\mathrm{CC}(\mathrm{P}_0) = -\nabla\nabla\mathrm{C}(\mathrm{P}_0) \cdot \Delta\mathrm{p} \tag{57}$$

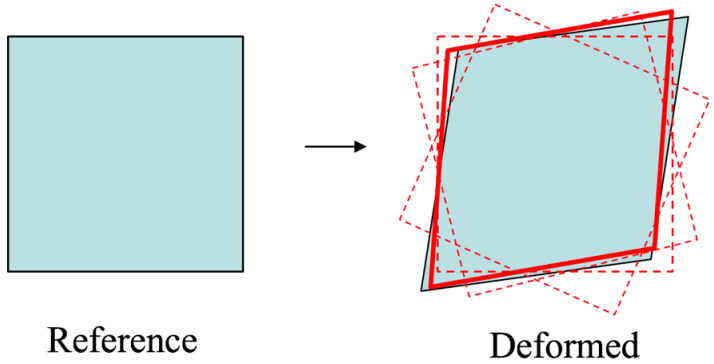


Figure 56 Gaussian Newton Minimization between Subsets

4.6.1 Deformable Subset Matching

In the sub-pixel accuracy, the deformable matrix can be applied to the reference subset to get the deformed subset, which can be used to further minimize the Error Cross-Correlation value. There are many existing algorithms to track the deformation of reference. Some of the algorithms have been used widely in Computer Vision (CV) for tracking the object in the video. In this work, we will present two famous algorithms, one is called Forward Additive method, another is the Inverse Compositional method. The former method is straightforward, and it is very intuitive and easy to understand. However, it is not very efficient for computing the deformation iteratively. The next method, the Inverse Compositional Method, is modified based on the deformation function, and it is relatively faster than the forward additive method (Pan B., 2014). However, it is hard to understand the idea and implement it. The following Figure 57 and Figure 58 show a reference picture and a deformed picture. The deformed image is derived from the reference image, in which a warp matrix is applied.

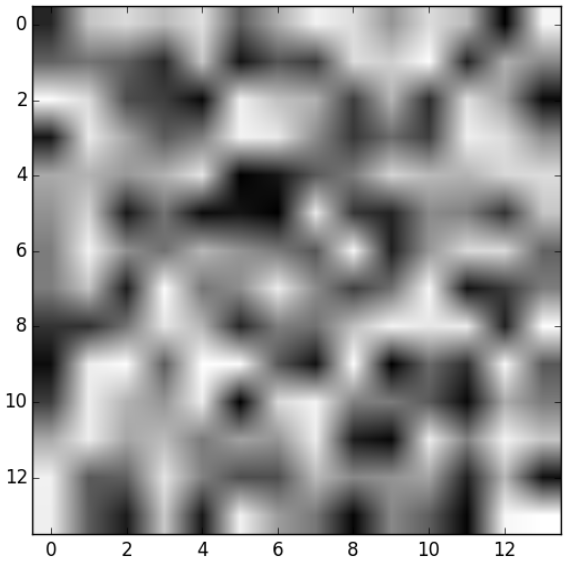


Figure 57 Reference Subset (14-14) from Random Greyscale Image

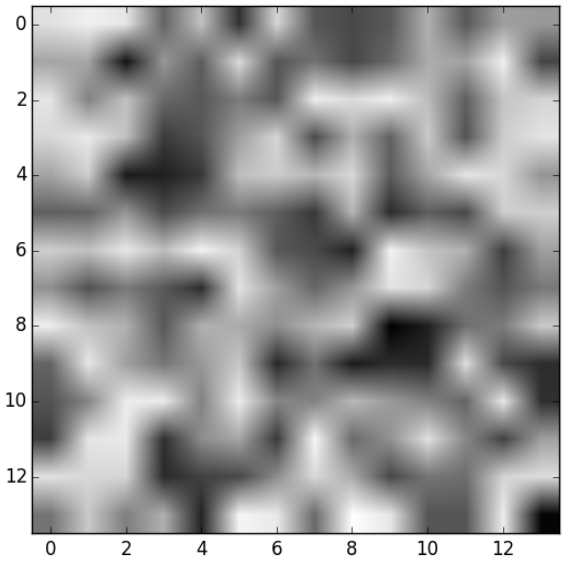


Figure 58 Deformed Subset (14-14) from Random Greyscale Image

4.6.2 Kanade-Lucas-Tomasi Feature Tracker:

In the world of image correlation, we have to talk something about the feature tracker that is utilized for image alignment (Lucas & Kanade, 1981) (BAKER & MATTHEWS, 2004). The beauty of KLT algorithm is that it translates a nonlinear problem of image alignment into a linear optimization problem, in which the deformation vector can be updated iteratively using the gradient-based methods (Pan, Li, & Tong, 2012). In the beginning, we have to introduce some backgrounds about the variation model assumptions in the images (Figueroa-O'Farrill).

(1) Greyscale value constancy assumption.

$$I(x, y, t) = I(x + u, y + v, t + 1) \tag{58}$$

(2) Greyscale value change over two scans if two scans are really close.

$$I(x + u, y + v, t + 1) \approx I(x, y, t) + I_x u + I_y v + I_t \tag{59}$$

$$I(x + u, y + v, t + 1) - I(x, y, t) = I_x u + I_y v + I_t \tag{60}$$

$$I_{,x}u + I_{,y}v + I_t \approx 0 \quad (61)$$

$$\nabla I \cdot [u, v]^T + I_t = 0 \quad (62)$$

I_t: Greyscale Value Difference between Two Scans

$I_{,x}$: Images' Derivative along x direction

$I_{,y}$: Images' Derivative along y direction

(3) Gradient constancy assumption.

$$\nabla I(x, y, t) = \nabla I(x + u, y + v, t + 1) \quad (63)$$

(4) Minimization approach.

$$E = \int_{\Omega} (|I(x + w) - I(x)|^2) \quad (64)$$

From above, we can see that the pixel movement in an image follows the equation:

$$0 = I_t(p_i) + \nabla I(p_i)\begin{bmatrix} u \\ v \end{bmatrix} \quad (65)$$

p_i: the pixel i

I_t: Greyscale Value Difference Between Two Scans

∇I: Images' Gradient along x direction and y direction

u: the displacement vector in x direction

v: the displacement vector in y direction

The u, v are motion vectors that describe the optical flow in the image (Warren & Strelow, 1985). If we consider all the pixel in the image, the following linear system matrix can be established (Lucas & Kanade, 1981).

$$\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ I_x(p_n) & I_y(p_n) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(p_1) \\ I_t(p_2) \\ I_t(p_n) \end{bmatrix} \quad (66)$$

$$A_{n\times 2}\, d_{2\times 1} = -b_{n\times 1} \quad (67)$$

$I_{,x}; I_{,y}; I_{,z}; I_t$: known quantities

u, v: unknown

n: number of pixels in image

The basic derivation for templates is the sum of square error over all pixels in the image (Baker & Matthews, 2001).

$$E(u, v) = \sum [I(x + u, y + v) - T(x, y)]^2 \quad (68)$$

$T(x, y)$: reference image

$I(x, y)$: deformed image

$I(x + u, y + v)$: moving deformed image

$E(u, v)$: sum of Error between two images

x, y: location of pixel

u, v: the displacement of pixel

I: deformed Image Intensity (Greyscale Values)

T(x,y): Reference Image Intensity at x,y (Greyscale Values)

T is a template image, and I is a current image. The u, v vector is hypothesized location of template in current frame. If we take derivative of error function, we can get the following approximation using Taylor series expansion:

$$E(u,v) = \sum [I(x+u, y+v) - T(x,y)]^2 \tag{69}$$

$$\approx \sum \left[I(x,y) + uI_x(x,y) + vI_y(x,y) - T(x,y) \right]^2$$

$$= \sum \left[uI_x(x,y) + vI_y(x,y) + D(x,y) \right]^2$$

$D(x,y)$: difference between $I(x,y)$ and $T(x,y)$

I_x: Images' Derivative along x direction

I_y: Images' Derivative along y direction

If we take the partial derivative and set it to zero, we can find the root of minimization for the error function:

$$\frac{\partial E}{\partial u} = \sum \left[uI_x(x,y) + vI_y(x,y) + D(x,y) \right] I_x(x,y) = 0 \tag{70}$$

$$\frac{\partial E}{\partial v} = \sum \left[uI_x(x,y) + vI_y(x,y) + D(x,y) \right] I_y(x,y) = 0 \tag{71}$$

$I_{,x}$: Images' Derivative along x direction

$I_{,y}$: Images' Derivative along y direction

This can be written as matrix form:

$$\sum \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\sum \begin{bmatrix} I_x D \\ I_y D \end{bmatrix} \tag{72}$$

In the KLT algorithm, it assumes the deformed image follows a warp function, and each pixel is undergoing a transformation through multiplying a warp matrix in homogeneous coordinator.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1+p_1 & p_3 & p_5 \\ p_2 & 1+p_4 & p_6 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (73)$$

x', y': grid point in deformed image

x, y: grid point in original image

p_i: image deformation coefficients (i.e. affine transformation)

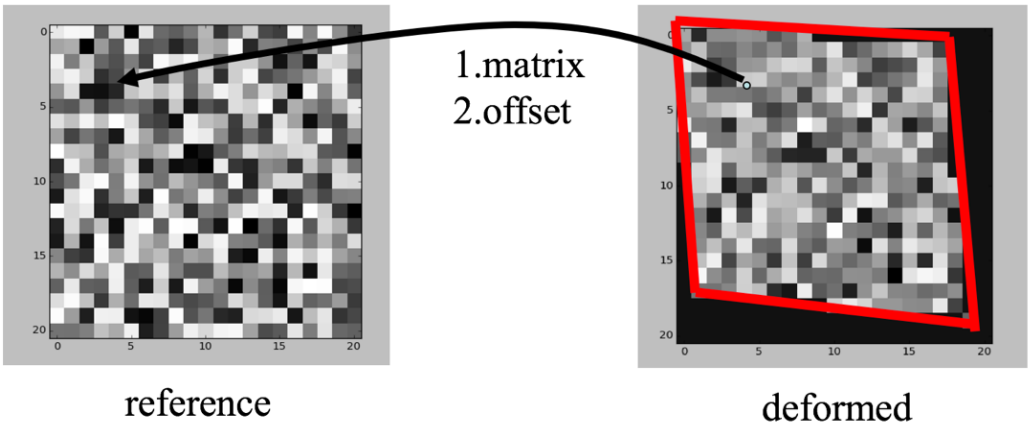


Figure 59 Image with Warp Matrix

In the KLT, the minimization of square error leads to the following function:

$$E(u,v) = \sum [I(x+u, y+v) - T(x,y)]^2 = \sum [I(W[x,y], P) - T(x,y)]^2 \quad (74)$$

Define:

$$W([x,y],p) = \begin{pmatrix} (1+p_1)\cdot x + p_3 \cdot y + p_5 \\ p_2 + (1+p_4)\cdot y + p_6 \end{pmatrix} = \begin{pmatrix} 1+p_1 & p_3 & p_5 \\ p_2 & 1+p_4 & p_6 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (77)$$

$$w_x = (1+p_1)\cdot x + p_3 \cdot y + p_5 \quad (75)$$

$$w_y = p_2 + (1+p_4)\cdot y + p_6 \quad (76)$$

The Taylor expansion for each pixel is:

$$I(W(x,y),p+\Delta p) \approx I(W(x,y),p) + \nabla I \frac{\partial W}{\partial p}\Delta p \tag{78}$$

∇I: Gradients in x,y,z direction of image

$\frac{\partial W}{\partial p}$: Derivative of Affine Transformation Matrix

Δp: Delta of Affine Matrix

If we use the chain rule to expand the above formula, we can get the following approximation using matrix multiplication:

$$\nabla I \frac{\partial W}{\partial p}\Delta p = \begin{bmatrix}\frac{\partial I}{\partial x} & \frac{\partial I}{\partial y}\end{bmatrix}\begin{bmatrix}\frac{\partial w_x}{\partial p_1} & \frac{\partial w_x}{\partial p_2} & \frac{\partial w_x}{\partial p_3} & \frac{\partial w_x}{\partial p_4} & \frac{\partial w_x}{\partial p_5} & \frac{\partial w_x}{\partial p_6} \\ \frac{\partial w_y}{\partial p_1} & \frac{\partial w_y}{\partial p_2} & \frac{\partial w_y}{\partial p_3} & \frac{\partial w_y}{\partial p_4} & \frac{\partial w_y}{\partial p_5} & \frac{\partial w_y}{\partial p_6}\end{bmatrix}\begin{bmatrix}\Delta p_1 \\ \Delta p_2 \\ \Delta p_3 \\ \Delta p_4 \\ \Delta p_5 \\ \Delta p_6\end{bmatrix} \tag{79}$$

$\frac{\partial I}{\partial x}$: Greyscale value gradient in x direction

$\frac{\partial I}{\partial y}$: Greyscale value gradient in y direction

The warp function can be calculated by solving the following linear system:

$$\Delta p = H^{-1}\sum_x \left[\nabla I \frac{\partial W}{\partial p}\right]^T [T(x) - I(W(x;p))] \tag{80}$$

If we consider the affine transformation matrix, the following relationship will stand:

$$W([x,y],p) = \begin{bmatrix}1+p_1 & p_3 & p_5 \\ p_2 & 1+p_4 & p_6\end{bmatrix} = \frac{\partial W}{\partial p} = \begin{bmatrix}x & 0 & y & 0 & 1 & 0 \\ 0 & x & 0 & y & 0 & 1\end{bmatrix} \tag{81}$$

From above, the Jacobian matrix for the reference image set looks like the following Figure 60:

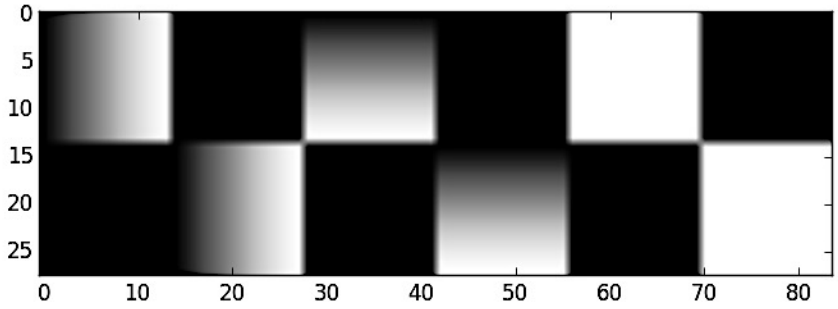


Figure 60 Jacobian Matrix for the Image

Then, the steepest descent images $\nabla \mathrm{I} \frac{\partial \mathrm{W}}{\partial \mathrm{p}}$ can be calculated from the Jacobian:

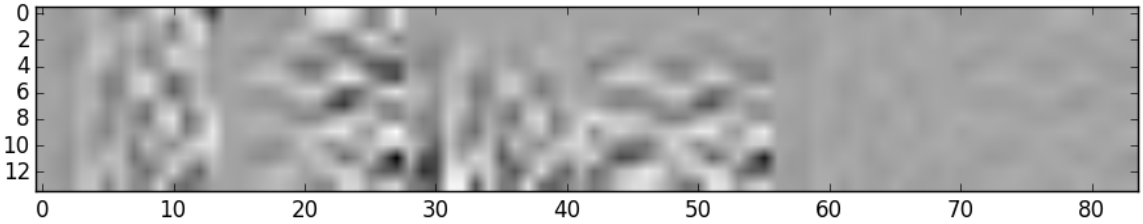


Figure 61 Steepest Descent Images

The hessian matrix is also calculated from the Jacobian:

$$\sum \left(\nabla \mathrm{I} \frac{\partial \mathrm{W}}{\partial \mathrm{p}}\right)^{\mathrm{T}} \left(\nabla \mathrm{I} \frac{\partial \mathrm{W}}{\partial \mathrm{p}}\right) \tag{82}$$

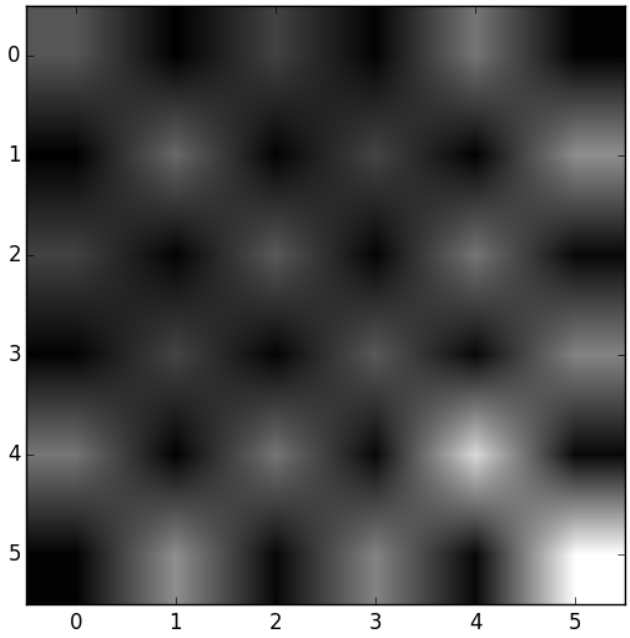


Figure 62 Hessian Matrix

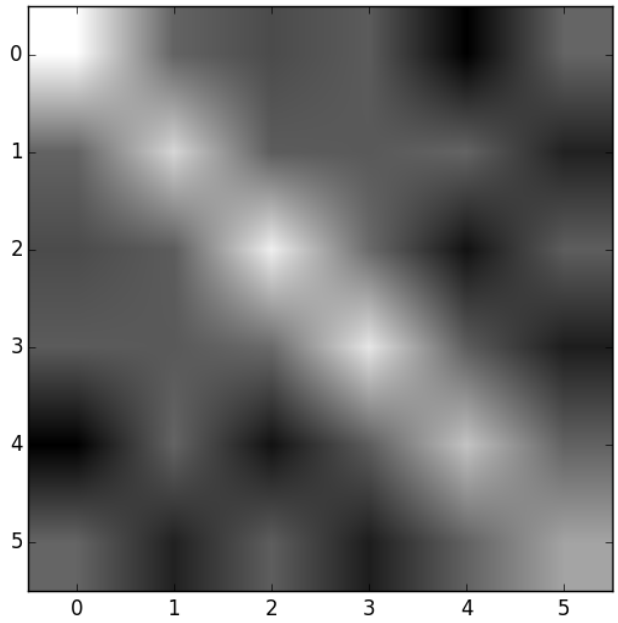


Figure 63 Inversed Hessian Matrix

4.6.3 Forward Additive Method:

The Forward Additive Method is an image alignment method, and it uses the warp matrix to find a deformed body from the image, and it can be directly compared with the template image. The deformation matrix 'p' is updated at each iteration (Blaber, Adair, & Antoniou, 2015).

$$p_{new} = p_{old} + \Delta p \tag{83}$$

In the Forward Additive Method, the least square fitting metric has been applied, and the target is to minimize the mean square between two images.

$$C(p_{new}) = \sum \left[\frac{f - f_m}{\sqrt{\sum [f - f_m]^2}} - \frac{g(p_{old} + \Delta p) - g_m}{\sqrt{\sum [g(p_{old} + \Delta p) - g_m]^2}} \right]^2 \tag{84}$$

f_m: mean of refernce subset

g_m: mean of refernce subset

For the reason of simplicity, we consider the following terms: the first assumption is the average grayscale value does not vary too much for the subset. Those assumptions can be verified empirically and be acceptable.

$$\frac{d}{d\Delta p}(g_m) \approx 0 \tag{85}$$

$$\frac{d}{d\Delta p}\left(\sqrt{\sum [g - g_m]^2}\right) \approx 0 \tag{86}$$

The gradient of least square error fit is:

$$\nabla C(p_{old}) = \frac{dC(p_{old})}{d\Delta p} \approx \frac{-2}{\sqrt{\sum [g - g_m]^2}} \sum \left[\left[\frac{f - f_m}{\sqrt{\sum [f - f_m]^2}} - \frac{g - g_m}{\sqrt{\sum [g - g_m]^2}} \right] \left[\frac{dg}{d\Delta p} \right] \right] \tag{87}$$

The hessian is:

$$\nabla\nabla C(p_{old}) = \frac{d^2C(p_{old})}{d\Delta p^2} \approx \frac{-2}{\sqrt{\sum[g-g_m]^2}}\left\{\sum\left[-\frac{\frac{dg}{d\Delta p}}{\sqrt{\sum[g-g_m]^2}}\right]\left[\frac{dg}{d\Delta p}\right]^T + \sum\left[\left[\frac{f-f_m}{\sqrt{\sum[f-f_m]^2}} - \frac{g-g_m}{\sqrt{\sum[g-g_m]^2}}\right]\left[\frac{d^2g}{dp^2}\right]\right]\right\} \quad (88)$$

The hessian will be used for the Newton-Raphson iterations, and it will become the Gaussian Newton using the following equation:

$$\sum\left[\left[\frac{f-f_m}{\sqrt{\sum[f-f_m]^2}} - \frac{g-g_m}{\sqrt{\sum[g-g_m]^2}}\right]\left[\frac{d^2g}{dp^2}\right]\right] \approx 0 \quad (89)$$

The final hessian estimation is the following:

$$\nabla\nabla C(p_{old}) = \frac{d^2C(p_{old})}{d\Delta p^2} \approx \frac{-2}{\sqrt{\sum[g-g_m]^2}}\left\{\sum\left[-\frac{\frac{dg}{d\Delta p}}{\sqrt{\sum[g-g_m]^2}}\right]\left[\frac{dg}{d\Delta p}\right]^T\right\} \quad (90)$$

The goal of forward additive method is to find the ultimate deformation field ‘p’ that deforms the body of reference image from the current image. The Figure 64 illustrates the forward additive method.

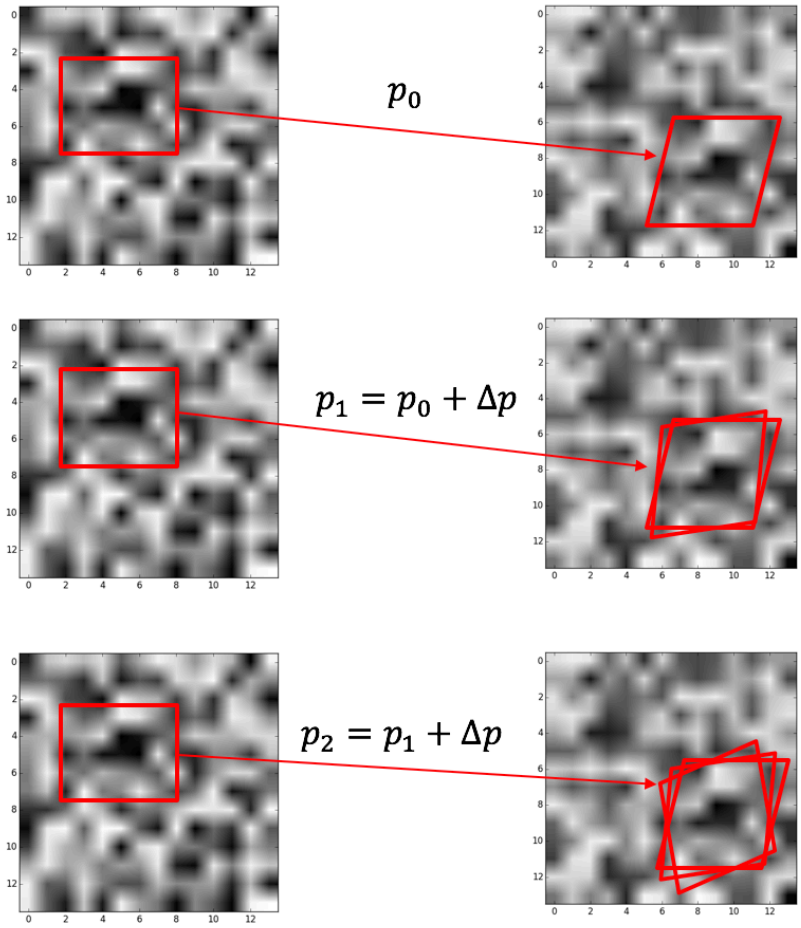


Figure 64 Forward Additive Method Illustration

4.6.4 Inverse Compositional Method:

The Inverse compositional method is reversed view from the forward additive method. It assumes the warp deformation is applied on the current image instead of the reference image, and the warp matrix is updated by multiplying the inversed the updating matrix at each time. The major part of algorithm is like the Forward Additive Method, the change is at the updating part. The following equation shows inversed updating matrix multiplication:

$$\begin{bmatrix} 1+\frac{du}{dx2} & \frac{du}{dy2} & u \\ \frac{dv}{dx2} & 1+\frac{dv}{dy2} & v \\ 0 & 0 & 1 \end{bmatrix} = \tag{91}$$

$$\begin{bmatrix} 1+\frac{du}{dx1} & \frac{du}{dy1} & u \\ \frac{dv}{dx1} & 1+\frac{dv}{dy1} & v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1+\Delta\frac{du}{dx} & \Delta\frac{du}{dy} & \Delta u \\ \Delta\frac{dv}{dx} & 1+\Delta\frac{dv}{dy} & \Delta v \\ 0 & 0 & 1 \end{bmatrix}^{-1}$$

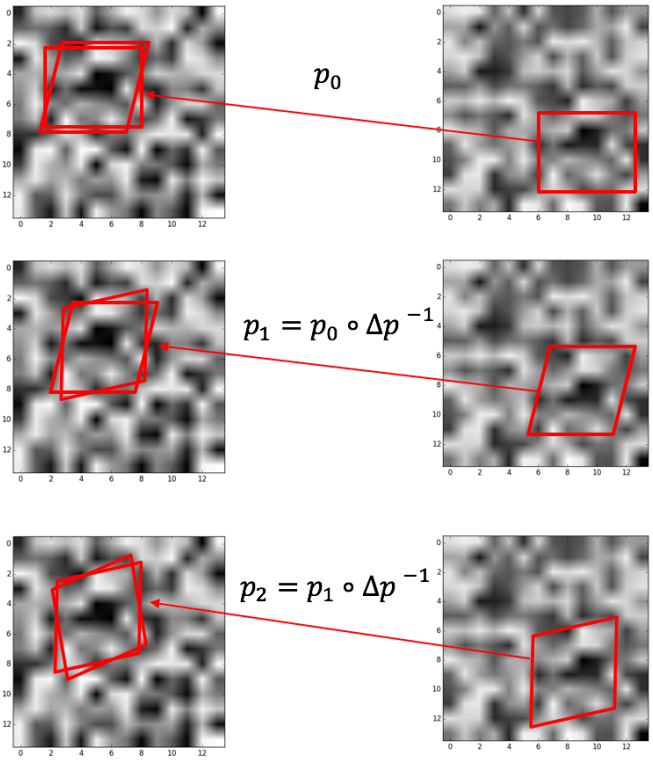


Figure 65 Inverse Compositional Method Illustration

The following Figure 66 shows the blue square window that has been selected for the subset.

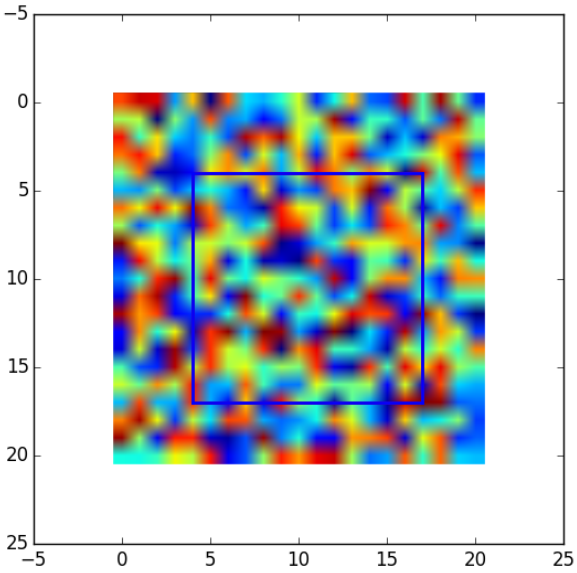


Figure 66 Subset Window Selection

The following Figure 67 shows the process of finding the optimal deformed subset, which gives the best estimation of deformed window.

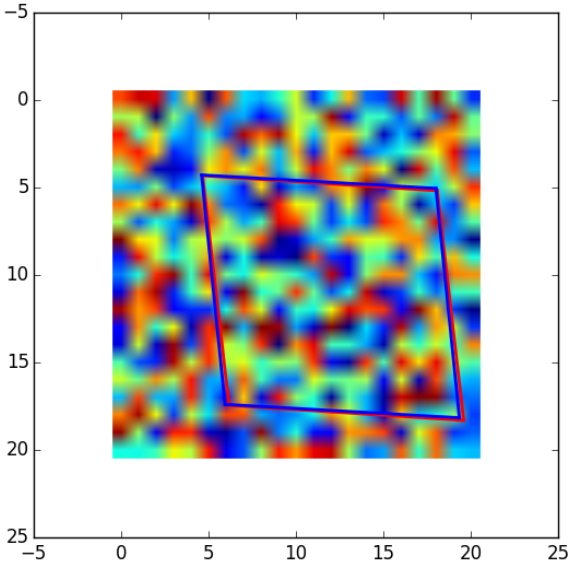


Figure 67 Deformable Search in the Subset

After the deformable calculation and comparison, the two image can be aligned together, we can see the errors and residuals after superimposing the two images.

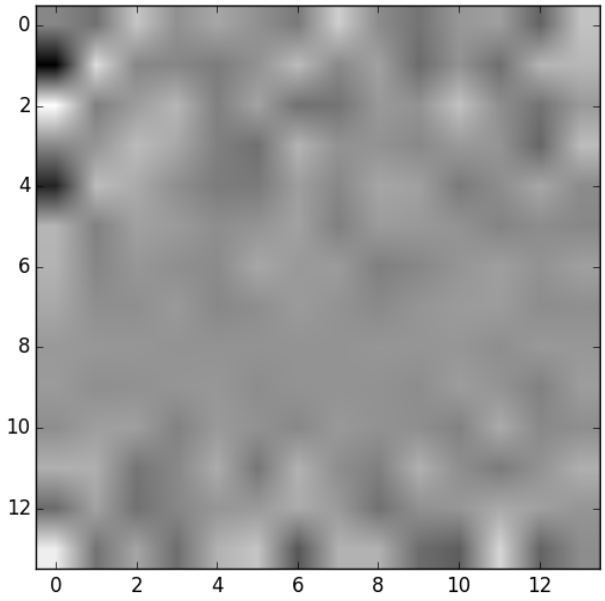


Figure 68 Small Residual Error after Two Subsets have been aligned

4.7 Digital Volume Correlation

Digital Volume Correlation (DVC) is a technique to calculate the full-field strains, and it is an extension for the Digital Image Correlation (DIC), which is used to calculate the in-plane stains or strains in the out-of-plane direction. Digital Volume Correlation matches a reference or undeformed image with deformed image by the different types of correlation functions. The deform images can be generated by introducing the loading such as thermal and mechanical stresses on the pristine condition.

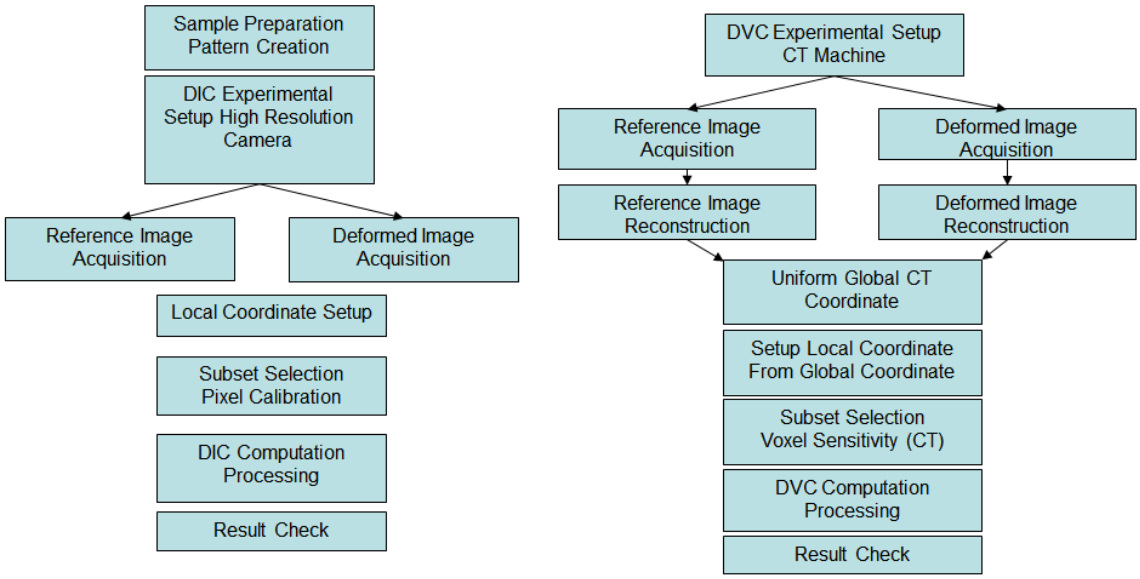


Figure 69 General DIC and DVC Procedures and Comparisons

Typically, the idea of Digital Volume Correlation measures interior displacements of an object. Both the DVC and the DIC use the concept of matching, which correlates each subset of the undeformed image to the deformed image. In the DIC, undeformed and deform images have been acquired by the DSLR camera that can have a high-resolution picture. Also, the random pattern has been created by scattering the uniform light on the sputtered surface (i.e. with rich random patterns). The DIC coordinator has been set up and based on choosing the zero displacements subset manually before the analysis. During the analysis, the searching algorithm starts at each pixel with its subset window, and searching direction will be moving only in-plane. (i.e. x-direction and y-direction) until the best matching subset has been found. This is a similar concept in the DVC. However, there is a complicated data structure and a complex implementation in the DVC, undeformed and deformed volumetric images have been generated by the X-ray micro-CT, which requires much more time to finish an entire 360 degrees CT scan. In the volumetric image, the random pattern has been naturally created by the property of X-ray

attenuation, which implies the X-ray will decay pass a dense material, and the density of material in the volume is entirely distinct in every direction at each rotation.

Then, those natural patterns have been recorded in the digital volume. Initially, the DVC does not have to manually choose the zero displacement subset, because there is an assumption that we assume the air around the scanned object does not change during the experiment, and the attenuation of the air is the same, so the X-ray can pass through the air freely with few attenuations. During the DVC analysis, each subset forms by a cubic window, which has enclosed the information in x, y and z-direction. Then, the searching algorithm will search the direction not only in the x-direction and y-direction but also search in the z direction. Therefore, the displacement field is complete and fully computed by finding the best matching subset in the 3D coordinator space. This images can be acquired by the X-ray CT machine in the CAVE3 lab. The following Figure 70 gives an example of 3D reconstruction object from the CT scan.

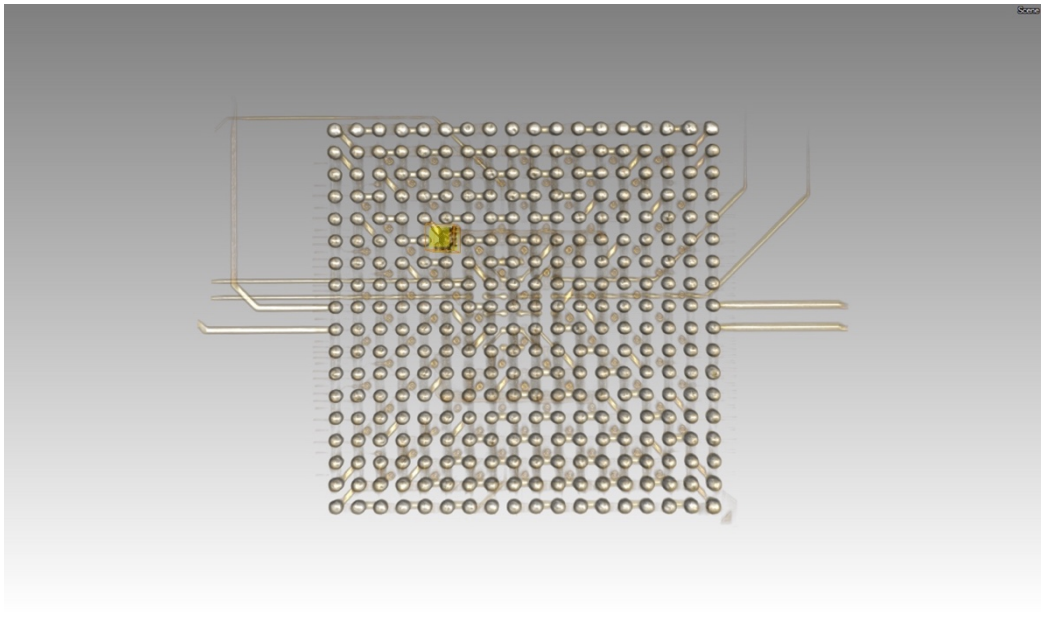


Figure 70 An X-ray CT Reconstructed Electronics Assembly

The idea for finding the correlation between the two images can be achieved by calculating the Cross-Correlation Coefficients between two subset image windows, and it decides the sensitivity of displacement component. Since the MATLAB/PYTHON-FORTRAN starts the iteration loops from the smallest index number, the reference point in the subset window can be chosen as the start index in the subset matrix for the simplification in programming. Ideally, the reference point would be chosen as the center of subset window. Initially, the corner reference point has been implemented, and then, we improved it with the center reference point for the accuracy of matching. Theoretically, the correlation algorithm starts at specific fixed subset window in the reference image, and it will search all possible subsets with correlation values in the deformed image. Therefore, the displacement components can be derived from the best matching point.

4.7.1 The Selection of Subset Size

Some scholars have done some researchers to achieve a high level of accuracy in the DVC, and they have tried to obtain an optimal subset size. Commonly, the larger subset sizes, the richer details are in each the subset, and the correlation value may be higher and less noisy. However, the large subset may lead to an error of deformation approximation by the first order and second order shape function (Pan & Xie 2008). Therefore, there is a trade-off for selecting the size of the subset. In the previous work, it shows that the small subset can reduce the systematic error of approximation to the shape function (Schreier & Sutton, 2002). Therefore, in the case of small deformation, the small subset size is used to estimate the shape function of deformation. In this work, the subset size will be chosen according to the experiments, and we had done some

calibration experiment to pick up the optimal subset size. Normally, we can choose the subset size from 19-19-19 to 41-41-41 for a small deformation measurement in the electronics.

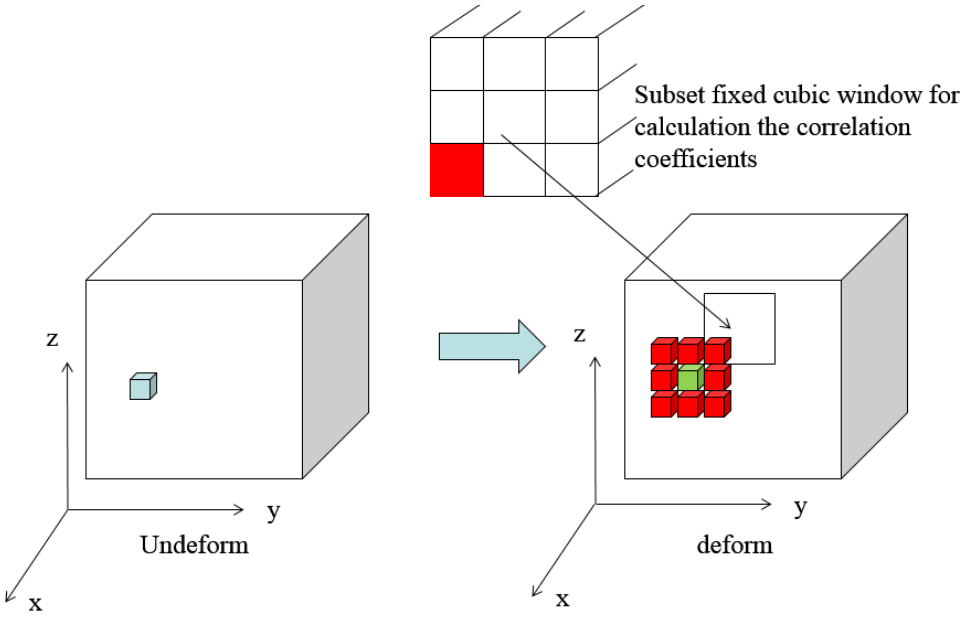


Figure 71 DVC Search Subset Size

4.7.2 Coarse Grid Search Algorithm

The coarse grid search is the technique to roughly find the grid displacement in x, y and z-direction, which provides the initial guess about how the object has been translated to the other locations. The goal is to map the deformed image to the undeformed image in the same global coordinate system. In the Figure 72, the undeformed subset shows in the blue grids, and the deformed subset shows in the green grids. The red arrow shows the deformation direction. In this case, for example, we take the smallest subset window size is 3-3-3, which has totally 27 voxels. The best-matched subset is shown in the deformed image. For each voxel and data point, we will repeat the same searching process to find the best match. The translation of the subset center

produces the displacement vectors. In the Figure 73 below, we can see that the P point is matched with the Q point in the same coordinator. Therefore, the relative displacement can be computed under the same coordinate system.

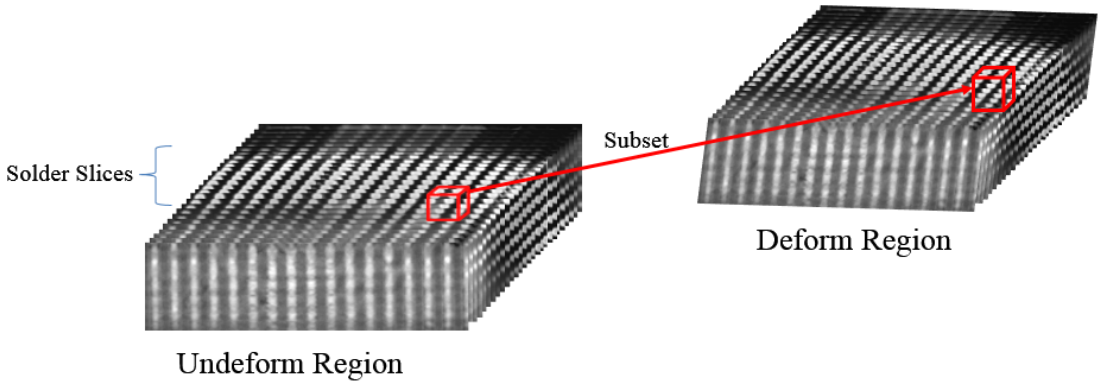


Figure 72 Search Illustrated in the Greyscale Image

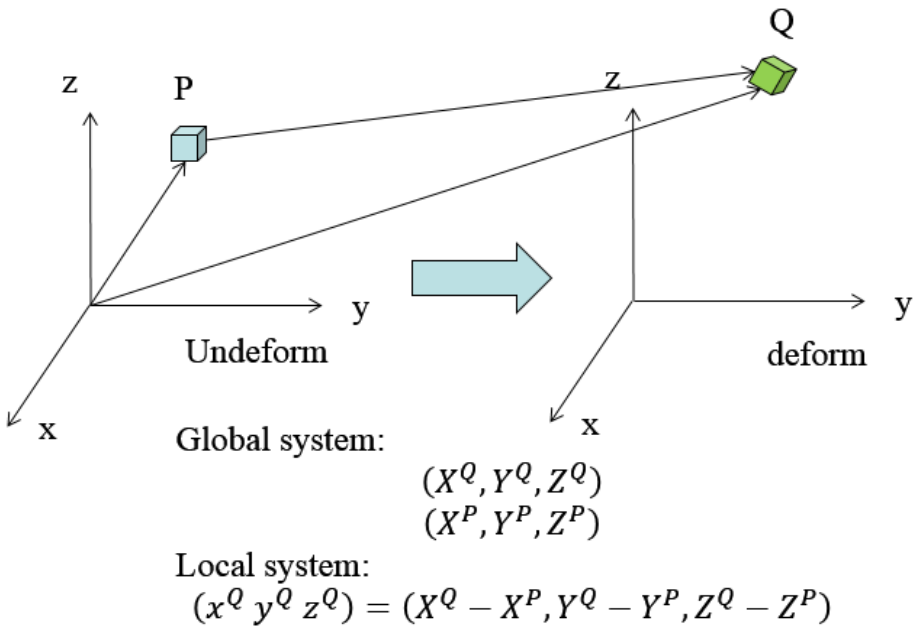


Figure 73 DVC Coordinator System Illustration

The basic search concept is that we use the subset to move in each direction: first in the y-axis, then in the x-axis, and later in the z-axis. In each direction, we evaluate and search for the smallest error of the Normalized Cross-Correlation Coefficients (i.e. 1-NCCCs), and find the best matching point (Xq, Yq, Zq), which is corresponded to (Xp, Yp, Zp) in the reference image (Figure 73).

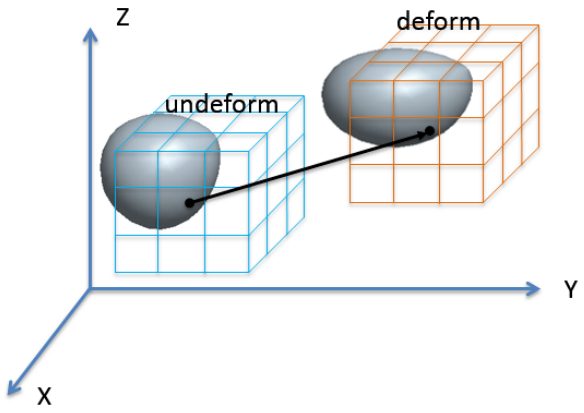


Figure 74 The Example 3-3-3 Subset Mapping

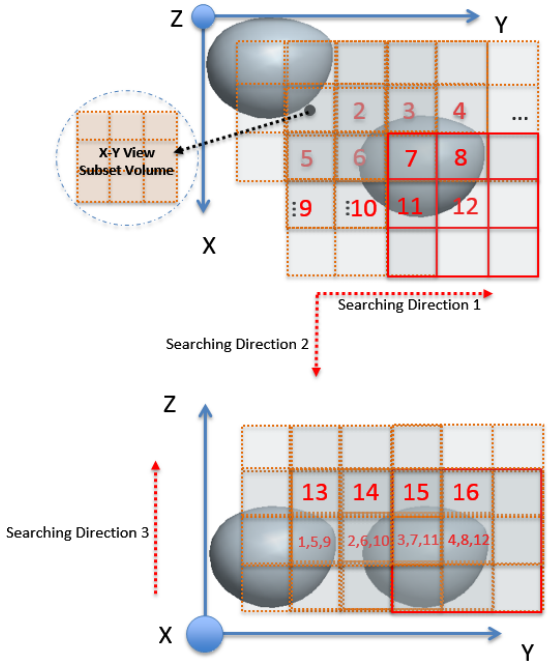


Figure 75 Subset Search Orientation

There are lots of algorithms to find the minimization of Error Normalized Cross-Correlation Coefficients: (1) exhaust searching algorithm (2) reduced searching algorithm (3) expanded searching algorithm (Gates & Lambros, 2011). The exhaust searching algorithm is computationally expensive, which it needs to examine each voxel in the deformed image. Usually, it is impossible to check every voxel, and this is because as the image size increases the searching time will increase as the 4th order of input image size (i.e.$O(N^4)$). Therefore, it is not feasible and practical to process a large volume with this search scheme. So, we can reduce the computation in each direction under the assumption that the maximum displacement in each direction is less than the searching range. It is less cost than the exhaust search algorithm, which the searching time is reduced dramatically. However, we still need an algorithm that can accommodate the searching range according to each searching step.

The expanded searching algorithm can automatically adjust the search range, and the search range will stop expanding once the optimal subset is found. The expanded searching algorithm works as the following: first it starts with the minimum cube size 1-1-1 and extends to 3-3-3, and then under the certain condition, it will expand the searching region to 5-5-5 cube size and further 7-7-7 cube size. We can see the following this algorithm in the following Figure76:

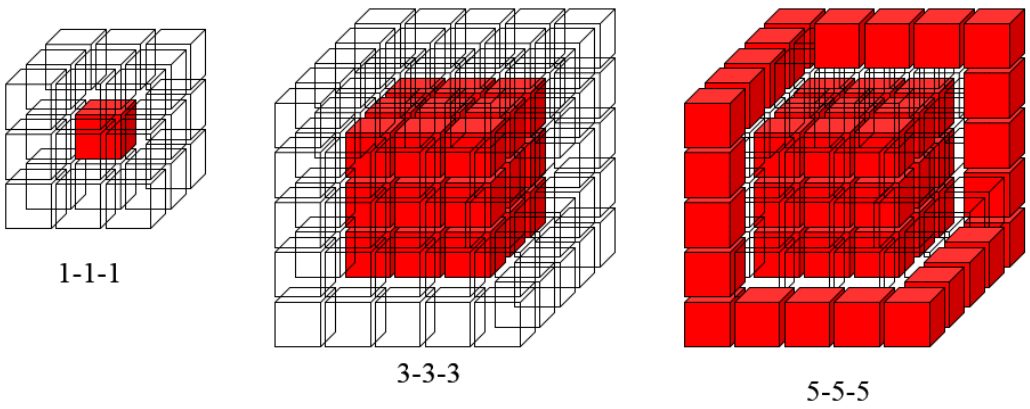


Figure 76 Expanded Search Algorithm

The extension condition is that we search for the minimum of Error Normalized Cross-Correlation Coefficients, which indicates the best matching between undeformed and deformed image. First of all, if the minimum value exists on the corner and edge of the searching voxels cubic region, we will further extend the searching region. In the following Figure 77, both the edge condition and corner condition demonstrate that there is possible that further minimizing the value (i.e. less than the current) of Error Normalized Cross-Correlation Coefficient exists in the outer region. Therefore, we continue to extend the search region until the minimum value of Error Normalized Cross-Correlation Coefficients is on the face of the region, which tells us that we should stop extending the searching area but searching the minimum value in the direction of the face. In a summary, this algorithm guarantees that the minimum value of Error Normalized Cross Correlation Coefficients has been contained and confined in the searching expanded voxels cubic region. The computational efficiency is higher than the exhausting search and reducing search. Notice that this algorithm does not guarantee the value is the global minimum.

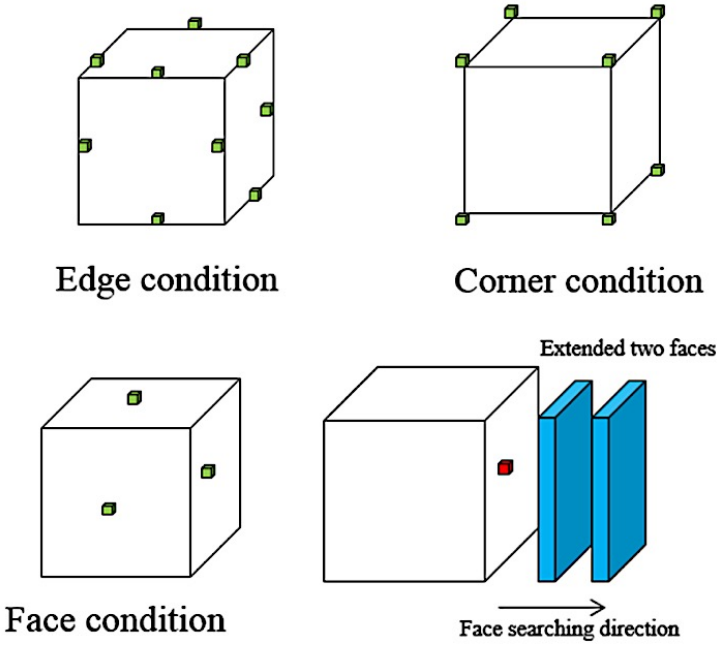


Figure 77 DVC Search Algorithm

The value for the Error Normalized Cross-Correlation Coefficient in the 7-7-7 voxel cube shows in the following Figure 78. It has been shown that ISO values of Error Normalized Cross-Correlation Coefficients. The values of Error Normalized Cross-Correlation Coefficients are not irregular and random, which there is a gradually decrement in the searching voxels cube. Therefore, the expanded searching algorithm will work well because of it contains the minimum value of Error Normalized Cross-Correlation Coefficients and guarantees the searching algorithm to find the global minimum. In the figure, the interior dark blue region shows the best match point.

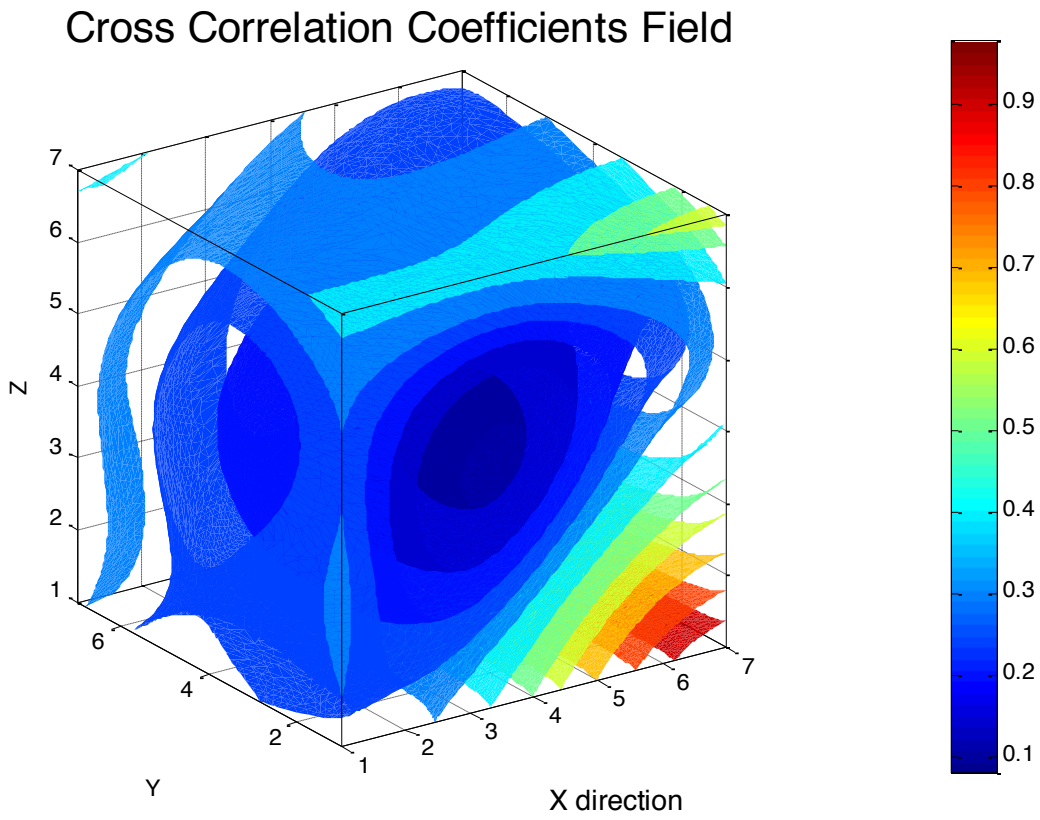


Figure 78 DVC Error of Normalized Cross Correlation Field

The goal of the initial search is to find a vector that roughly aligns the subsets, which provides the start point for the optimization process. A good initial displacement vector yields a

reliable start point for optimization. As we mentioned before, the basic scheme of initial search applies the block matching method. Once a close guess has been found, the sub-voxel resolution can be calculated. Also, a good initial alignment shows a better convergence for the optimization part, and if two sub-volumes are not matched and close enough, the gradient-based optimization method may not converge. Therefore, it is very important and critical to find a close deformed subset that matches our reference. The following Figure 79 shows one of the best initial matched subsets.

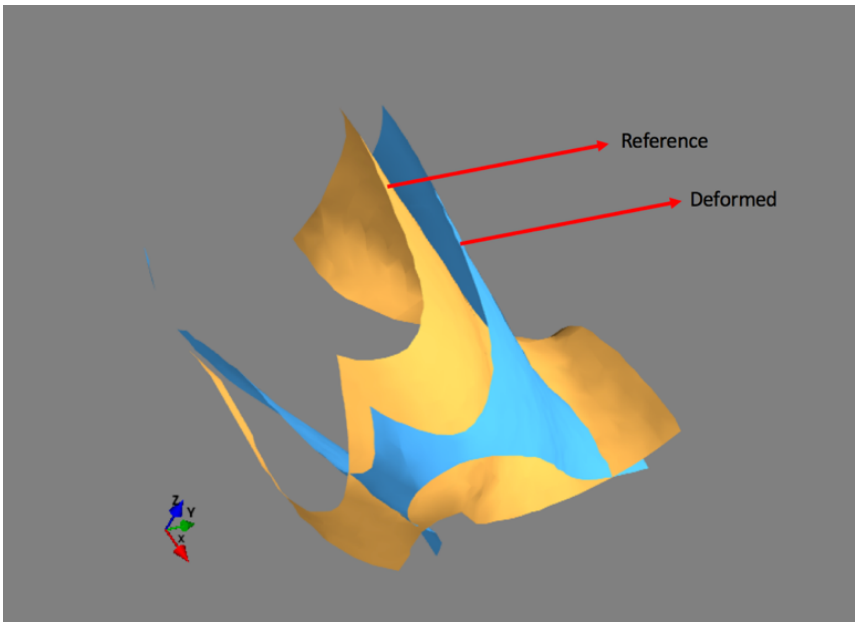


Figure 79 Best Initial Matched Two Subset Surfaces

4.7.3 Fine Displacement and Cross Correlation Coefficient

There is transformation function for the 3D material. The following Figure 80 can be used to illustrate the coordinate and match subset window in the DVC. The CT scanning axis provides the global fixed coordinate, in which two images have been acquired at the same fixture

configuration. The global coordinator has been labeled in the plot. The target subset window labeled as 'D', and the center of the subset labeled as 'X0'. The best matching subset window has been shown as the ' D' ' and its center reference point is ' X0' '. In the deformation, we express the material deformation as following equation:

$$\Phi(x) = X + U(X) = X + U(X_0) + \frac{\partial U}{\partial X}(X_0)(X - X_0) \tag{92}$$

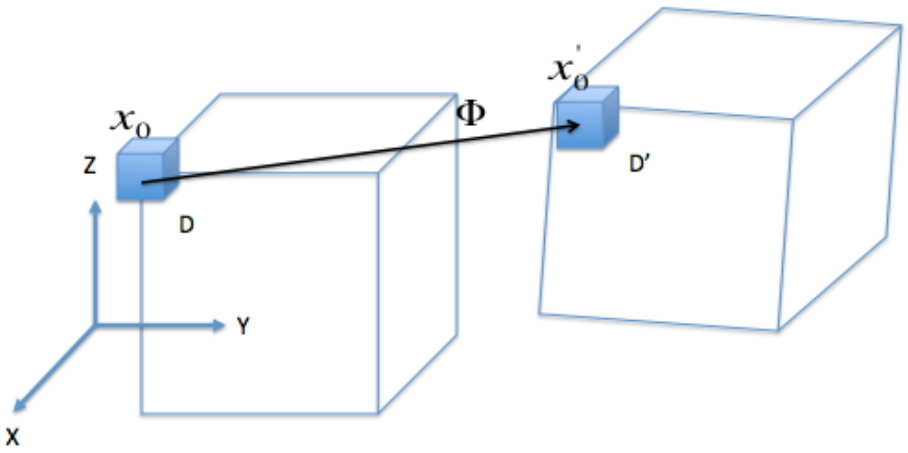


Figure 80 Subsets Mapping with Material Transformation

The displacement of the subset center defines as a parameter (u, v, w) in the rigid body translation. The local displacement gradient includes the rigid body rotation and the local stretch of the subset volume, and is characterized by the nine parameters (Smith, Bay, & Rashid, 2002):

$$(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial u}{\partial z}, \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y}, \frac{\partial v}{\partial z}, \frac{\partial w}{\partial x}, \frac{\partial w}{\partial y}, \frac{\partial w}{\partial z}) \tag{93}$$

Especially, if we ignore the strain or it is a small strain, we can approximate the displacement to the pure translation. Usually, if we take all the translation, stretch, and rotation into consideration, and we have three translation components (u, v, w) and nine rotation and stretch

components $(\frac{\partial u}{\partial x},\frac{\partial u}{\partial y},\frac{\partial u}{\partial z},\frac{\partial v}{\partial x},\frac{\partial v}{\partial y},\frac{\partial v}{\partial z},\frac{\partial w}{\partial x},\frac{\partial w}{\partial y},\frac{\partial w}{\partial z})$, which can be combined in a single vector form in the P:

$$\mathrm{P} = (\mathrm{u},\mathrm{v},\mathrm{w},\frac{\partial \mathrm{u}}{\partial \mathrm{x}},\frac{\partial \mathrm{u}}{\partial \mathrm{y}},\frac{\partial \mathrm{u}}{\partial \mathrm{z}},\frac{\partial \mathrm{v}}{\partial \mathrm{x}},\frac{\partial \mathrm{v}}{\partial \mathrm{y}},\frac{\partial \mathrm{v}}{\partial \mathrm{z}},\frac{\partial \mathrm{w}}{\partial \mathrm{x}},\frac{\partial \mathrm{w}}{\partial \mathrm{y}},\frac{\partial \mathrm{w}}{\partial \mathrm{z}}) \qquad (94)$$

The best parameters characterizing the approximation includes twelve components (three translations and nine rotations and stretches), which minimize the Error Normalized Cross Correlation Coefficient. It measures the degree of similarity of gray levels distributions in subset window D and its transformed one by the $\mathrm{X}' = \Phi(\mathrm{X})$. The following formula demonstrates how to calculate the Error Cross Correlation between two subsets, and this criterion is selected and is insensitive to the light fluctuations in the experiment (Xu, Moussawi, & Lubineau, 2015).

$$C = 1 - \frac{\sum_{X\in D}(f(X)-\bar{f}_D)\cdot(g(X')-\bar{g}_{D'})}{\sqrt{\sum_{X\in D}(f(X)-\bar{f}_D)^2}\cdot\sqrt{\sum_{X\in D}(g(X')-\bar{g}_{D'})^2}} \qquad (95)$$

$$C = 1 - \frac{\Sigma\Sigma\Sigma[f(x,y,z)-f_m]\cdot[g(x+u,y+v,z+w)-g_m]}{\sqrt{\Sigma\Sigma\Sigma[f(x,y,z)-f_m]^2\cdot\Sigma\Sigma\Sigma[g(x+u,y+v,z+w)-g_m]^2}} \qquad (96)$$

where : $\mathrm{X}' = \Phi(\mathrm{X})$

f(X): the greyscale value in the reference window;
g(x′): the greyscale value in the deformed window;
f_m: the mean value of reference window;
g_m: the mean value of deformed window;
u, v, w: search displacements in x, y, and z direction;

The general DVC algorithm can be described as following Figure 81: firstly, the grid-searching algorithm can find the initial grid translation, the second step is to search the neighbor

fractional voxels around the best match point. The optimization algorithm is introduced in the second step.

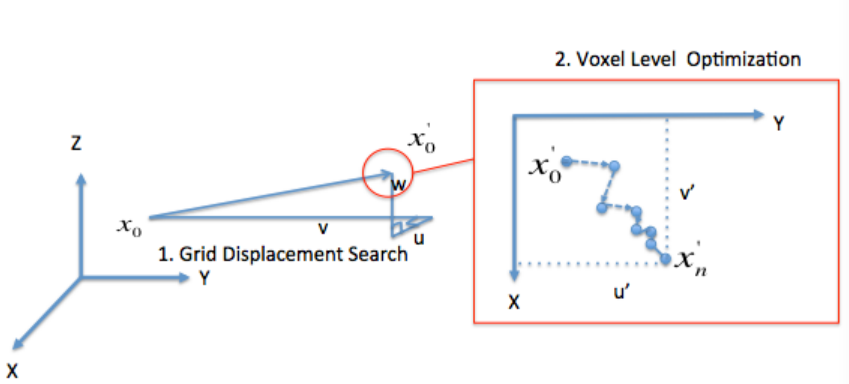


Figure 81 DVC Grid Search and Optimization

In the optimization part, a finer solution of P is searched with an automatic first-gradient minimization procedure by considering X_0' as a start point. Then, the minimization algorithm starts to search the neighbor of this point to find the best solution. The general gradient based method for the optimization algorithm is shown as following equation:

$$X_{n+1} = X_n - \alpha \nabla F_n \qquad (97)$$

∇F_n: first derivatives at the current location X

α: the step size coefficient

The 2D KLT optimization, explained previously in the Digital Image Correlation(DIC), can also be implemented in the 3D space for the optimization process of the Digital Volume Correlation (DVC). In this work, it shows the derivation of 3D KLT algorithm, and show its connection with the DIC as following derivation process.

The warp function in the 3D space should looks like the following matrix:

$$W_p = \begin{bmatrix} p_0 & p_3 & p_6 & p_9 \\ p_1 & p_4 & p_7 & p_{10} \\ p_2 & p_5 & p_8 & p_{11} \end{bmatrix} \tag{98}$$

From above matrix, there are 12 coefficients to be minimized during the optimization part, and the common way to find the solution is through the KLT method, which is based on the gradient descent. Another way is used the discrete optimization method such as heuristic search method including the Nelder-Mead Method and the Genetic Algorithm, and in this work, the basic Nelder-Mead algorithm has also been demonstrated in the next section.

4.7.4 3D KLT Implementation

In 3D case, the KLT tracker will be implemented, and one more dimension will be added and analyzed. The voxel movement in the 3D case would be like following (Wang & Jiang, 2016):

$$0 = I_t(p_i) + \nabla I(p_i) \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{99}$$

p_i: the pixel i

I_t: Greyscale Value Difference Between Two Scans

∇I: Images' Gradient along x direction, y direction and z direction

u: the displacement vector in x direction

v: the displacement vector in y direction

w: the displacement vector in z direction

All voxel movement in the 3D image would follow the linear system as following:

$$\begin{bmatrix} I_x(p_1) & I_y(p_1) & I_z(p_1) \\ I_x(p_2) & I_y(p_2) & I_z(p_2) \\ I_x(p_n) & I_y(p_n) & I_z(p_n) \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = - \begin{bmatrix} I_t(p_1) \\ I_t(p_2) \\ I_t(p_n) \end{bmatrix} \tag{100}$$

I_x: image gradient in x direction

I_y: image gradient in y direction

I_z: image gradient in z direction

It is equivalent to the following matrix multiplication:

$$A_{n\times 2} \cdot d_{2\times 1} = -b_{n\times 1} \tag{101}$$

The basic idea for matching two templates is to minimize the sum of square error over all voxels in the template, which gives the error cost function:

$$E(u,v,w) = \sum [I(x+u, y+v, z+w) - T(x,y,z)]^2 \tag{102}$$

$T(x,y,z)$: template image

The derivative of error function can be showed as Taylor series expansion:

$$E(u,v,w) = \sum \left[I\left(x+u,\ y+v, z+w\right) - T(x,y,z)\right]^2 \approx \sum \left[I(x,y,z) + \right. \tag{103}$$

$$\left. uI_x(x,y,z) + vI_y(x,y,z) + wI_z(x,y,z) - T(x,y,z)\right]^2 = \sum \left[uI_x(x,y,z) + \right.$$

$$\left. vI_y(x,y,z) + wI_z(x,y,z) + D(x,y,z)\right]^2$$

Similarly, we take partial derivative with vector (u, v, w), and set the equations to zero:

$$\frac{\partial E}{\partial u} = \sum \left[uI_x(x,y,z) + vI_y(x,y,z) + wI_z(x,y,z) + D(x,y,z)\right] I_x(x,y,z) = 0 \tag{104}$$

$$\frac{\partial E}{\partial v} = \sum \left[uI_x(x,y,z) + vI_y(x,y,z) + wI_z(x,y,z) + D(x,y,z)\right] I_y(x,y,z) = 0 \tag{105}$$

$$\frac{\partial E}{\partial w} = \sum \left[uI_x(x,y,z) + vI_y(x,y,z) + wI_z(x,y,z) + D(x,y,z)\right] I_z(x,y,z) = 0 \tag{106}$$

Then, we rewrite the above into a matrix form:

$$\sum \begin{bmatrix} I_x^2 & I_xI_y & I_xI_z \\ I_xI_y & I_y^2 & I_yI_z \\ I_xI_z & I_yI_z & I_z^2 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = -\sum \begin{bmatrix} I_xD \\ I_yD \\ I_zD \end{bmatrix} \tag{107}$$

$$D(x,y,z) = I(x,y,z) - T(x,y,z) \tag{108}$$

In 3D, we apply the same technique, which is in the Inverse Compositional Method as we mentioned in the last chapter. Every time the warp transformation is applied onto the reference subset:

$$E(u,v,w) = \sum[I(x+u, y+v, z+w) - T(x,y,z)]^2 = \quad (109)$$

$$\sum\left[I\left(W[x,y,z],\ P\right) - T(x,y,z)\right]^2$$

$$I(W(x,y,z), p+\Delta p) \approx I(W(x,y,z), p) + \nabla I \frac{\partial W}{\partial p}\Delta p \quad (110)$$

There are twelve warp constants to be evaluated as each pair of subsets:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1+p_1 & p_4 & p_7 & p_{10} \\ p_2 & 1+p_5 & p_8 & p_{11} \\ p_3 & p_6 & 1+p_9 & p_{12} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (111)$$

If we applied a chain rule and expand the Taylor as a matrix multiplication, we can get the following matrix multiplication:

$$I(x,y,z,p_1,p_2,p_3 \dots p_{12}) = \quad (112)$$

$$\begin{bmatrix} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial x} & \frac{\partial I}{\partial x} \end{bmatrix} \begin{bmatrix} \frac{\partial w_x}{\partial p_1} & \frac{\partial w_x}{\partial p_2} & \frac{\partial w_x}{\partial p_3} & \frac{\partial w_x}{\partial p_4} & \frac{\partial w_x}{\partial p_5} & \frac{\partial w_x}{\partial p_6} & \frac{\partial w_x}{\partial p_7} & \frac{\partial w_x}{\partial p_8} & \frac{\partial w_x}{\partial p_9} & \frac{\partial w_x}{\partial p_{10}} & \frac{\partial w_x}{\partial p_{11}} & \frac{\partial w_x}{\partial p_{12}} \\ \frac{\partial w_y}{\partial p_1} & \frac{\partial w_y}{\partial p_2} & \frac{\partial w_y}{\partial p_3} & \frac{\partial w_y}{\partial p_4} & \frac{\partial w_y}{\partial p_5} & \frac{\partial w_y}{\partial p_6} & \frac{\partial w_y}{\partial p_7} & \frac{\partial w_y}{\partial p_8} & \frac{\partial w_y}{\partial p_9} & \frac{\partial w_y}{\partial p_{10}} & \frac{\partial w_y}{\partial p_{11}} & \frac{\partial w_y}{\partial p_{12}} \\ \frac{\partial w_z}{\partial p_1} & \frac{\partial w_z}{\partial p_2} & \frac{\partial w_z}{\partial p_3} & \frac{\partial w_z}{\partial p_4} & \frac{\partial w_z}{\partial p_5} & \frac{\partial w_z}{\partial p_6} & \frac{\partial w_z}{\partial p_7} & \frac{\partial w_z}{\partial p_8} & \frac{\partial w_z}{\partial p_9} & \frac{\partial w_z}{\partial p_{10}} & \frac{\partial w_z}{\partial p_{11}} & \frac{\partial w_z}{\partial p_{12}} \end{bmatrix} \begin{bmatrix} \Delta p_1 \\ \Delta p_2 \\ \Delta p_3 \\ \Delta p_4 \\ \Delta p_5 \\ \Delta p_6 \\ \Delta p_7 \\ \Delta p_8 \\ \Delta p_9 \\ \Delta p_{10} \\ \Delta p_{11} \\ \Delta p_{12} \end{bmatrix}$$

If we apply an affine transformation warp function, the partial derivative of warp matrix is given as following:

$$\frac{\partial \mathrm{W}}{\partial \mathrm{p}} = \begin{bmatrix} \mathrm{x} & 0 & 0 & \mathrm{y} & 0 & 0 & \mathrm{z} & 0 & 0 & 1 & 0 & 0 \\ 0 & \mathrm{x} & 0 & 0 & \mathrm{y} & 0 & 0 & \mathrm{z} & 0 & 0 & 1 & 0 \\ 0 & 0 & \mathrm{x} & 0 & 0 & \mathrm{y} & 0 & 0 & \mathrm{z} & 0 & 0 & 1 \end{bmatrix} \tag{113}$$

The hessian matrix is:

$$H = \sum (\nabla \mathrm{I} \frac{\partial \mathrm{W}}{\partial \mathrm{p}})^{\mathrm{T}} \left(\nabla \mathrm{I} \frac{\partial \mathrm{W}}{\partial \mathrm{p}} \right) \tag{114}$$

The increment at each step is:

$$\Delta \mathrm{p} = \mathrm{H}^{-1} \sum \left[\nabla \mathrm{I} \frac{\partial \mathrm{W}}{\partial \mathrm{p}} \right]^{\mathrm{T}} [\mathrm{T}(\mathrm{x}) - \mathrm{I}(\mathrm{W}(\mathrm{x}; \mathrm{p}))] \tag{115}$$

Again, to implement this, we apply the Inverse Compositional Method as it has already been implemented in the 2D DIC.

If we apply the BFGS nonlinear algorithm for solving this system. The BFGS is more efficient compared to the Newton's gradient method. The BFGS named for the inventors (Broyden, Flecher, Goldfarb, Shanno). It builds an approximation B to the Hessian H, using only the first derivative information in the $\nabla \mathrm{F_n}$. In the BFGS algorithm, it updates the B matrix at each step, which guarantees that B is symmetric positive definition matrix at the search descent direction s. For quadratic function, BFGS will guarantee the convergence of B matrix at most n steps. BFGS often converges linearly and is widely considered one of the best general-purpose optimization methods.

BFGS Algorithm:

1.Input function C(u,v,w) and initialize the start point U=(u0,v0,w0)
2.Initialize the B matrix: B = I
3.For i = 1 to N do
4. Solve B*s = -∇C(U_n)
5. Let alpha =1: Fixed length in the searching direction
6. U(i+1) = U(i) + alpha*s;
7. If Converged then
8. Return U(i+1)
9. End if
10. $\mathbf{y = \nabla c(U_{i+1}) - \nabla(U_i)}$
11. $\mathbf{d = alpha \cdot s}$
12. $\mathbf{B_{i+1} = B_i + \frac{yy^T}{d^T y} - \frac{(Bd)(Bd)^T}{d^T Bd}}$
13.End for loop

The convergence region for the BFGS is much larger than the Newton method, which can allow the initial guess to be off to several voxels from the final minimum value. In general, the BFGS method is much more robust than the Newton method. For example, if we are trying to minimize the function below, in the figure it shows a quick convergence as the red dash line:

$$f(x, y, z) = (x + 3)^2 + (y + 6)^2 + (z + 11)^2 - 1 \qquad (116)$$

The x, y and z range for this function is:

$$x \in [-100\ 100];$$

$$y \in [-100\ 100];$$

$$z \in [-100\ 100];$$

The function field can be shown as ISO-surface with identical colors in Figure 82:

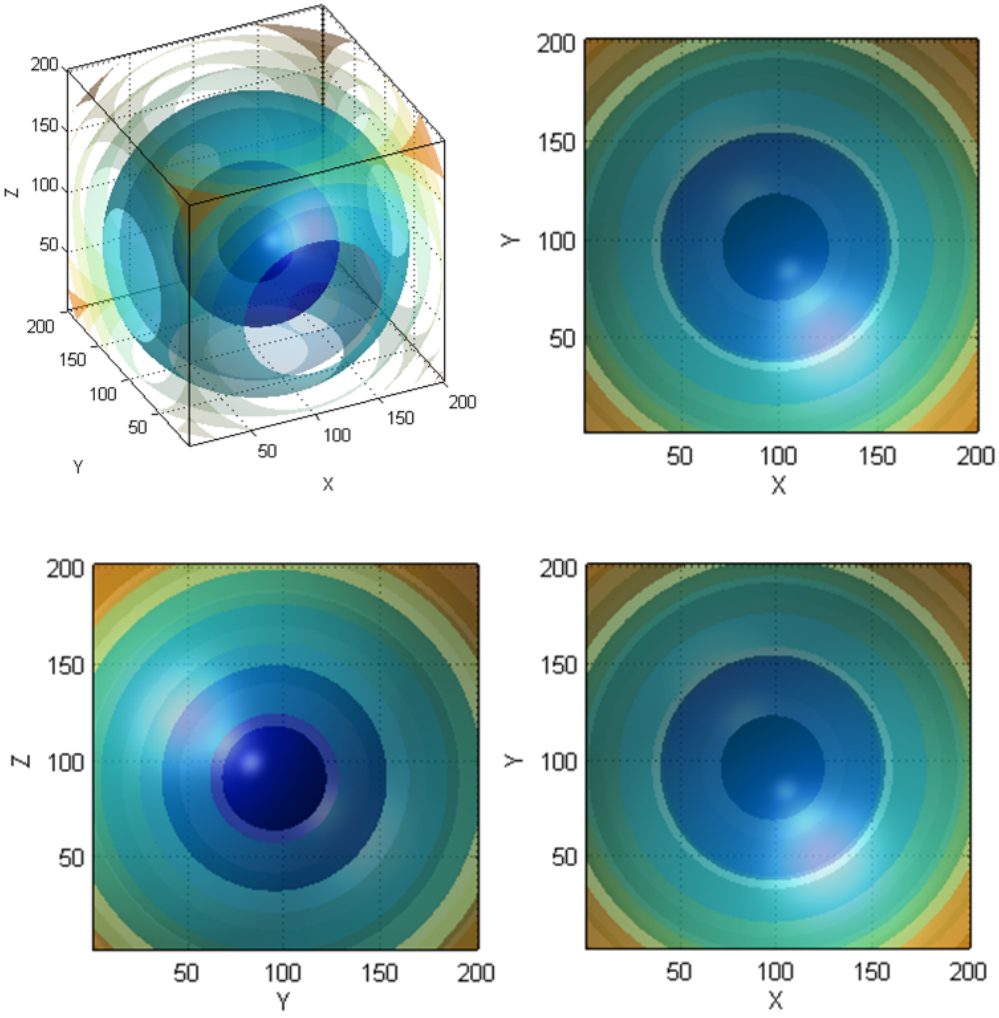


Figure 82 Sphere Field BFGS Example

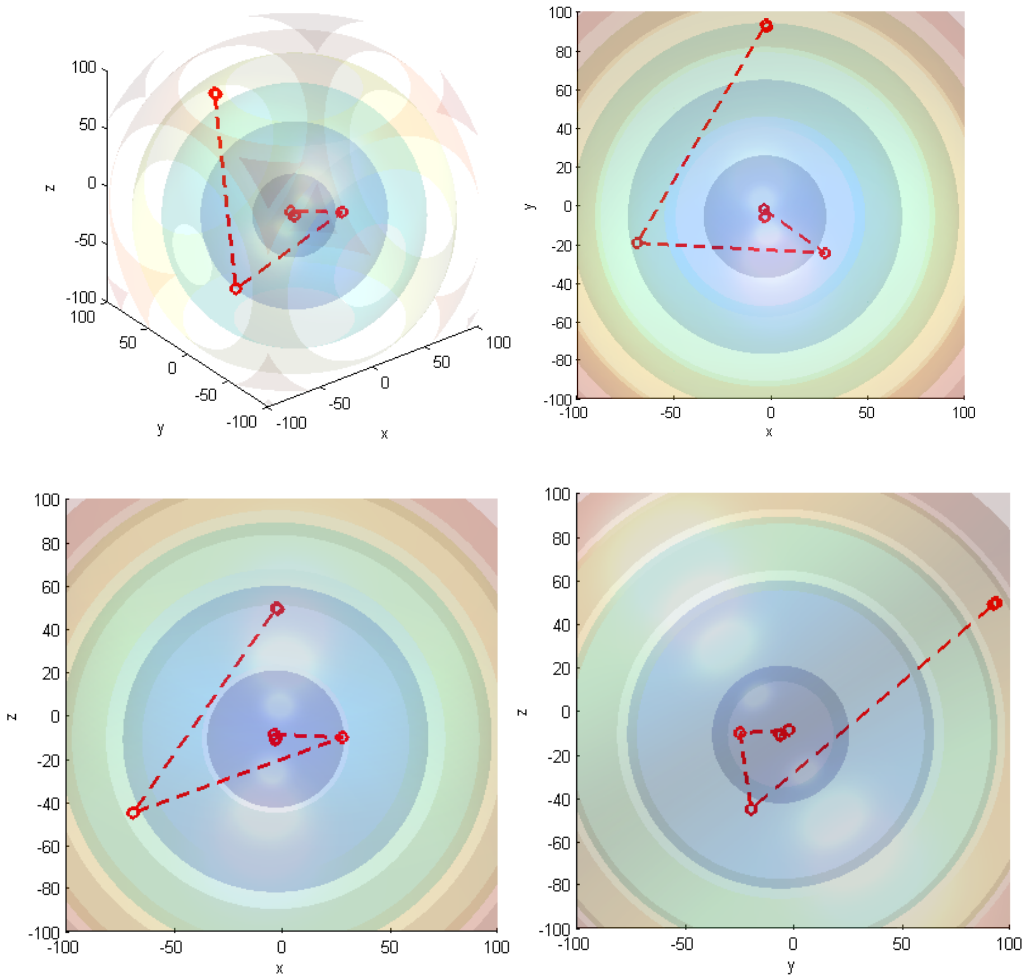


Figure 83 BFGS Convergence Plot

The Nelder-Mead (NM) method is another commonly used nonlinear optimization algorithm, which the Jacobian Matrix cannot be computed. The Nelder–Mead technique is a heuristic technique converging at a non-stationary point. Compared to the BFGS nonlinear method, the Nelder-Mead method convergence is much stable. Also, for the Nelder-mead, the computation is less than the Newton method, which often needs to compute interpolation and updates Jacobian matrix step by step.

The NM method uses the concept of a simplex, which is a special polytope of (N + 1) vertices in N dimensions. The examples include a line segment on a line, a triangle on a plane, a tetrahedron in three-dimensional space. In geometry, a simplex (e.g. plural simplexes) is a generalization of the notion of a triangle or tetrahedron to arbitrary dimensions. Specifically, a k-simplex is a k-dimensional polytope which is the convex hall of its vertices.

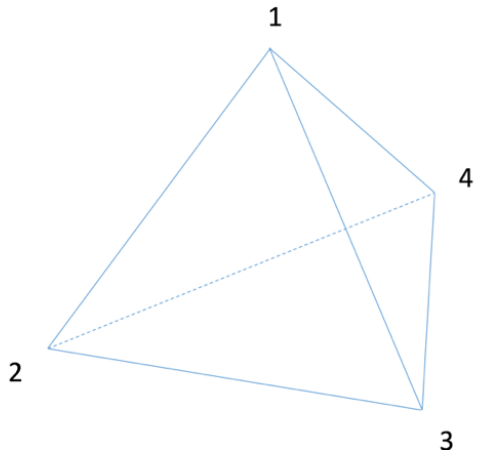


Figure 84 A regular-3-simplex or tetrahedron

A simplex method was first introduced by the Nelder and Mead (Nelder, Mead, 1965). In the plane x-y coordinate system, the shape of the simplex is a triangle, and three vertices value will be used and compared in the search. The worst vertex has largest value of f(x, y), which will be replaced as the new vertices. The searching process generates a sequence of triangles, from which the function value at the vertices get smaller and smaller. The size of the triangles is shrinking and reduced, and the coordinates of the minimum point are computed. The algorithm finds the minimum of a function with N variables. It is effective and computationally compact.

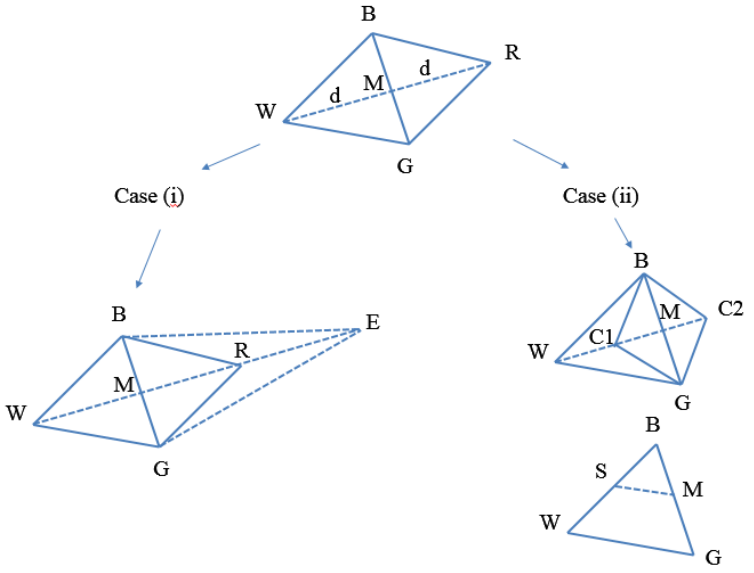


Figure 85 Nelder-Mead Algorithm Illustration

IF f(R) < f(G), Then Perform Case(i){either reflect or extend}
ELSE Perform Case(ii) {either contract or shrink}

```
BEGIN {Case(i)}
IF f(B) < f(R) THEN
    Replace W with R
ELSE
    Compute E and f(E)
    IF f(E) < f(B) THEN
    Replace W with E
        ELSE
    Replace W with R
        ENDIF
ENDIF
END {Case(i)}
```

```
BEGIN {Case(ii)}
IF f(R) < f(W) THEN
    Replace W with R
Computer C = (W + M)/2
or C = (M + R)/2 and f(C)
    IF f(C) < f(W) THEN
    replace W with C
    ELSE
    computer S and f(S)
    replace W with S
    replace G with M
    ENDIF
END {Case(i)}
```

Figure 86 NM-Optimization Algorithm

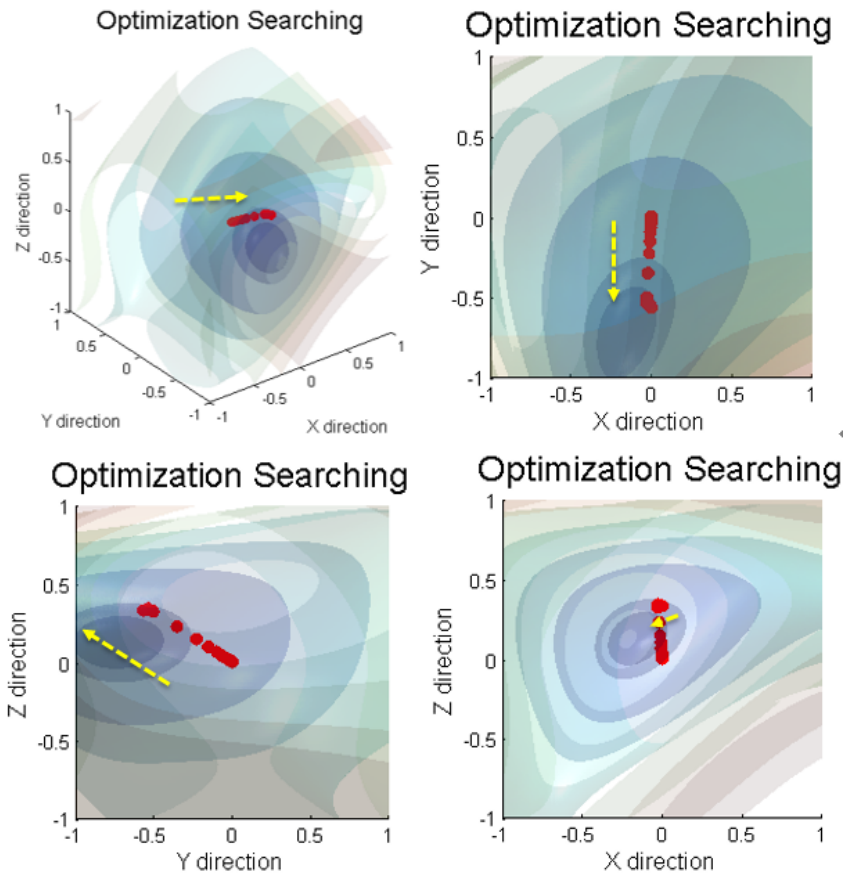


Figure 87 The Converge of NM

In the Figure 87, the yellow arrow is the direction of minimization search by the Nelder-Mead Method, and the red dots show the migration of the searched points. The surfaces in the figure show the ISO-values of the error cross-correlation field, and we can also see the gradients of the field from the different layers. The core in the middle shows the minimum value region of error normalized cross-correlation field. In each direction of the view window, we can visualize and check how the optimization works from time history and in the 4D (X-Y-Z-V) space.

The Non-linear Optimization searching time and converge property can be described as: it initiates a start point in the center of window as [0,0,0]. The optimization process terminates as the current x satisfies the termination criteria using tolerance of 1.0e-04 and f(x) satisfies the

convergence criteria using tolerance of 1.0e-04. The final evaluation is at coordinate (-0.0000, -0.5644, 0.3304), which function value is 0.0985. At the end of this chapter, we summary the DVC algorithm as the following flowchart:

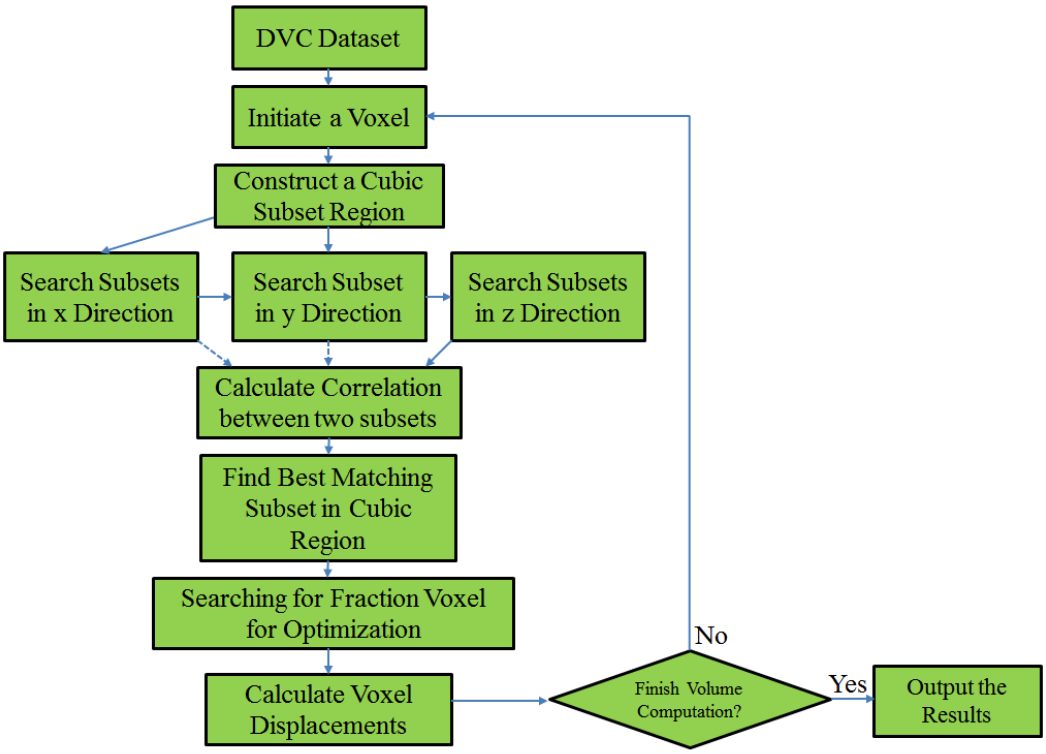


Figure 88 Illustration for the Digital Volume Correlation Algorithm

4.7.5 Strain Field Calculation

In the continuum mechanics (Bower, 2016), the object deformation follows a material transformation function. In the following Figure 89, the points P and Q are deformed to the new points.

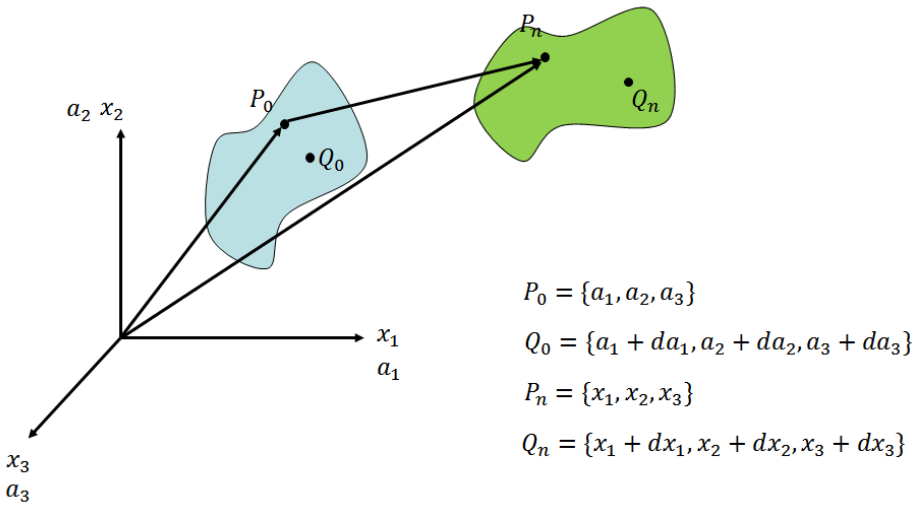


Figure 89 Object Transformation in Continuum Mechanics

The Strain Tensor and Green Strain are the following:

$$E_{ij} = \frac{1}{2}\left(\frac{\partial x_k}{\partial a_i}\frac{\partial x_k}{\partial a_j} - \delta_{ij}\right) \tag{117}$$

$$E_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial a_j} + \frac{\partial u_j}{\partial a_i} + \frac{\partial u_k}{\partial a_i}\frac{\partial u_k}{\partial a_j}\right) \tag{118}$$

The Almansi Strain shows as the following:

$$\mathrm{e_{ij}} = \frac{1}{2}\left(\delta_{\mathrm{ij}} - \frac{\partial \mathrm{a_k}}{\partial \mathrm{x_i}}\frac{\partial \mathrm{a_k}}{\partial \mathrm{x_j}}\right) \tag{119}$$

$$\mathrm{e_{ij}} = \frac{1}{2}\left(\frac{\partial \mathrm{u_i}}{\partial \mathrm{x_j}} + \frac{\partial \mathrm{u_j}}{\partial \mathrm{x_i}} + \frac{\partial \mathrm{u_k}}{\partial \mathrm{x_i}}\frac{\partial \mathrm{u_k}}{\partial \mathrm{x}}\right) \tag{120}$$

The Cauchy's Infinitesimal Strain shows as the following:

$$\mathrm{E_{ij}} = \mathrm{e_{ij}} = \frac{1}{2}\left(\frac{\partial \mathrm{u_i}}{\partial \mathrm{x_j}} + \frac{\partial \mathrm{u_j}}{\partial \mathrm{x_i}}\right) \tag{121}$$

The Gradient of Transformation:

$$F = \frac{\partial U}{\partial X} + I \tag{122}$$

$$\begin{aligned} dx_i &= F_{xx}dX_i + F_{xy}dY_i + F_{xz}dZ_i \\ dy_i &= F_{yx}dX_i + F_{yy}dY_i + F_{yz}dZ_i \\ dz_i &= F_{zx}dX_i + F_{zy}dY_i + F_{zz}dZ_i \end{aligned} \tag{123}$$

$$F = \begin{bmatrix} F_{xx} & F_{xy} & F_{xz} \\ F_{yx} & F_{yy} & F_{yz} \\ F_{zx} & F_{zy} & F_{zz} \end{bmatrix} \tag{124}$$

Then, the Green strain can be calculated as the following:

$$E = \frac{1}{2}(F^T F - I) \tag{125}$$

The field of displacement gradient that uses for calculating the strain components shows as the following:

$$\frac{\partial u}{\partial x} = u'_x = \frac{f(x + l_0) - f(x - l_0)}{2l_0} \tag{126}$$

$$\frac{\partial u}{\partial y} = u'_y = \frac{f(y + l_0) - f(y - l_0)}{2l_0} \tag{127}$$

$$\frac{\partial u}{\partial z} = u'_z = \frac{f(z + l_0) - f(z - l_0)}{2l_0} \tag{128}$$

$$\frac{\partial v}{\partial x} = v'_x = \frac{g(x + l_0) - g(x - l_0)}{2l_0} \tag{129}$$

$$\frac{\partial v}{\partial y} = v'_y = \frac{g(y + l_0) - g(y - l_0)}{2l_0} \tag{130}$$

$$\frac{\partial v}{\partial z} = v'_z = \frac{g(z + l_0) - g(z - l_0)}{2l_0} \tag{131}$$

$$\frac{\partial w}{\partial x} = w'_x = \frac{h(x + l_0) - h(x - l_0)}{2l_0} \tag{132}$$

$$\frac{\partial w}{\partial y} = w'_y = \frac{h(y + l_0) - h(y - l_0)}{2l_0} \quad (133)$$

$$\frac{\partial w}{\partial z} = w'_z = \frac{h(z + l_0) - h(z - l_0)}{2l_0} \quad (134)$$

$2l_0$: Gauge Length

The procedure of strain calculation follows the procedure:

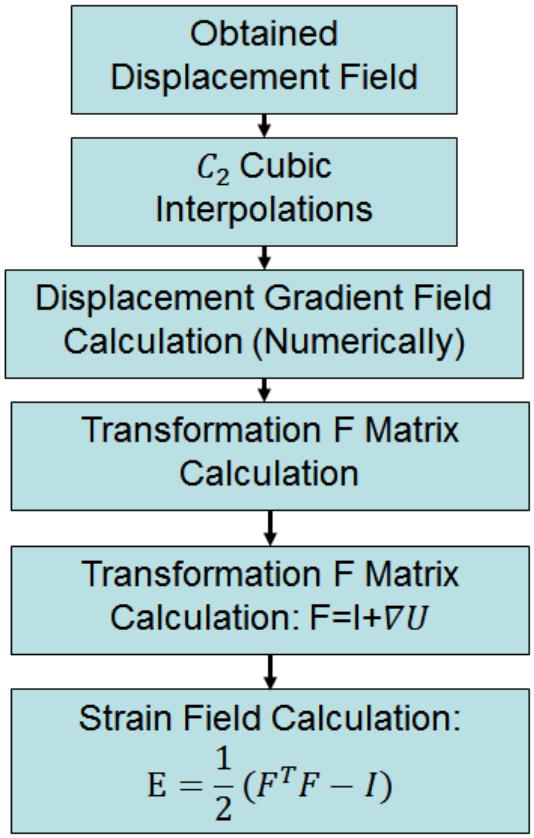


Figure 90 Strain Tensor Calculation Procedures

4.7.6 Digital Volume Correlation Software Construction

The Digital Volume Correlation (DVC) software has been constructed originally using the MATLAB. The MATLAB is a high-level programming language, which has abundant functions

and toolboxes. However, its efficiency is not very high. To improve computational efficiency in the DVC, we need to implement the Parallel Computation scheme in the MATLAB, and the following Figure 91 shows a parallel computing scheme in design the data separation. According to the Amdahl's law (David P, 1985), the theoretical speedup of the task can be calculated as two parts: (1) a parallel part (2) serial part. The parallel formula has been shown in the following:

$$S_{\text{latency}}(s) = \frac{1}{(1-p) + \frac{p}{s}} \tag{135}$$

S_{latency}: theoretical speedup of the execution of the whole work

s: the speedup of the parts benefited from parallel

p: percentage of execution time in parallel

According to this law, we divide the volume of 300-150-50 as the following Figure 91, in which more divisions in the large axis. So, the 300 in the x-axis has been divided into six parts. The y-axis has been divided into three parts, and z-axis has been divided into two parts. In this way, the division has been balanced and the minimized. Therefore, the parallel speedup could be high. The division may be set up manually according to different input volume size.

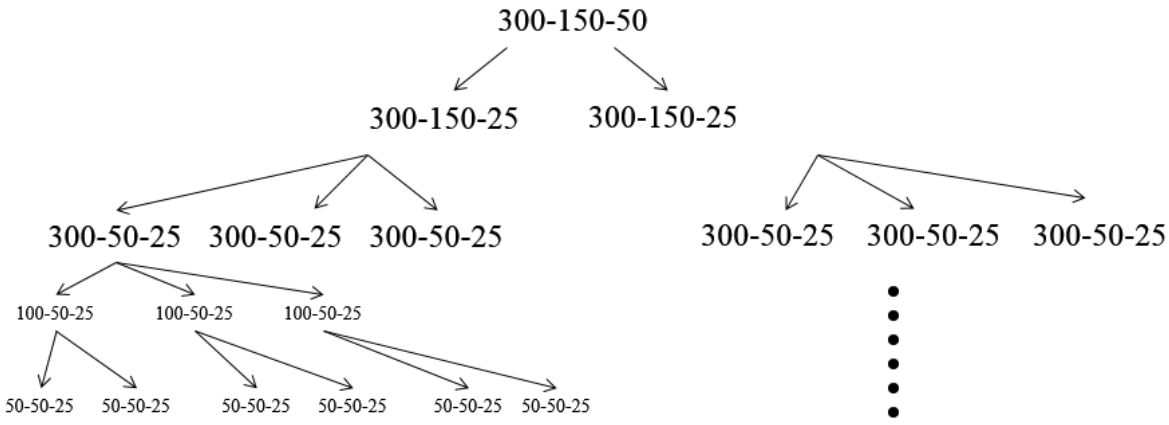


Figure 91 Example of Volume Division for Parallelism

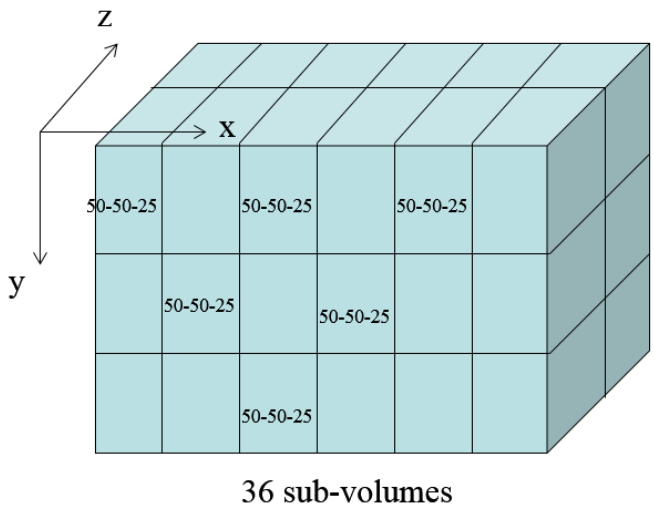


Figure 92 Sub-volumes Division Result

To implement the parallel computation, we use multiple MATLAB computational kernels, and each kernel takes charge of small volume in parallel. The number of the kernel can be decided by the power of the computational platform. In the following figure, it shows four kernels in the desktop computer, which has 3.2GHz CPU, 32GB RAM workstation. Figure 93 and Figure 94 below show the modules in the software.

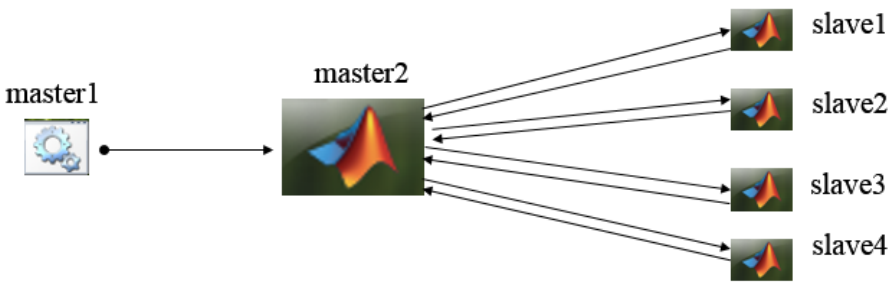


Figure 93 MPI Parallel Computation for MATLAB Implementation

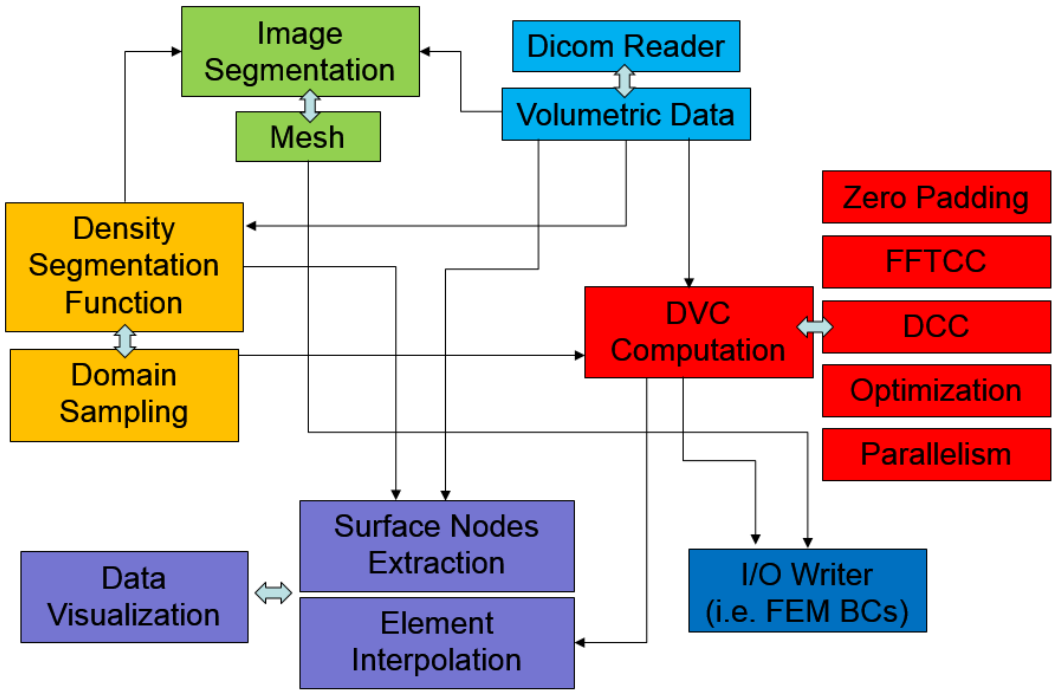


Figure 94 Digital Volume Correlation Software Modules

4.7.7 DVC Algorithm Virtual Calibration and Tuning:

In the development of DVC, we need to perform two-stage calibrations, and the first part is the process noise calibration. Another part of calibration is the measurement noise calibration. The CT data acquisition system may have variability and the CT reconstruction gray scale level

image can also have a white noise for the calculation. Therefore, we need to know and quantify those two types of noises in the DVC technique that we have developed.

At first, the process noise calibration needs to use the duplicate images for the DVC software. So, we use the same control and experimental volume, which is identical to each other. Therefore, we can predict the displacement field would be ideal zero- displacement. Secondly, the measurement noise calibration needs to use two different images, but those images would be generated under the same experimental conditions. Ideally, there is not displacement between those images. However, the data acquisition system will produce the white noise, which would be imbedded into the image data. Therefore, we also need to perform the measurement noise quantification to check the variability of the DVC processed results.

To verify the DVC algorithm, we have used the simulated random texture volume size 50-50-50, and we applied a uniform tensile strain 0.1 in each direction x, y and z. From the computational, we got the maximum displacement in each direction is five voxels, which matches with the input is known deformation. Moreover, we also input the mixed strains to verify the algorithm, and the combination of E12, E13, and E23 has been shown in the following Figure 97, Figure 98 and Figure 99.

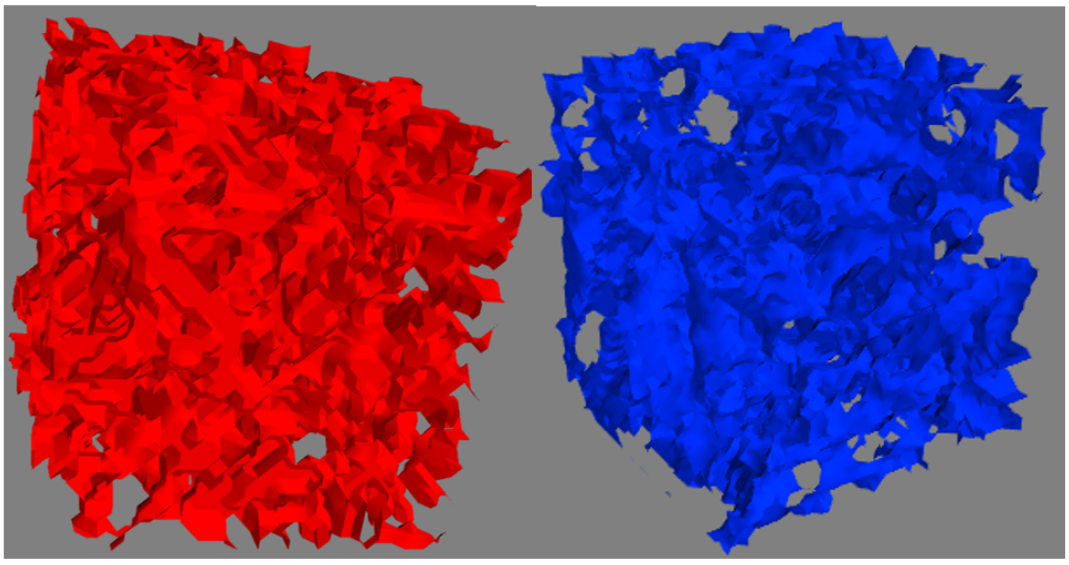


Figure 95 Two Simulated Local Material Noisy Volumes size 50-50-50

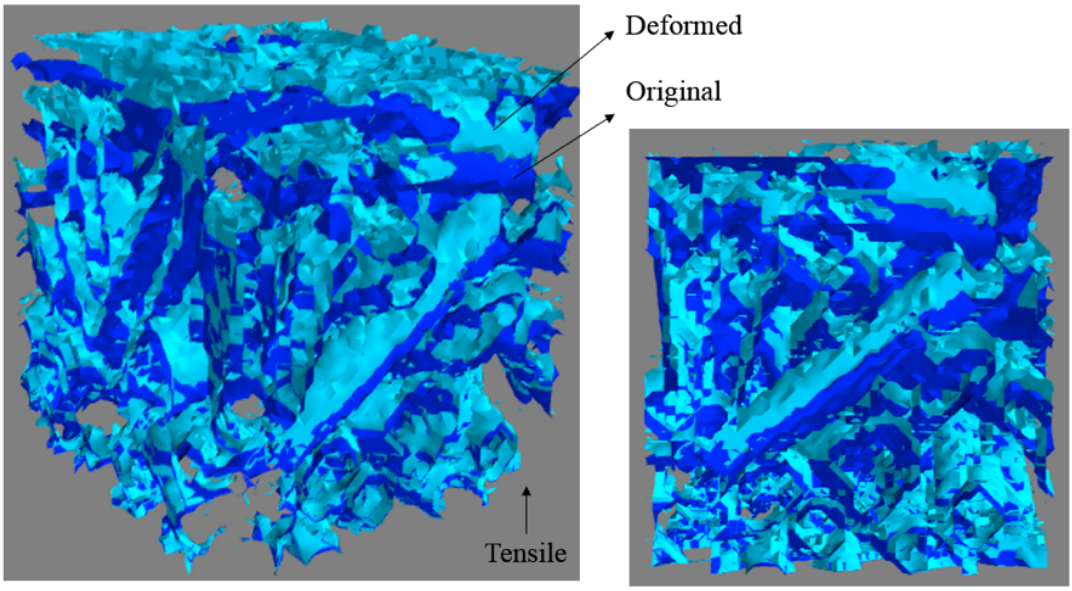


Figure 96 Applied Uniform 0.1 Tensile Strain

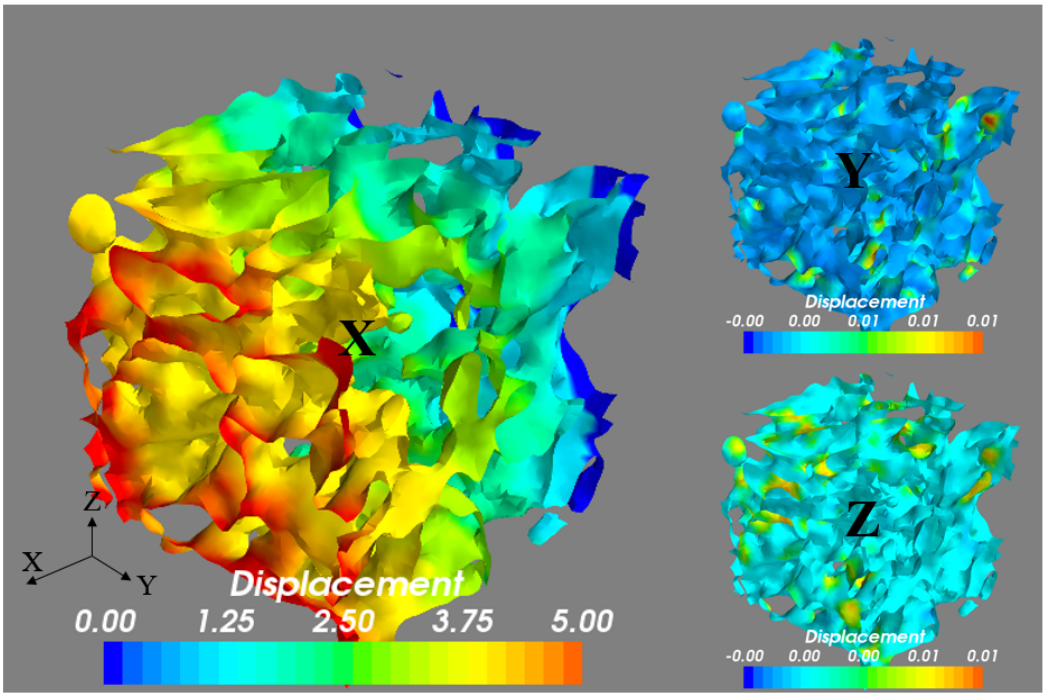


Figure 97 Calculated Deformation in X direction (e_{11}=0.1)

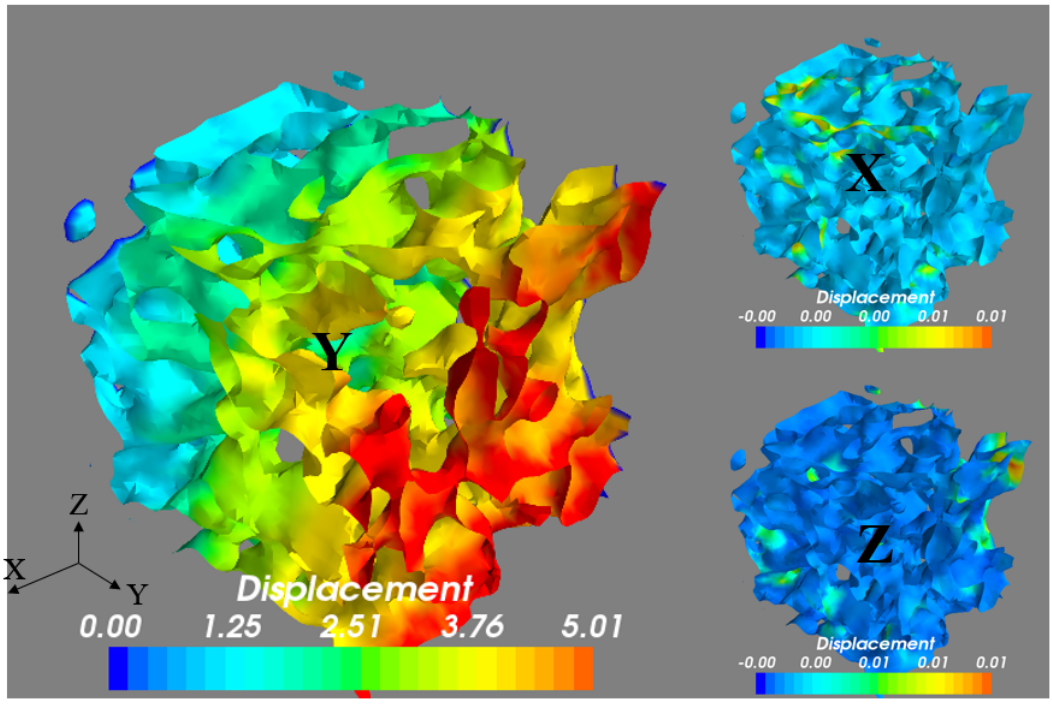


Figure 98 Calculated Deformation in Y direction (e_{22}=0.1)

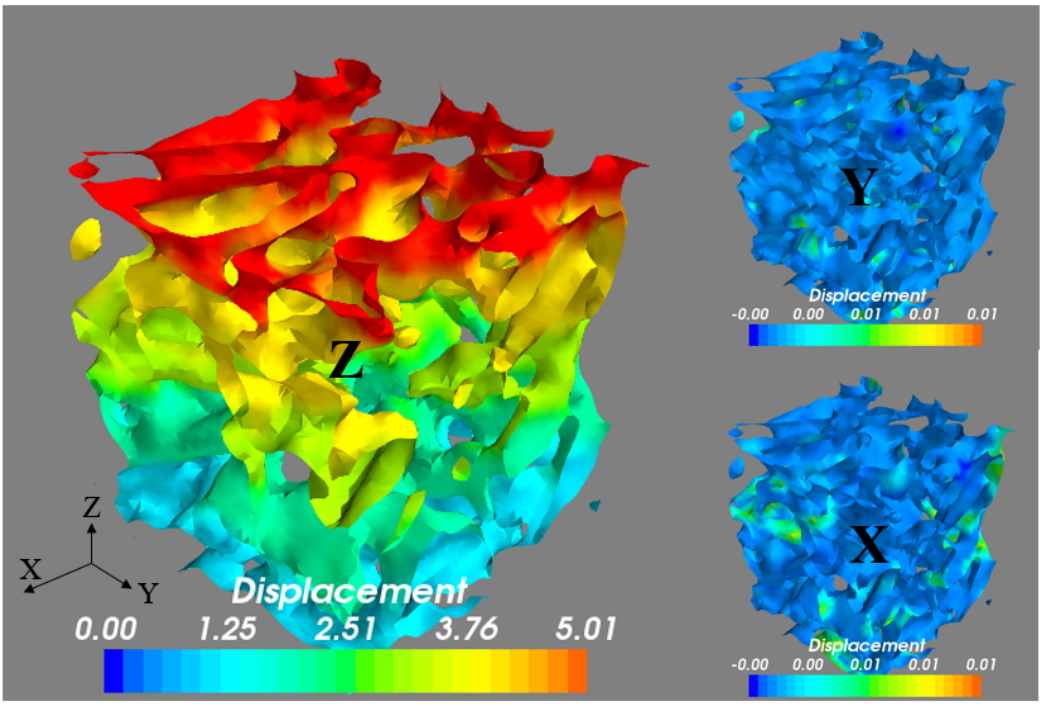


Figure 99 Calculated Deformation in Z direction (e_{33}=0.1)

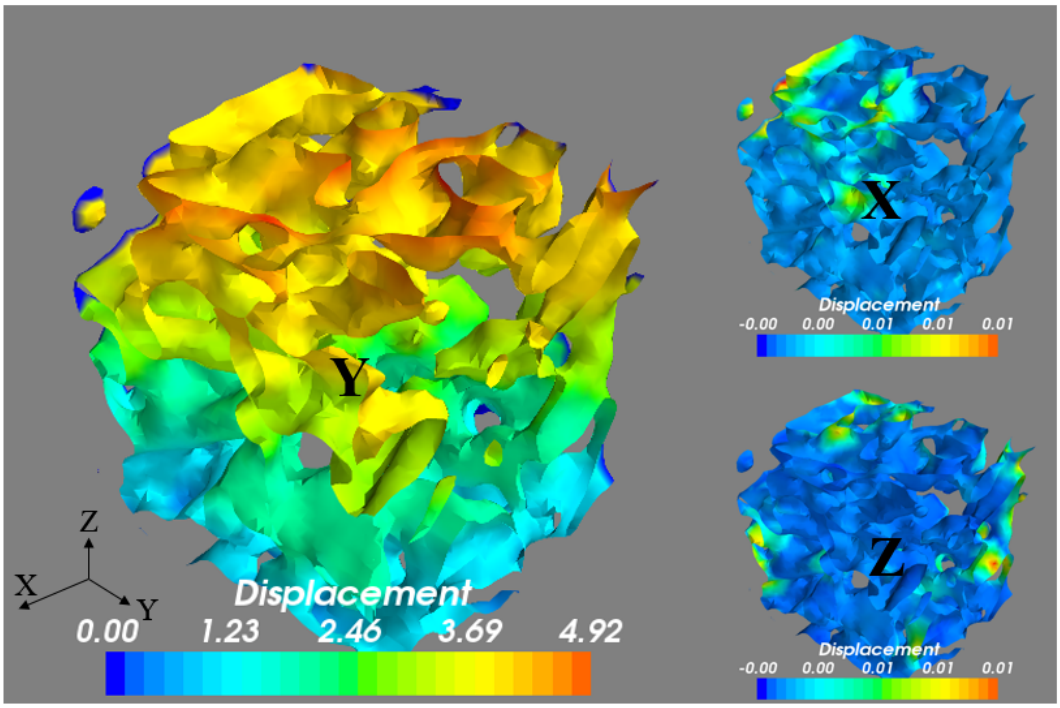


Figure 100 Calculated Deformation in Y direction (e_{23}=0.1)

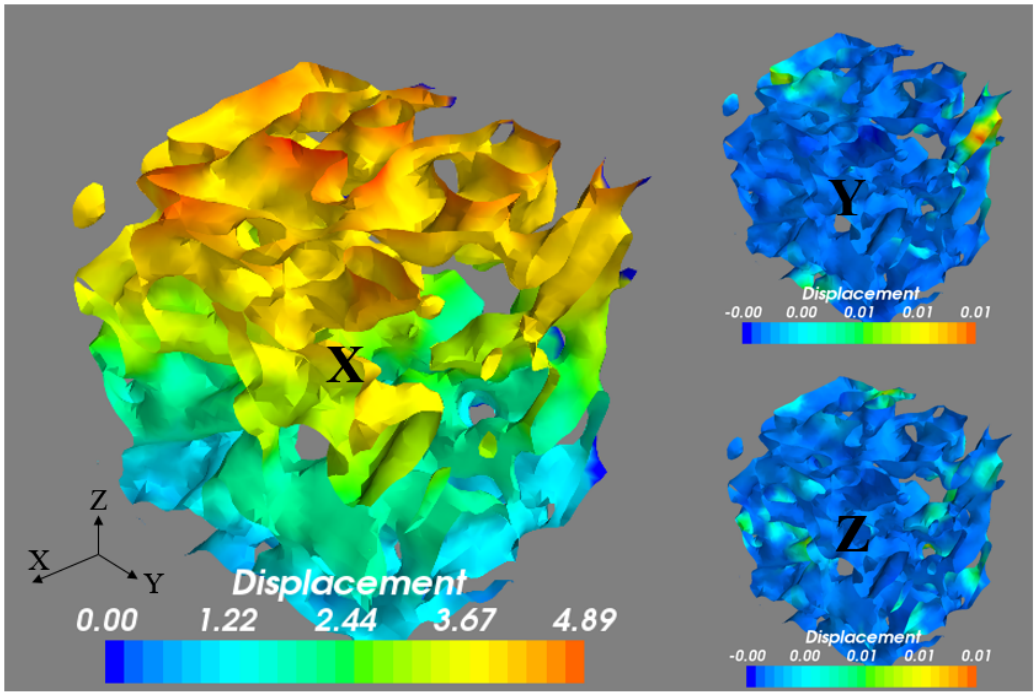


Figure 101 Calculated Deformation in X direction (e_{13}=0.1)

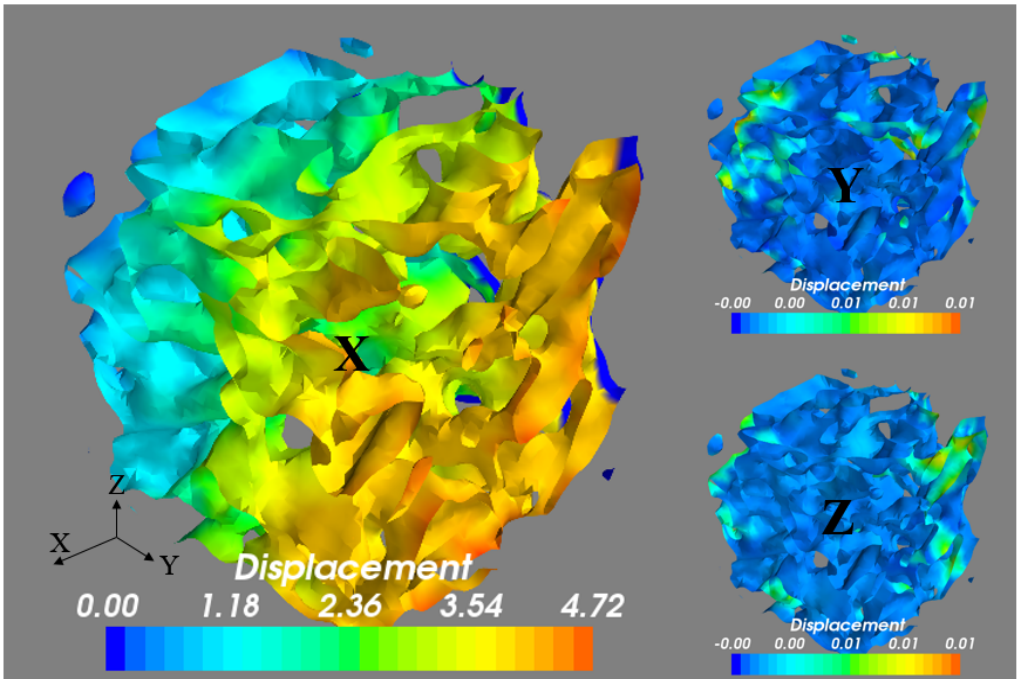
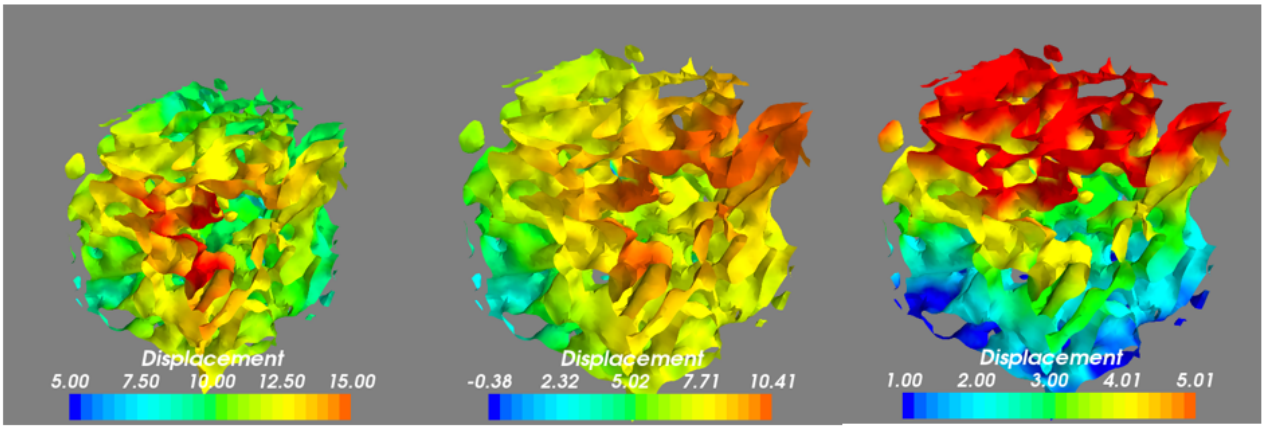


Figure 102 Calculated Deformation in X direction (e_{12}=0.1)



X **Y** **Z**

$$\begin{bmatrix} 1.1 & 0.1 & 0.1 \\ 0 & 1.1 & 0.1 \\ 0 & 0 & 1.1 \end{bmatrix}$$

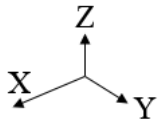
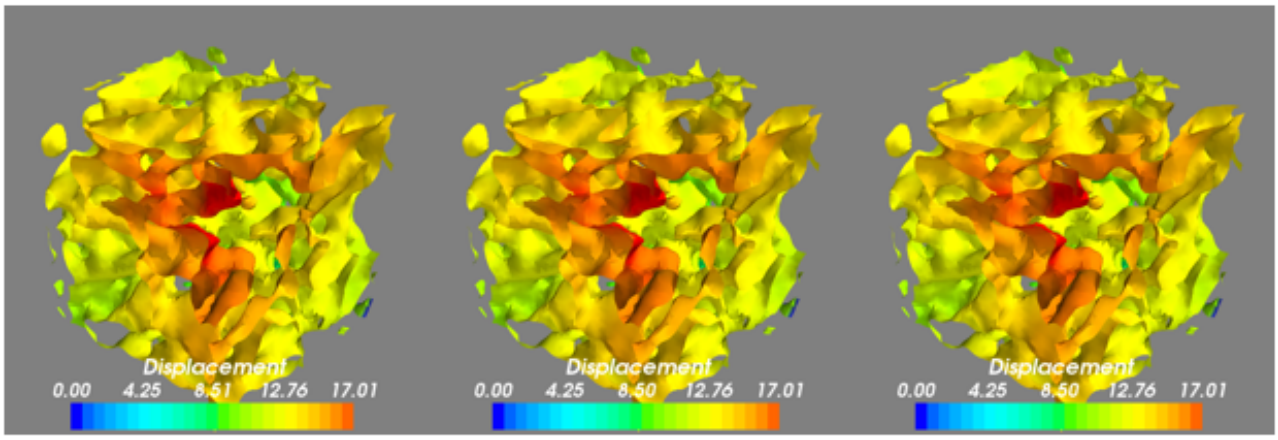


Figure 103 Calculated Mixed Deformation in X, Y and Z Direction1



X **Y** **Z**

$$\begin{bmatrix} 1.1 & 0.1 & 0.1 \\ 0.1 & 1.1 & 0.1 \\ 0.1 & 0.1 & 1.1 \end{bmatrix}$$

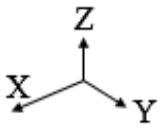


Figure 104 Calculated Mixed Deformation in X, Y and Z Direction2

4.7.8 DVC In-Situ Experimental Calibration:

In the DVC calibration, we have applied two types of DVC calibration to verify the DVC algorithm and program. The first one is experimental zero displacement images correlation, which is the correlation between two zero loading images. This process of calibration provides us how much noise that has been contained in the experiment. We called this experimental noise quantification. The second one is pseudo-displacement calibration, which uses the duplication from uniform one as a deformed image, which is ideal and perfect zero displacement between the two images. This process of calibration provides us the information about how the DVC algorithm works, and we check it as process noise verification.

First of all, we design a bending test fixture to test the DVC program, and the testing sample is a pure aluminum bar. The testing configuration shows below:

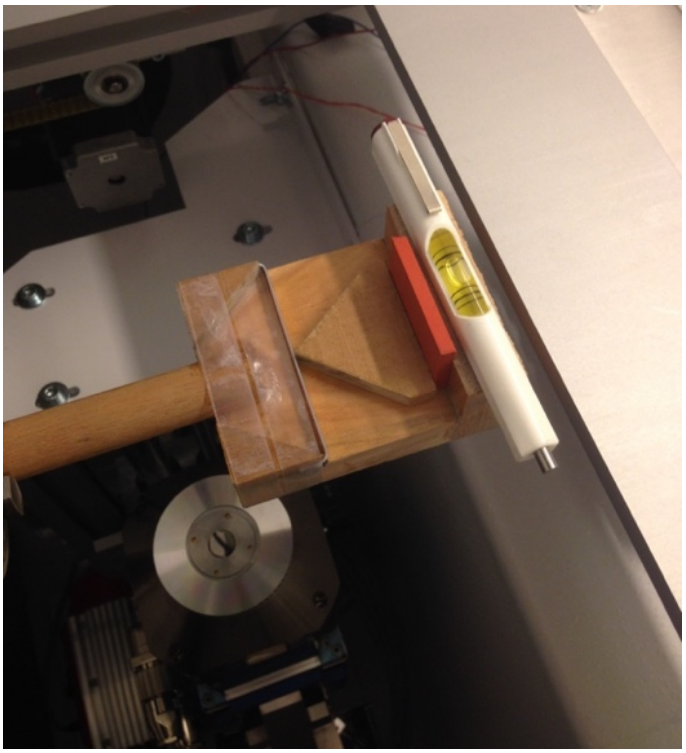


Figure 105 DVC Scan Preparation 1

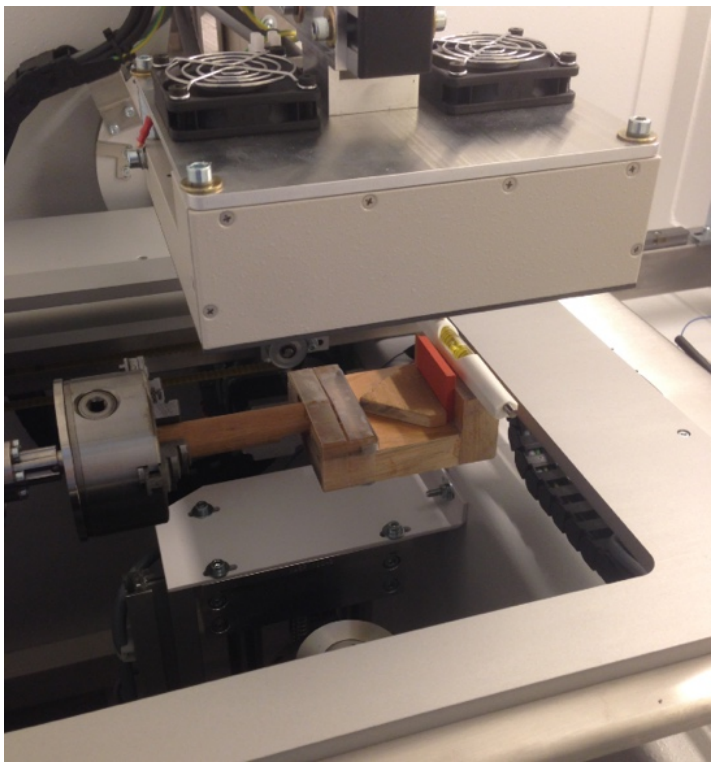


Figure 106 DVC Scan Preparation 2

Calibration Stage	**Calibration Target**
In-Situ Zero Displacement Calibration	Experiment Noise
In-Situ Pseudo Zero Displacement Calibration	Process Noise

Table 5 DVC Calibration Guide Table

At first, the process noise calibration needs to use the duplicate images for the DVC software. So, we use the same control and experimental volume, which is identical to each other. Therefore, we can see the displacement field would be ideal zero. Secondly, the noise calibration requires to use two different images, but those images would be generated at the same experimental condition. So, there is not displacement between those images. However, the data acquisition system will generate some white noises, and it would be added to the image. Therefore, we also need to perform the noise quantification to check the variability of the DVC computation. Also,

there are two types of configuration for the calibration process; the first one is fixture based calibration. We assume that the scan including the fixture would pose some noise into the reconstruction process. Because of rotation during the CT scan, only a part of the fixture will be seen in the scan projection images, which introduces some errors to the object.

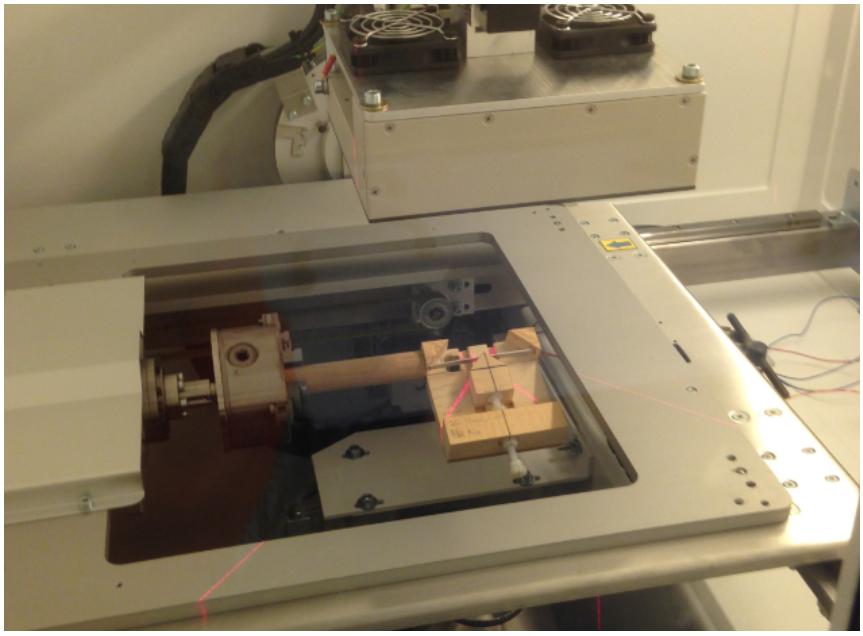


Figure 107 Fixture Base Calibration

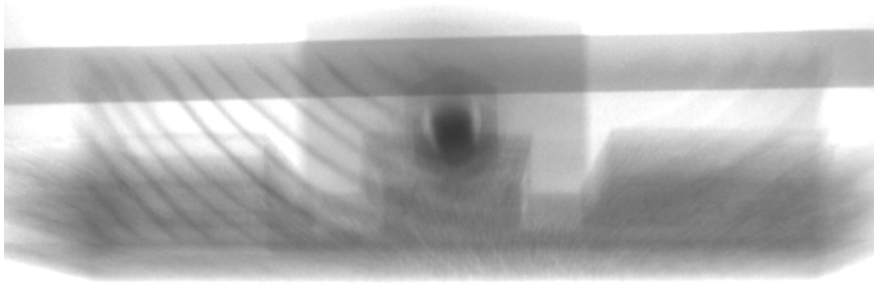


Figure 108 Fixture Base Scan

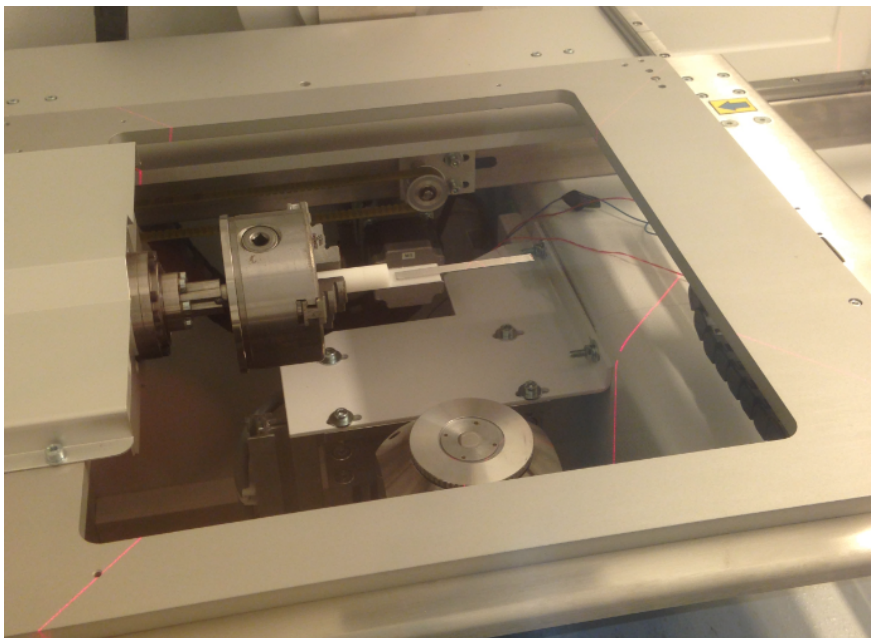


Figure 109 Air Base Calibration

Figure 110 Air Base Scan

4.7.8.1 Fixture Base Calibration:

In the first stage, the displacement calibration images show below(e.g. ISO surface value = 15000). We can see that in the most places the displacement is equal to zero except some variance on the edge of the boundary. This experimental noise may come from the Volume Graphics volume reconstruction, and we use the quick scan with about 500 projections. In the meantime, along with the edge, there is less information, which creates confusion during the subset correlation.

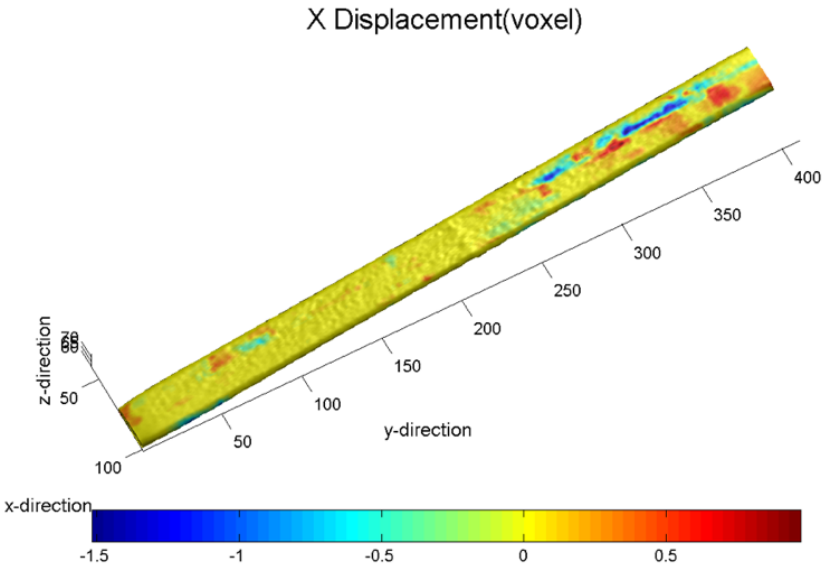


Figure 111 X Displacement of Fixture Base Calibration

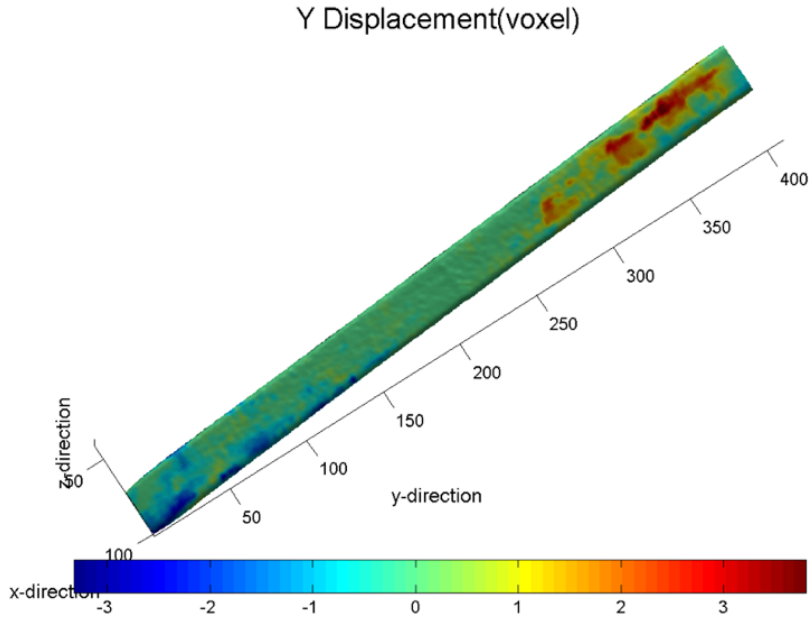


Figure 112 Y Displacement of Fixture Base Calibration

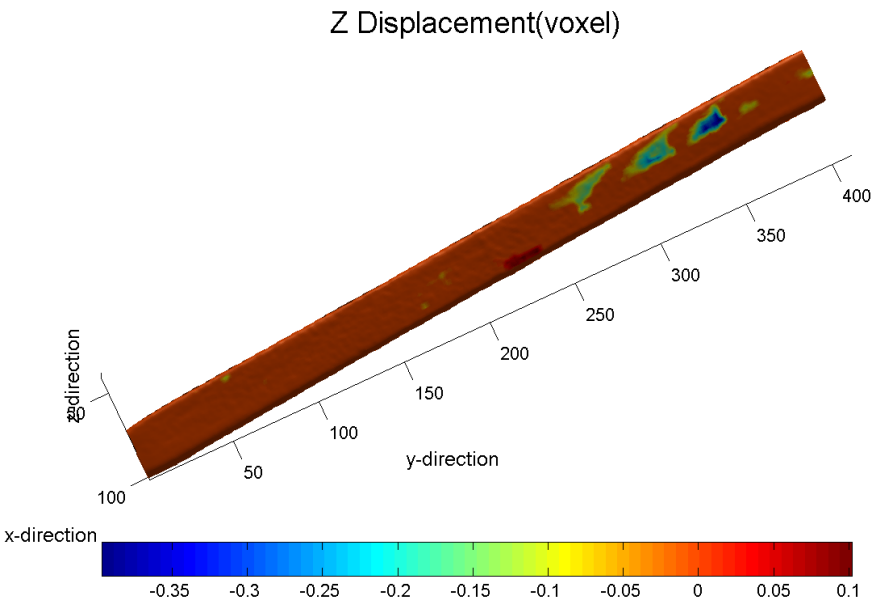


Figure 113 Z Displacement of Fixture Base Calibration

The second stage calculation shows below, which quantifies the process noise. In the second stage, the process noise quantification, we see that the DVC algorithm captured the zeros very well, which all of the displacements in the three directions are in the perfect condition.

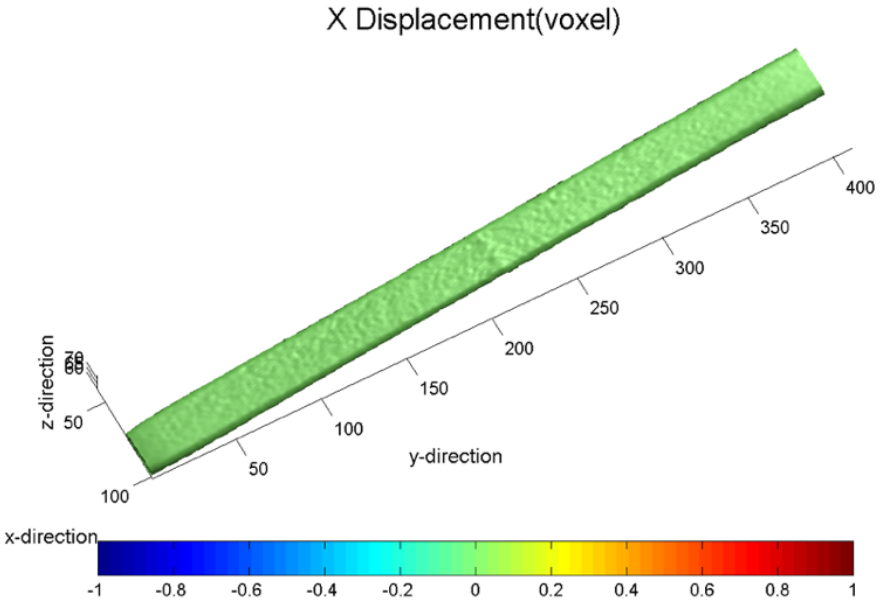


Figure 114 X Displacement of Pseudo Zero Calibration

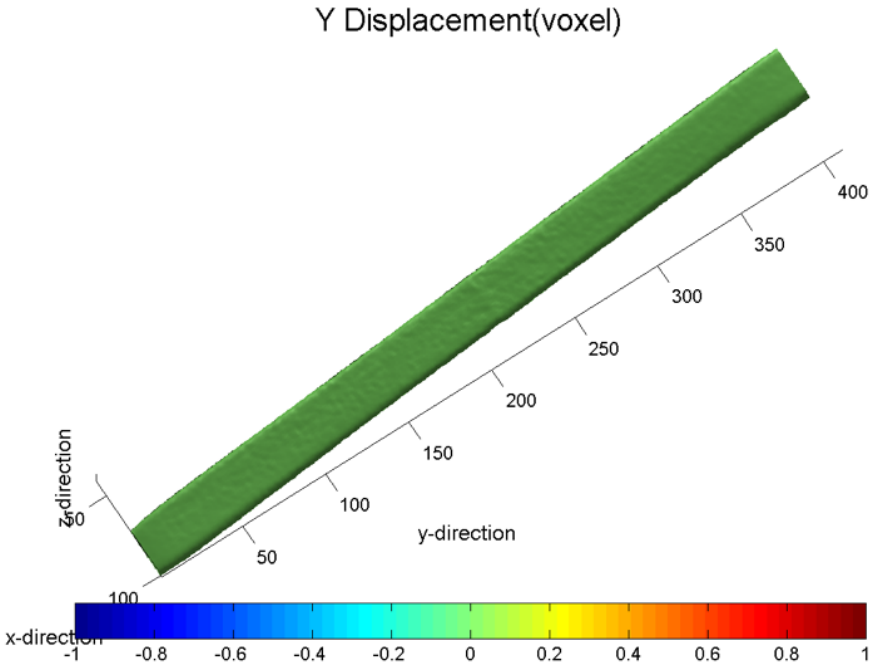


Figure 115 Y Displacement of Pseudo Zero Calibration

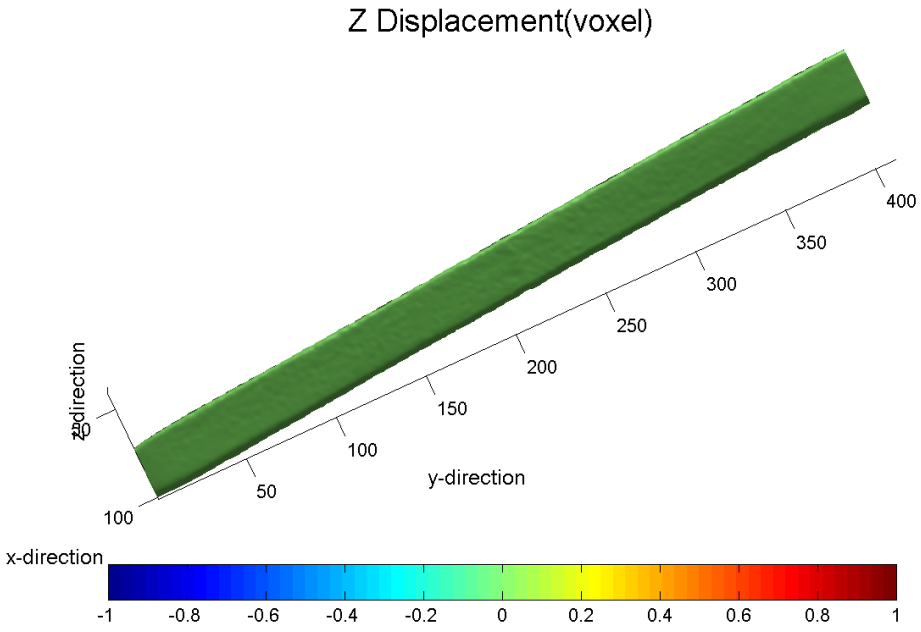


Figure 116 Z Displacement of Pseudo Zero Calibration

4.7.8.2 Air base Scan Calibration:

The Air-base Configuration sample has the Height 22 voxels, width 129 Voxels, and the length is 1992 voxels, which is approximate 0.81mm, 4.81mm, and 74.51mm (i.e. H, W, and L), but we only take the middle portion of the computation, which has 450-150-150 (L-W-H). The scan dataset is 16-bit single type. The dataset has total 10125000 voxels.

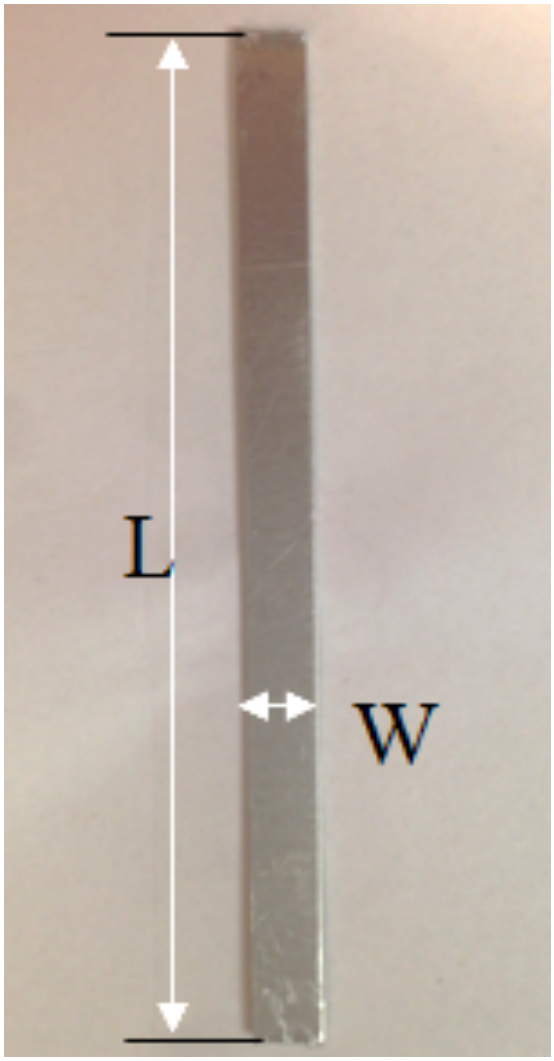


Figure 117 The Length and Width of the Sample

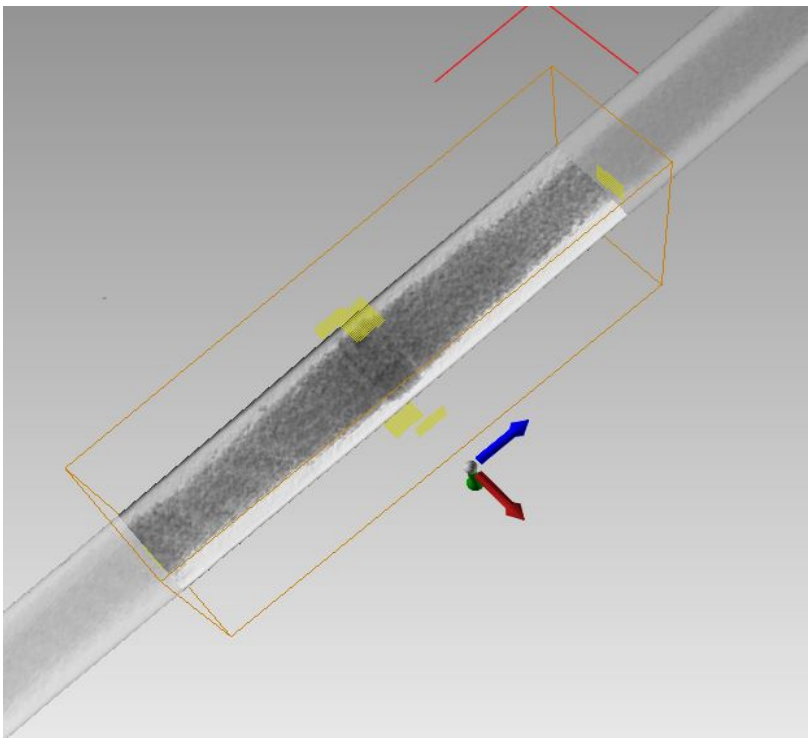
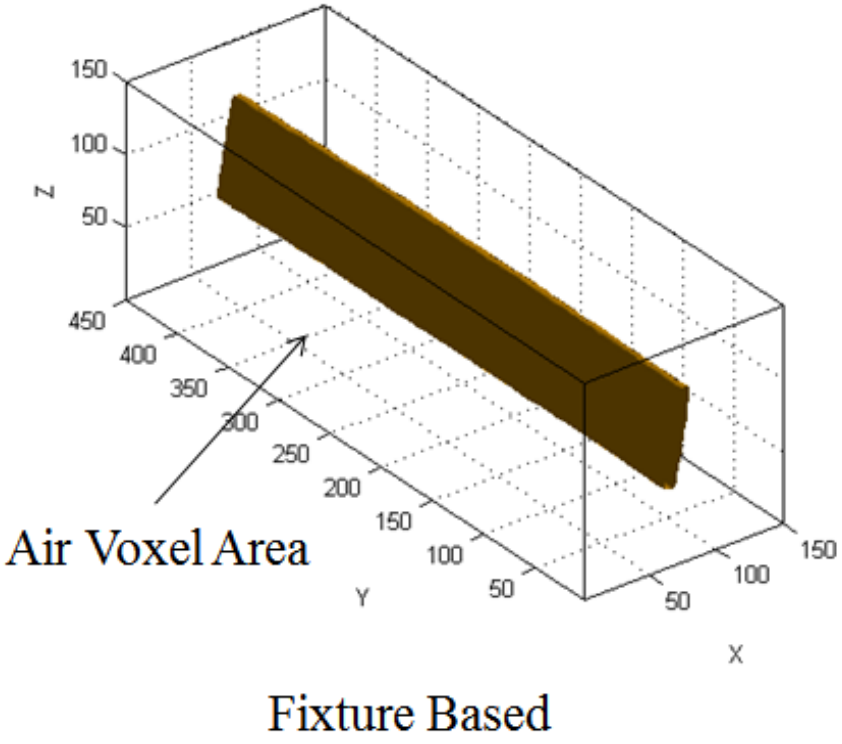


Figure 118 The Analysis Region 450-150-150 of the 3D Reconstruction Sample



Fixture Based

Figure 119 Fixture Based Calibration Configurations

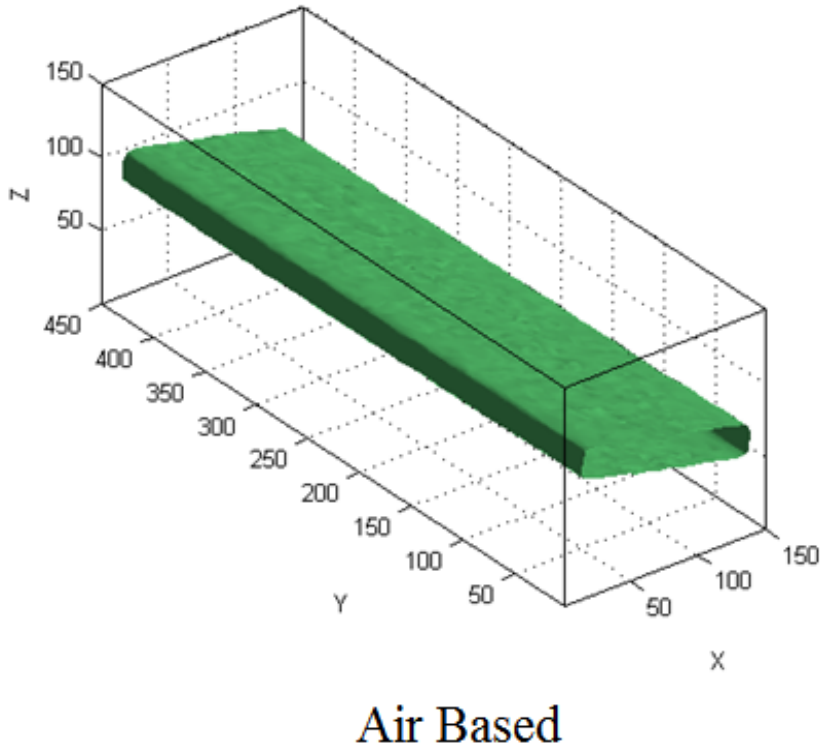


Figure 120 Air Based Calibration Configurations

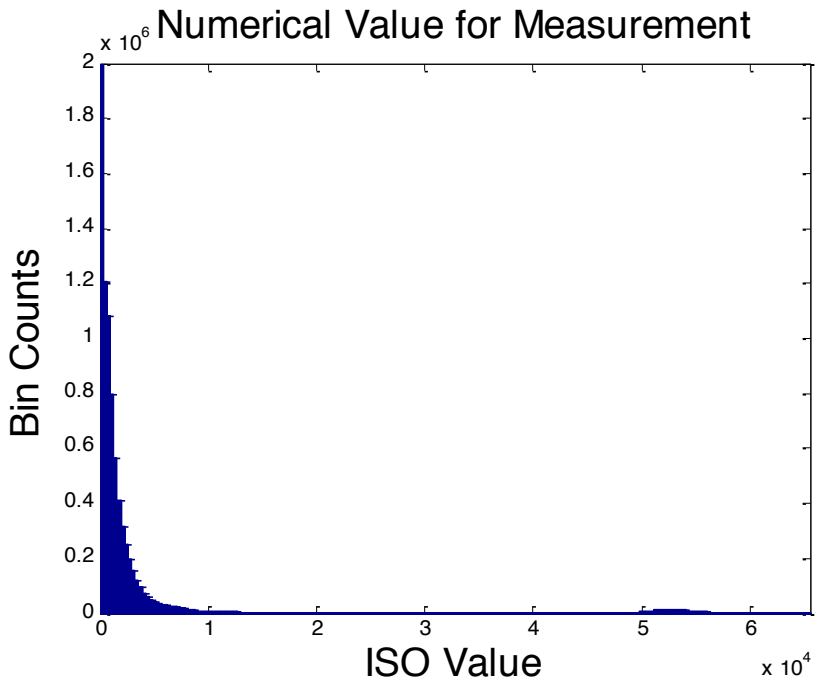


Figure 121 Fixture Based Scan Histogram

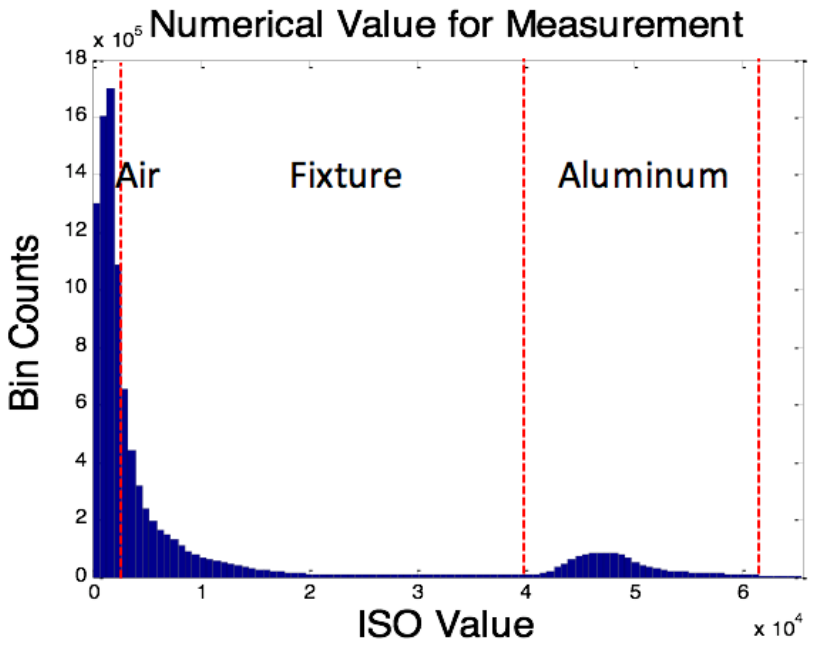


Figure 122 Air Based Scan Histogram

The Air Base Scan has been focused on the middle region of aluminum bar, which has less air voxel units than the fixture based scan. The scanning accuracy is shown in the following figure. We take the two scans under the same condition, which are consecutive two scans. Then we can compare those two scans and find the relative accuracy. The relative accuracy of voxel formula has been given by the following:

$$Relative\ Accuracy = 1 - \frac{|V1 - V2|}{\max(|V1 - V2|)} \qquad (136)$$

V1: voxel in the reference image

V2: same position in the deformed image

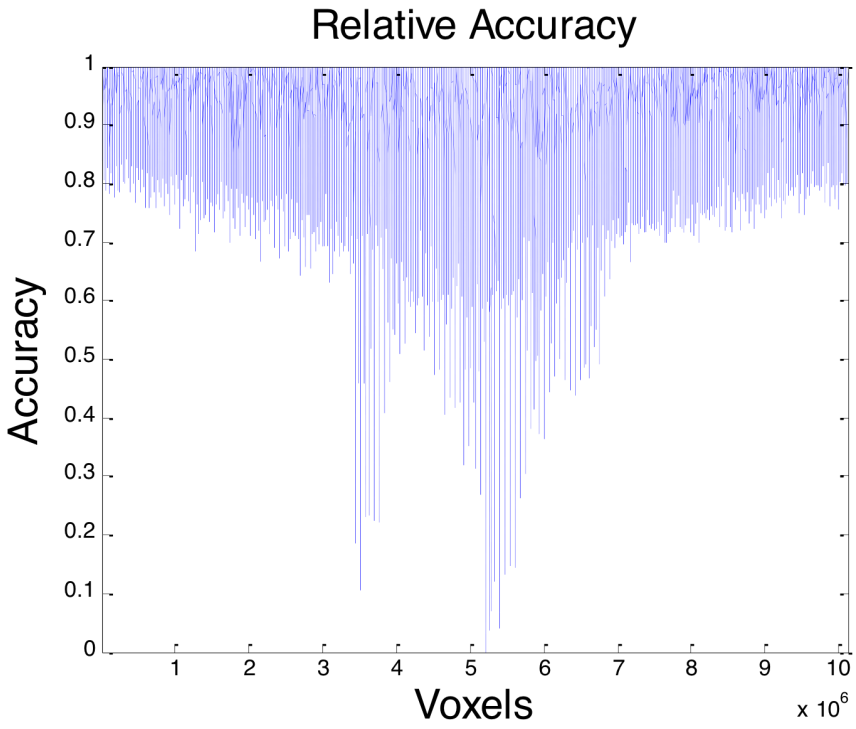


Figure 123 Accuracy of the Identical Scans

The voxel accuracy distribution has been shown as the following figure:

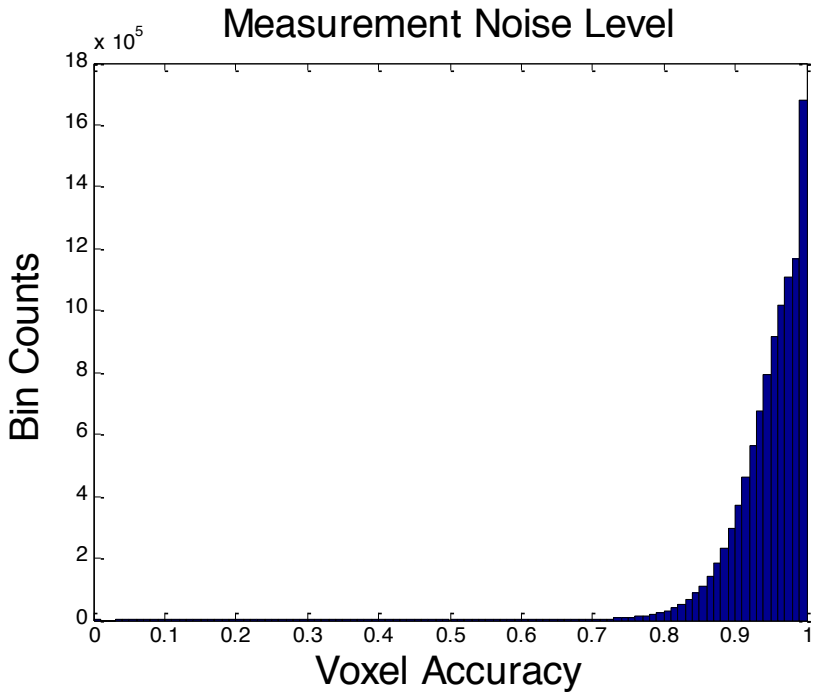


Figure 124 The histogram of accuracy for all Voxels

The scan precision can also be represented by the noise between the two scans, which is simple subtraction according to each voxel. The following Figure 125 shows the noise level, and it is normally distributed, the mean is close to zero (-46.7), and variance is 1141.7, and the accuracy is $8.97e-4$.

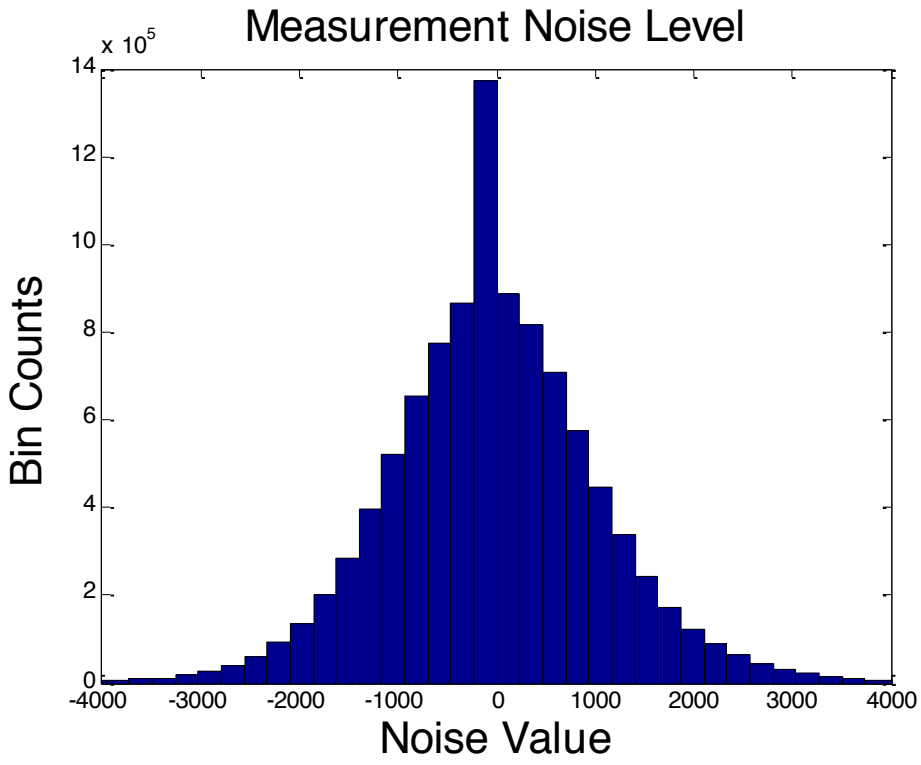


Figure 125 The noise histogram distribution

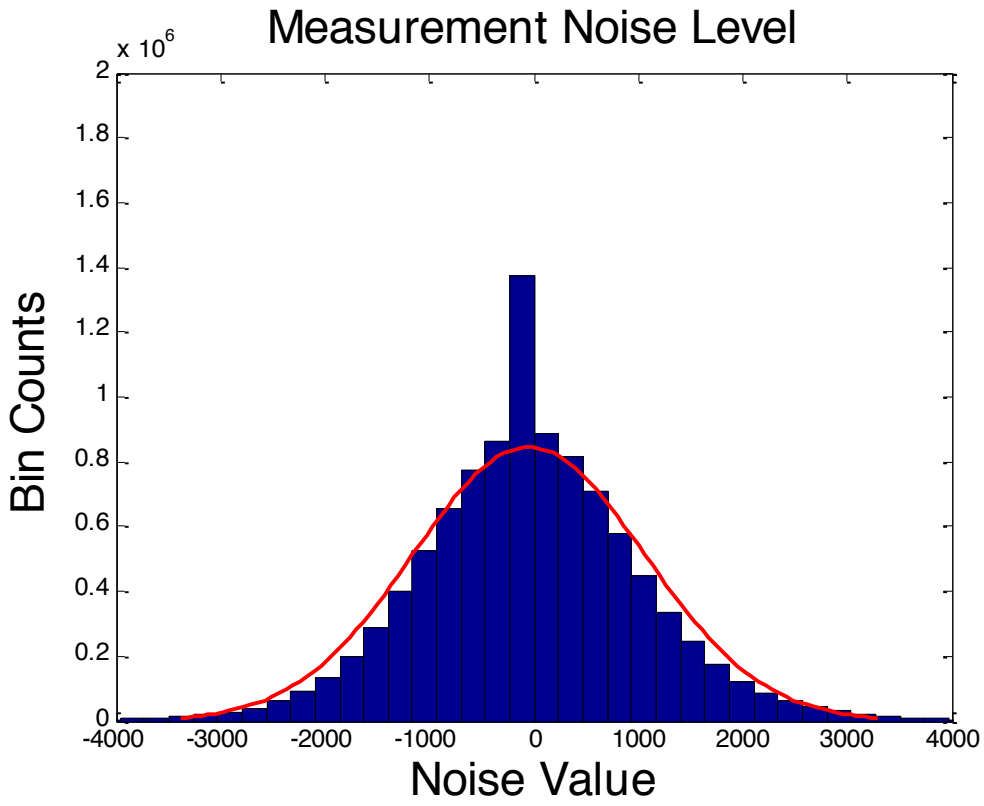


Figure 126 The Noise Level Fitted by the Normal Distribution

$$u = -46.7 \qquad (137)$$
$$\sigma = 1114.7$$
$$Scan(P) = \frac{1}{\sigma} = 8.97e - 4$$

The air based scan calibration contours have been shown as following. There is little variance on the edge of the aluminum bar, which is less accurate than the middle part.

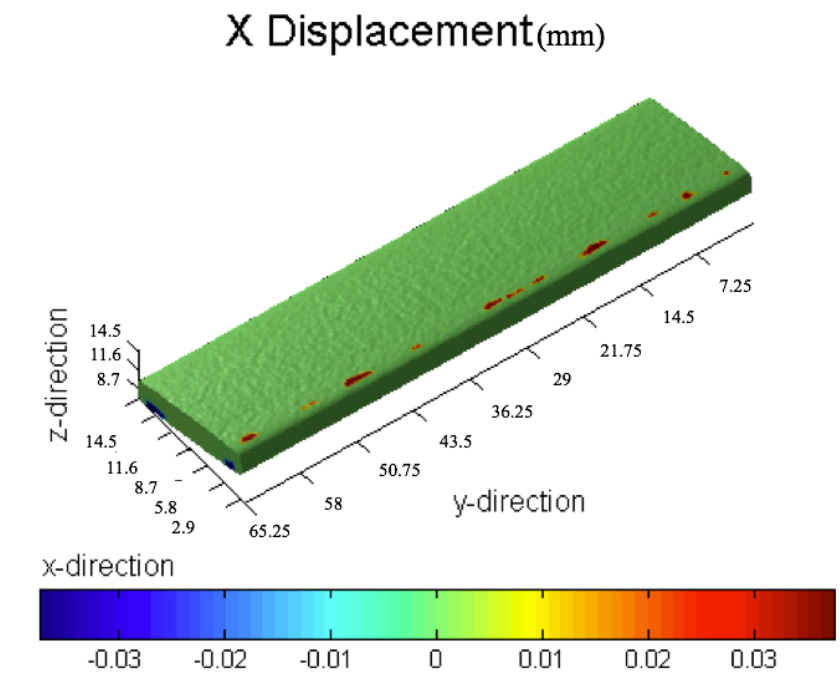


Figure 127 Air Base Calibration X Displacement

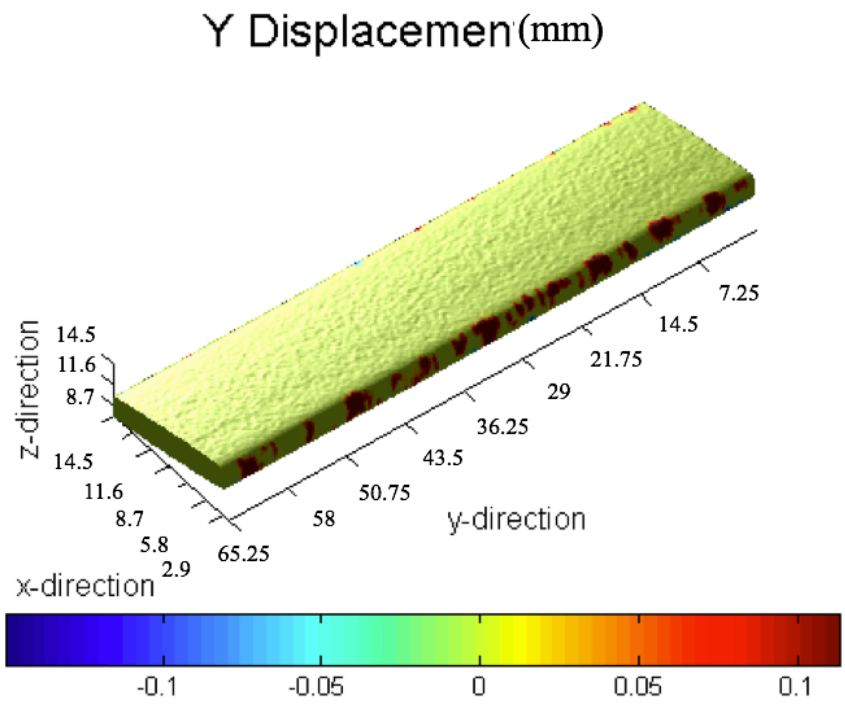


Figure 128 Air Base Calibration Y Displacement

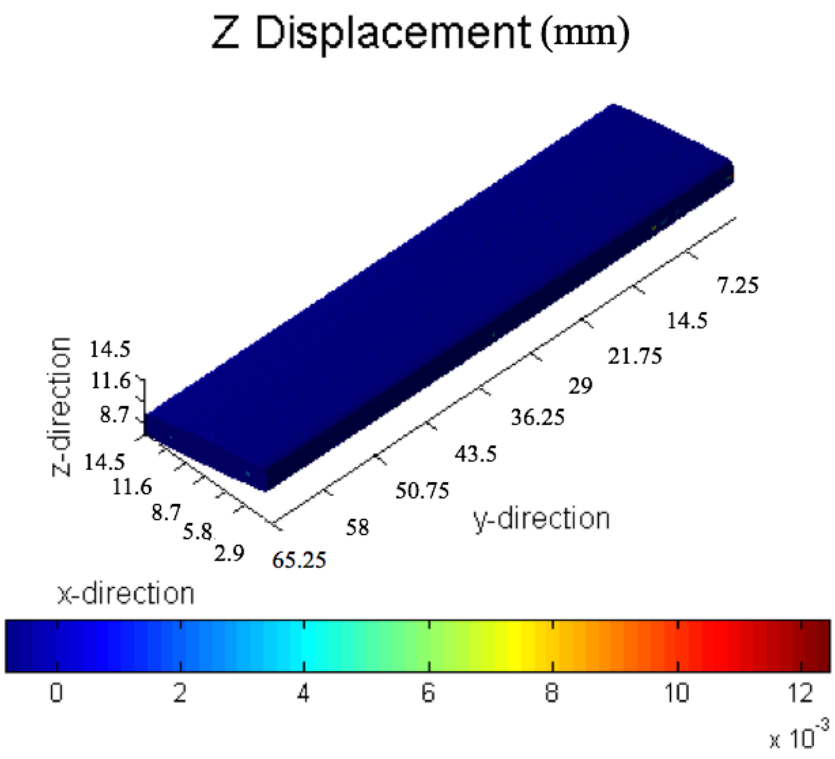


Figure 129 Air Base Calibration Z Displacement

We can also find the histogram of the DVC computed displacements, and then the accuracy of computation at each direction can be known from the results.

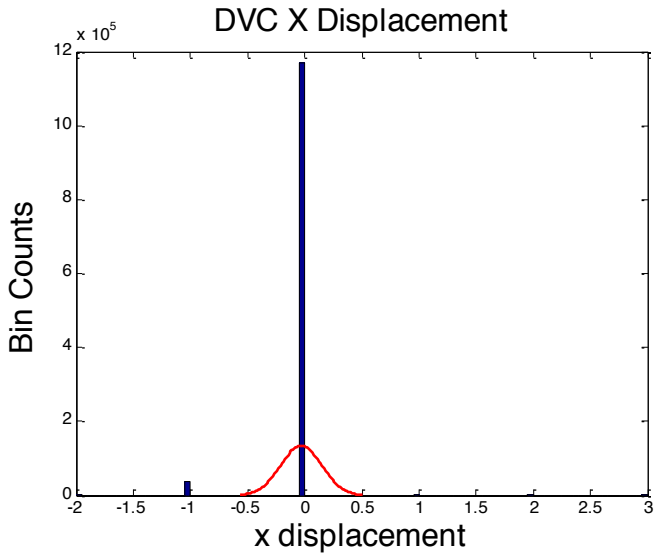


Figure 130X direction Displacement Variation

$$u = -0.0304$$
$$\sigma = 0.1792 \tag{138}$$
$$\mathrm{DVC(P)} = \frac{1}{\sigma} = 5.5803$$

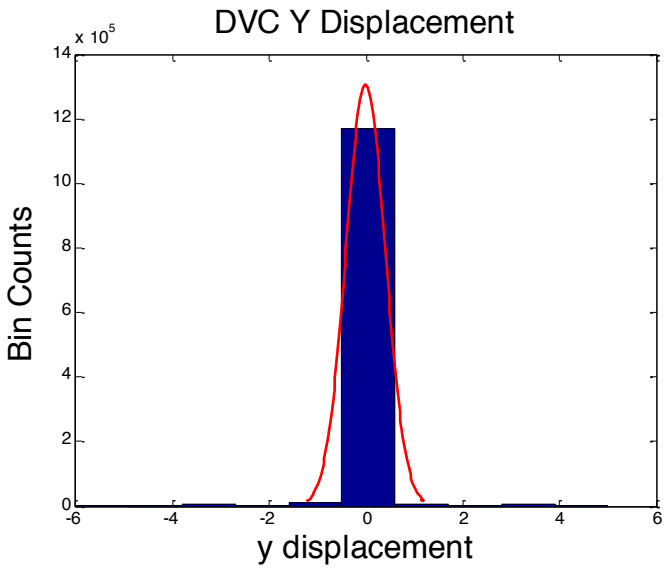


Figure 131 Y direction Displacement Variation

$$u = -0.0093$$
$$\sigma = 0.4058 \tag{139}$$
$$\mathrm{DVC(P)} = \frac{1}{\sigma} = 2.4642$$

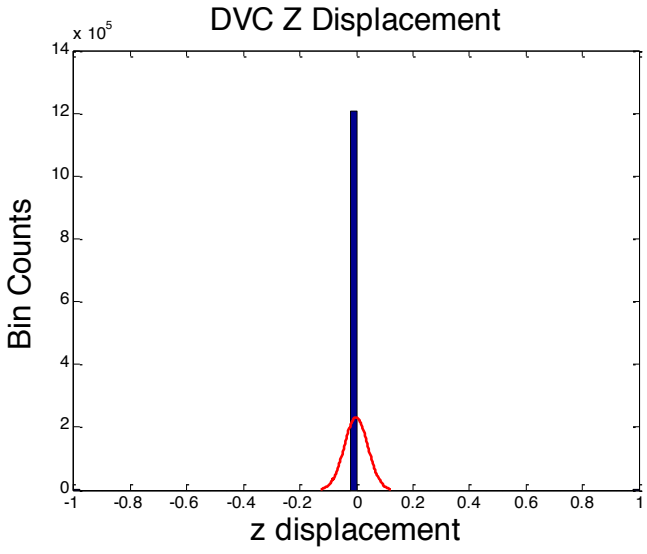


Figure 132Z direction Displacement Variation

$$u = -0.0011 \quad (140)$$
$$\sigma = 0.0417$$
$$DVC(P) = \frac{1}{\sigma} = 23.9808$$

4.7.8.3 DVC Experimental Validation with Single Strain Gauge:

We have performed several stages of validation to test the technique. The following block shows the flowchart of validations we have developed. There are two milestones for the DVC development and validation, and the first one is the algorithm validation, which contains the stages of algorithm development and validation. The second one is in-situ experimental validation. In the algorithm validation part, we first acquire a reference image from the CT scan, and we deform the image with known displacements field to create a deformed image. Through the volume correlation, we want to recover the displacement we applied initially to check the residual and accuracy. In the meantime, we add some artificial white noises to test the robust of the algorithm. Once the algorithm is verified, we further move to the experiment validation which brings more interest and real application of this technique. In the second stage, we have done in-situ experiment test, which is a three-point bend test of the aluminum beam. During the test, we have calculated the deflection of the beam analytically using the Beam Equations, and the theoretical deflection is compared with the DVC calculations. Meanwhile, we have attached three strain gages to retrieve the strain field to verify the correctness of beam equation.

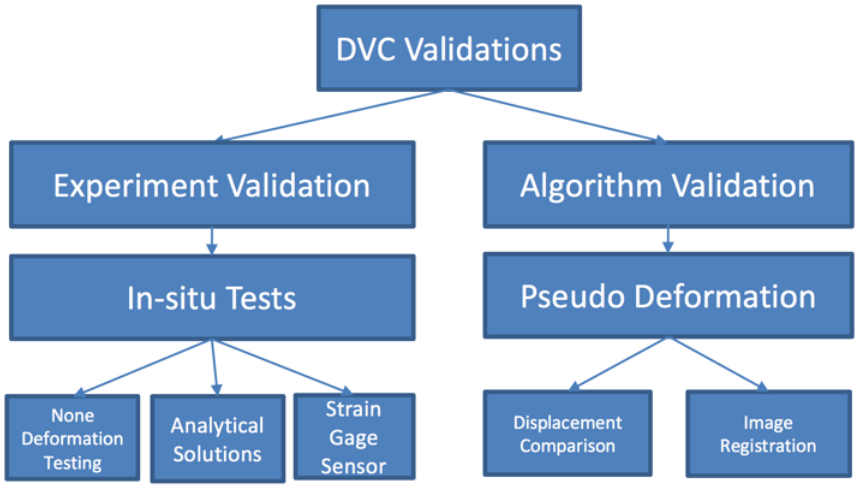


Figure 133 DVC Experiment Validation Flowchart

It is necessary to use the physical measurement from the strain sensor to validate the calculation from the Digital Volume Correlation method. In the validation stage, we applied single strain bridge pasted on the surface of aluminum bar to measure the surface strain. The applied strain specification listed in the following Table 6:

Specification Name	**Value**
Grid Resistance in Ohms	120.0±0.3%
TC OF Gage Factor %100ºC	(±1.3±0.2)
Gage Factor @ 24°C	2.055±0.5%
Transverse Sensitivity	(±1.1±0.2)%
Material	2024-T3 Aluminum
Foil Lot Number	A62AF814
Manufacturer	VISHAY-6828

Table 6 Strain Gage Specification

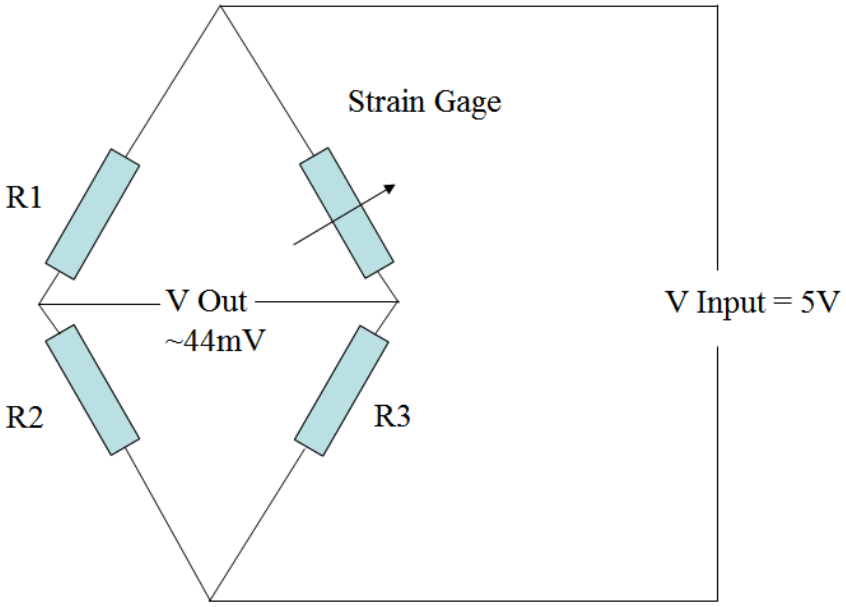
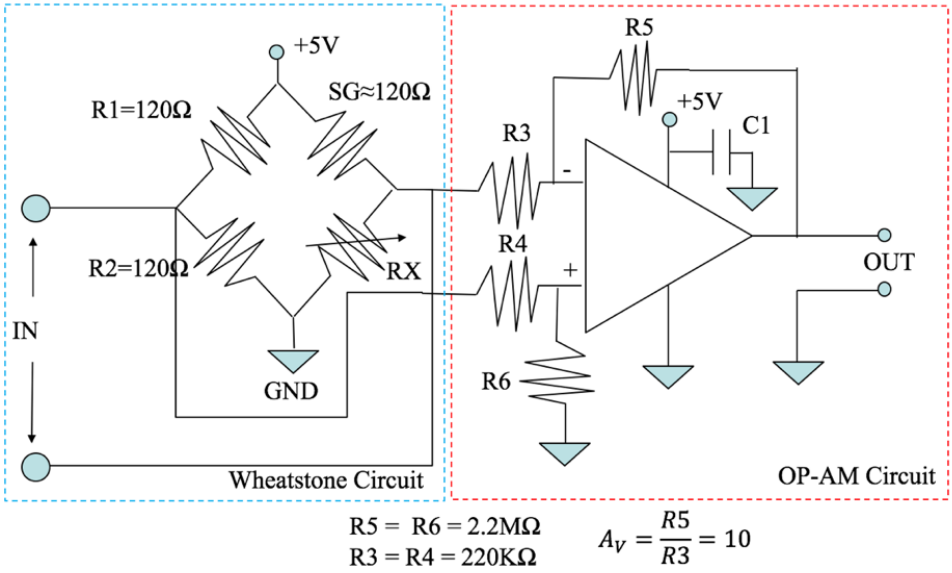


Figure 134 Quarter Bridge Circuit Diagram



R5 = R6 = 2.2MΩ
R3 = R4 = 220KΩ

$A_V = \frac{R5}{R3} = 10$

Figure 135 Strain Gauge Experiment Signal Amplification Circuit Diagram

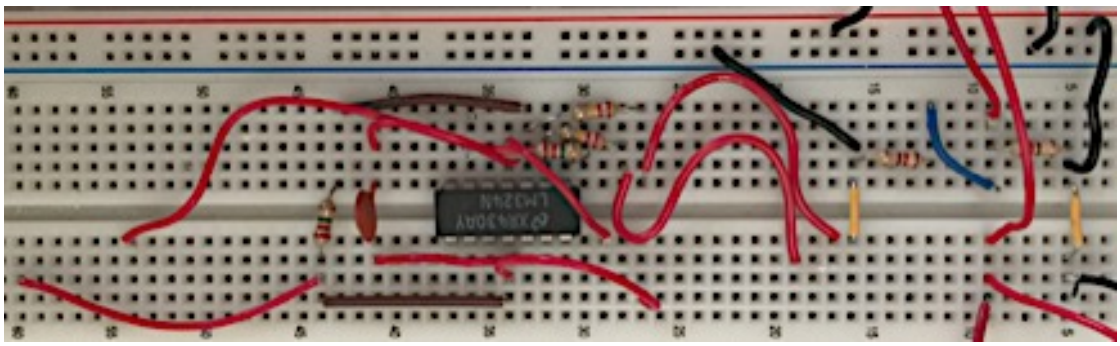


Figure 136 Experiment Physical Circuit

To build one gage system and simplify configuration inside of micro-CT, we applied the quarter bridge measurement system using three the identity 120 Ω resistors to balance the bridge. Since there is little variation in the bridge, we adopt the unbalanced bridge system measurement calculations. The equation for the relation between the bridge system input and output can be written as:

$$\frac{V_{out}}{V_{in}} = \left[\frac{R_3}{R_3 + R_g} - \frac{R_2}{R_1 + R_2} \right] \tag{141}$$

R_g is the Resistance of Strain Gage

The equation holds for both the strained and unstrained conditions. If we consider that in the strained system, the resistance of strain gages changes to $(R_g + \Delta R_g)$. Also, the output voltage change percentage can be expressed as in the following form:

$$V_r = \left[(\frac{V_{out}}{V_{in}})_{Strained} - (\frac{V_{out}}{V_{in}})_{Unstrained} \right] \tag{142}$$

Using the previous equation and the equation above, we can solve for the $(\Delta R_g / R_g)$ regarding V_r , and we use the Gage Factor formula, we can find the strain formula regarding voltage change percentage:

$$GF = \frac{\Delta R_g / R_g}{\varepsilon} \tag{143}$$

$$\varepsilon = \frac{-4 \cdot V_r}{GF \cdot (1 + 2 \cdot V_r)} \tag{144}$$

In the validation, we put the pasted strain gage aluminum bar in into the micro-CT scan. We have introduced two-stage strains, which are corresponding to the two reconstruction deformed volumes. The second stage strain has been used to validate the output from the strain gage sensor and DVC measurement. The threaded screw for introducing the displacement is about 0.7 mm/pitch.

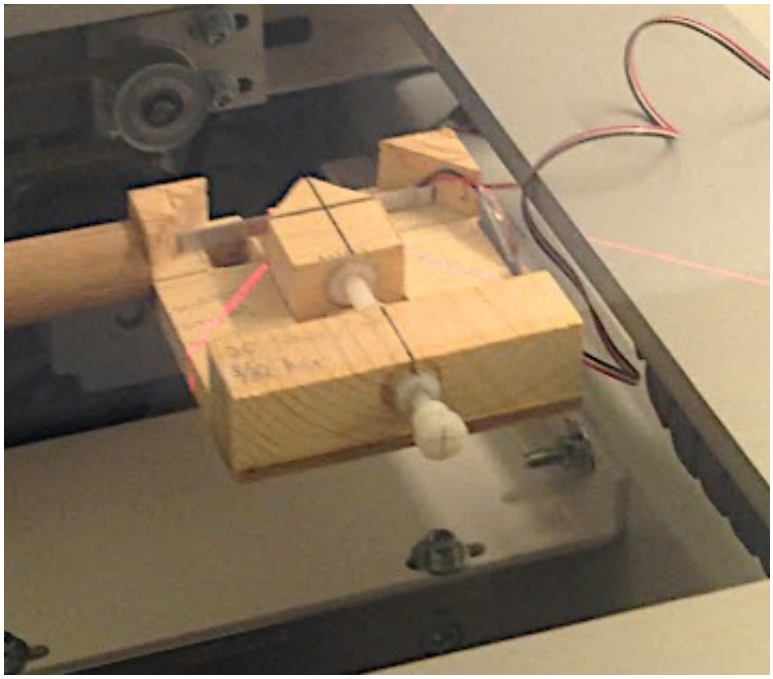


Figure 137 The Aluminum Bar with Fixture

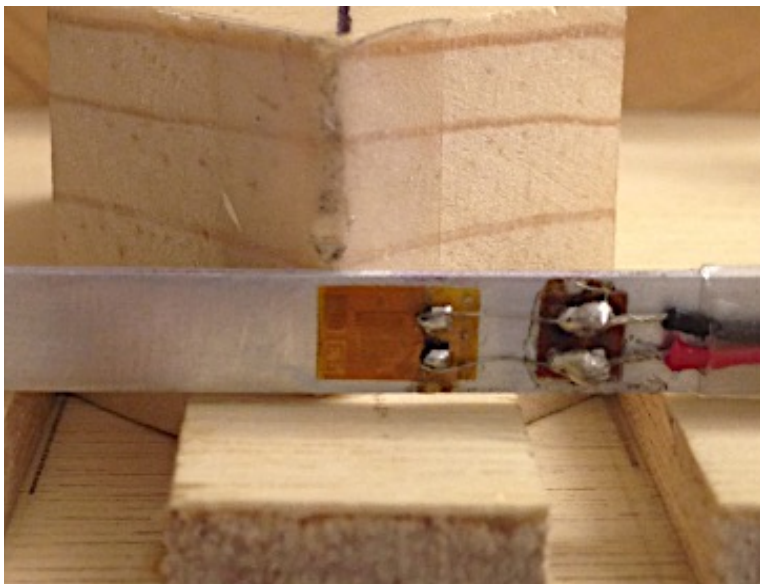


Figure 138 The Aluminum Bar with Strain Gage Measured In-Situ

The validation experiment table has been shown in the following table.

Voltage Input	Voltage Output	Loading Screw Turns	Reconstruction Volume
5V	54.09 mV	0	Reference
5V	55.62 mV	¼	1
5V	56.37 mV	½	2

Table 7 The validation experiment table

Used the unbalance bridge calculation formula, we can compute the strain at each step. Since we will use the second step for the validation, the output at the second step is 450 μ strains, and we can see that the strain introduced by each step is linearly related.

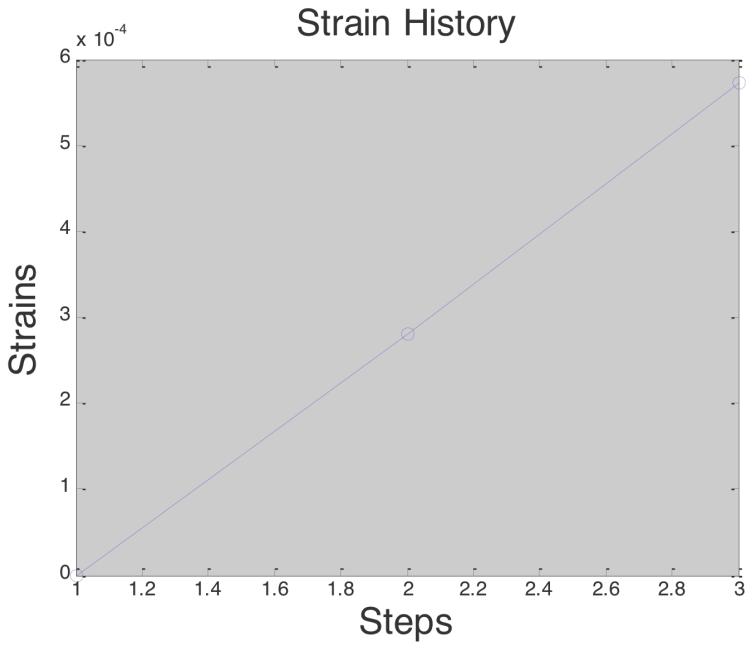


Figure 139 Experiment Loads History

To calculate the strain along the bar length direction, we will use the Euler-Bernoulli beam equation for strain E11. The statement of the equation is that if we know the deflection of the beam, we can calculate the strains of the beam. The strain defined along the beam length direction defined in the following:

$$\varepsilon_x = \frac{dx' - dx}{dx} = -\frac{z}{\rho} \tag{145}$$

$$\frac{1}{\rho} = \frac{d\theta}{dx} = \frac{d^2\omega}{dx^2} \tag{146}$$

$$\varepsilon_x = -z\frac{d^2\omega}{dx^2} \tag{147}$$

In the equation, we have set the parameters: z = -h/2 = - 0.4mm, ω is the z direction displacement, which is the beam deflection. The $\frac{d\omega}{dx}$ is the 1st derivative of the beam deflection, and the $\frac{d^2\omega}{dx^2}$ is the 2nd derivative of the beam deflection.

The Digital Volume Correlation computation parameters have been shown in the following table: We applied the subset window is 19-19-19, and the physical length of each voxel is 0.0584mm, the digital volume correlation rendering ISO gray scale value is 5000, and the total DVC computation window is 990-150-75.

Specification Name	Value
Subset Radius	9
Physical Length Per Voxel	0.0584 mm
Rendering Iso-Value	5000
Computation Window	990-150-75

Table 8 DVC Calculation Parameter

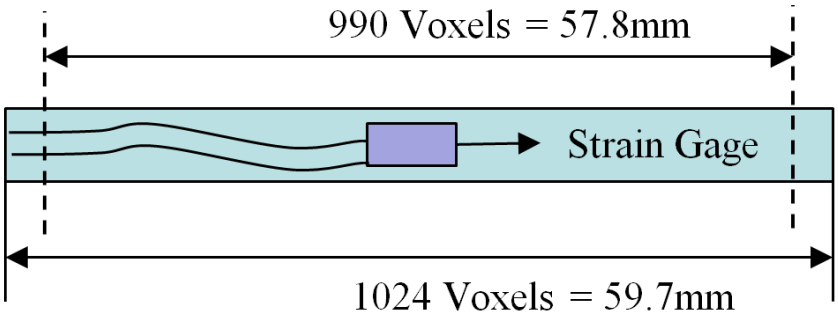


Figure 140 DVC Computational Area

The Beam deflection contour calculated by the DVC has been shown as following:

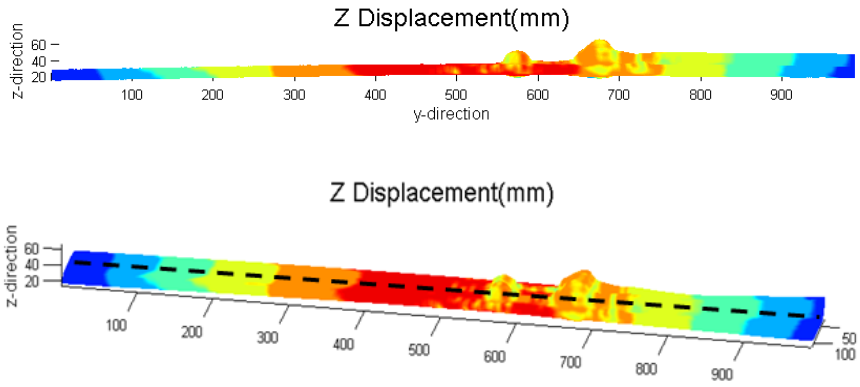


Figure 141 Beam Deflection Contour

The DVC computed displacement along the z direction can be viewed in the following plot.

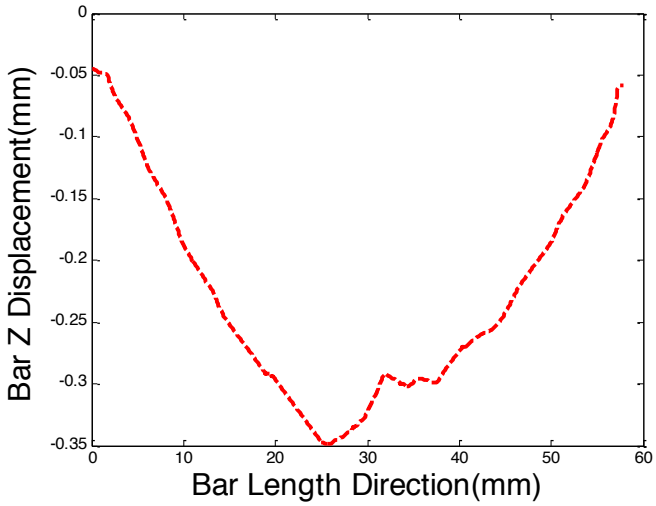


Figure 142 Beam Deflection from the DVC

Measurement Error Comparison: we can compare the difference between the strain gage measurement and DVC computation measurement.

$$ErrorPercentage = \left|\frac{m1 - m2}{m1}\right|\% \tag{148}$$

m1: measurement 1

m2: measurement 2

Measurement Method	Strain Value
Strain Gage	450μ
DVC	440μ
Percentage Error	2.22%

Table 9 Measurement Comparison

4.7.8.4 DVC Experimental Validation with Multiple Strain Gauges:

The following Figure 143 shows the DVC calibration set-up, where the sample, aluminum beam is put on the three-point bending fixture. The strain gages are connected by the electronic wires during the experiment. Then, the strain value for each strain gage is measured by the Wheatstone quarter bridge, and the output signal is amplified by an op-amp circuit shown in the figure.

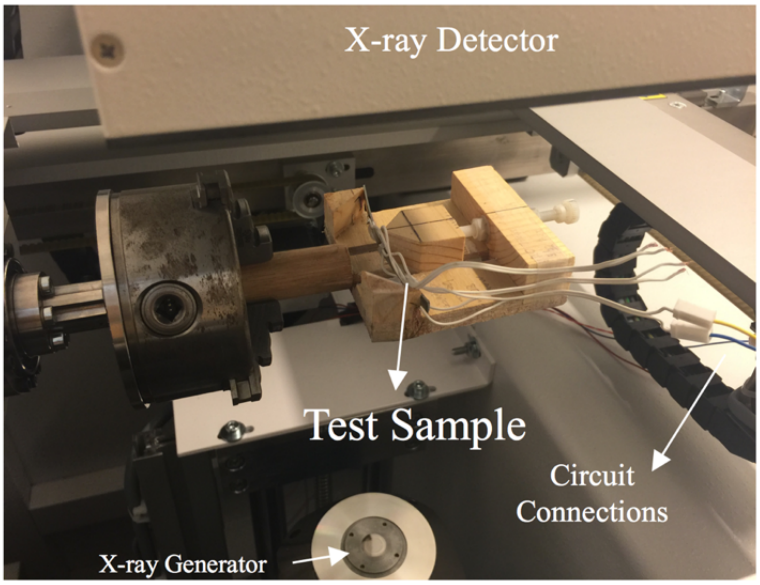


Figure 143 Multi-strain Gauges In-situ Calibration

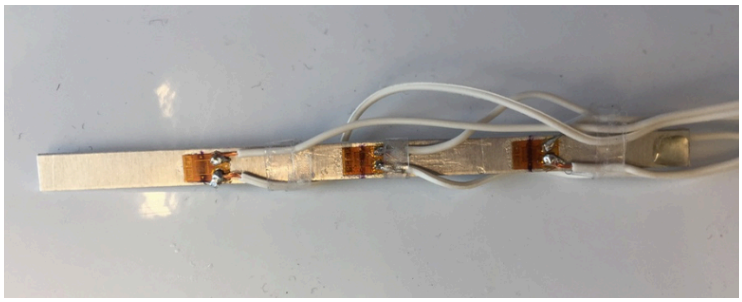


Figure 144 Multi-strain Gauges Sample

The beam deflection has been computed using the Euler-Bernoulli Theory, in which the simple support boundary condition is assumed and used in the governing equation. In the three-point bending test, the maximum beam deflection can be found by the theory. Therefore, the calculated force applied is 4.35N. Thus, the beam deflection can be computed.

The beam model for this test is shown in the following figure, and it also shows the locations where the strain gages are attached. The beam is simply supported by the two ends, and on the middle of the beam, the displacement of 2.33mm is applied. We use the Euler-Bernoulli beam theorem to compute the beam deflection, and since we know the center maximum deflection, therefore, the applied is about 4.35N, and the theoretical deflection equations are calculated separately in the section [0, L/2] and [L/2, L] in the Figure 146.

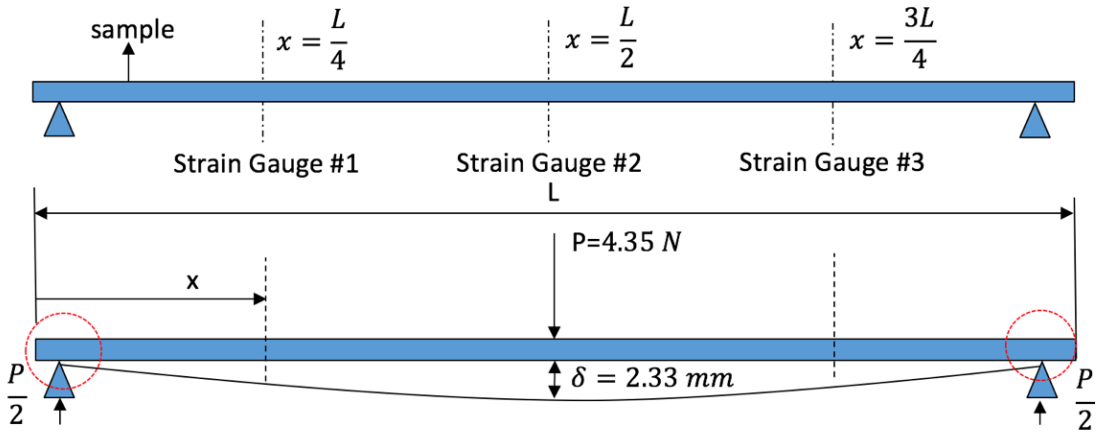


Figure 145 Location of Strain Gauge and Mathematic Beam Model

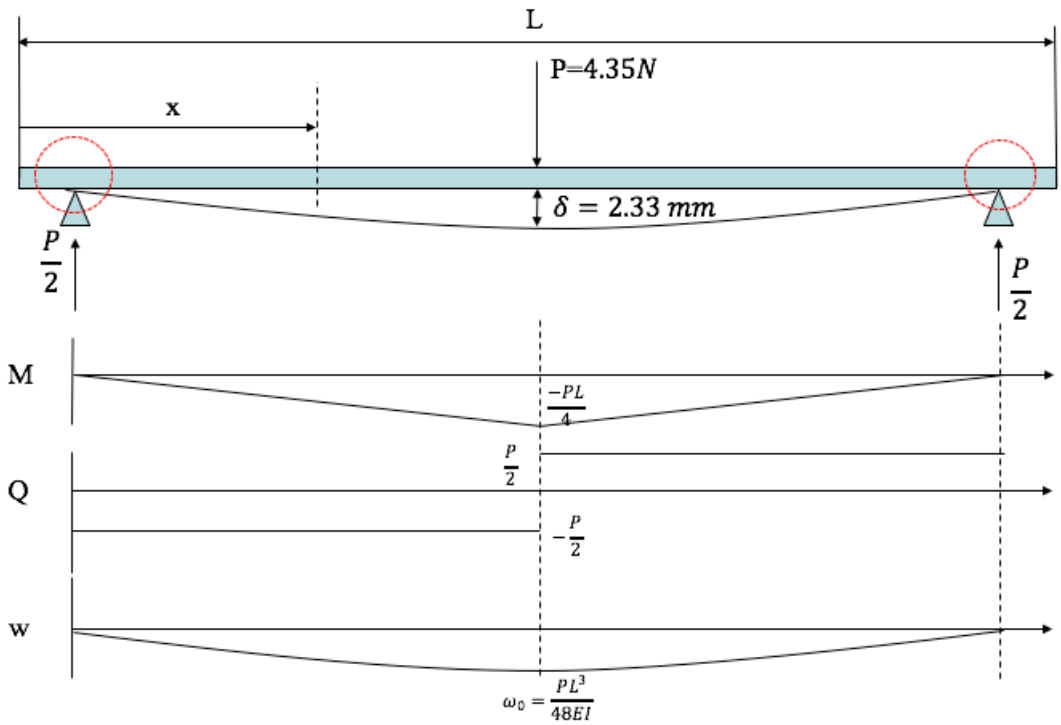


Figure 146 Beam Model Theoretical Bending and Deflection Plot

The bending moment is given by:

$$M = \frac{-Px}{2};\ x \in [0,\ \frac{L}{2}] \tag{149}$$

It follows the Governing Equation:

$$EI\frac{d^2y}{dx^2} = \frac{-Px}{2} \tag{150}$$

We can solve the PDE above using appropriate boundary conditions:

$$EI\frac{dy}{dx} = -\frac{Px^2}{4} + C_1;\ @\ x = \frac{L}{2},\ \frac{dy}{dx} = 0: C_1 = \frac{PL^2}{16} \tag{151}$$

$$EIy = \frac{-Px^3}{12} + \frac{PL^2x}{16} + C_2;\ @\ x = 0,\ y = 0: C_2 = 0 \tag{152}$$

We can find the deflection equation for the left half beam [0, L/2] is the following:

$$y = \frac{Px}{48EI}(3L^2 - 4x^2) \tag{153}$$

Using the similar procedure for the right half beam, the bending moment is given by:

$$M = \frac{P}{2}(x - L);\ x \in [\frac{L}{2}, L] \tag{154}$$

It follows the Governing Equation:

$$EI\frac{d^2y}{dx^2} = \frac{P}{2}(x - L) \tag{155}$$

We can solve the PDE above using appropriate boundary conditions:

$$EI\frac{d^2y}{dx^2} = \frac{P}{2}(x - L);\ @\ x = \frac{L}{2},\ \frac{dy}{dx} = 0: C_1 = \frac{3PL^2}{16} \tag{156}$$

$$EIy = \frac{Px^3}{12} - \frac{PLx^2}{4} + \frac{3PL^2x}{16} + C_2;\ @x = L,\ y = 0: C_2 = \frac{-PL^3}{48} \tag{157}$$

We can find the deflection equation for the right half beam [L/2, L] is the following:

$$y = \frac{Px^3}{12EI} - \frac{PLx^2}{4EI} + \frac{3PL^2x}{16EI} - \frac{PL^3}{48EI};\ x \in [\frac{L}{2}, L] \tag{158}$$

The following Figure 147 shows the deflection of beam with the solution of PDE.

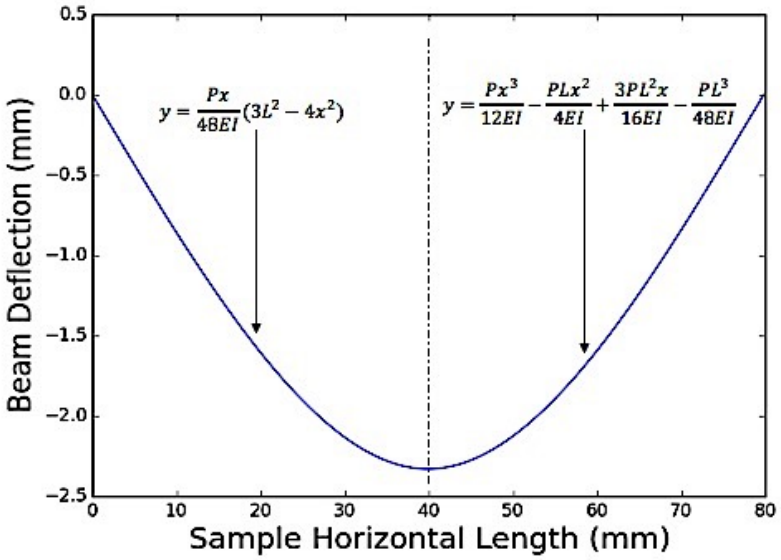


Figure 147 Theoretical Beam Deflection

In the DVC, there are several models of the computation, and the Geometry Contour Model is the surface contour of displacement field. The Computational Mesh Model is the element contour of displacement field, and the Mixed Model shows the discretization within the geometry, in which the aluminum beam is shown in the middle dash line and the strain gage wire is labeled.

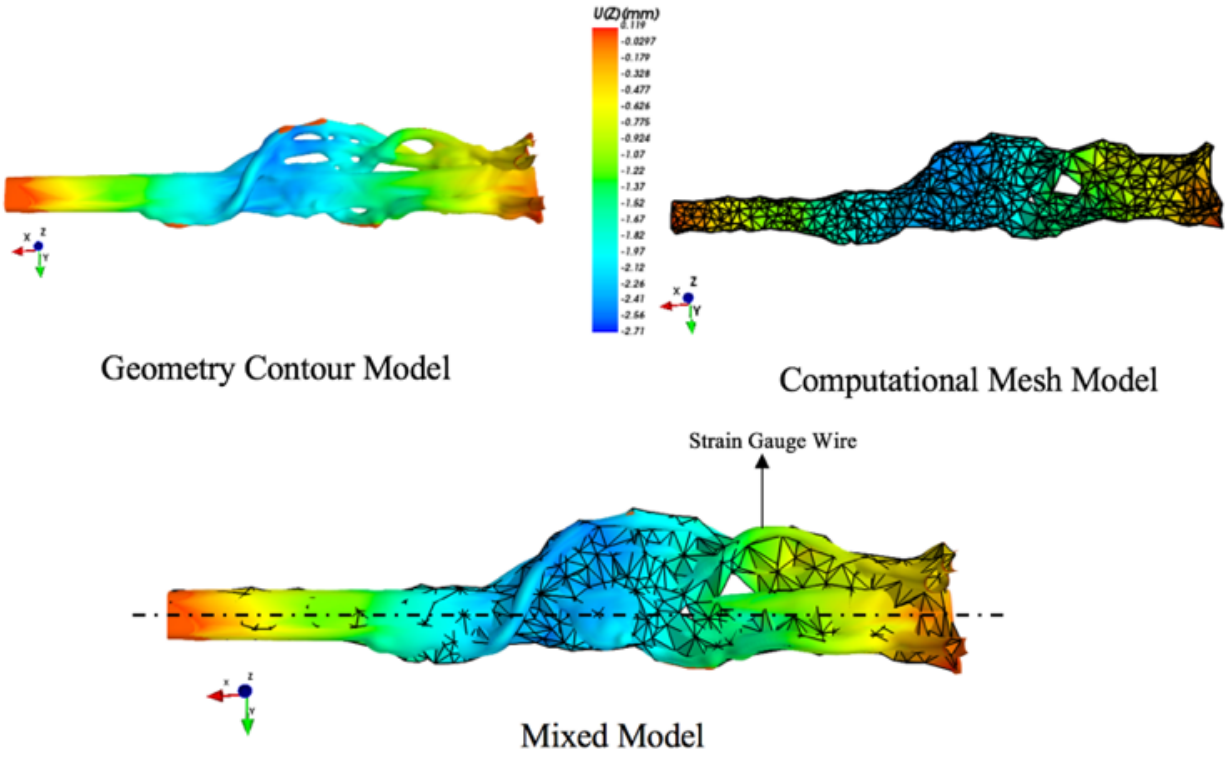


Figure 148 DVC Computation Models

In the plot, we can see that the blue line is the beam theory and the red line is the DVC computations, and the experimental data matches with our theory and assumption model.

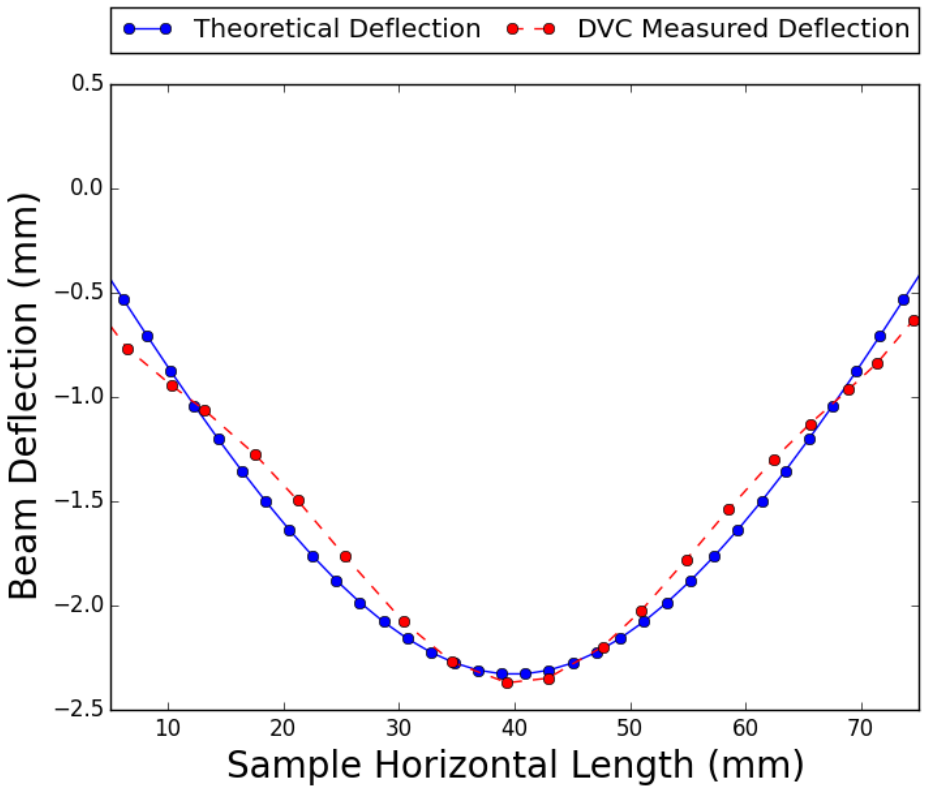


Figure 149 Experiment and Theory Deflection Comparison

From the beam equation, we can find the theoretical strain from the governing equation. The calculation procedure can be expressed as the following:

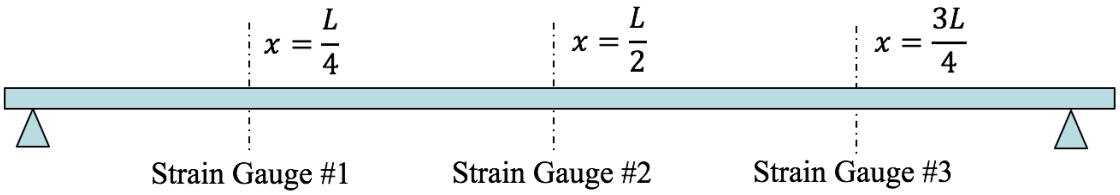


Figure 150 Strain Gauges Location

The strain is calculated as the following second derivative of the deflection:

$$y = \frac{Px}{48EI}(3L^2 - 4x^2);\ \frac{d^2y}{dx^2} = -\frac{Px}{2EI};\ x \in [0,\ \frac{L}{2}] \tag{159}$$

$$y = \frac{Px^3}{12EI} - \frac{PLx^2}{4EI} + \frac{3PL^2x}{16EI} - \frac{PL^3}{48EI}; \frac{d^2y}{dx^2} = \frac{P(x-L)}{2EI}; x \in [\frac{L}{2}, L] \quad (160)$$

The strain can be calculated as the following:

$$\varepsilon_x = -z\frac{d^2y}{dx^2} \quad (161)$$

Therefore, the strain output at each location shows as the following:

$$\varepsilon_x = -z\frac{d^2y}{dx^2}|_{x=\frac{L}{4}} = -z\frac{PL}{8EI} = 0.000877 \quad (162)$$

$$\varepsilon_x = -z\frac{d^2y}{dx^2}|_{x=\frac{L}{2}} = -z\frac{PL}{4EI} = 0.001750 \quad (163)$$

$$\varepsilon_x = -z\frac{d^2y}{dx^2}|_{x=\frac{3L}{4}} = -z\frac{PL}{8EI} = 0.000877 \quad (164)$$

The strain gages raw data from the circuit shows in the following block:

Strain Gage 1		Strain Gage 2		Strain Gage 3	
Unstrained Voltage (mV)	Strained Voltage (mV)	Unstrained Voltage (mV)	Strained Voltage (mV)	Unstrained Voltage (mV)	Strained Voltage (mV)
5.87	7.85	5.21	10.18	2.97	5.37
4.14	6.62	4.68	10.01	2.93	5.37
3.96	6.44	4.63	9.94	2.93	5.21
3.85	6.26	3.83	9.02	3.03	5.4
3.15	5.58	3.78	8.98	2.92	5.15

Table 10 Strain Gauge Voltage Outputs

Strain Gage 1	**Theoretical**	**Relative Errors**	**Strain Gage 2**	**Theoretical Strains**	**Relative Errors**	**Strain Gage 3**	**Theoretical**	**Relative**
ϵ_{xx} Strains	**Strains**		**ϵ_{xx} Strains**			**ϵ_{xx} Strains**	**Strains**	**Errors**
0.000770193	0.000877	0.00010681	0.001930954	0.00175	0.00018095	0.00093341	0.000877	0.00005641
0.000964493	0.000877	8.7493E-05	0.002070525	0.00175	0.00032053	0.000948952	0.000877	0.000071952
0.000964493	0.000877	8.7493E-05	0.002062772	0.00175	0.00031277	0.000886782	0.000877	0.000009782
0.000937296	0.000877	6.0296E-05	0.002016252	0.00175	0.00026625	0.000921754	0.000877	0.000044754
0.000945067	0.000877	0.00068067	0.002020129	0.00175	0.00027013	0.000867353	0.000877	0.000009647
Mean Error Strain		**82.0μ (9%)**	**Mean Error Strain**		**270.1μ (10%)**	**Mean Error Strain**		**38.5μ (4%)**

Table 11 Theoretical Strain and Experimental Strain Comparison

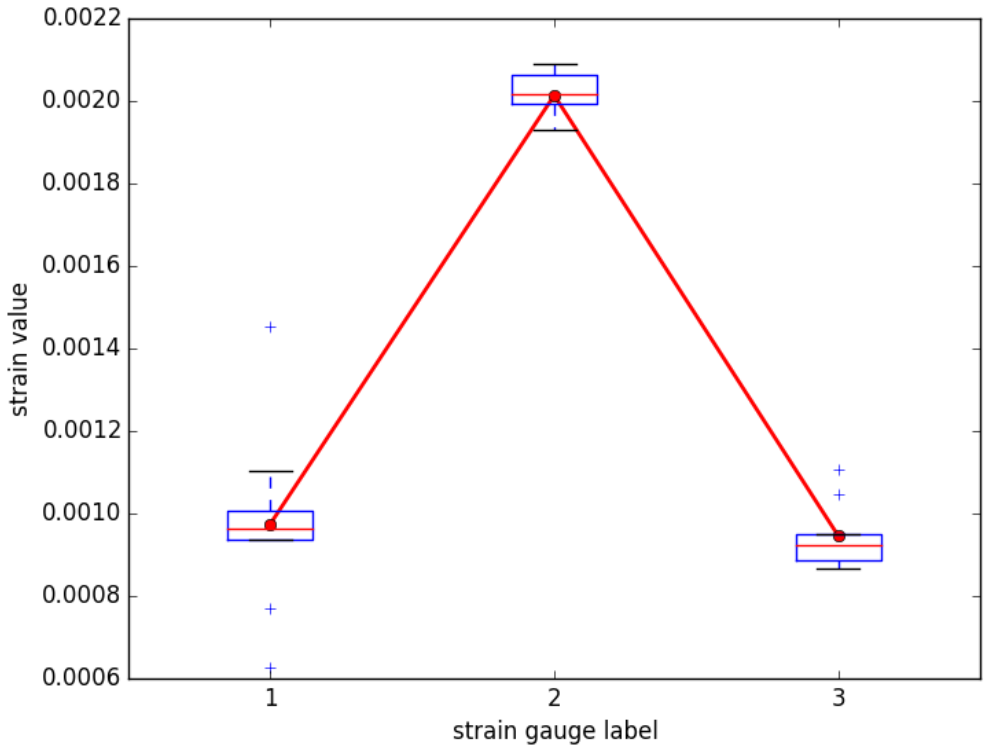
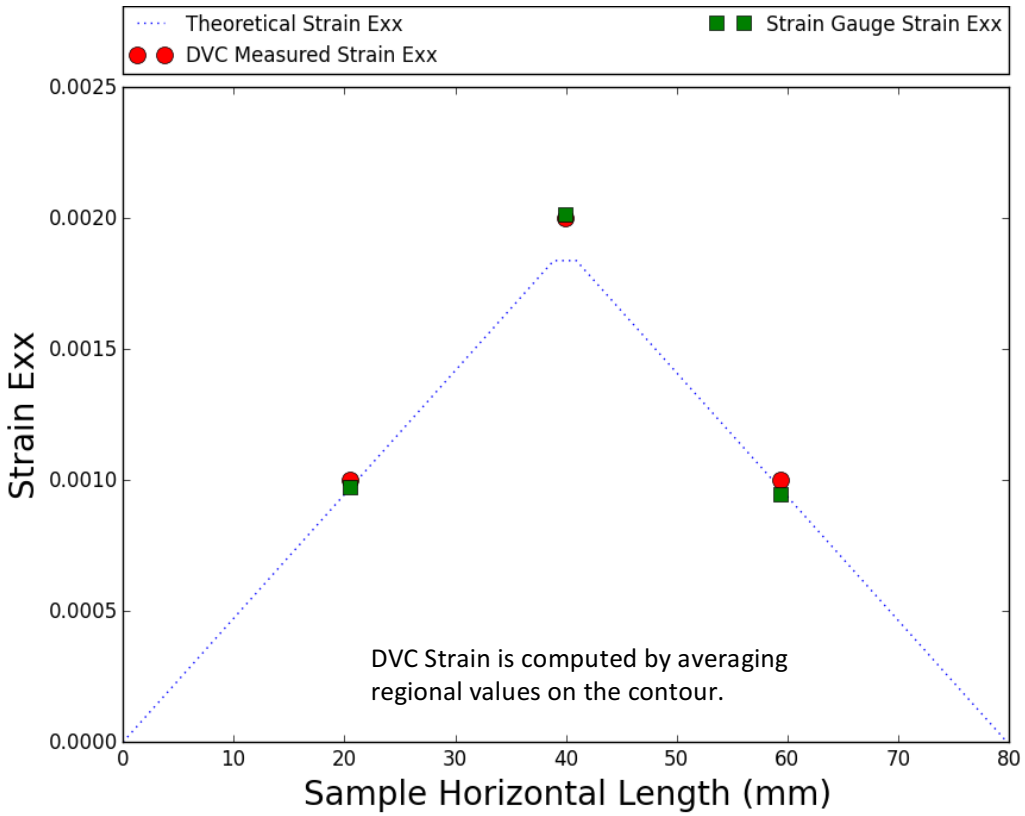


Figure 151 Strain Gauges Output on the Beam

The following figure presents the strain values at specified gage location to demonstrate the comparison between the strain gage measurement, the theoretical calculation, and the DVC measurement. After comparison, we found the average difference between the theory and the experiment is 7.67%.



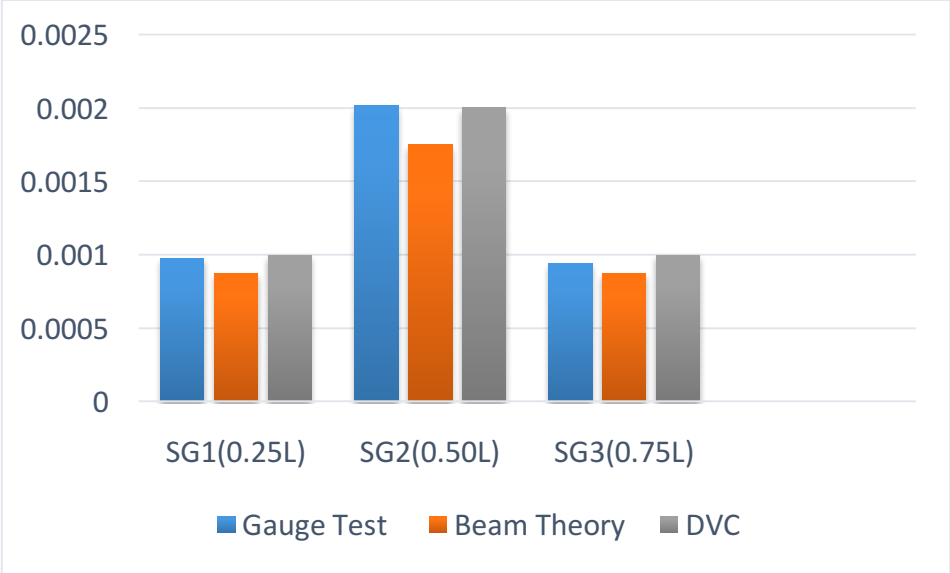


Figure 152 Computed Strain Comparison

The following flowchart shows the procedure of post-DVC validation. In the figure, the source volume is the reference image, and the target volume is the deformed image, and after the

DVC computation, we have the displacement field vectors (U, V, W). Each U, V, W component has the same size of the volume image, and each element shows the displacement in each direction x, y and z respectively. The displacement field can be applied to the reference image to create a similar transform image to get the deformed image. Therefore, we can compare the created transform image with the deformed image, and the closeness of images indicates the correctness of displacement field. The iterations are required if the residual of the image is above the threshold.

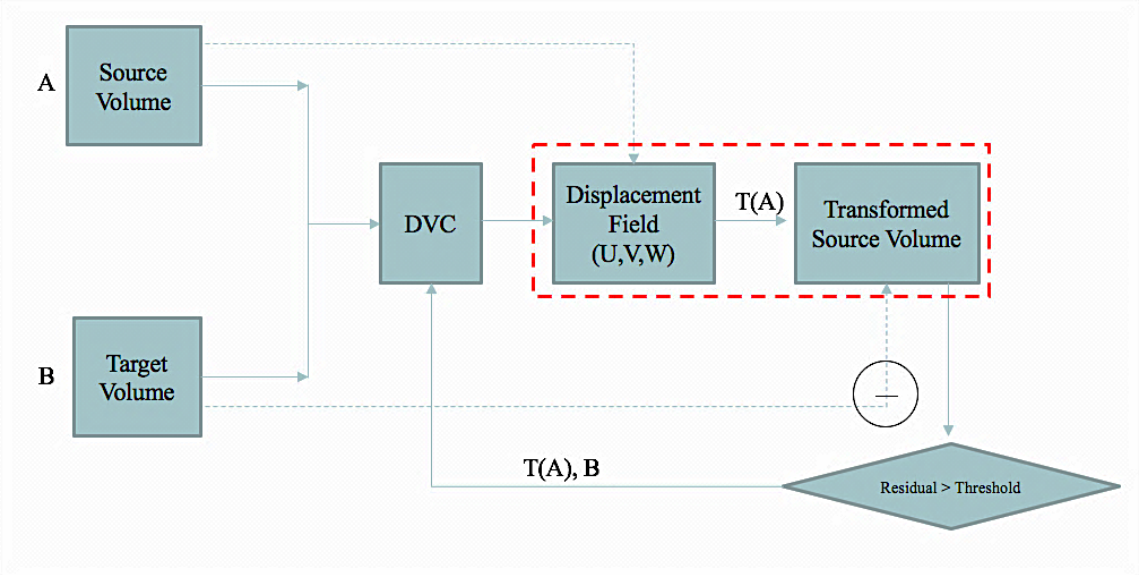


Figure 153 DVC Validation Flowchart

The following Figure 154 shows the comparison of images. The image ‘A’, colored in red, represents the reference image. The image ‘B’, colored in green, represents the deformed image, and the T(B) image, colored in blue, represents the transformed image. In the figure, the first image shows the 3D experiment geometry; the load is applied in the center. In the second image, both the reference and deformed image have been shown the differences in the cross-section view. In

the third image, additional transform image T(B) has been shown, which is matched with the reference image. In the fourth image, the closeness and difference of A and T(B) have been shown.

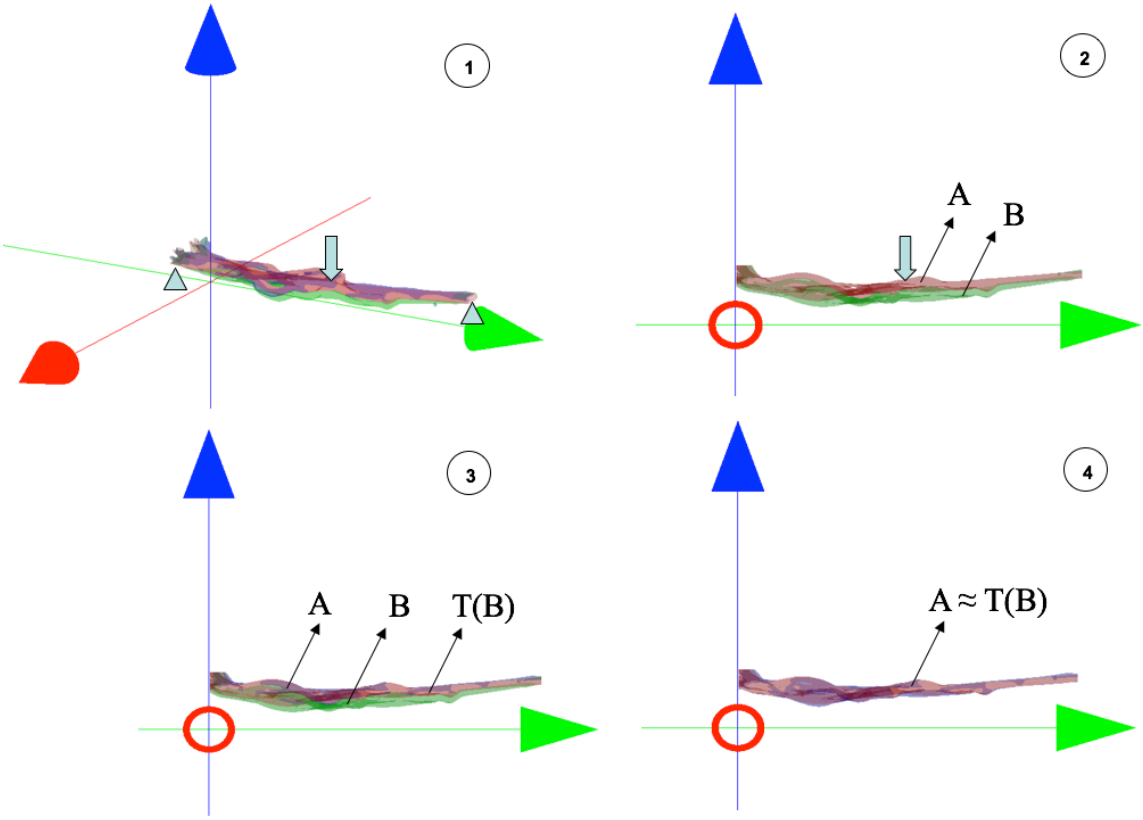


Figure 154 Registration Contour Visualization

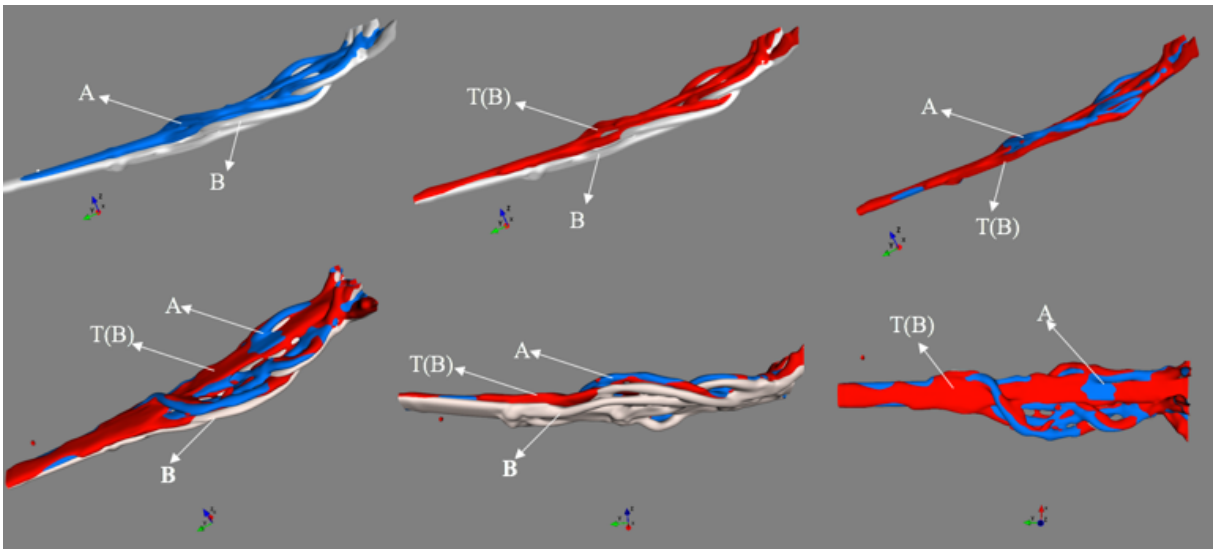


Figure 155 DVC Volumetric Registration Validation

The following Figure 156, the cross-section grayscale image, has been shown. The cross section of reference image A is subtracted by the cross section of transform image T(B). In the residual plot, we can see both the images are aligned well in the cross-section, which indicates the good match and correctness of computation in displacement field.

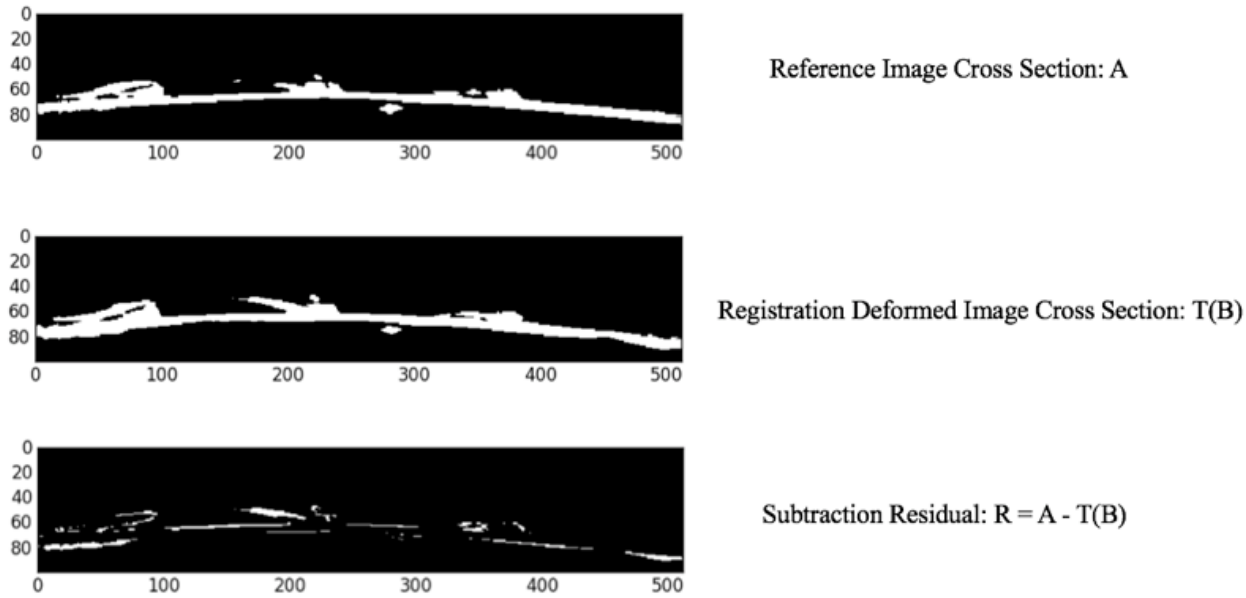


Figure 156 Difference of Images in the Cross-section Grayscale View

The detailed registration algorithm is shown in the following blocks. In the first block, the forward image registration is shown, in which each voxel, given by an integer index, in the reference image is moved by the displacement vectors to the location of empty sample volume. Due to the inhomogeneous displacement vectors, the index in the empty sample volume is not an integer. Therefore, the fraction index is rounded into the nearest integer index, and this would cause some image aliasing. In the second block, we introduce a better method, backward registration method, to perform image registration. In the algorithm, we are going to find back the

"reference" image with the displacement vectors from the deformed image. At each integer index, given the displacement values, the corresponding grayscale value can be retrieved from the deformed volume. An interpolation function is created in the deformed image space, and the queried value at fraction index can be approximated by the interpolation. Therefore, the backward image registration algorithm removes the voxel aliasing and will be better fitted with this problem.

Volumetric Forward Validation:
Input reference volume: A
Initialize a empty VOLUME, size of [x, y, z];
For (i, j, k) in Total Voxels of Volume:
    Find displacement @ A(i, j, k) with (u,v,w)
    Get new coordinator in VOLUME using displacements: i', j', k' = i+u, j+v, k+w
    if (i' < 0) : i' = x + i' ; if (i' > x): i' = i' – x;
    if (j' < 0) : j' = y + j' ; if (j' > y): j' = j' – y;
    if (k' < 0): k' = z + k' ; if (k' > z): k'= k'- z;
    Assign the greyscale value @ location(i', j', k') in VOLUME with greyscale value@ location A(i, j, k)
Return T(A) = VOLUME
Compare T(A) and B

Volumetric Backward Validation:
Input reference volume: B
Initialize a empty VOLUME, size of [x, y, z];
For (i, j, k) in Total Voxels of Volume:
    Find displacement @ VOLUME (i, j, k) with (u,v,w)
    Get coordinator in the B using displacements: i', j', k' = i+u, j+v, k+w
    if (i' < 0) : i' = x + i' ; if (i' > x): i' = i' - x
    if (j' < 0) : j' = y + j' ; if (j' > y): j' = j' - y
    if (k' < 0): k' = z + k' ; if (k' > z): k'= k'- z
    Assign the greyscale value @ location(i, j, k) in VOLUME using the interpolated value over neighbor of B(i' , j', k')
Return T(B) =VOLUME
Compare A and T(B)

In the following Figure 157, it illustrates the algorithm details of forward image moving and resampling algorithm. The template image is the reference image 'A', and using the displacement vectors, for each voxel located at (i, j, k) in the 3D image, the new location index can be found by adding the displacement vector to the old location index as (i+u, j+v, k+w). Nonetheless, there is no guarantee that the new location will be an integer location. Therefore the

fraction index will be round to the nearest integer point, and this would cause the image aliasing for comparison. A better solution is to use the backward image registration technique shown below, instead of using reference as a template, we use the deformed image as a template. In this case, we will try to cover a pseudo reference image T(A), which can be compared with the origin reference image A. For each integer index in the T(A), its greyscale value is calculated by an interpolation from the corresponding nearest points in the deformed image. In this way, the image aliasing is solved, and the comparison between two images is more realistic and intuitive.

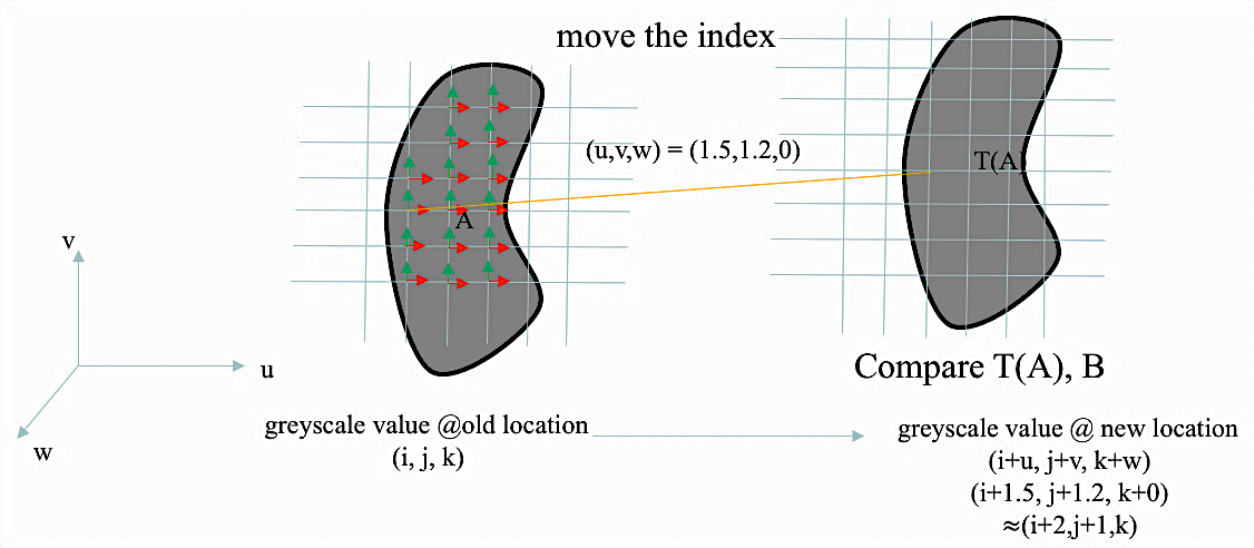


Figure 157 Forward Image Moving and Resampling

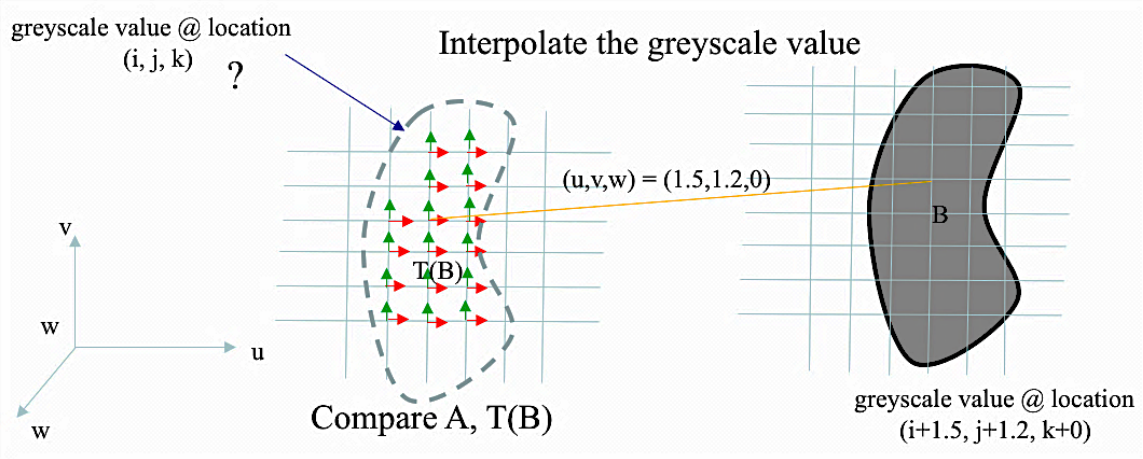


Figure 158 Backward Image Moving and Resampling

The following Figure 159 shows the reference image with deformation vectors, and the reference image A is compared with the image B. It shows the major deformed region is in the center of the object. In the front view of the image, we can see the boundary and shape changes between the reference and deformed image.

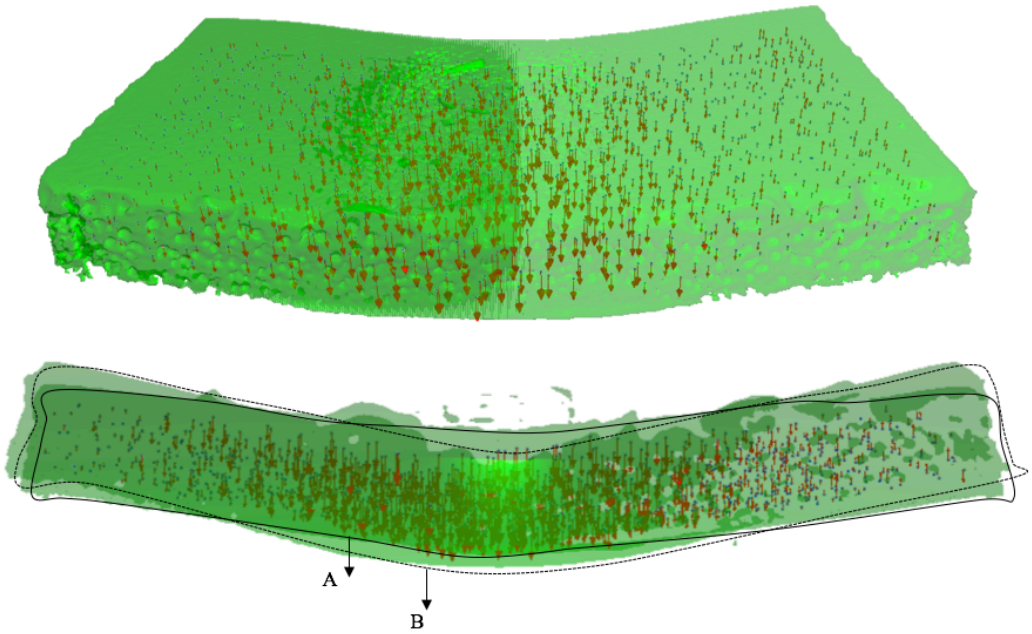


Figure 159 Displacement Vectors in Images

Chapter 5 Digital Volume Correlation Applications and Analysis

5.1 Aluminum Beam Three Point Bending

We also perform displacement loading on an aluminum bar, which provides us how the aluminum bar deforms under the displacement 0.58mm. The bending condition has been shown below figures.

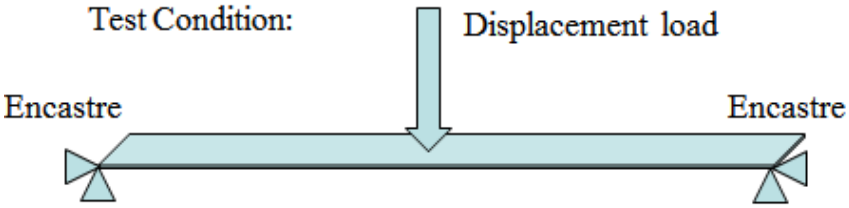


Figure 160 Load Condition

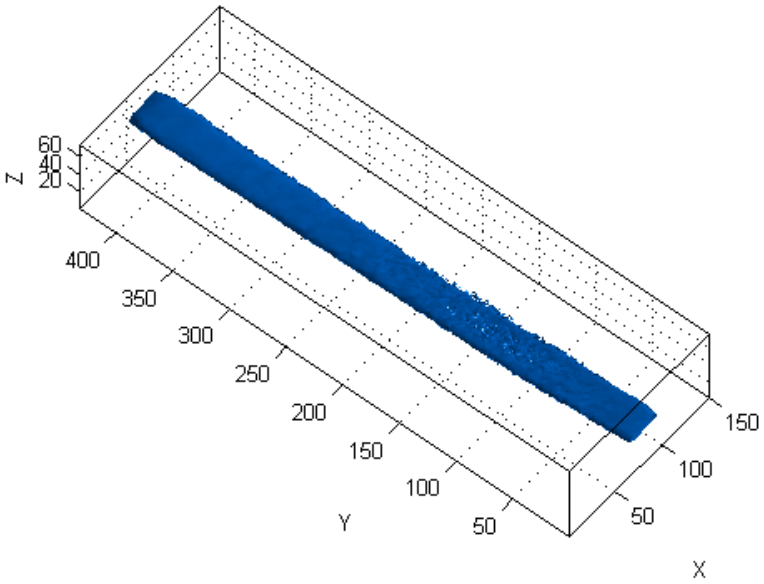


Figure 161 the Dimensionality of Sample in Voxels

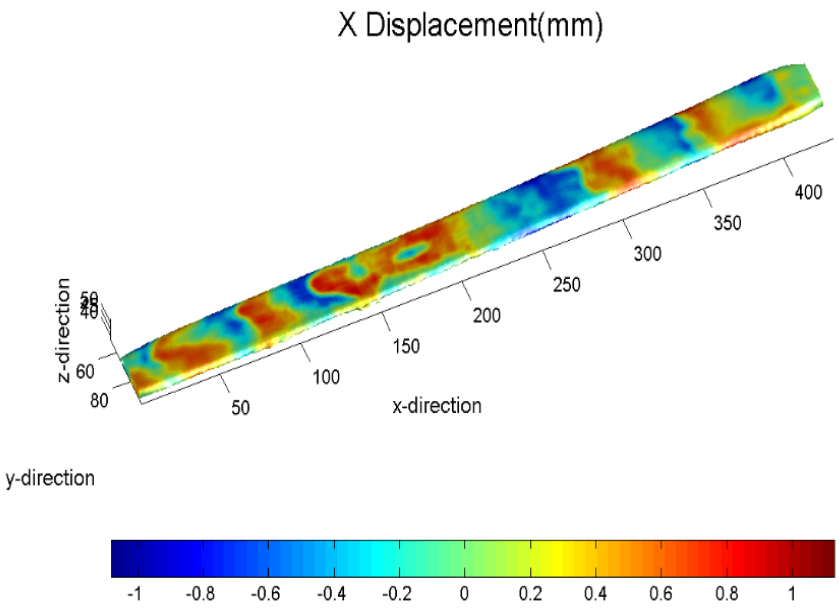


Figure 162 Beam X Displacement Contour

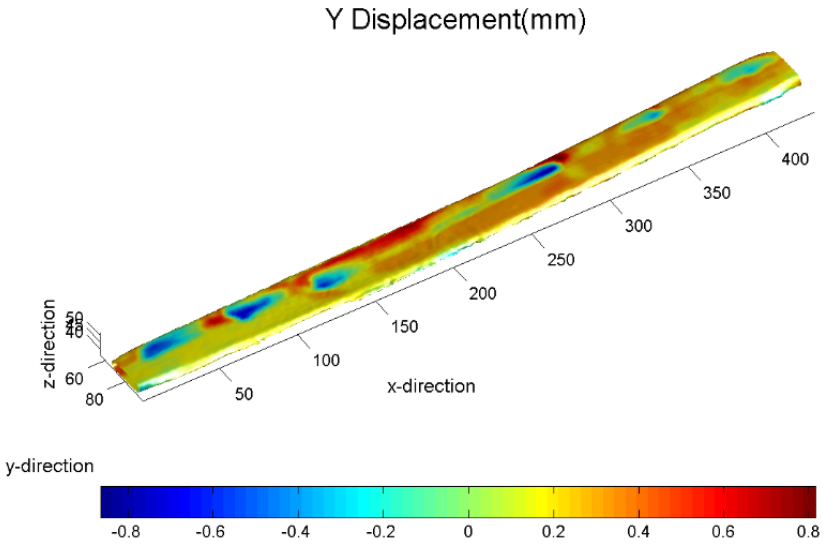


Figure 163 Beam Y Displacement Contour

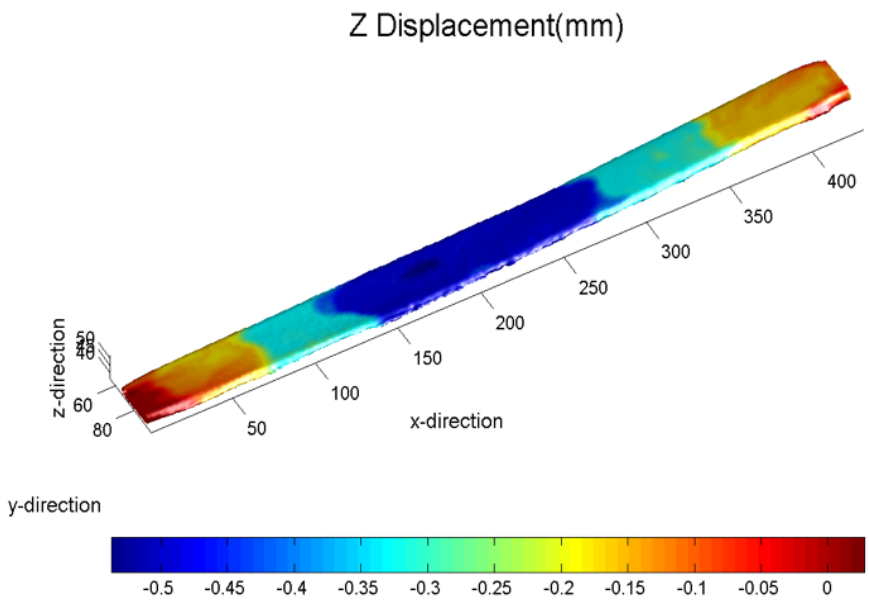


Figure 164 Beam Z Displacement Contour

We extract the displacement data of the middle plane in the Z direction (out of plane direction), then we apply the moving average to smooth the data curve. The displacements along the Z direction show in the following picture.

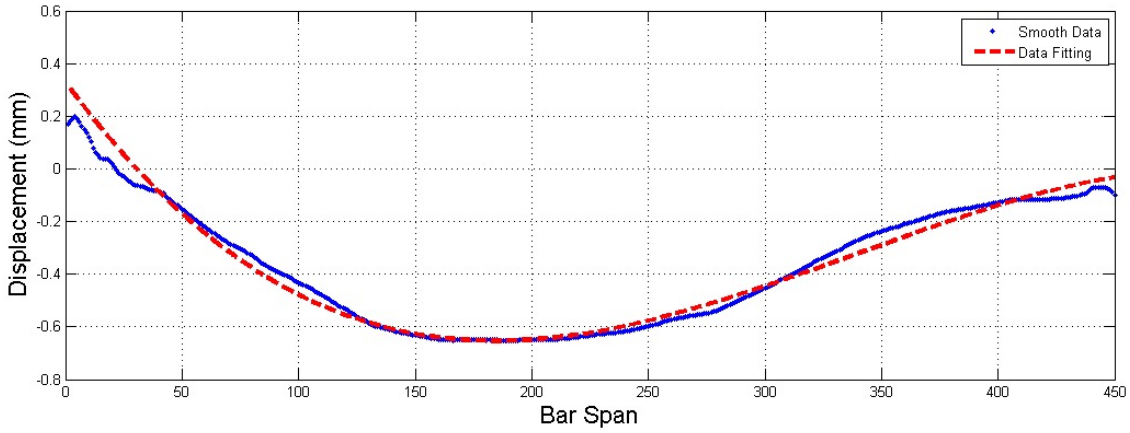


Figure 165 Aluminum Bar Out-of-Plane Displacement

5.2 LED Deformation Measurement:

We have developed a roadmap on how to quantify the deformation of LED chip area. The following flowchart shows the experiments that have been conducted to find the appropriate deformation magnitude of displacement on the LED chip.

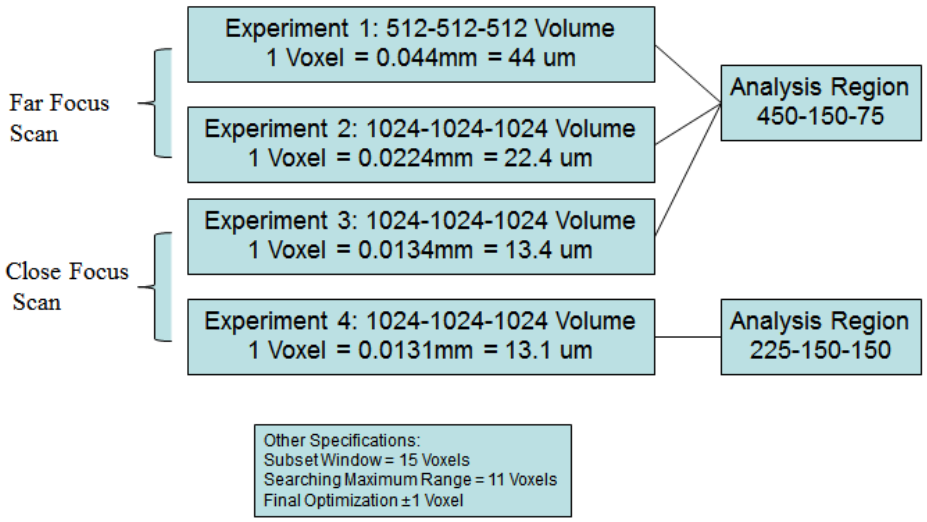


Figure 166 LED Test of Different Resolutions

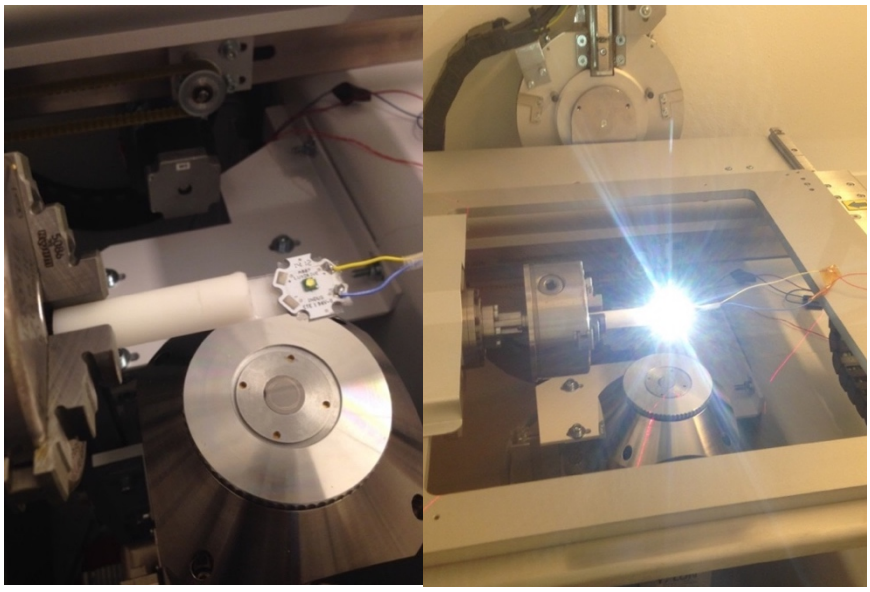


Figure 167 The undeform and deformed state of the testing condition

The LED chip area has been shown by the following 3D CT reconstruction images. The left shows the original scan volume, and the right shows the reduced computation area with the LED chip and wire bonds

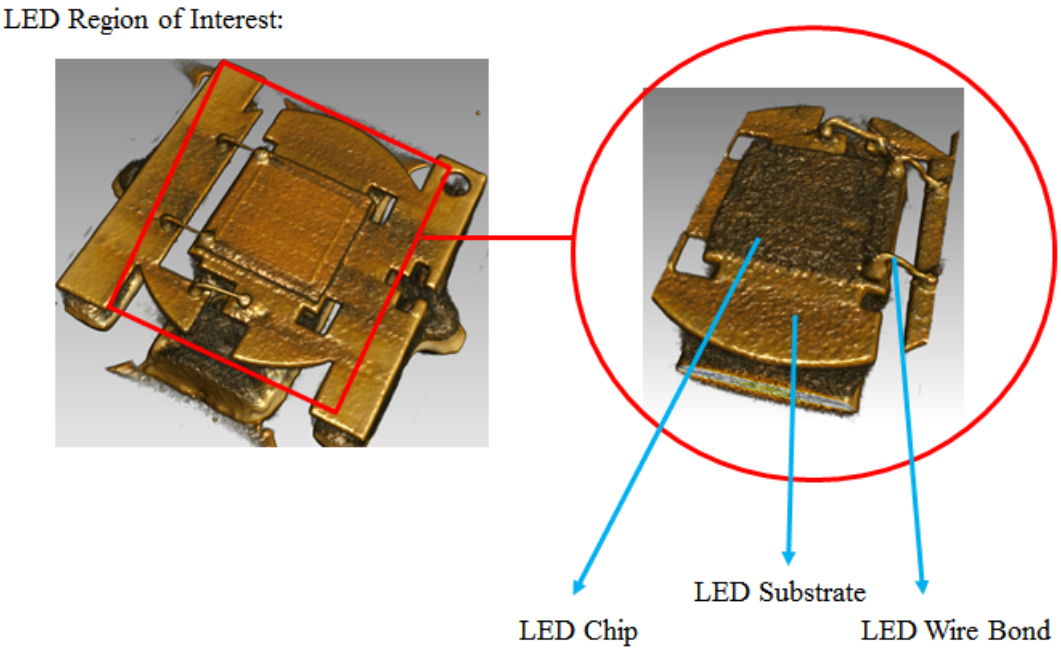


Figure 168 LED Region of Interest

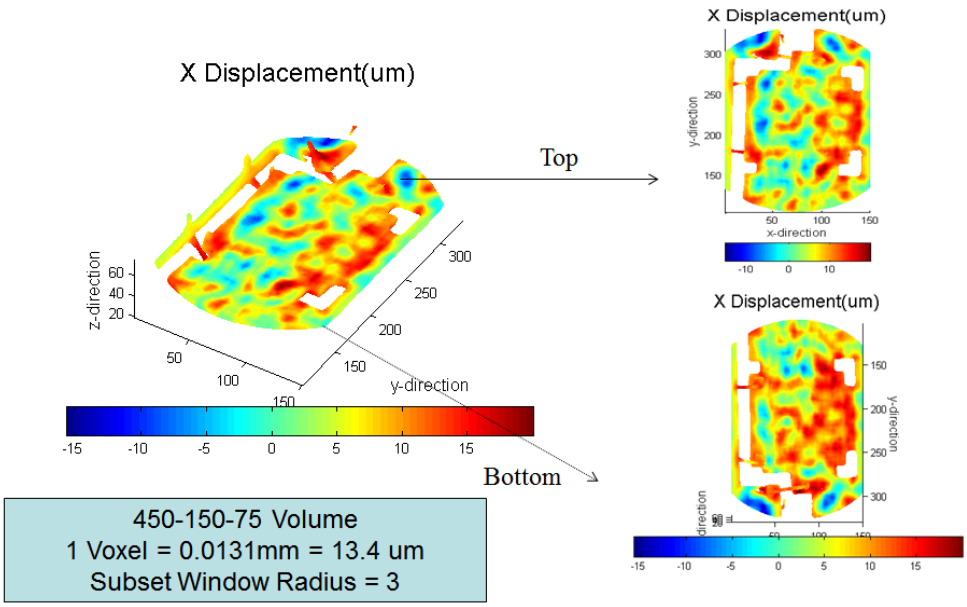


Figure 169 The LED chip displacement in the X direction

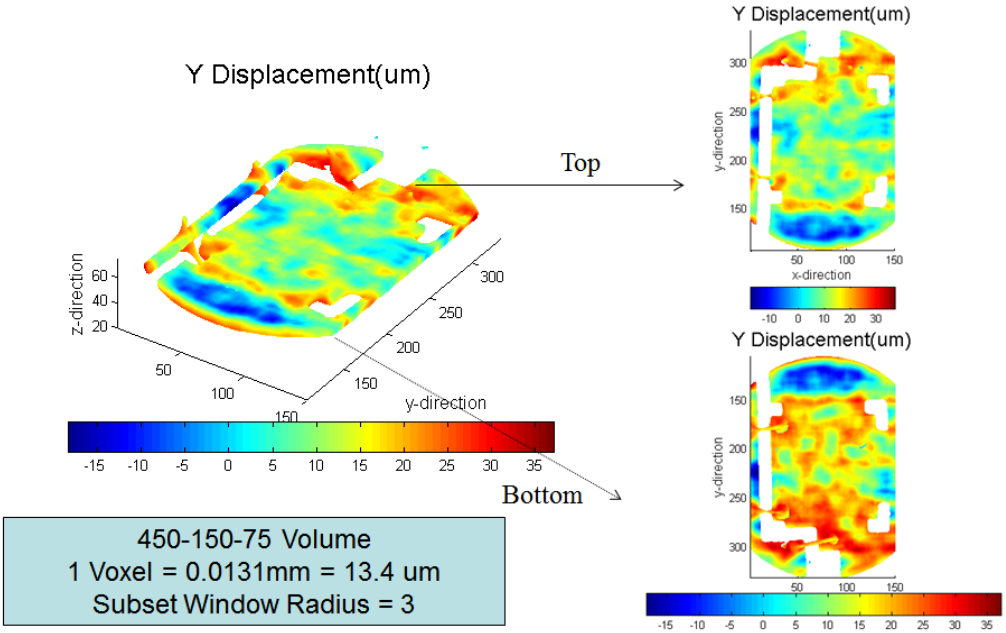


Figure 170 The LED chip displacement in the Y direction

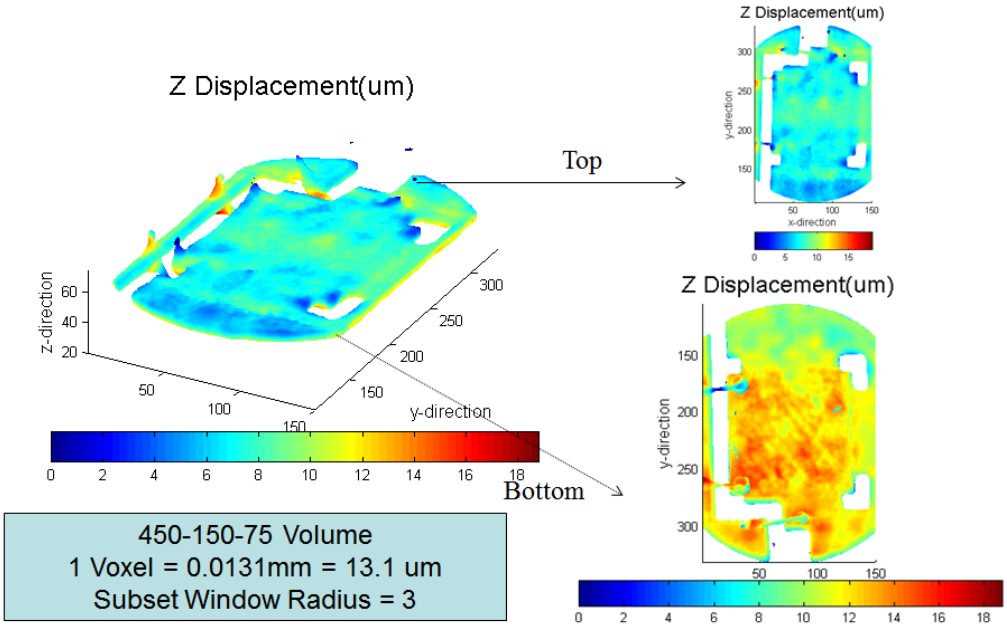


Figure 171 The LED chip displacement in the Z direction

The LED Chip Deformation Pattern Analysis:

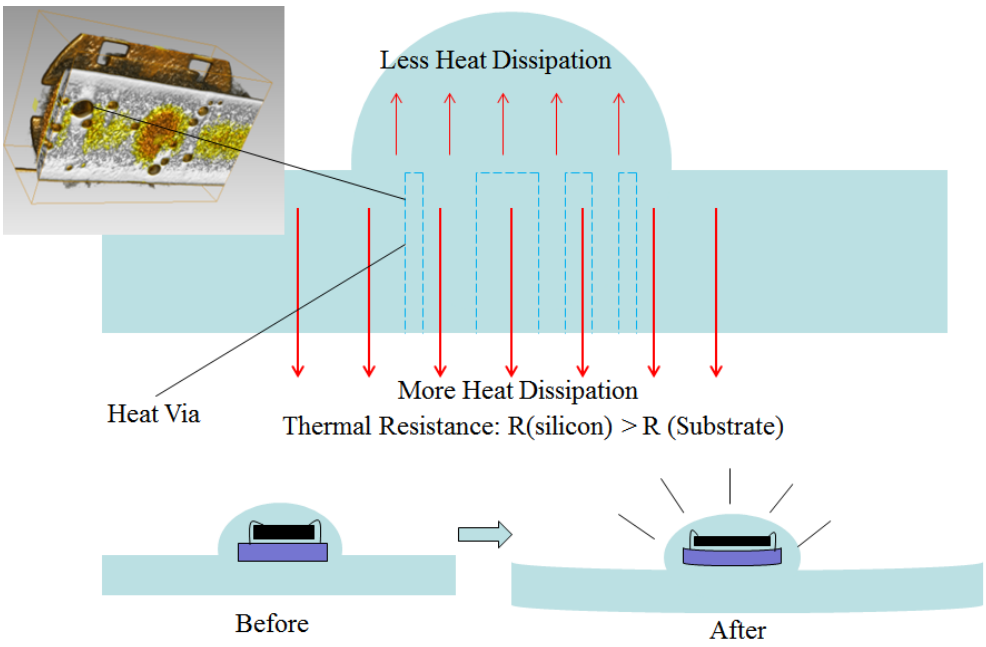


Figure 172 LED Thermal Dissipation

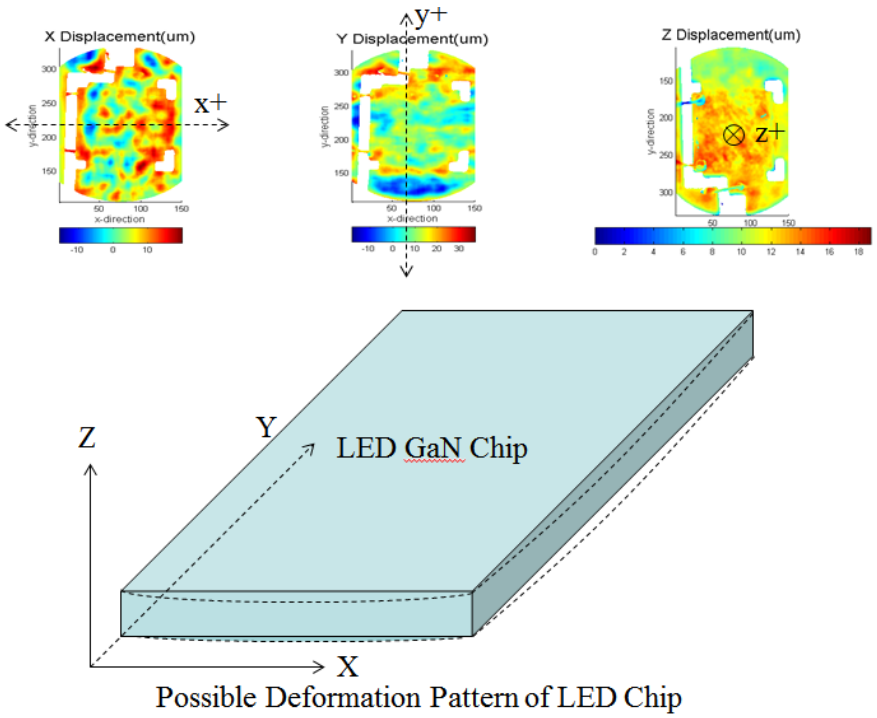


Figure 173 LED Thermal Deformation Pattern Analysis

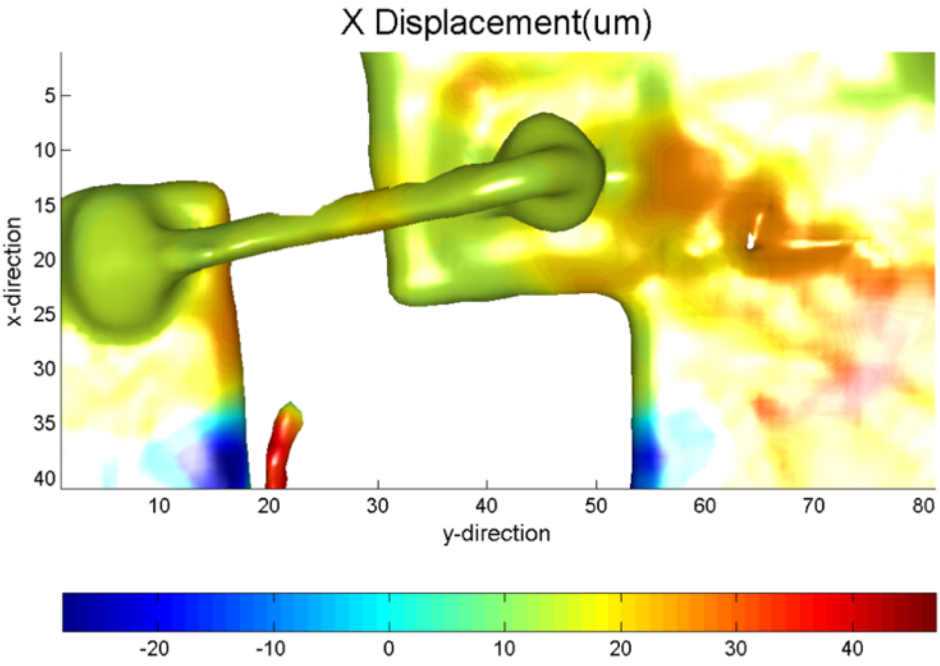


Figure 174 Wire-bond X Direction Displacement

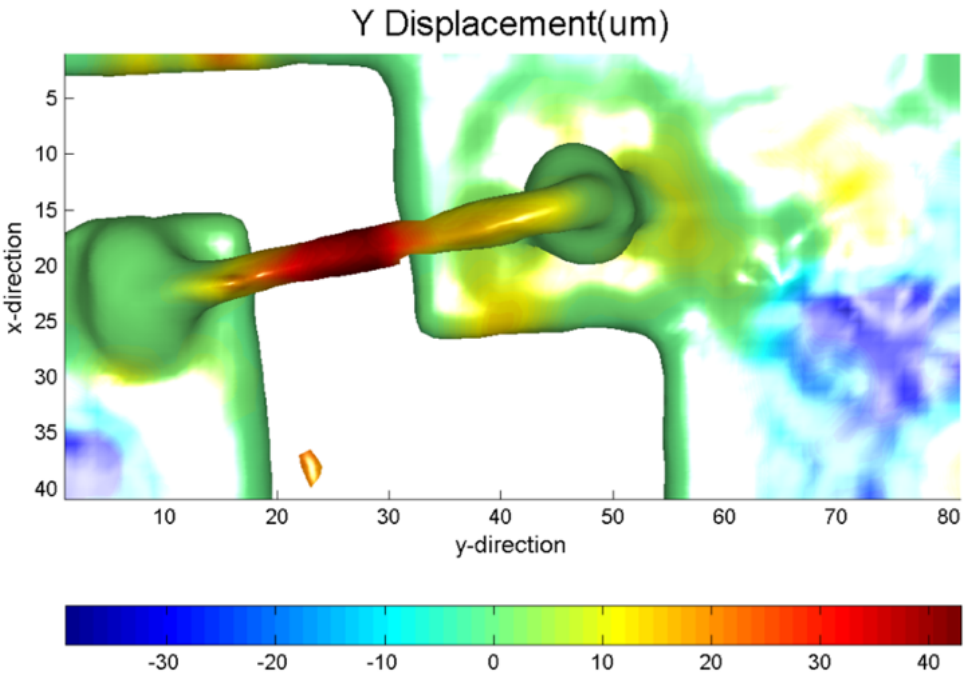


Figure 175 Wire-bond Y Direction Displacement

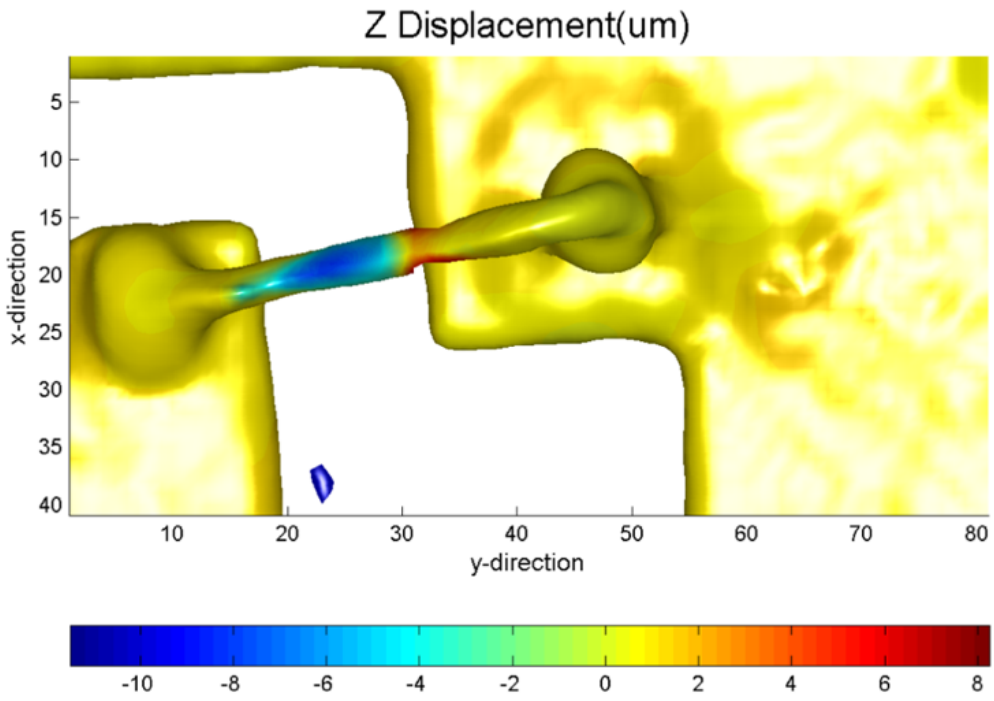


Figure 176 Wire-bond Z Direction Displacement

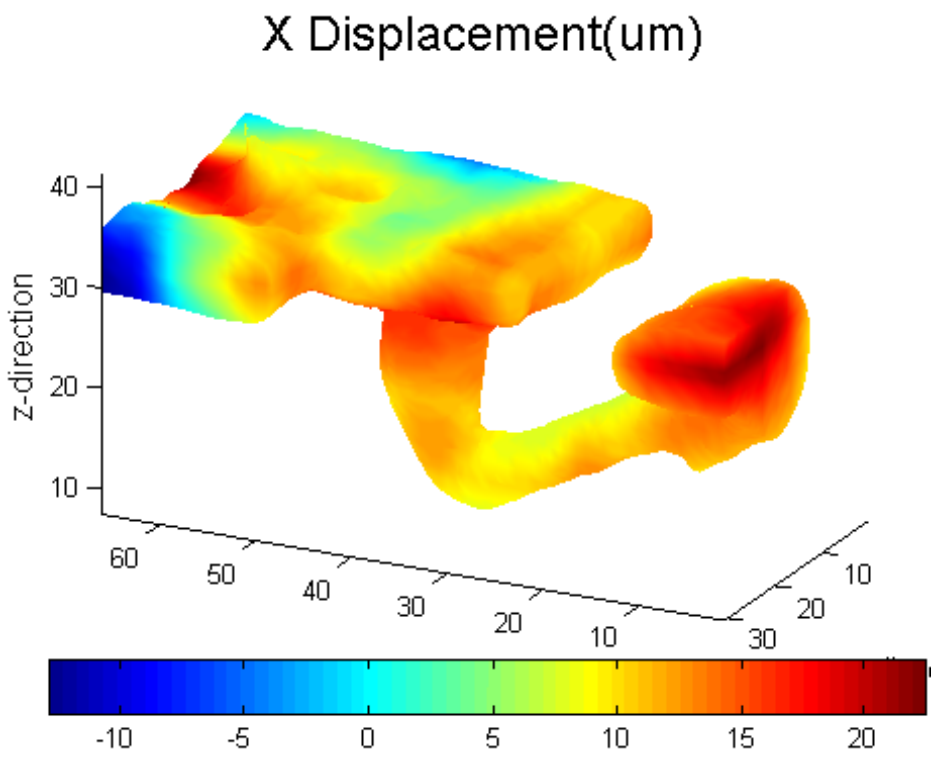


Figure 177 Slice View of Wire-bond X Direction Displacement

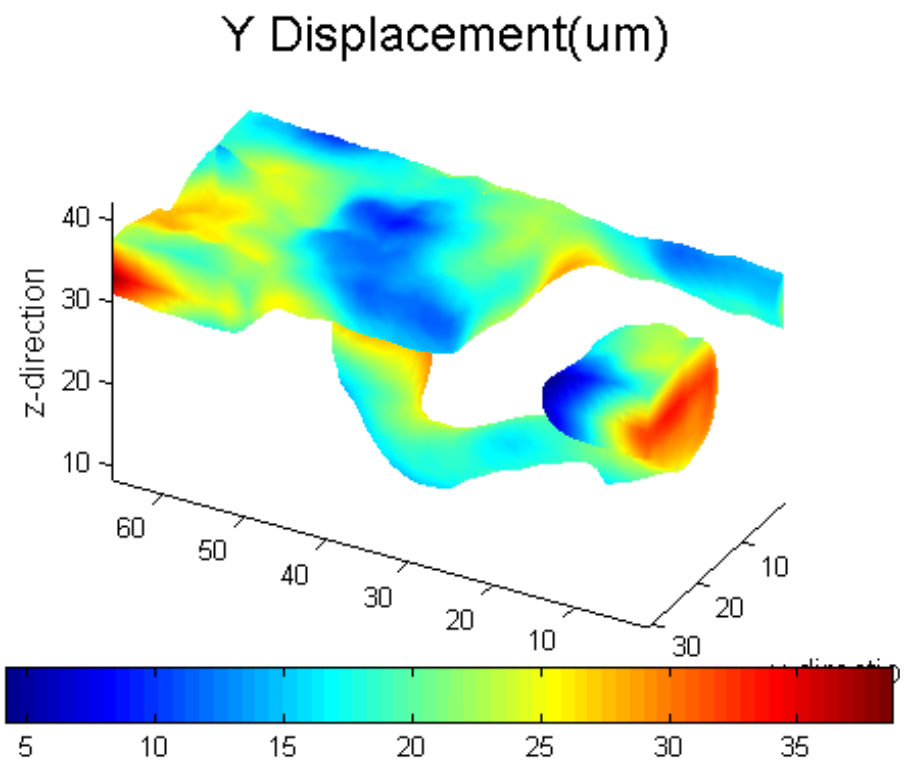


Figure 178 Slice View of Wire-bond Y Direction Displacement

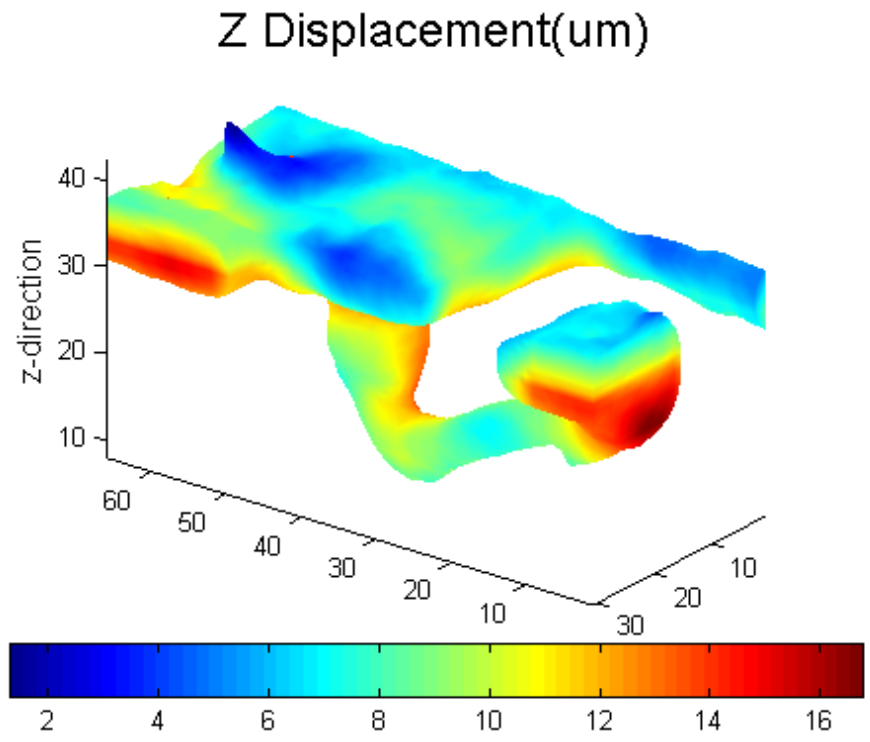


Figure 179 Slice View of Wire-bond Z Direction Displacement

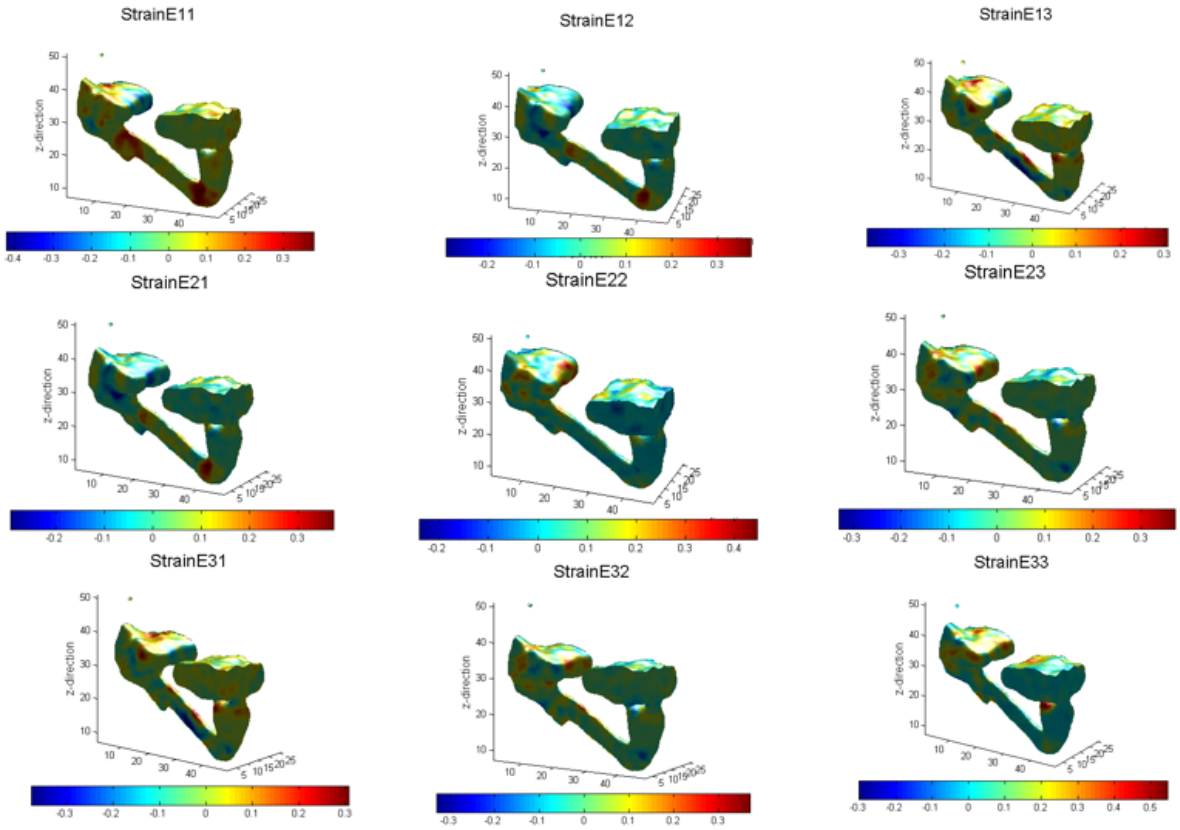


Figure 180 Wire-bond Strain Components

5.3 BGA Assembly Solder Joint Measurement:

The test package is PBGA 324 ball-grid array package with package parameter shows in the Table1. The package is daisy chain package with 324 I/O counts, and the body size is 19 mm. The ball matrix is 18 by 18, and it is full array package.

I/O Count	324
I/O Pitch	1mm
Body Size	19mm
Ball Matrix	18×18
Ball Alignment	Full Array
Package Type	PBGA
Board Finish	ImAg
Substrate Pad Type	SMD
Solder Ball Material	Sn3Ag0.5Cu
Substrate Pad Dia	0.45mm
Ball Dia	0.4mm
Mold Cap Thickness	1.17mm

Table 12 BGA 324

Figure 181 Physical Picture of the Test Vehicle

Experiment Setup: The setup has been used the cross-sectioning method to reach the silicon chip area of the package, which can easily introduce the deformation on solders under the die shadow region. The experiment configuration shows in the following figure. We applied the 0.25 mm displacement load at the top of PBGA.

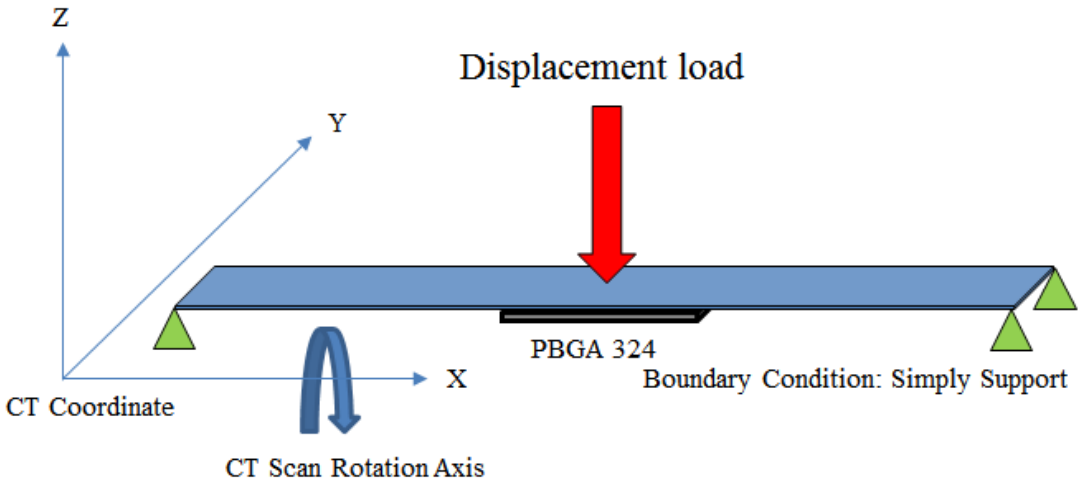


Figure 182 Experiment Configuration

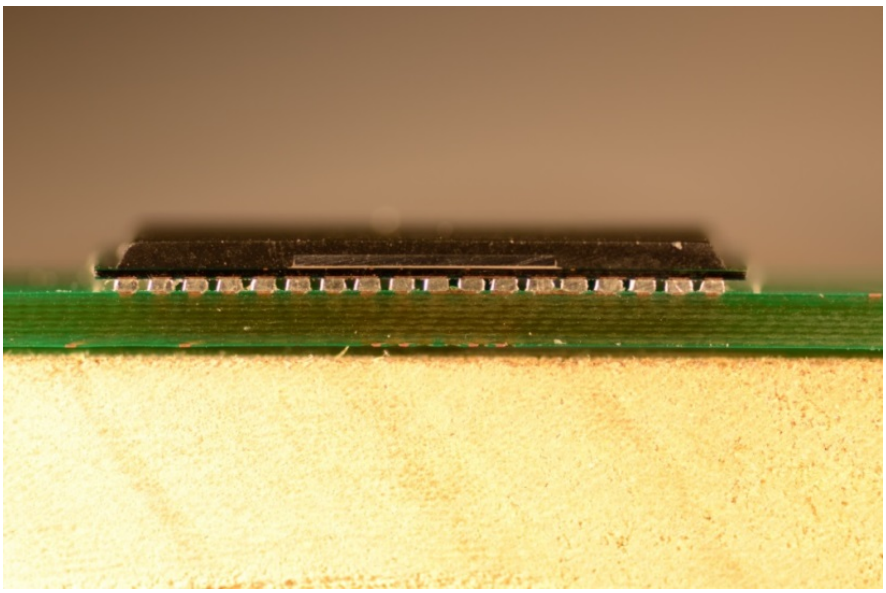


Figure 183 PBGA 324 Cross-Sectioning Optical Image

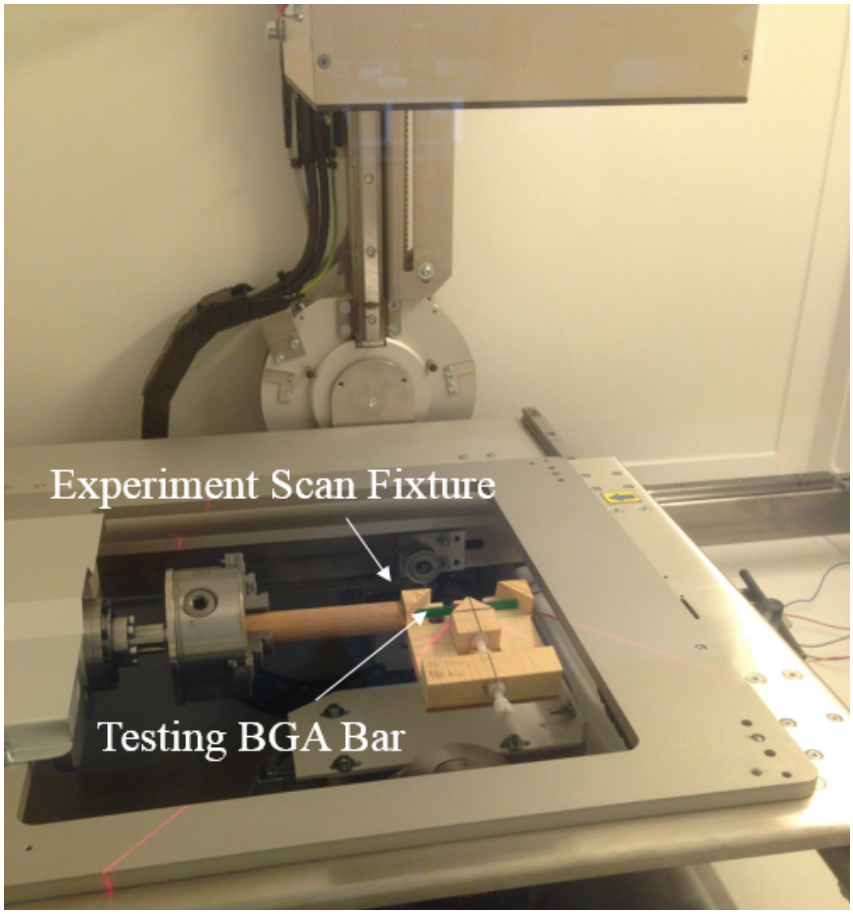


Figure 184 Micro-CT Scan BGA In-situ Bending Test

Micro-CT Scan Digital Volumes: The scanned PBGA digital volume has different regions, and they are: (1) silicon chip region (2)solder region (3)BGA substrate region (4)molding compound region. In the CT scan, the solder region, and silicon chip region is relative opaque to the X-ray, and the BGA substrate and molding compound is almost transparent to the X-ray. Therefore, the solder region and silicon chip region will be a primary region of analysis in the digital volume of CT reconstruction.

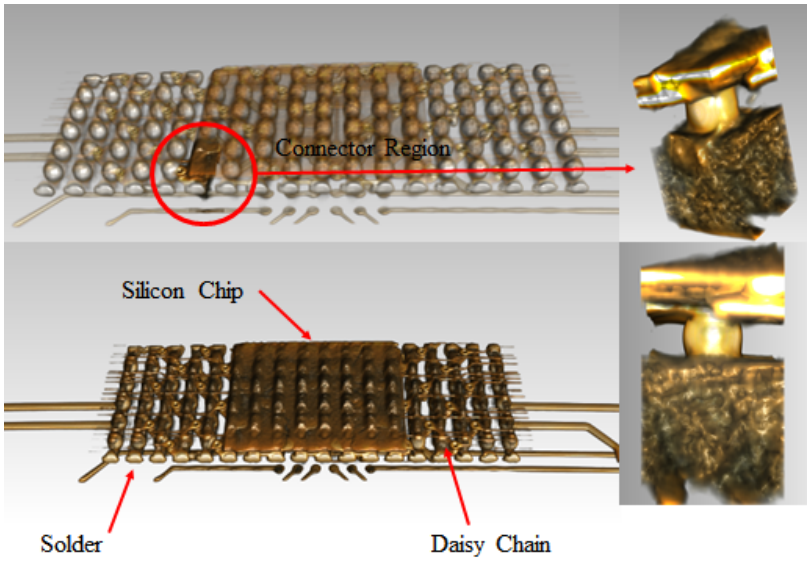


Figure 185 PBGA 324 Volume Graphics View with Solder Connection Region

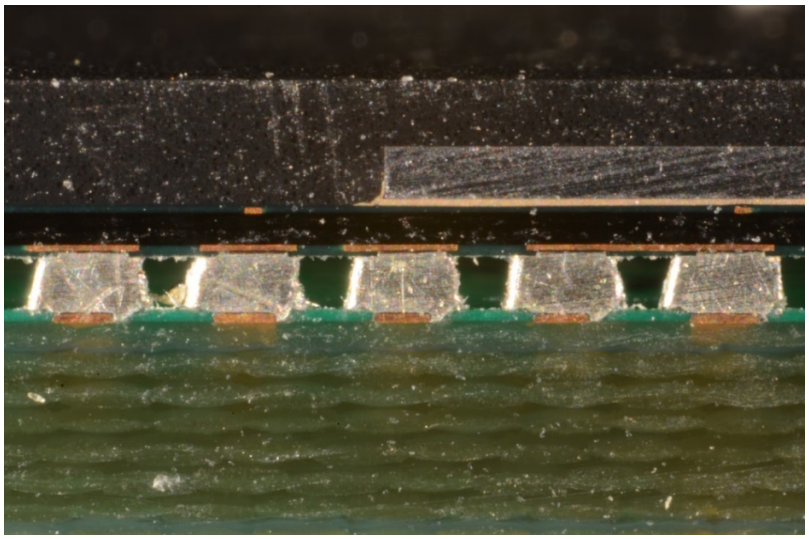


Figure 186 PBGA 324 Cross-Sectioning Die Shadow Region Solders

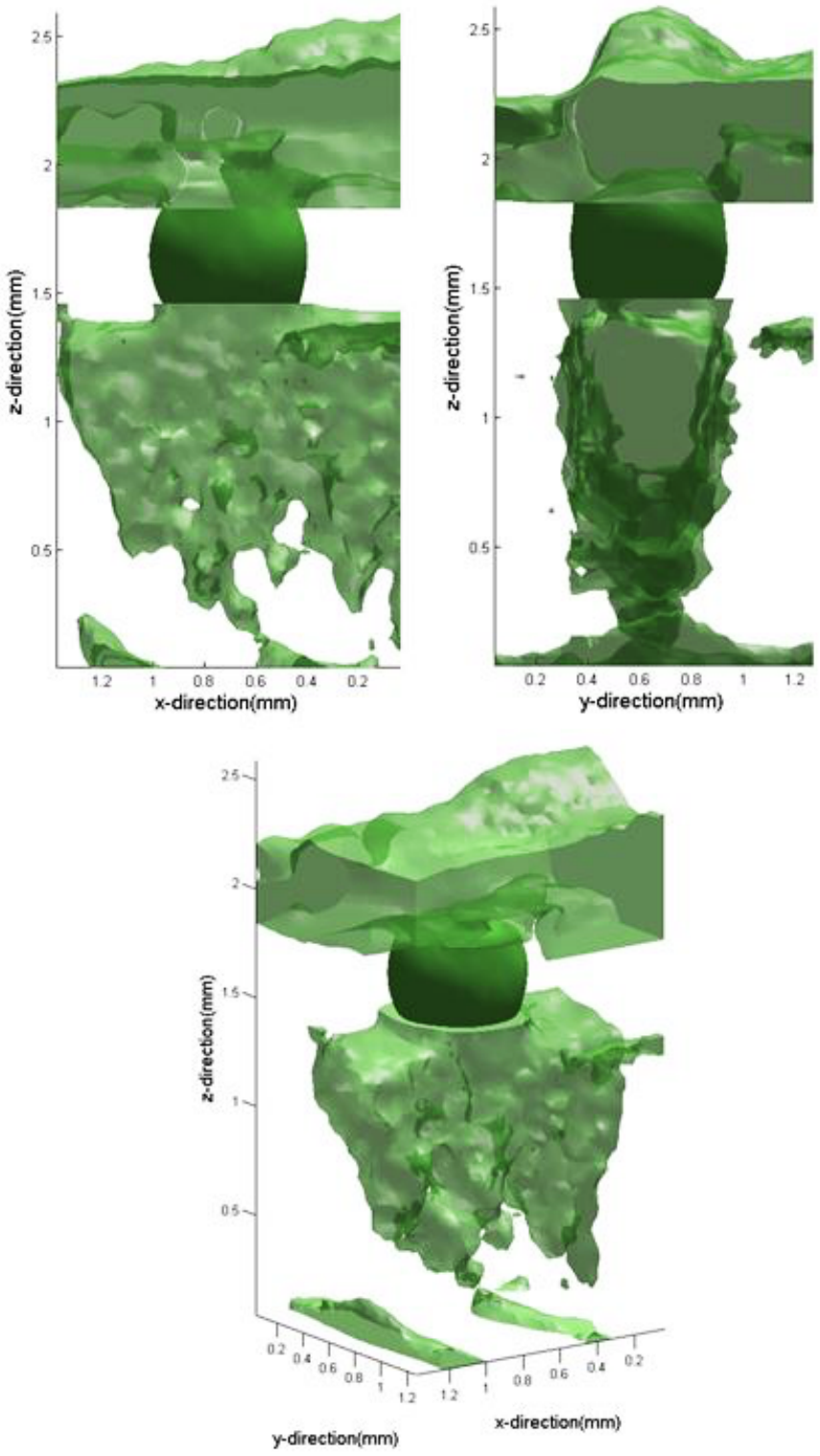


Figure 187 PBGA 324 Solder Connector Region X-ray CT Scan 3D Reconstruction Plot

It has been shown in the following Figure189, Figure190 and Figure191 that the displacements in each direction of the solder region. In the x-direction displacement, we find that there is 45° line contour, which there are many positive and negative displacements at the corner of solder ball due to the bending moment applied to the BGA board. Also, those displacements of the corner are symmetrical and are forming the 45° contours. In the y direction, we also find the similar contours, but it has much large displacement value than that in the x-direction since the BGA board is mainly bending in the y direction. Therefore, it can be concluded that the displacement in the y-direction is bigger than that in the x direction. Also, the 45° contour line is much more clear and continuous in the y-direction. In the z-direction of force applied, we can find the displacement is gradually increased from the bottom to the top of the solder ball. The biggest displacement is at the top solder ball, which is the almost equal to the displacement load 0.25mm. Those contours provide us the intuition of the displacements field of BGA undergoing the mechanical bending. The highest displacements at the bottom corner of the solder joints reduce the reliability of solder balls and pose the potential cracks from that point.

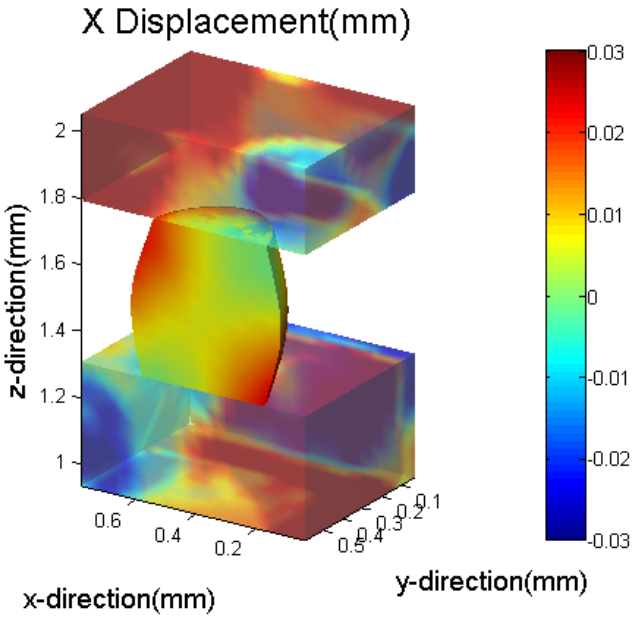


Figure 188 X Direction Displacement on Cross-Sectioning of X Direction

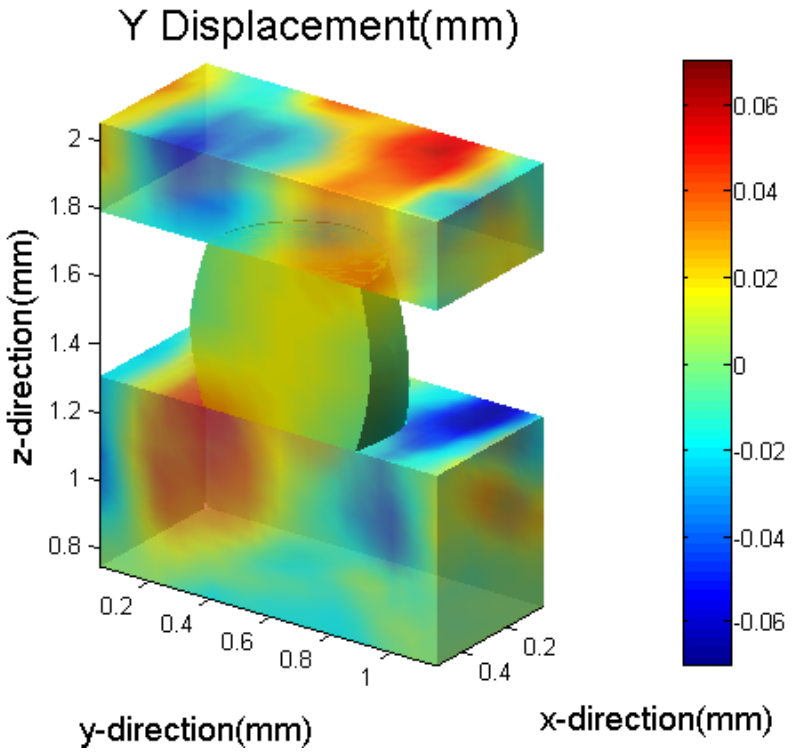


Figure 189 Y Direction Displacement on Cross-Sectioning of Y Direction

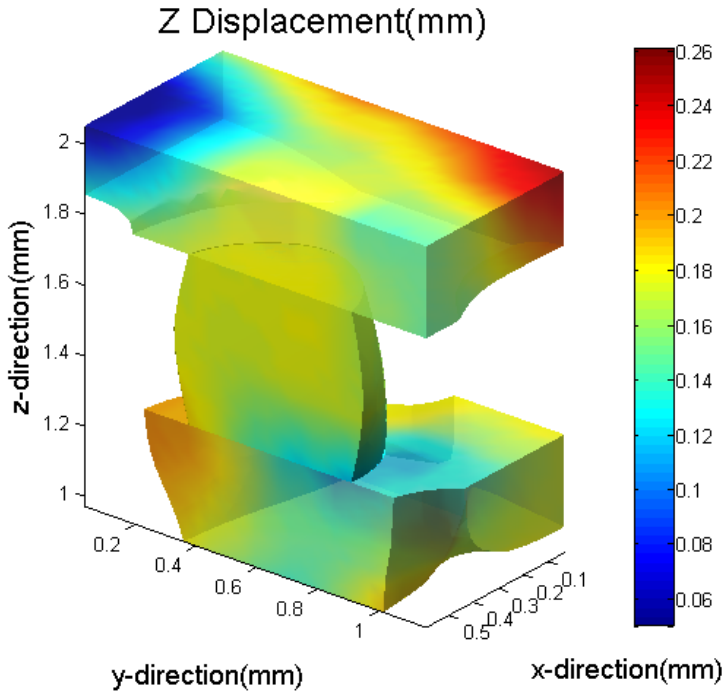


Figure 190 Z Direction Displacement on Cross-Sectioning of Z Direction

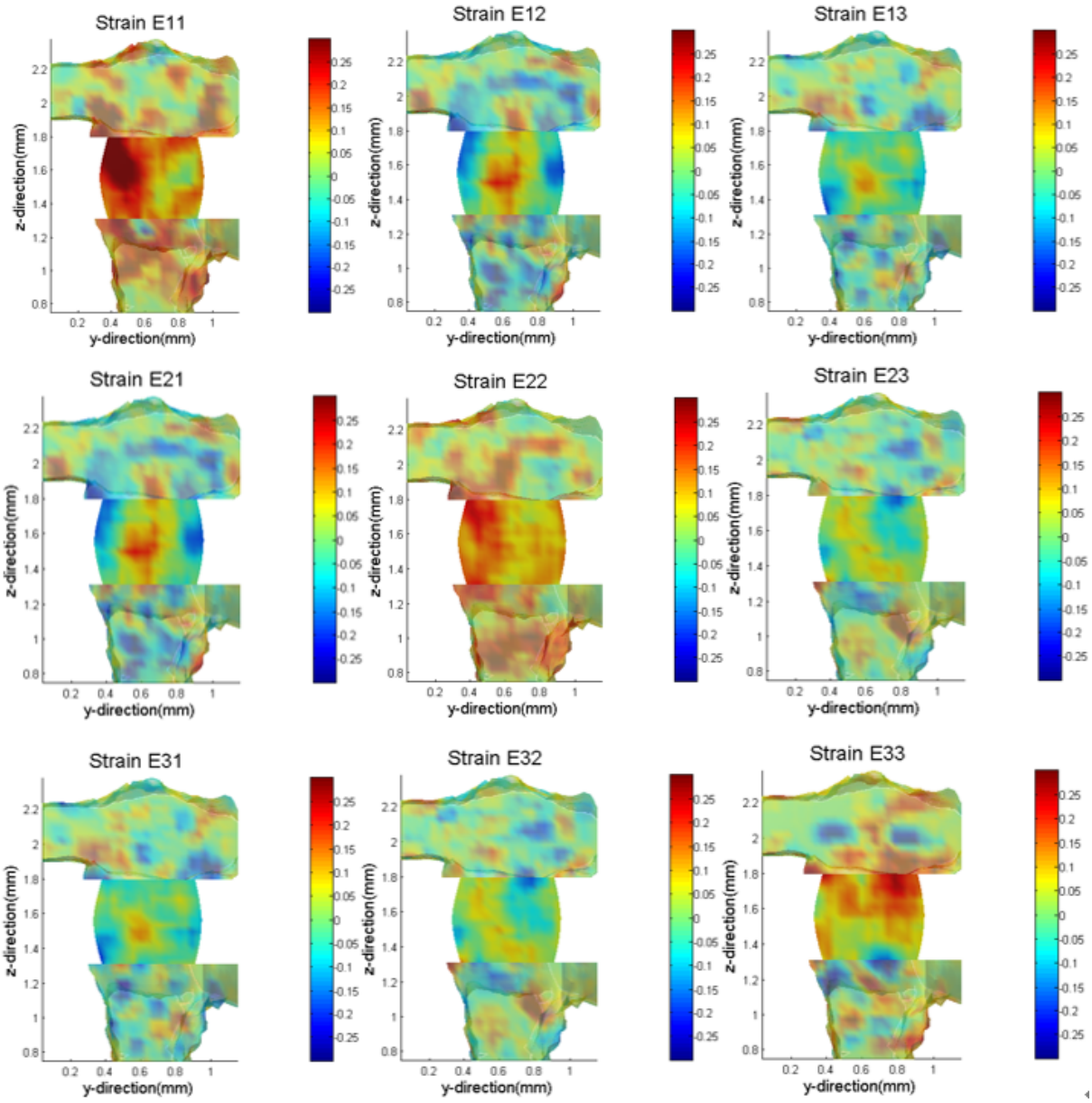


Figure 191 Strain Components Plots for Solder Joint

5.4 Soft Material and Flexible Electronics Deformation

As the flexible electronics become more and more important in the market, and the related techniques are becoming advanced and mature, it can be envisioned that many fields such as sports area, spacing navigation, and exploration, military as well as the medical area would be under

rapid development and change. To provide a solution for measuring significant deformation in the product testing, in this section, we introduce an improved technique, the Mesh-based Digital Volume Correlation, to bridge the gap between the real applications and in-situ lab testing. In this method, the deformation of the object has been computed by comparing two consecutive X-ray Based micro-CT scans. The reference scan is the baseline scan without any pre-history load, and the deformed scan is the target scan with identified mechanical loads to create a significant deformation about the baseline scan. Through the subset template matching algorithm, voxel by voxel, the deformation field containing a displacement vector at each voxel can be calculated. We improved the speed of original Digital Volume Correlation algorithm by creating a discretization in the spatial domain, and the vector field is calculated by the interpolation over the closest neighbor nodes. Also, the feature points matching is introduced into the optimization part to help the convergence. In the meantime, the validation of deformation field has been shown, in which the experimental computation matches with the analytical solution, and the error between the two has been compared and discussed. In the post-processing, the aligned two volumes have been plotted to show the closeness of the transformed deformed scan with reference scan using the given displacement field. In the end, the strain field has been computed basing on the input displacement field. This work demonstrates that there are the potential applications in flexible electronics with soft material base and high precision sensor system, where the improved geometrical design could be done, and more reliable materials could be tested under this system.

5.4.1 Introduction to Flexible Electronics

The flexible electronics is the future, and there is increasing demands on the market. In the last decade, the flexible manufacture has experienced an exponential increment, by the investigation and report of a professional website, it is expected that the total wealth of flexible

market is reaching 40 billion dollars in 2018. Many new devices and products have been made each year, which covers many areas including wearable sports band, electronics embedded medical device and smart watch. This huge demanding on the flexible electronics gives us the challenge to make more reliable electronics, in the meantime, this also provides us an opportunity to promote the development of market further. The challenge asks us to make a long-lasting electronics for people used in different situation. At the moment, the flexible electronic, in general speaking, is not totally "flexible," and the battery cannot be stretched and compressed due to the concern of reliability and safety. Moreover, the microprocessor has to be embedded into the flexible material substrate, therefore, making the flexible electronics little rigid in those applications. The new generation of flexible electronics, to some extent, has to be totally flexible, and this also leaves us a challenge to choose a more stretchable material like the PDMS or similar, and the electronics has to be well tested before it releases to the market.

The mechanical reliability is key to the success of flexible electronics, and we cannot sell the unreliable product to our customer. Until today, many manufacturers did not release the life span and expectancy of flexible electronics, because it is hard to know the life of flexible electronics in various situations including high strain, high temperature, humidity, etc. In the meantime, we carry those flexible electronics everywhere and every day, and it is vital to know the life expectancy of it for a particular environment. Therefore, the in-situ experimental testing must be done to meet the requirement of the industrial standard. In this section, we focus on the in-situ high deformation testing of flexible electronics, in which the components may fail instantly during to the enormous strains. The limitations are varied according to different materials, for example, the PDMS material is widely used in the future manufacturing due to its highly stretchable and extendable. However, this large stretchable material could create a tremendous

pressure to the embedded components like the high precision sensor or processor, and traditional destructive inspection could not be fitted because of the integrity of the system. Therefore, we introduce a non-destructive way to inspect each component reliability in the flexible electronics, and the state of interconnector and sensor could be monitored.

In the latest, more and more scholars begin to conduct research in this new area. In the literature, (Magdalèna Couty, Sébastien Nazeer 2012), it illustrates the process of manufacturing micro-antennas system with PDMS substrate. They applied these flexible electronics with the MRIs to study the rat brain during the scan. This low density and flexible material give high flexibility with the skin. Also, in the section, it shows a figure of the ultra-flexible and transparent feeling of the fabrication, and the microscope view of tiny alignment conductor line that width is about 100 microns has been presented. For the reliability study, in another related paper, (Quayle Chen, Anping Zhao 2010), it runs a FEM simulation on the PDMS substrate electronics to study the large strain effects. Recently, there is a developing surface measuring technique, which has been described in the paper (Matthew D. Parker 2015), and it develops a surface tracking algorithm using stereoscope. The highest resolution of the measurement is about 2.3 mm, and the error of measurement is about 10% comparing with the actual. Previously, we have successfully applied the Digital Volume Correlation technique with traditional electronics with hard material, in the paper (Lall, Wei 2015), the measurement has been made non-destructively by the CT scan of flip-chip BGA package and LEDs. Later in the article (Lall, Wei 2016), it makes a further development of the DVC algorithm to utilize the spatial discretization to increase the computation efficiency and pseudo real-time applications. However, there is a possibility to adapt this technique to the flexible electronics, and a solution of large deformation compensation is required to be sought.

In this study, we demonstrate several flexible assemblies to give a test and show "real-time" measurement of displacement and strain from the test. The scope of flexible parts ranges from the large flexible band with electric connector lines to the micro-electronic-mechanical systems (MEMS), a tiny high precision strain gauge. The deformation can be made in the global bird's-eye view, or a local high-precision measurement can also be achieved, which are according to the region of interest and the density of computational nodes. Initially, we take two measurements, and each one is converted into a digital volume containing the features by using different grayscale values. By statistical comparison and registration technique, the relative "difference," the volume translation and rotation at each voxel, can be found computationally. The experimental resolution is determined by the geometry of scan specification, which the smaller object usually gets high resolution in the scan. In order to validate the computational result, the analytical solution of beam deflection has been compared with the experimental computation, and the strain values have also been compared to the analytical solution and experiment. Lastly, we demonstrate the displacement and strain calculation for the flexible electronics' system. To better visualize the result, we show the 3D deformation surface contour plot and deformation vector plot within the 3D object.

5.4.2 Test Flexible Samples

The test samples in this study are made by the silicone flexible material in the electronics fabrication. The silicone material is liquid-like material, and after several hours curing the small processors and sensors could be mounted onto the sample. The following figure shows several samples: In the A, it shows the fabricated flexible electronics with Band-Aid on the top, and there are several connected gages beneath it. In the B, it shows the large extended trace strain gage that spans over the entire soft material substrate. In the C, it shows a little strain gage, representing a

high precision MEMS on the soft material. All the samples are electronically connected to track the state of the sample during the test. We have tested each of those samples under the X-ray Micro-CT to show the digital 3D reconstruction and measurements basing on the grayscale information.

We designed and printed fixtures using a 3D printer to hold the samples for the CT scan, in the following figure (A), it is showing the 3D printed fixture in a high-resolution scan where the distance between the object and X-ray generator is only about 1 inch. In the B and C, it shows the three-point bending test fixture with flexible electronics and soft material on it.

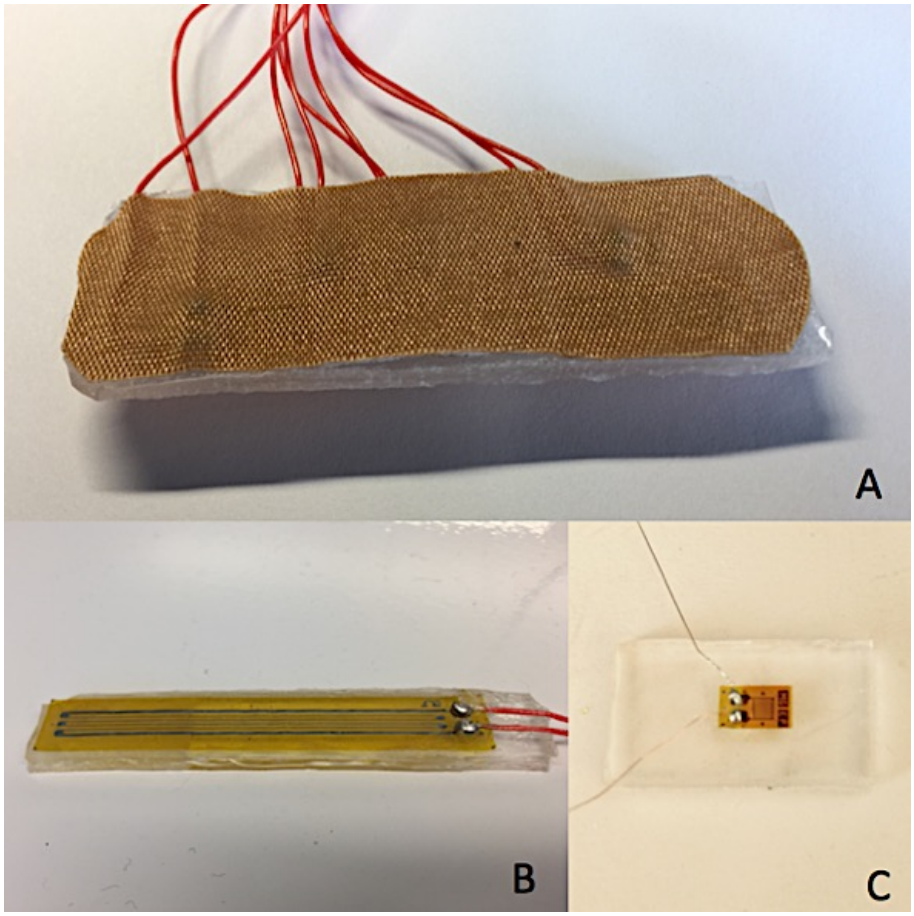


Figure 192 Silicone Soft Material Samples; A: Flexible Electronics with Band-Aid; B: Soft Material with a long trace strain gage; C: Soft Material with tiny high precision strain gage

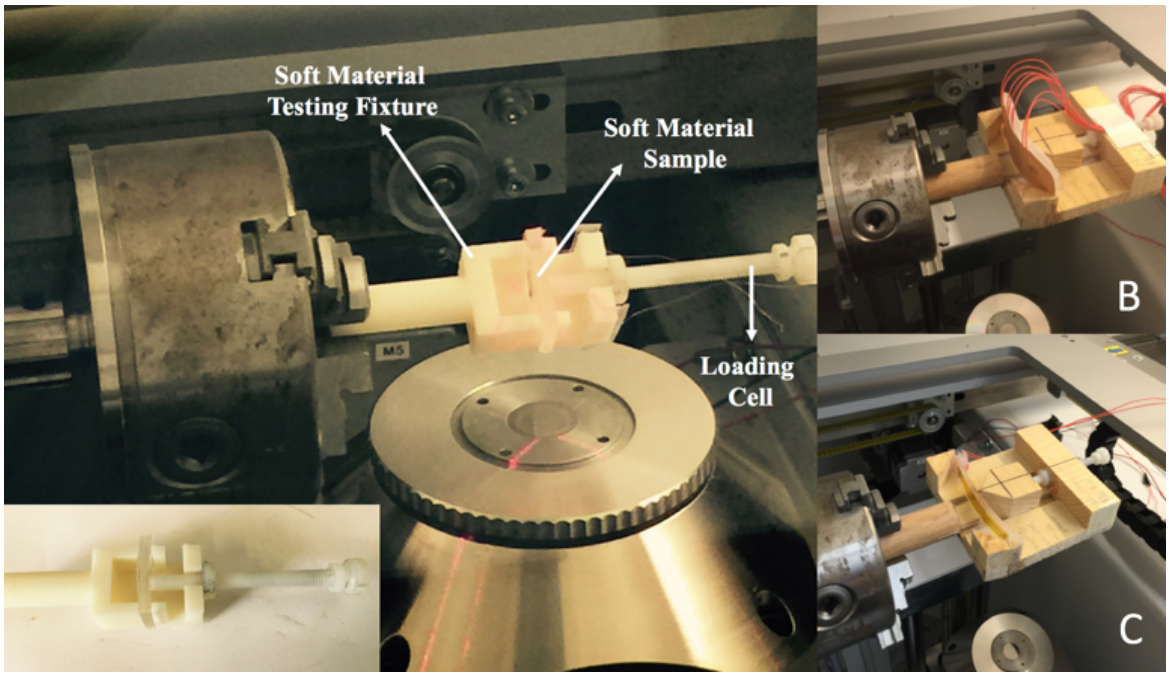


Figure 193 The Soft Material Tests and Sample and Fixtures; A: High Resolution In-situ Testing; B: Bending Testing for Flexible Electronics; C: Bending Testing for Silicone Soft Material

In the soft and low-density material testing, the density of fixture has to be lower than the sample material. Otherwise, the information about the low-density material will be lost. In the following figure, it shows the 3D printed ABS fixture and the test sample. We choose the ABS material for the substrate due to the lower X-ray density compared to the silicone material, and in the projection images, we can see that the sample is the middle dark region, which has much higher grayscale value than the fixture region. Also, in the figure, the projection of flexible electronics has been shown, and the sensors on the sample are connected by the electric wires.

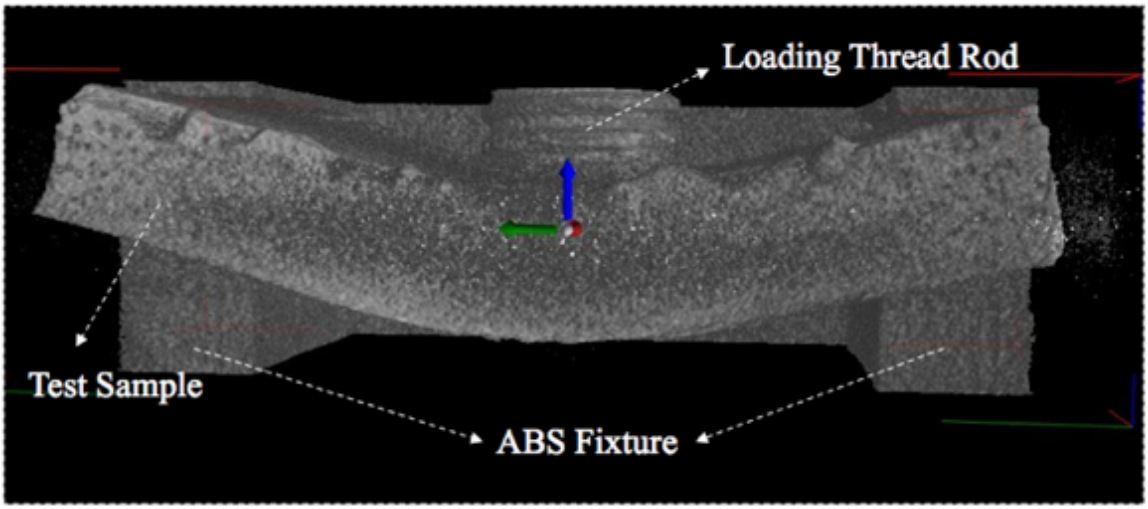


Figure 194 Soft Material and Fixture 3D Reconstruction

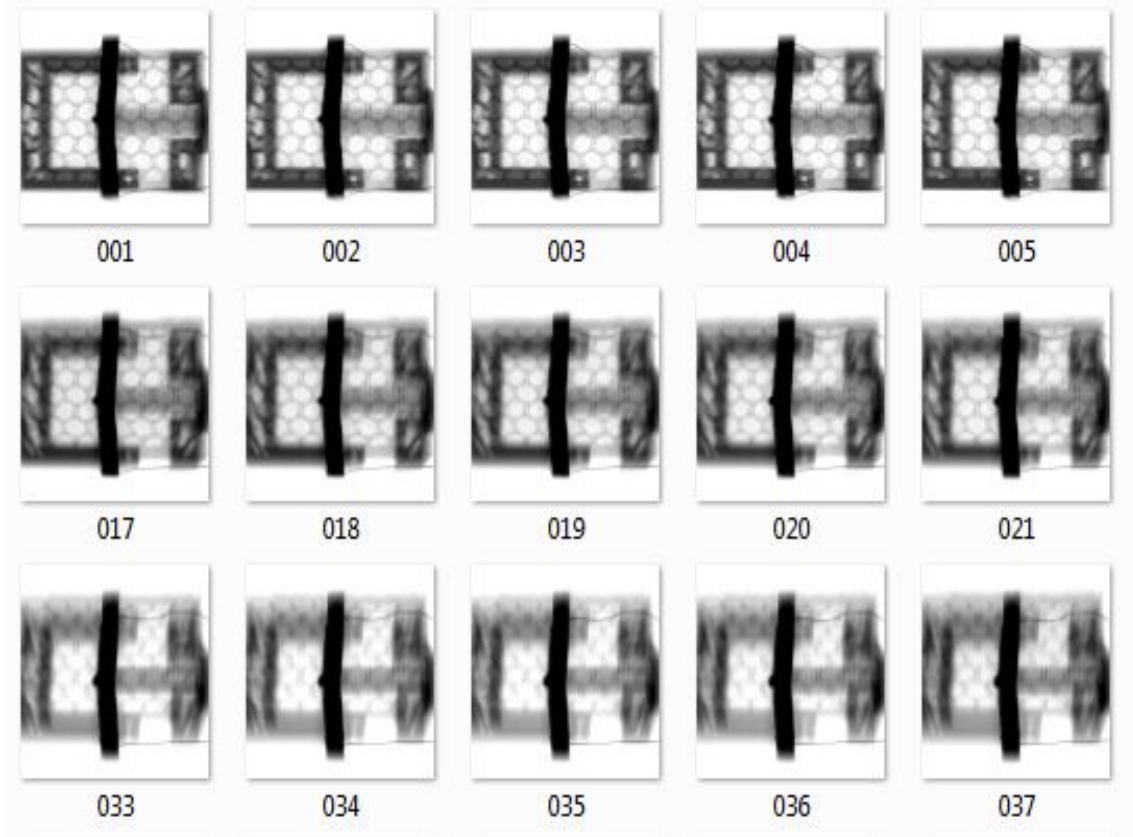


Figure 195 The Test Geometrical Set up and X-ray Projections

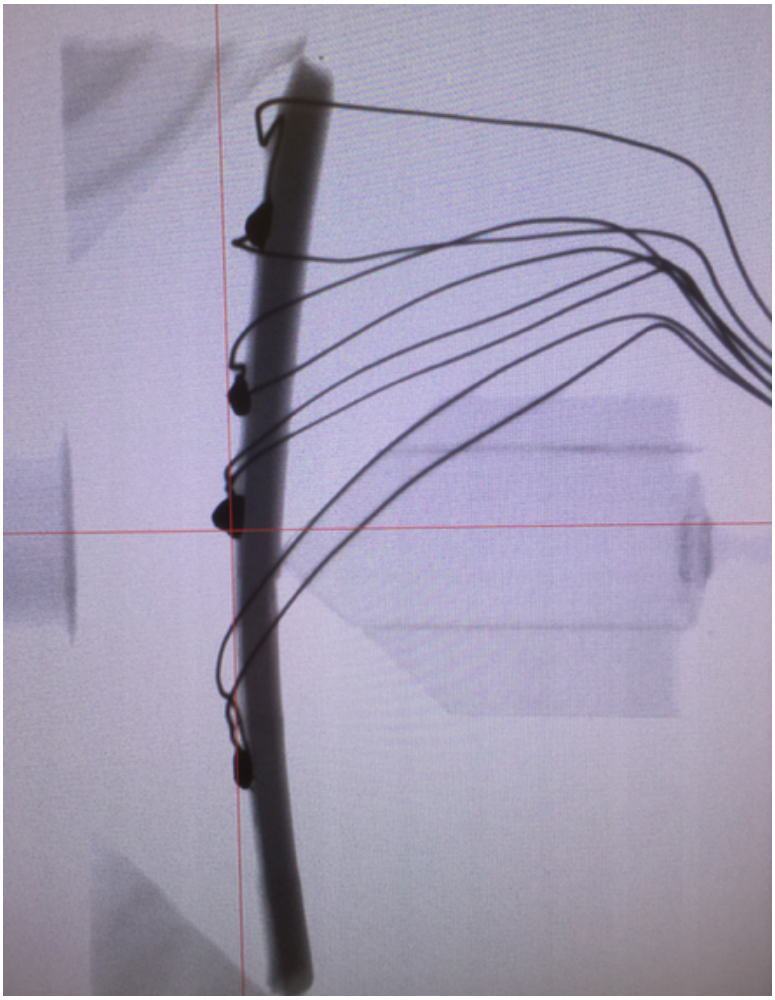


Figure 196 Flexible Electronics with Electric Wire Connections

5.4.3 Soft Material X-ray CT 3D Reconstruction:

The histogram of reconstruction has many features and local peaks, in the following figure, there are several regions, in which the air region has lowest grayscale value and the solder and wire region has highest grayscale value. To properly render the soft material, the medium level grayscale values, the dynamic range of histogram should be adjusted accordingly. To lower the X-ray power, the specified current, and voltage, can increase the dynamic range of histogram, but, its contrast will drop. In the experiment, we reach a balance to choose the 75KV voltage and 10 A

current, in the meantime, we eliminate the air voxels by the partial selecting histogram in the X-ray projection images. Therefore, the shape of low-density material can be rendered, and material textures are clear to be seen. In the following figure, we can see the textures of our lab fabricated silicone substrate in flexible electronics, and in the image, the surface concaves that are about 0.21 mm in diameter have been shown. Those surface concaves can be imaged using the high-resolution micro-optical DSLR camera, and we made a comparison of 3D reconstruction image with the optical image to verify the righteousness of CT reconstruction.

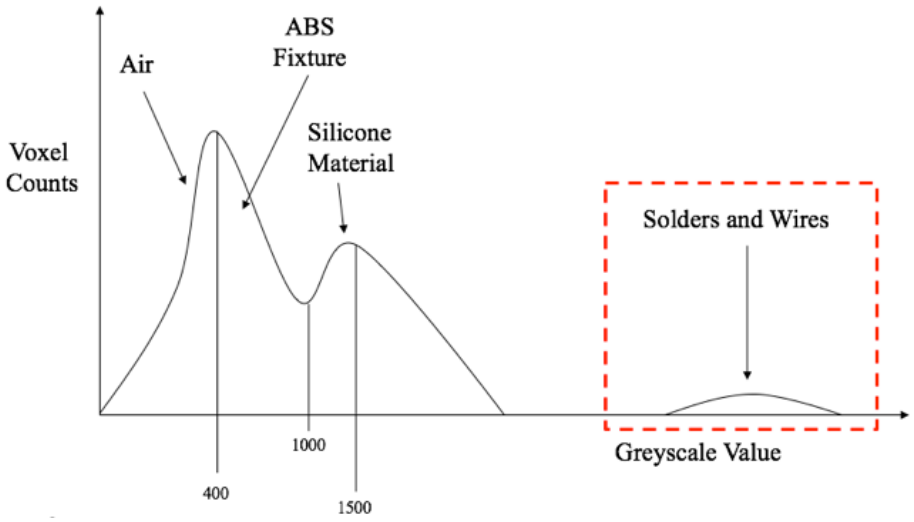


Figure 197 Histogram of Flexible Electronics Scan

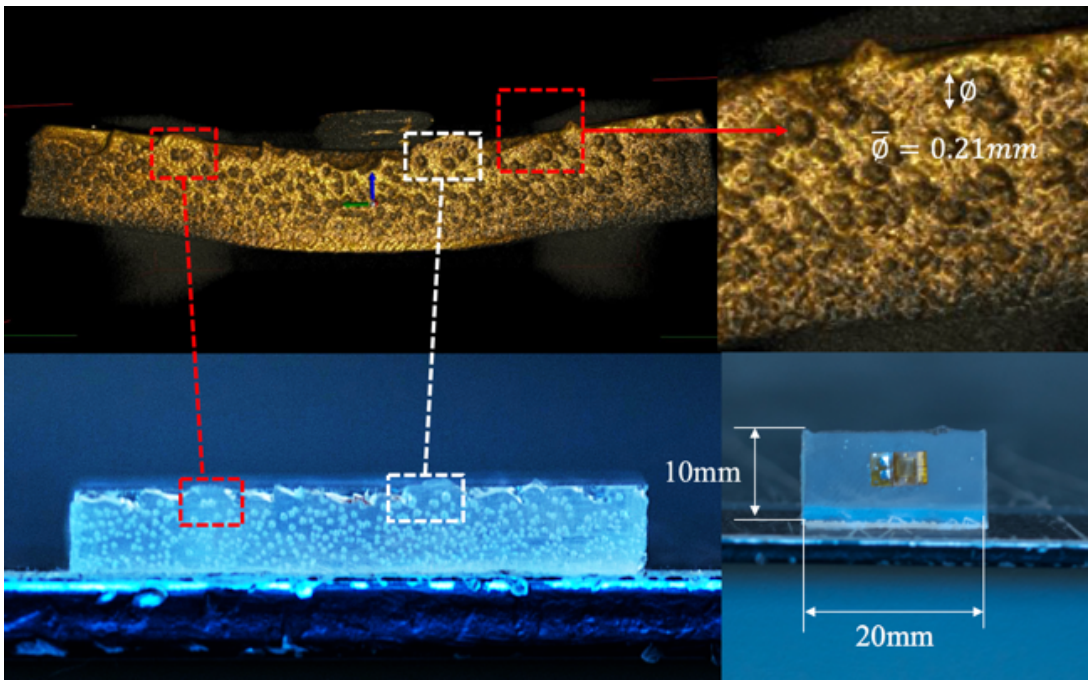


Figure 198 Soft Material Texture Verification

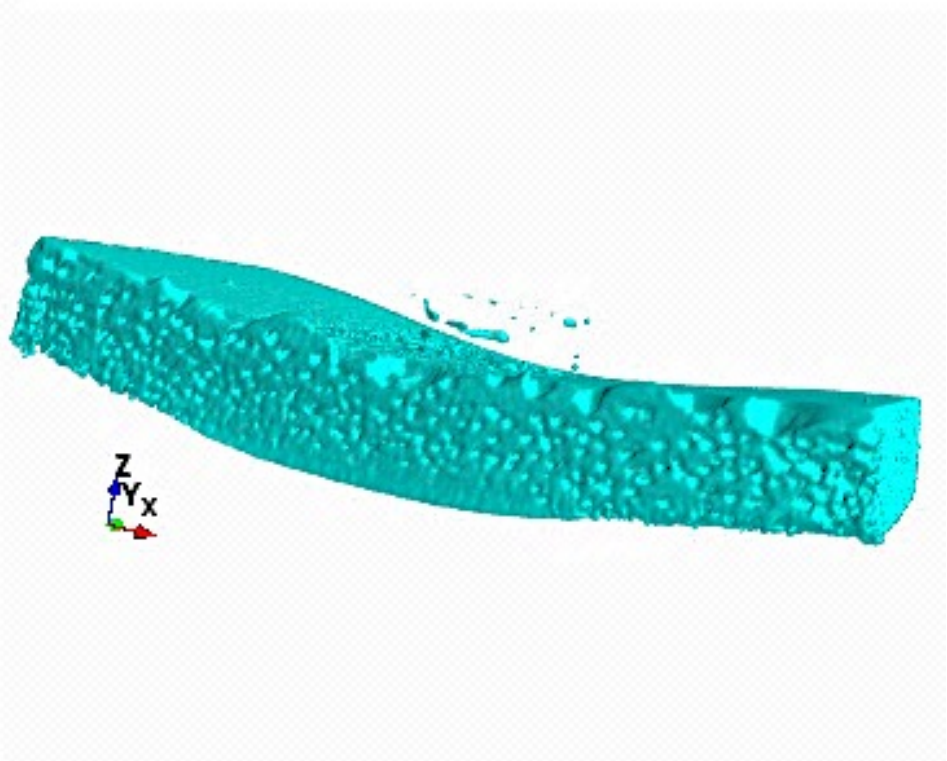


Figure 199 Soft Material Reconstruction Compared with High Resolution Optical Image and 3D Reconstruction Model

5.4.4 FEM-Mesh Based Digital Volume Correlation in Soft Material:

In the Mesh-based Digital Volume Correlation, the displacement between two volumes is calculated by the Zeroed-Normalized Cross-Correlation (ZNCC) criterion, which was original from the 2D Digital Image Correlation (DIC) algorithm (P.E.Anuta 1970). The technique has been used for measuring the engineering deformation, displacement, strain, and optical flow (S.S. Beauchemin 1995). In the original algorithm, at each point of interest, the block template has been formed using the block voxels, which contains the features to be compared with the neighbor blocks. The best estimation block has been selected by the highest ZNCC value, which the center is characterized as an optimal match point. The formula shows the following: it calculates one correlation value by looping over all the voxels in one template, and the computation can be improved by vector calculation using matrix.

$$C(u,v,w) = \frac{\sum\sum\sum[I_1(x,y,z) - \overline{I_1}][I_2(x+u,y+v,z+w) - \overline{I_2}]}{\sqrt{\sum\sum\sum[I_1(x,y,z) - \overline{I_1}]^2 \sum\sum\sum[I_2(x+u,y+v,z+w) - \overline{I_2}]^2}} \tag{165}$$

$$\overline{I_1} = \frac{1}{(2M+1)^3}\sum_{i=-M}^{M}\sum_{j=-M}^{M}\sum_{k=-M}^{M} I_1(x+i,y+j,z+k)$$

$$\overline{I_2} = \frac{1}{(2M+1)^3}\sum_{i=-M}^{M}\sum_{j=-M}^{M}\sum_{k=-M}^{M} I_2(x'+i,y'+j,z'+k)$$

$$(x,y,z) - \text{spatial coordinate}$$

$$(u,v,w) - \text{displacement vector}$$

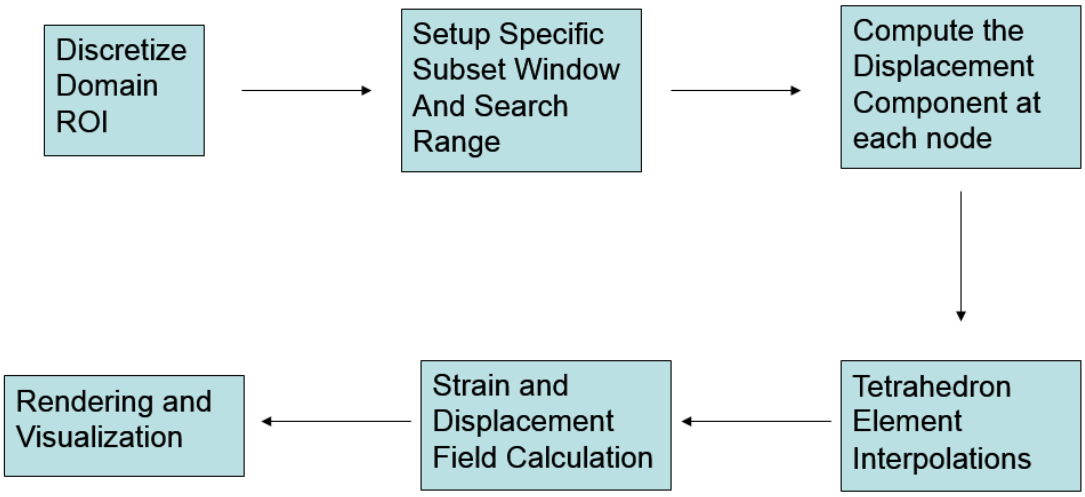


Figure 200 Flowchart for Advanced Digital Volume Correlation

The original block-matching algorithm is not very efficient way to find the displacement for each voxel. We realized that the algorithm is $O(N^4)$, which the computations grow as the fourth order of input size. If the volume is 512^3, the computation is about $(512^3)^4 = (512)^{12}$ times. If we use 3.2GHz quad core computer, it will take us about 44 hours to finish the computation of the whole volume. In the previous implementation of the DVC, the search could be done within the reduced region if we have an estimation about the maximum of displacement. Also, we only compute the voxels identified as the object, and we ignore the air part in the volume. Moreover, we have implemented the algorithm in a parallel computation way, which applies the Message Passing Interface (MPI) scheme using the Hopper supercomputer on campus. In this section, the new algorithm is basing on the meshwork, at each computational node the displacement vector will be found, and the displacement field is calculated as interpolation over the critical nodes.

During the meantime, the spatial computation can be transferred into complex domain. In the paper named “Fast Template Matching” (J.P.Lewis 1995) and “Template Matching Using Fast Normalized Cross Correlation.” (Uwe D. Hanebeck 2001), it transfers the computation from the

spatial domain into the complex domain, which says the multiplication in the frequency domain equals to the convolution in the spatial domain (Lewis, 1995). If we use the Fast Fourier Transform (FFT) for the transformation, we can reduce the computation from $O(N^4)$ to $N^2\log(N)$. However, the use of FFT-based cross-correlation should be careful because of the quality of the result. In the literature named "PIV: Direct Cross-Correlation compared with FFT-based Cross-Correlation" (Oliver Pust, 2000), it states the usage of FFT-CC should be cautioned in the Particle Images Velocimetry (PIV) where the objects are formed by the speckle particles. We have done some experiments to compare the efficient between the spatial DVC and FFT based-DVC, the nine regions are used for the graph. In the figure, we can see for the spatial DVC algorithm; we need about 6500 computations while there are only 180 computations if we compute over the complex domain. This improvement would put this implementation in a real application.

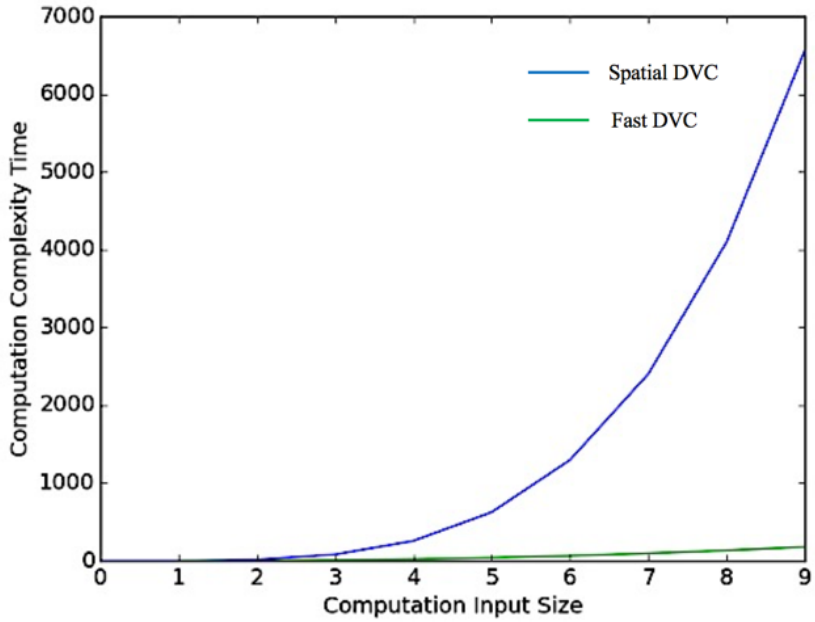


Figure 201 Spatial DVC and Fast DVC Computational Comparison

The 3D Poisson Disk Sampling algorithm has been used for the geometry discretization (Cavalcanti & Mello, 1970), and then the FEM-like mesh has been generated by those random

nodes using mesh generator (Cunha & Canann, 1997), from which the surface mesh could be constructed using the Alpha Shape Algorithm (Ying & Xin, 2011) (Cline & Jeschke, 2009) (Edelsbrunner, Kirchkpatrick, & Seidel, 1983). To smooth the mesh, it iteratively applies diffusive equation at each node until the convergence. This FEM-mesh work also gives us a connection with the Finite Element Method using the real geometry conversion from the X-ray CT images (Lall, Wei 2016), and the mesh can be input into the interface of FEM software such as ANSYS or ABAQUS. The following nodes cloud shows the two different density and discretization. In general, the denser of the cloud, the higher accuracy, but the solve time would increase. After computing the displacement for each node, the displacement field is calculated as interpolation over the closest neighbor nodes using interpolation function. The general interpolation function is shown below; the (x,y,z) is the coordinator at the point of interest. The basis function is $\emptyset_j$, and the C_j is degree of freedom. The domain mesh model is shown in the figure below.

$$U(x,y,z) = \sum_{=1}^{N} C_j \emptyset_j(x,y,z) \tag{166}$$

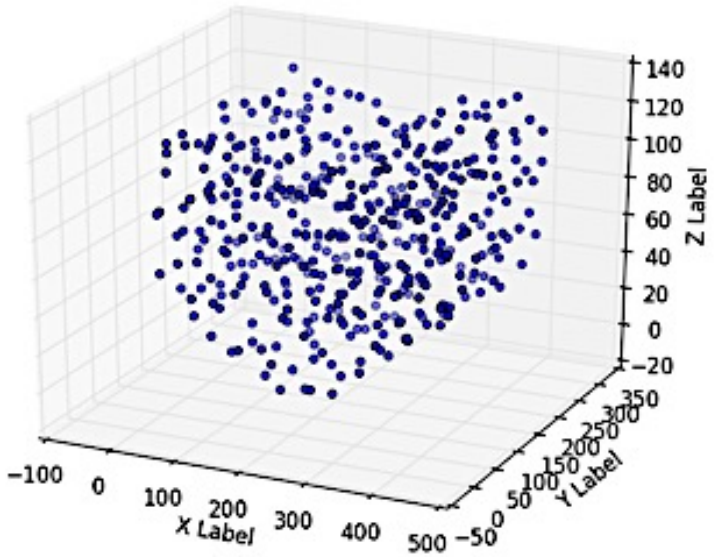


Figure 202 Sparse Random Samples Generation in the Object Domain

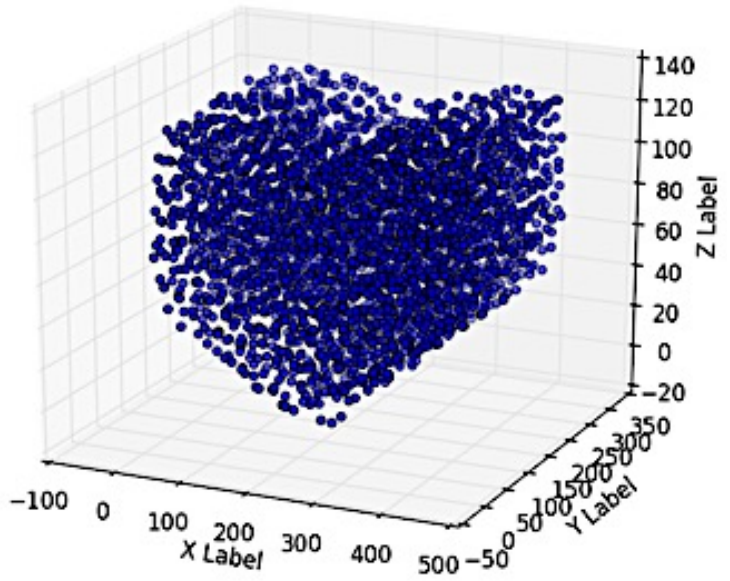


Figure 203 Dense Random Samples Generation in the Object Domain

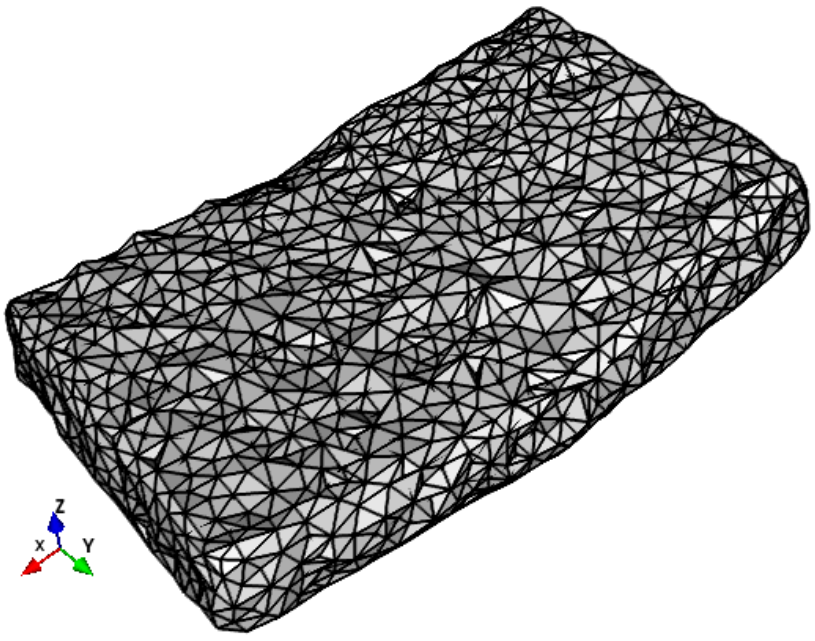


Figure 204 DVC Mesh Example

5.4.5 Soft Material and Flexible Electronics Deformation:

In the correlation, it assumes that the transformation of material follows the local translation and rotation. Firstly, the rough translation vector, a grid displacement, is found by the block-matching, which gives the approximation. Then, during the optimization process, the refined translation containing a fractional displacement is found, and the rotation matrix for the blocks is also calculated. The optimization between two blocks has been calculated by the points cloud samples using the Singular Value Decomposition(SVD). The convergence of optimization is decided by the Mean Square Error(MSE) of two points cloud samples. The 'R' is the three by three rotation matrix, and the 'U' is the three by one translation column vector. 'V' is the final location of a material index.

$$V' = RV + U = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (167)$$

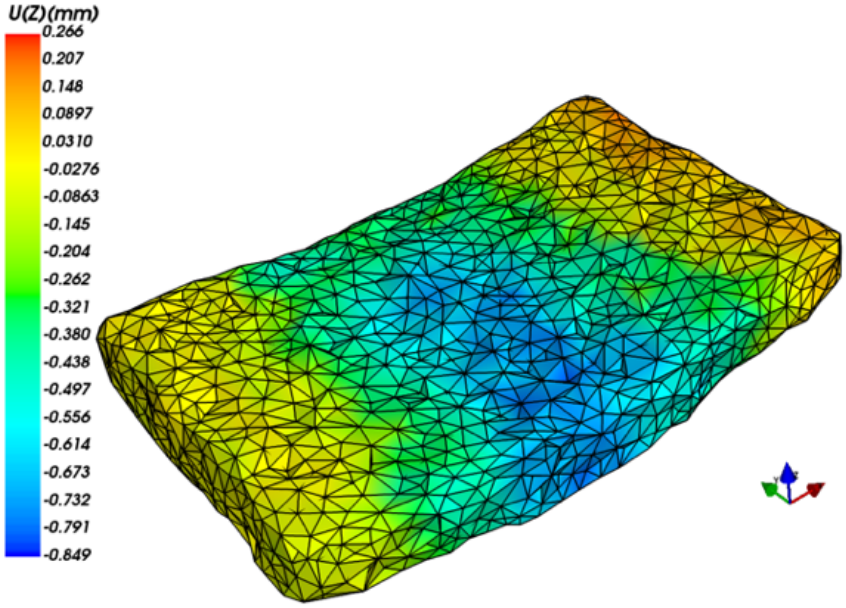


Figure 205 the Out-of-Plane Displacement in the Mesh Computational Model

The following view mode shows the material deformation contour, and the deformation is plotted onto the physical surface features. In this view mode, it provides an intuition about the deformation field with material geometry, which helps us to identify the target component in the deformation field.

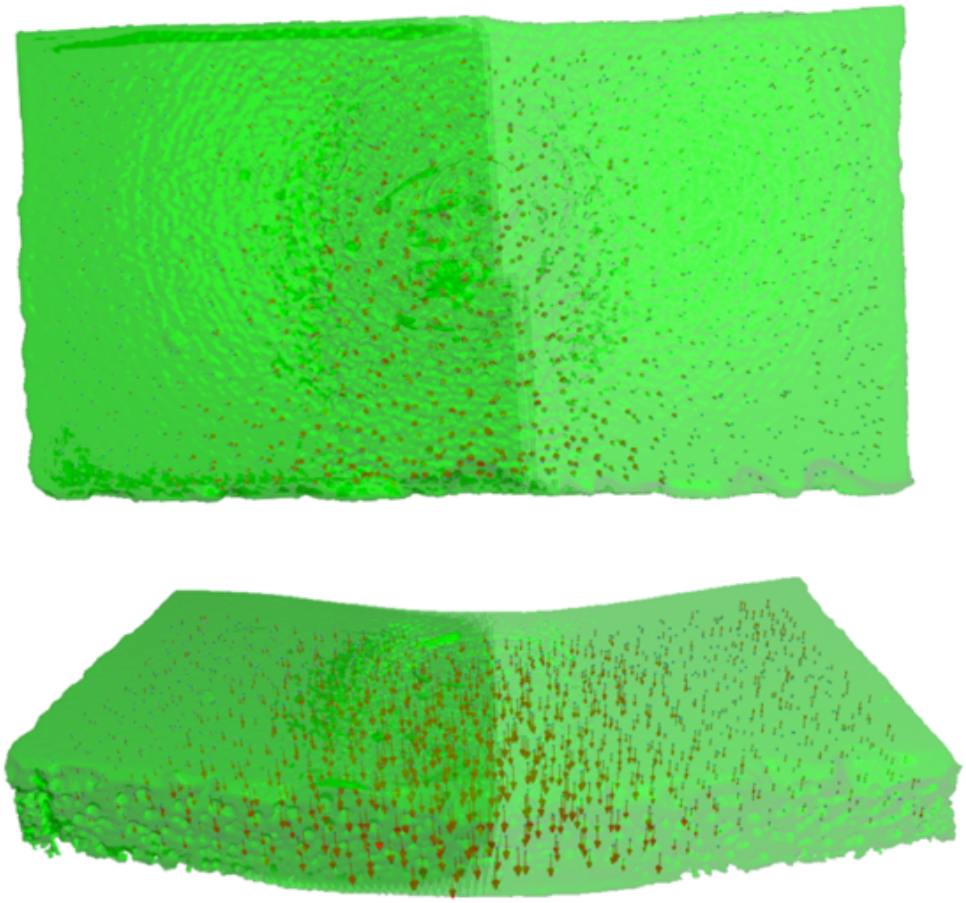


Figure 206 the Displacement Vectors in the Reference and Deformed Samples

The Lagrangian strain components can be computed from the displacement field, from which the displacement gradient in each direction is calculated, and the Figure 207 below shows the nine symmetric Lagrangian strain components.

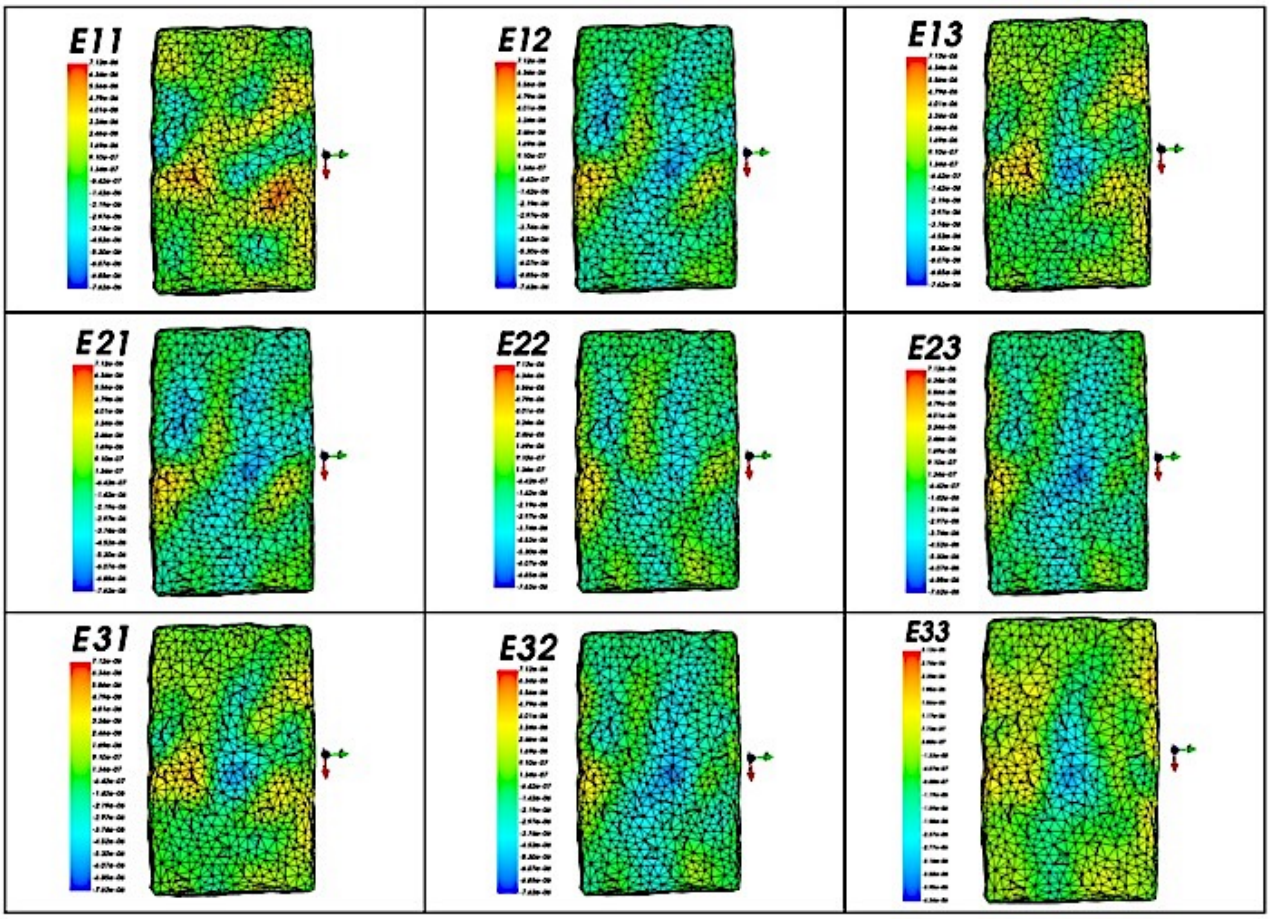


Figure 207 Strain Components

5.4.6 Flexible Electronics Band Deformation Measurement

In this study, we also demonstrate the way to compute the deformation in the flexible electronics with silicone soft material substrate. The hard material (i.e. electronic component) is attached and covered by the soft material, and in the CT scan, in the following Figure 208, the flexible electronics band contains the low density part and high density part.

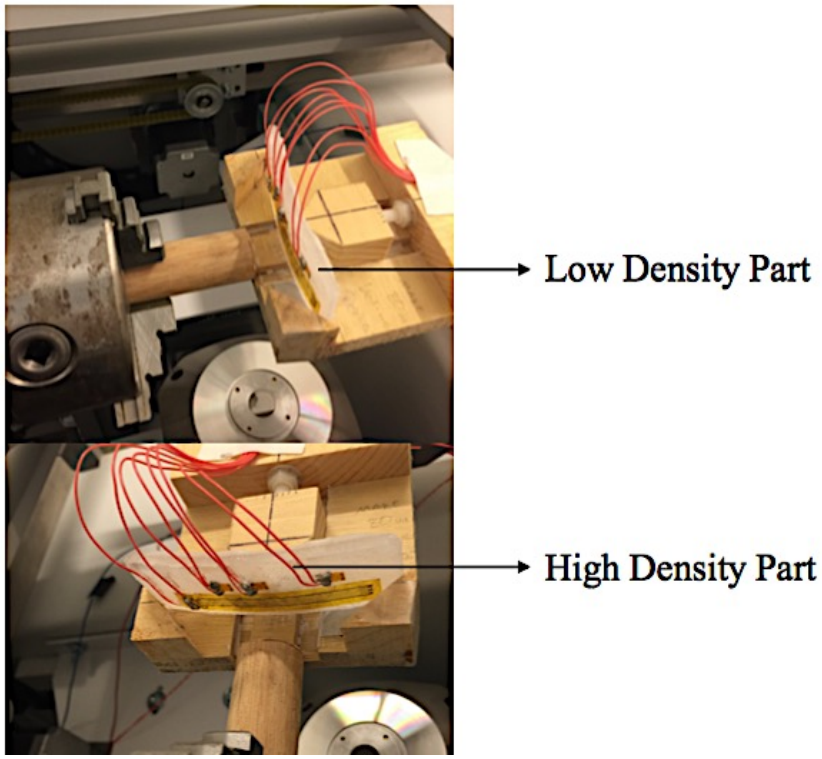


Figure 208 Flexible Electronics Band Scan

The following Figure 209 shows the historical deformations in the flexible electronics band mesh.

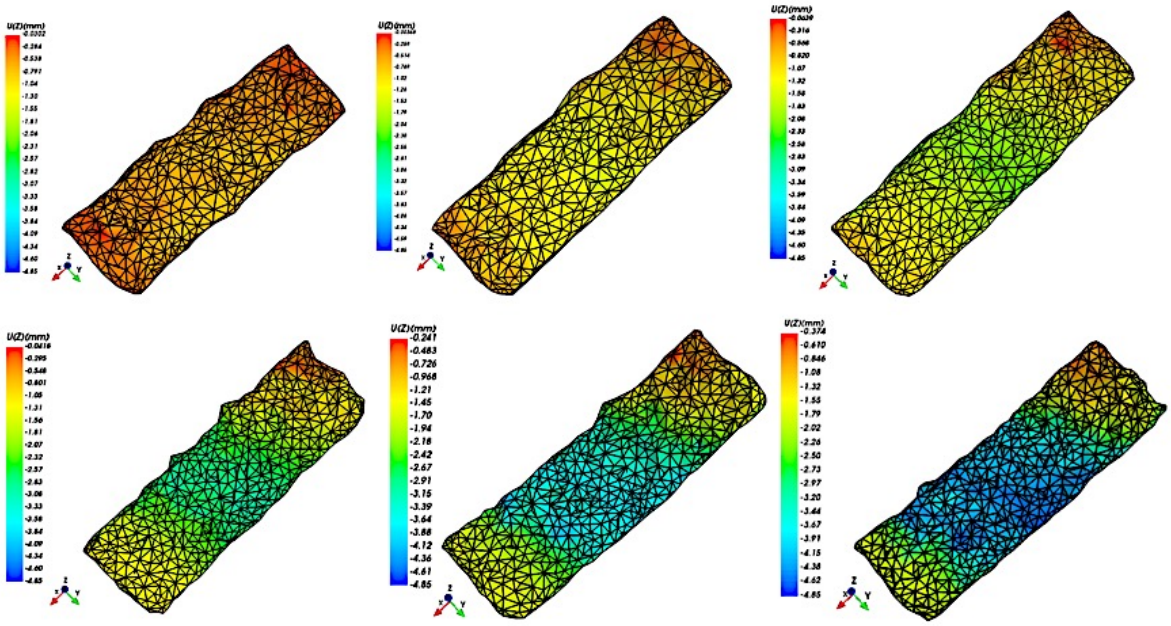


Figure 209 Flexible Electronics Band Deformation History with Increment Loads

The effective strain calculates the magnitude of all strain components, and we show the effective strain computed in the deformed band. After computation and visualization, we found that there is tremendous strain existing on the edge of the flexible band, which is shown in the red region. The strains on the two band ends, in the blue region, are negligible.

$$E_{effective} = \sqrt{\frac{2}{3} * (\varepsilon_{11}^2 + \varepsilon_{22}^2 + \varepsilon_{33}^2 + 2\varepsilon_{12}^2 + 2\varepsilon_{13}^2 + 2\varepsilon_{23}^2)} \quad (168)$$

ε_{ij}: strain components

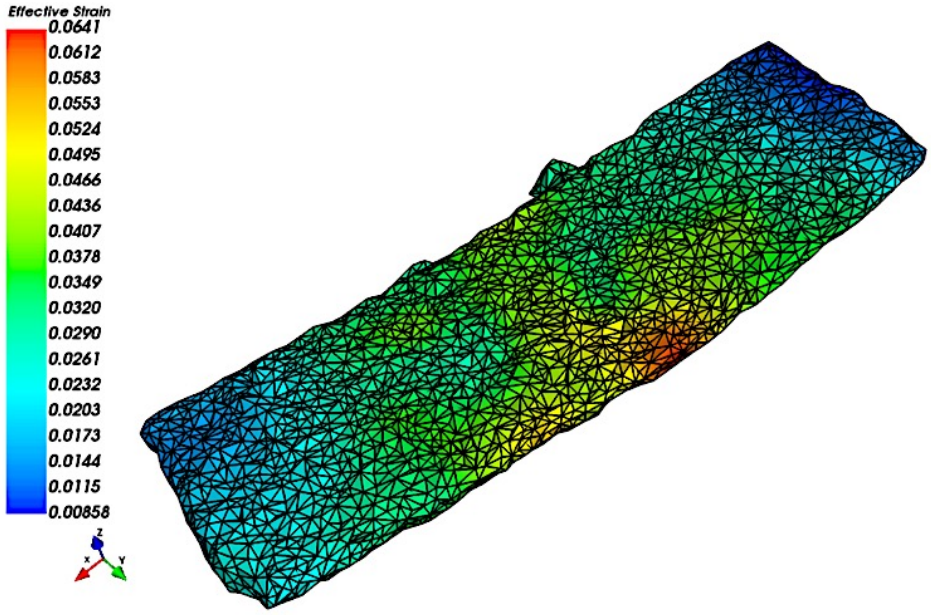


Figure 210 Effective Strain of Flex Electronic Band

In the flexible electronics, we have attached a tiny strain gage sensor that is 3mm length, and the strain is measured by its long wired electric trace, showing in the following figure. In the 3D reconstruction figure, we can see the Strain Gage trace region and its two solder pads are shown clearly in the picture. The voxel deformation in each direction, x, y and z-direction are shown.

From the picture, we can see the deformation on the center of strain gage is significant, and the deformation of strain gage is distinct and easily to be visualized.

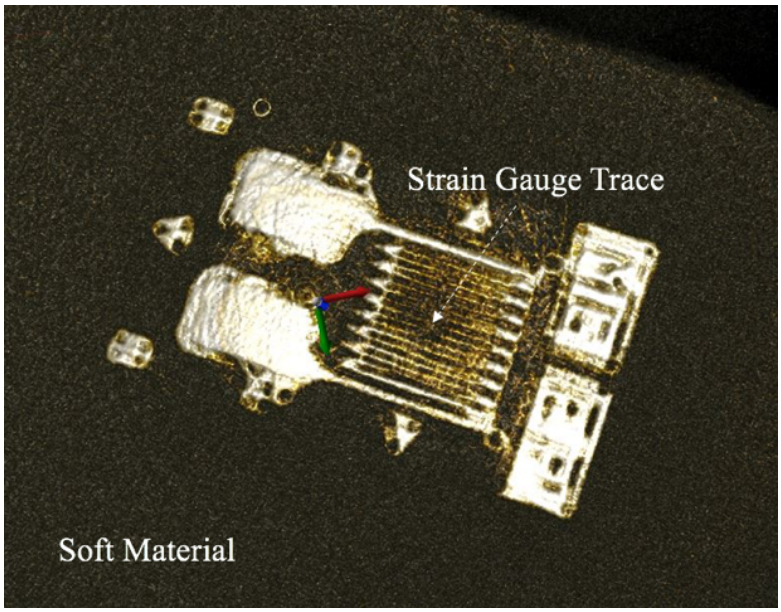


Figure 211 3D Reconstruction of 5mm Strain Gage

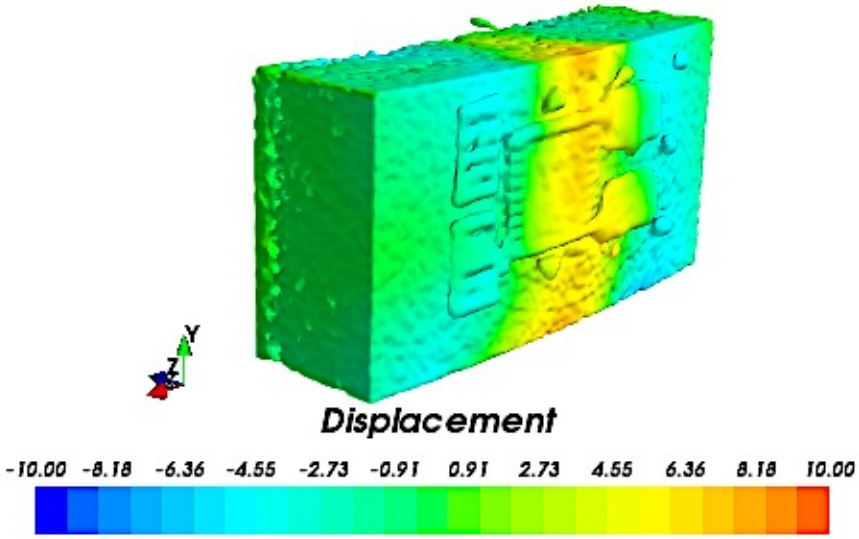


Figure 212 Strain Gage X Voxel Displacement Deformation

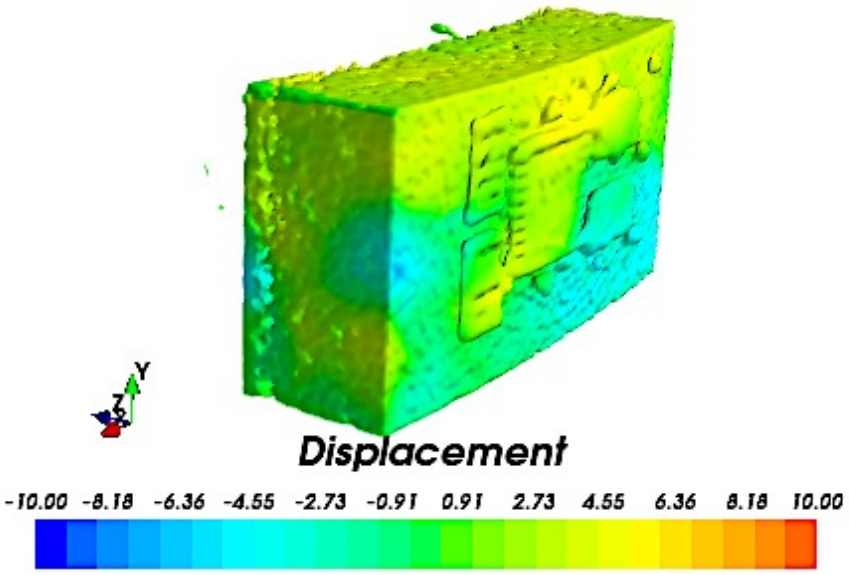


Figure 213 Strain Gage Y Voxel Displacement Deformation

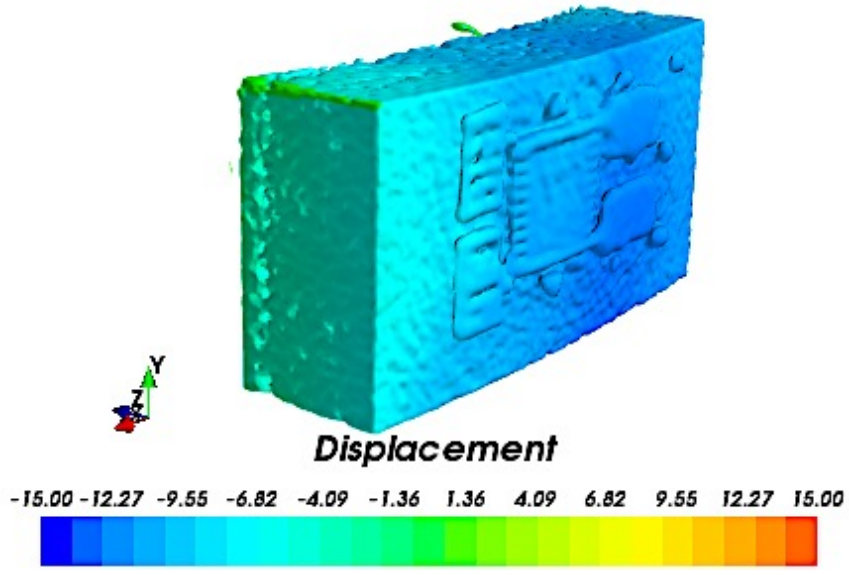


Figure 214 Strain Gage Z Voxel Displacement Deformation

5.4.7 Summary and Conclusions

In this chapter, we have demonstrated a technique to measure the deformation in the soft material and flexible electronics. First of all, we have a solution for imaging low-density material

using the X-ray based Micro-CT scan. The scan uses 75Kv and 10mA, which is low voltage and medium current in the specification. The experimental low-density sample was made from silicone packaging soft material, and it is transparent and highly stretchable. In the flexible electronics testing, a high precision strain gage has been attached to the soft material substrate and the electric connector wires have also been attached to the sample to demonstrate a metallized flexible electronics. Then, the deformation is calculated using the mesh based Digital Volume Correlation method, and the deformation has been verified by an analytical solution and strain gage measurement. In the validation, the computed displacement field is verified by the image registration algorithm, in which a reversed deformed image is compared with the reference image to show the closeness. The strain values from strain gages have been compared with the theoretical and experimental computations. The mismatch about 4% has been demonstrated. After that, we have performed several tests to measure the deformation of the flexible electronics sample. The displacement has been shown using the 3D surface contour onto the geometry, and the nine Lagrangian strain components have been computed from the gradients of displacement. Lastly, we show a high-resolution measurement in the strain gage senor using the DVC, and the displacement contour shows the deformation in the center of strain gage reaches the maximum. This work could be easily applied and adapted to any soft and low-density material based flexible electronics, and the in-situ deformation can be quantified with given loads to investigate the reliability of flexible electronics.

Chapter 6 Finite Element Simulation Technique for Prognostic Health Management

6.1 Finite Element Method with the CT Data

The Ball-grid array (BGA) package can fail under thermo-mechanical loading such as thermal cycling. It is important to study the strain and stress of an electronic package using Finite Element Analysis (FEA) that is basing on physical geometry and dimension. We can precisely model the real package dimension under the CT scan. Therefore, this study will not only help us improve the reliability of the BGA package, but it will provide us with a better design of a package. Currently, many people try to take geometrical measurements with calipers or micrometers, which only measure the surface and outline. However, the surface measurement is not precise or accurate because there is often an error in variance during each measurement. Also, some curvatures and irregular shapes cannot be directly measured, and assumptions and approximations often needed to model complex geometry. Therefore, it is important to apply a reconstructed geometry to the FEA analysis. Currently, there are many medical applications using CT scan reconstruction. Previously, people used a medical CT-based femur model to find stress concentration in the femur under mechanical stress (Vulovic, Korunovic, & Trajanovic, 2011), and D.Jovanovic (D.Jovanovic & LJ. Jovanovic, 2010) has used the CT-Scan data to model the vertebra geometry and its properties in the FEA. Also, some have scanned foot and used it into the FEA (Antunes & Dias, 2008), (Muller, R1, Ruegsegger, P 1995), (Silva MJ1, Keaveny, TM, 1998). The model transformation from CT to FEA can be done with an STL conversion, which has been mentioned earlier by Liu, (Liu, Zhang, & Ju). In the electronic packaging field, it is worthwhile to introduce the finite element model based on a real geometry rather than the simplified geometry. Also, it is

necessary for engineers who work in the electronics package area, and it will benefit researchers to study the package. In this chapter, a generalized scheme of FEA using the X-ray micro-CT has been put forward. Firstly, a procedure for a micro-CT scan and a technique for volume reconstruction have been developed. Secondly, the CT data structure has been demonstrated, and the FEA global and local model from the scan has been incorporated into the analysis. Thirdly, the Annand Visco-plasticity model of SAC-305 has been utilized and merged into the FEA software to model material behavior. Lastly, a model of voided solder joint under thermal cycling has been presented. The cumulative plastic work and stress-strain hysteresis loop have been plotted.

The Finite Element Analysis can be performed using physical geometry from the micro-CT scan and 3D reconstruction. The scan geometry is represented as ISO-Surface in grayscale volume data, in which each voxel in the volume image has its value between 0 and 255. The volumetric image has different regions of gray scale value including the distinct density sections such as PCB, Solders, Die and Molding Compound. The appropriate image segmentation techniques will be required to acquire the different sections (Felzenszwalb, 2004) (Schwemmer & Prummer, 2010). Often, it is convenient for us to transform a grayscale volume image into a binary image with sub-volumes, which followed by morphological operations such as dilation and erosion to smooth the image boundary. Then, we can directly convert the binary volume into both the surface mesh and body mesh with tetrahedrons. Lastly, we can import a genuine meshed 3D model into the ABAQUS to perform the Finite Element Analysis. One of the advantages using a tetrahedron based mesh model is that it can adapt complicated geometry and shape to irregular surface and curvature. This X-ray CT-based FEA can also be applied to multi-physics and computational fluid dynamics analysis. Previously, the Finite Element Analysis heavily depends on the CAD file geometry, which has a simplified boundary and ideal shape. The CAD file

modeling also requires the geometrical measurement to be precise and accurate. However, this cannot always be true if we have various measurements. Also, the inner features of an object cannot be measured directly. Moreover, the X-ray CT modeling procedure could be fully automatic and much faster than the manual CAD creation. Therefore, it is necessary to invent and utilize a new method to run the Finite Element Analysis with an accurate geometry information. From this X-ray micro-CT 3D reconstruction, both out and inner features can be acquired directly and modeled accurately. Nevertheless, to produce the X-ray micro-CT scan and 3D reconstruction, it demands much more precise machine work and expensive equipment, and it requires image processing and excessive CPU/GPU computation for processing 3D volume. However, those problems can be relieved and solved by the development of modern super-computer and fast growing parallel computational algorithms.

In the past, there have been many X-ray CT scan medical applications. In hospitals, diseases can be diagnosed using the X-ray CT scan image. Because the X-ray has attenuations when passing through a body, and the high-density tissue will absorb much more beam energy than the lower density tissue so that the projections could be distinct at each degree of rotations. From those projections, the volume reconstruction could be done by a reconstruction algorithm. (Antunes & Dias, 2008), Antunes and Dias converted an ISO-layer of volumetric data into a CAD model, and then converted it into a 3D tetrahedron FEA model. Also, (D.Jovanovic & LJ. Jovanovic, 2010), D.Jovanovic and LJ.Jovanovic applied to image segmentation technique to volumetric data and modeled different Region of Interest (ROI) with distinct densities. This image segmentation technique is made under the assumption that the grayscale value distributions, representing the ROI histogram, are varied, and its target is to resolve the hard tissue (i.e. vertebra bone) from the surrounding soft tissue (i.e.skin). Particularly, the author makes an X-ray

calibration by using a substitute material, which is similar to the vertebra. Moreover, in the latest paper (Vulovic, Korunovic, & Trajanovic, 2011), the author firstly creates a polygonal model, and after image improvement and modification, this polygonal model can also be translated into a CAD model. Then, the Finite Element model will be generated from the CAD mode.

In this chapter, a direct and straight meshing technique, converting the ISO-surface into a tetrahedron model, has been developed for the FEA. The Region of Interest (ROI) can be determined from ISO-values. Secondly, we can convert the gray scale ROI volume into a binary ROI volume, and in the binary image, the jagged and discontinued surface can be smoothed and stitched using image morphological operations. Lastly, the tetrahedron creator can generate a mesh from the MATLAB connecting to the ABAQUS. In this application, the Volume Graphics software has been used to reconstruct 3D volume, MATLAB and CGAL library has been used to generate a mesh. The finite element software ABAQUS has been implemented for thermo-mechanical analysis on a BGA324 package to prove the concepts and show the feasibility in an electronics application.

6.2 Volume Processing

In the X-ray CT scan, the detector has a high resolution on a particular window region, which is Region of Interest (ROI). In the ROI, the reconstruction model is based on the vertical layer construction. The different components will be separated using the distinct ISO values. At each layer, a smoothing filter will be applied to the sub-volume to smooth the boundary.

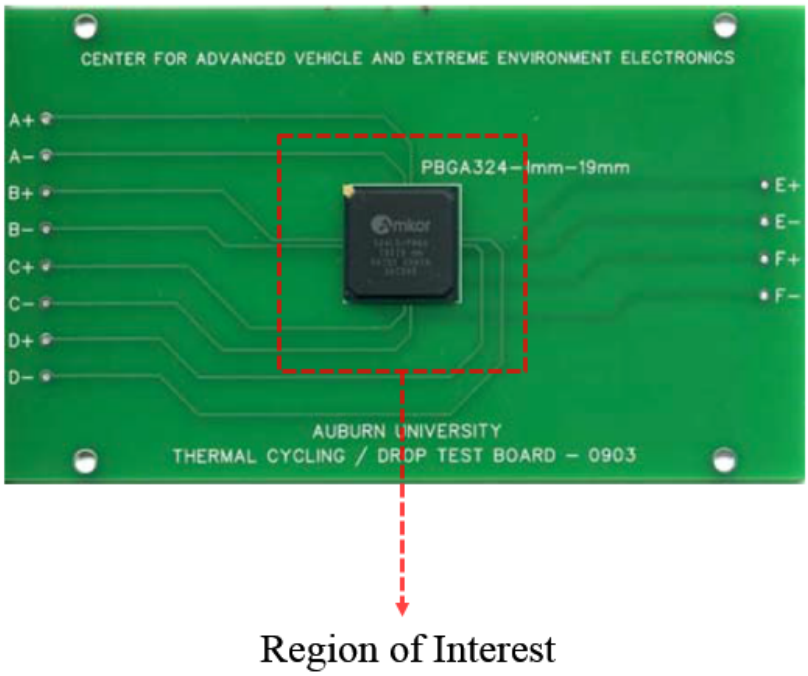


Figure 215 the Region of Interest

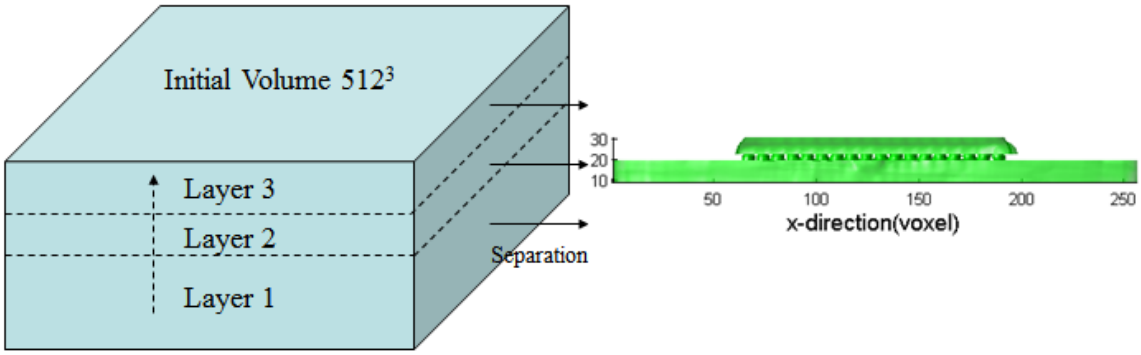


Figure 216 Region of Interest for BGA324 Modeling

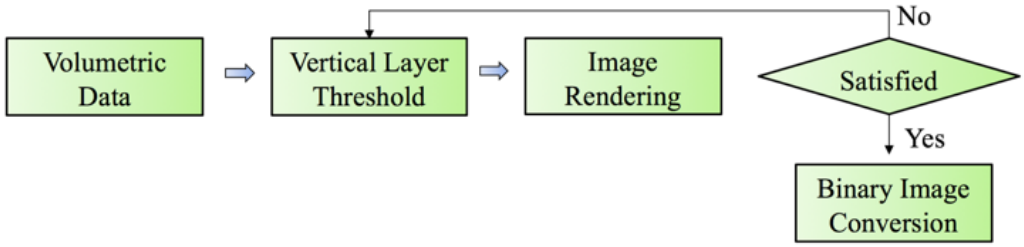


Figure 217 Volumetric Data Processing Flow Chart

6.3 Implicit Geometry Mesh Generation

In this section, the package level mesh has been generated, which uses a 3D tetrahedron mesh generator from Computational Geometry Algorithm Library (CGAL), and it is open-source C/C++ library (Dey & Levine, 2007). It can transfer a binary volume or gray scale ISO-surface into a solid tetrahedral body mesh (Desbrun & Meyer, 1999). Therefore, we can directly create a mesh model from the micro-CT scan. To find an outline of the package, the gray scale image will be transformed into a binary image using appropriate gray scale value threshold. In a vertical layer direction, we can identify different thresholds in the BGA 324 such as BGA substrate, the solder joint, and the molding compound. The following Figure 218 shows the 3D reconstruction image of BGA324, whose dimension is about 250×250×25 Voxels.

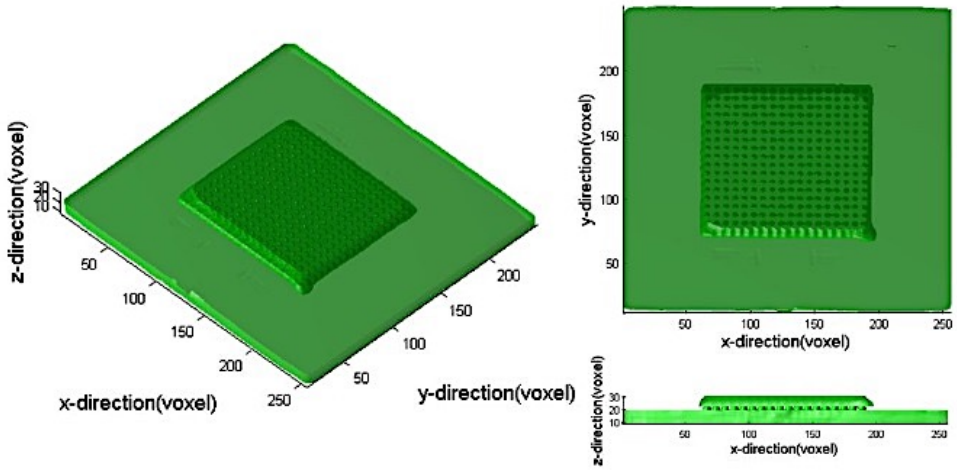


Figure 218 BGA 324 CT Scan Reconstruction 3D Rendering

In the X-ray micro-CT scan, there are several scan specifications needed to be determined. In this section, the reconstruction parameters for the BGA 324 have been listed in the following Table 13:

Specifications	Value
Scan Projections	1440 Projects
Reconstructed Volume Size	10243 Voxels
Rendering Size	2563 Voxels
Voxel Unit Length (X-Y-Z)	(0.140,0.155,0.156) mm
Scan Type	Quality Scan
Source-Object Distance	114.136mm
Object-Detector Distance	299.833mm
Result Physical Size(X-Y-Z)	35.1553
Volumetric Data Type	Unsigned 16 bit

Table 13 CT Scan Specification

As it mentioned earlier, the 3D reconstruction has been performed by Volume Graphics®, which works on a raw 'TIFF' image data. Before outputting to a DICOM file, the object registration has been used to find the appropriate and aligned X-Y-Z axis of the BGA324 package. Because a large size volume, approximately one to 2GB, the data will be compressed to get a shrink volume, which can be loaded in MATLAB.

In MATLAB interface, the grayscale volumetric data has been first transformed into a binary data with a sub-volume, and then a morphological image operation will be used. To decide a threshold of transformation, 3D image segmentation tool has been used. In the literature, it uses the Otsu's method to find a gray scale threshold that maximizes the variance between the groups. Then, the best threshold will be decided by trials to find an optimal value around the mean. After calculation and trial, we set up ISO-values to identify the different components in an image of electronics. For instance, the value for a substrate is chosen as 100, the value for a molding compound is chosen as 1500, and the value for a solder joint is 27000. Therefore, using those values, the outline of BGA324 can be determined and captured in Figure 222.

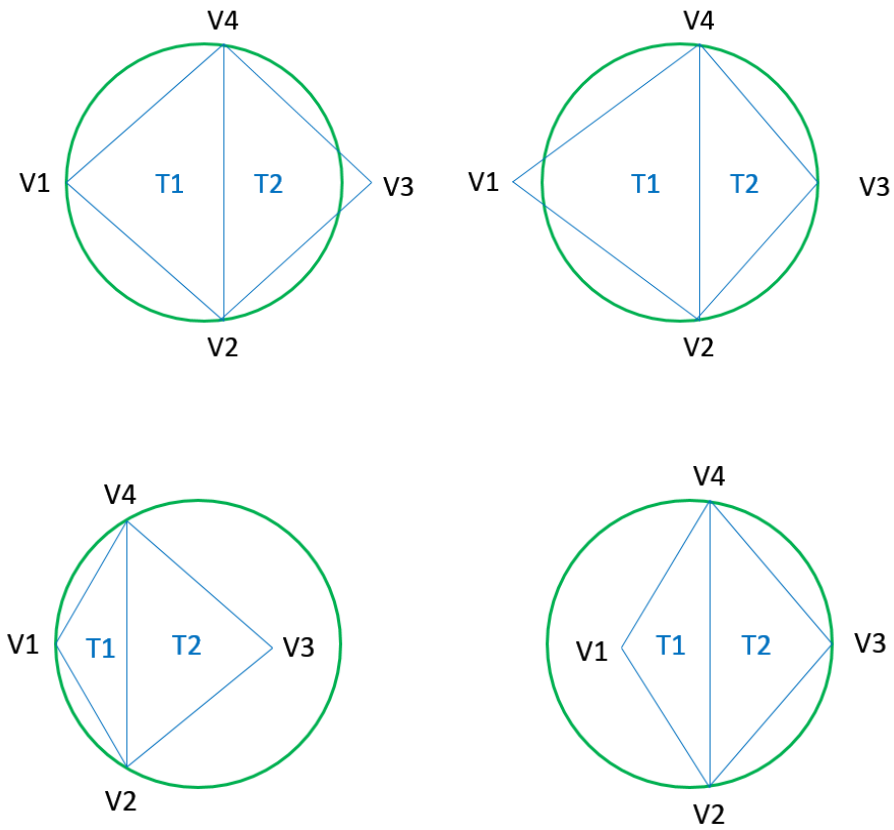


Figure 219 Delaunay Triangulation ‘Empty Circles’ Property

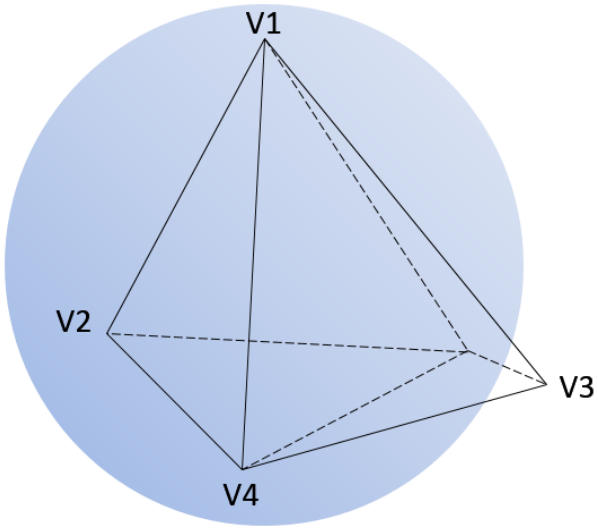


Figure 220 3D Delaunay Triangulation Tetrahedron

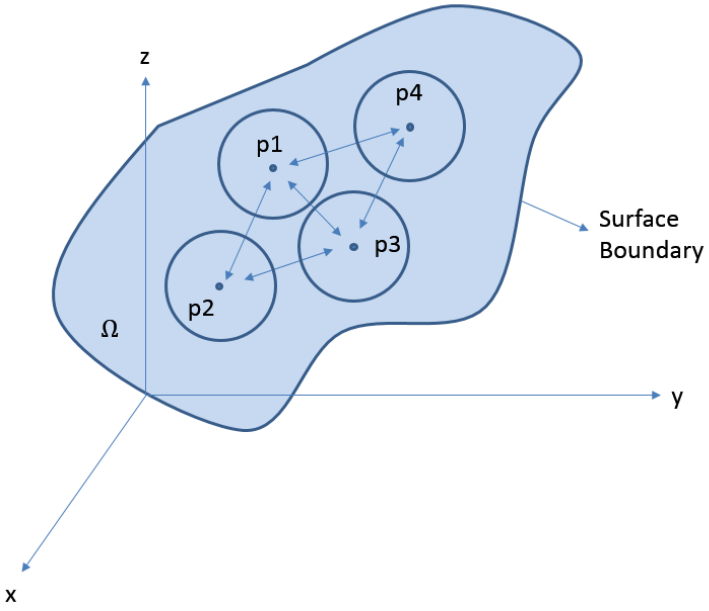


Figure 221 Geometrical Distance and Object Boundary

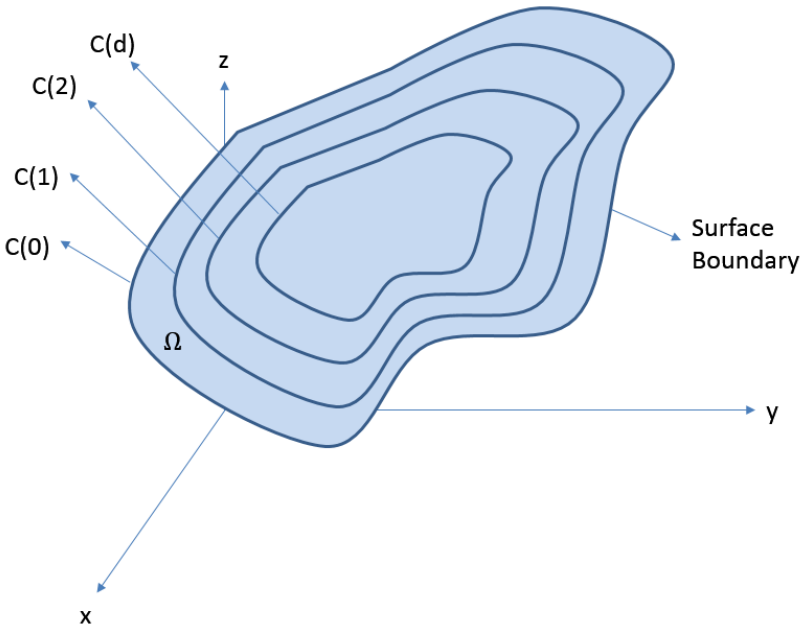


Figure 222 Isosurfaces and Object Boundaries

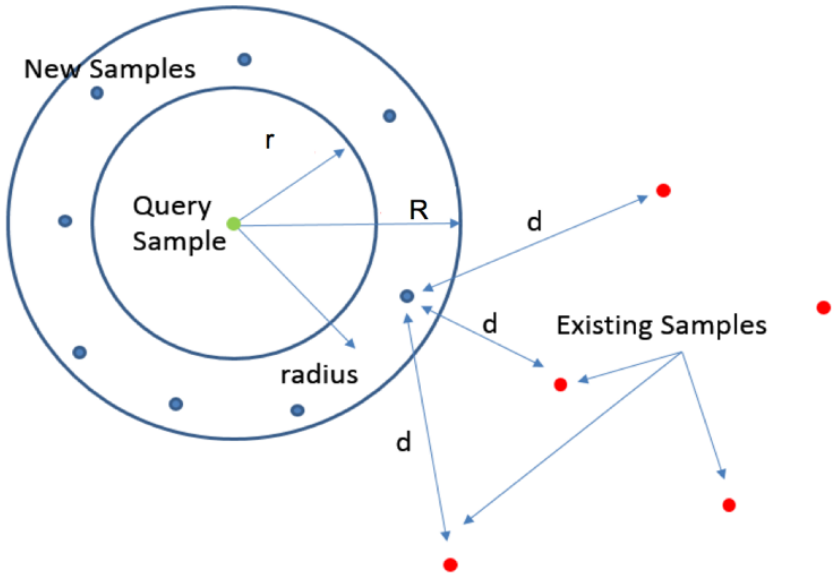


Figure 223 Poisson Disk Sampling in the Boundary

6.4 Global Meshed Model

Because the package's geometrical model is complicated, so the element must be small enough to adapt to the geometry. However, great number elements in a model could increase the time running on the FEA software. Therefore, to keep both efficiency and accuracy, we developed a global and local model to perform the analysis. The global model has a very tiny element size on its surface, which could fit the curvatures and the angles. Also, in the middle of the model, the element size will increase as the distance from the surface increases. The input geometrical model will be meshed using tetrahedrons, which have four vertices, four faces. The global mesh has 138398 nodes and 409269 tetrahedrons. The global mesh has been shown below Figure 224, and we can see that there is a dense mesh on its surface.

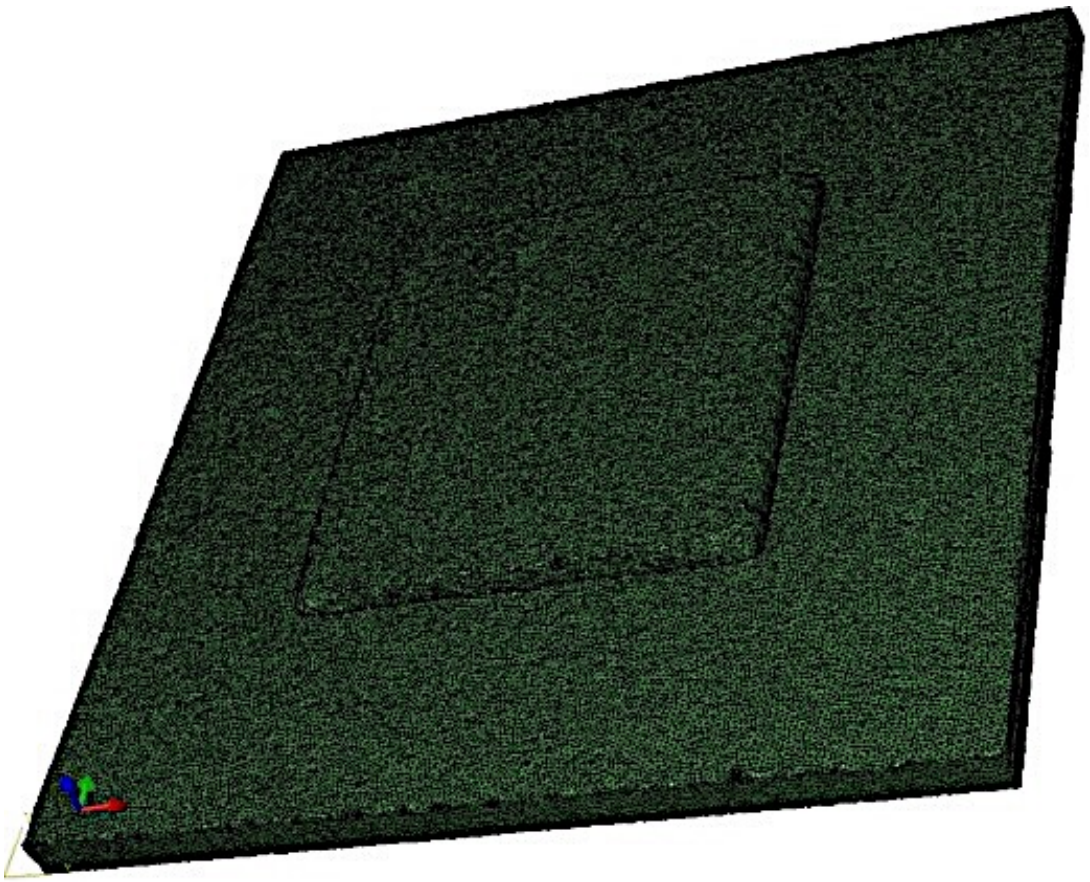


Figure 224 BGA 324 Global Mesh Model

6.5 Local Mesh Model

The local mesh model has been generated basing on a regional geometrical model that reduced from its global geometrical model, and it has much more detail. The local mesh model must be compatible with the global model, which its edge nodes have the boundary conditions imported from the global model. In the local mesh model, the element size can be as small as one-tenth of the global meshed model. The regional geometrical model is chosen from a region under the die shadow, which apparently suffers a higher stress than other regions. The local mesh mode has dimensions that are about 1.1mm by 1.1 mm by 3.5 mm (i.e. length, width, and height). The

geometrical relationship between the global and the local model has been shown in below Figure 225, and they are consistent.

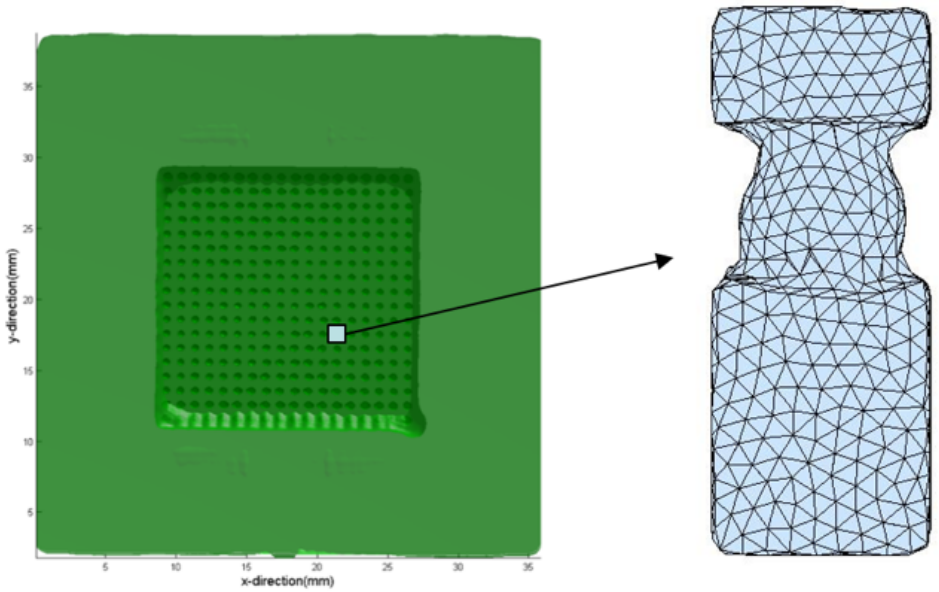


Figure 225 Die Shadow Region Model

In following Figure 226, we can see that there is an original void, a natural and manufactured defect, in the solder. The crack is growing fast and penetrating the void, and it will finally separate the whole solder neck region. In this section, the local mesh model has 1283665 elements and 220924 nodes, which is three times of the total number of elements in the global model.

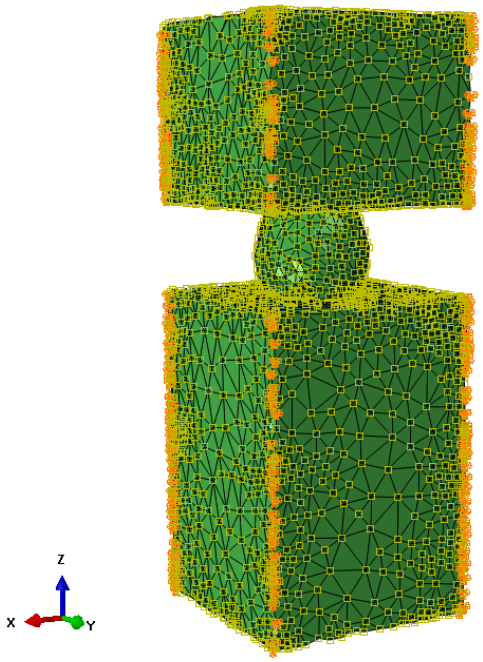


Figure 226 Local Model

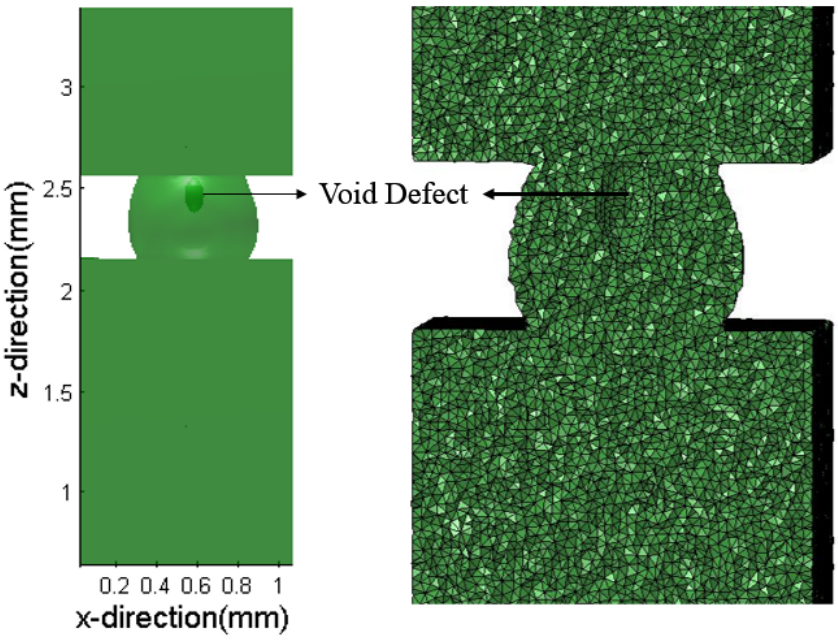


Figure 227 Region Representation

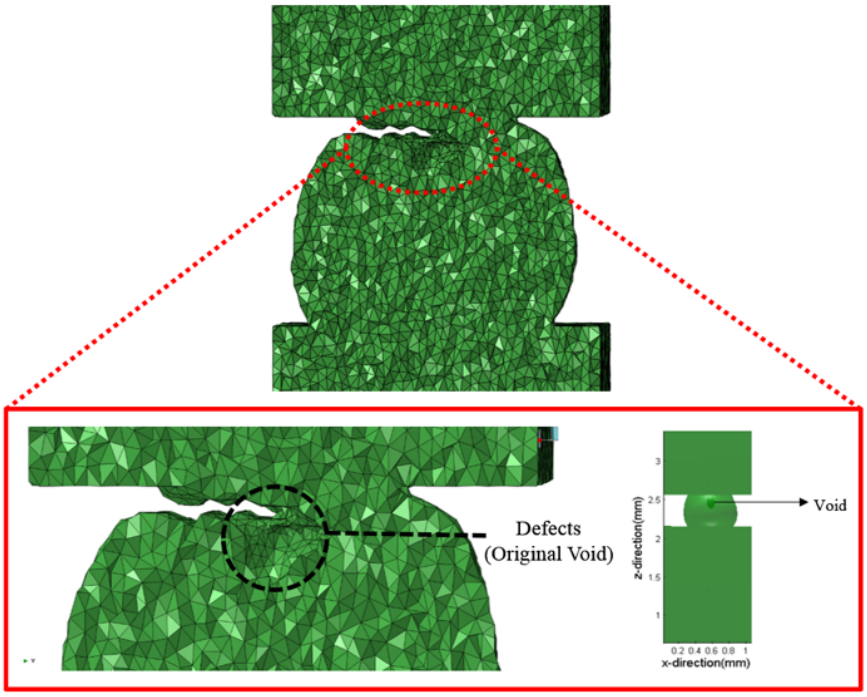


Figure 228 Solder Crack Discretization

6.6 The Global-Local Mesh Node Compatibility

In the global model, each node has an output. Typically, the displacement at each direction x, y and z have been obtained and written into a '.txt' file. The file that will be loaded into a local model. The local model has been re-meshed and all the nodes have been regenerated, the global node and local node are not identical. Therefore, we used an interpolation method to find corresponding displacement values that will be written as boundary conditions in the local model. The following Figure 229 shows the global and local nodes' compatibility. The blue nodes are from the global model, and the green nodes are from the local model. Under a homogenous coordinator, the 3D interpolation method creates interpolation boundary condition values, which could be written into local CAE model.

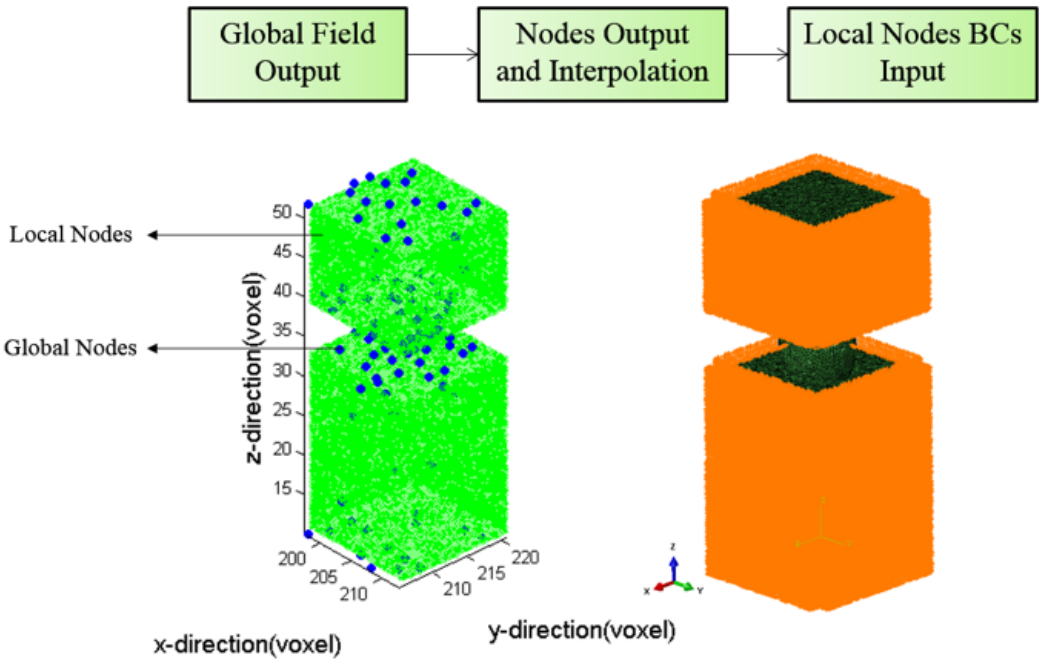


Figure 229 Global and Local Nodes BCs Connectivity

6.7 Model Finite Element Analysis

In the Finite Element Analysis, both the global model and local model share the same section properties and material definitions. The material properties used to perform Finite Element Analysis has been shown tables Table 14 and Table 15 below:

Name	Property		
PCB	Ex=25000MPa	Ey=7436MPa	Ez=20000MPa
	Exy=3324.6MPa	Eyz=3324.6MPa	Exz=7623.4MPa
	μxy=0.39	μyz=0.39	μxz=0.11
Solder Mask	E=3100MPa		μ=0.30
Cu Pad	E=128.93GPa		μ=0.34
Solder Ball	E=54GPa		μ=0.35
BT Substrate	Ex=17890MPa	Ey=7846MPa	Ez=17890MPa
	Exy=2822MPa	Eyz=2822MPa	Exz=8061MPa
	μxy=0.39	μyz=0.39	μxz=0.11
Adhesive	E=6769MPa		μ=0.35
Die	E=162700MPa		μ=0.28
Mold	E=23520MPa		μ=0.30

Table 14 Elastic Modulus and Poisson Ratio [Lall,P, DR,Panchagade, 2006]

Name	**Linear Coefficient (ppm/K)**		
	CTEX	**CTEY**	**CTEZ**
PCB	14.5	67	14.5
Solder Mask	30		
Cu Pad	16		
Solder Ball	20.8 (SAC305)		
BT Substrate	12	57	12
Adhesive	52		
Die	2.5		
Mold	15		

Table 15 CTE Property Table

In the Finite Element Analysis, the BGA324 is subjected to a thermo-mechanical load. It has been applied to two thermal cycling loops. The loop is from 25ºC to 125ºC, and the mean temperature 75ºC is chosen as a reference temperature. In the ABAQUS, the relative strain generation is based on the following equation:

$$\varepsilon = \alpha \cdot (\theta - \theta_{ref}) - \alpha \cdot (\theta^{I} - \theta_{ref}) \quad (169)$$

θ^{I}: Initial Temperature

θ_{ref} : Reference Temperature

α: CTE Coefficient

If we choose $\theta^{I} = \theta_{ref}$, we can calculate the strain as $\varepsilon_{relative} = \alpha \cdot (\theta - \theta_{ref})$

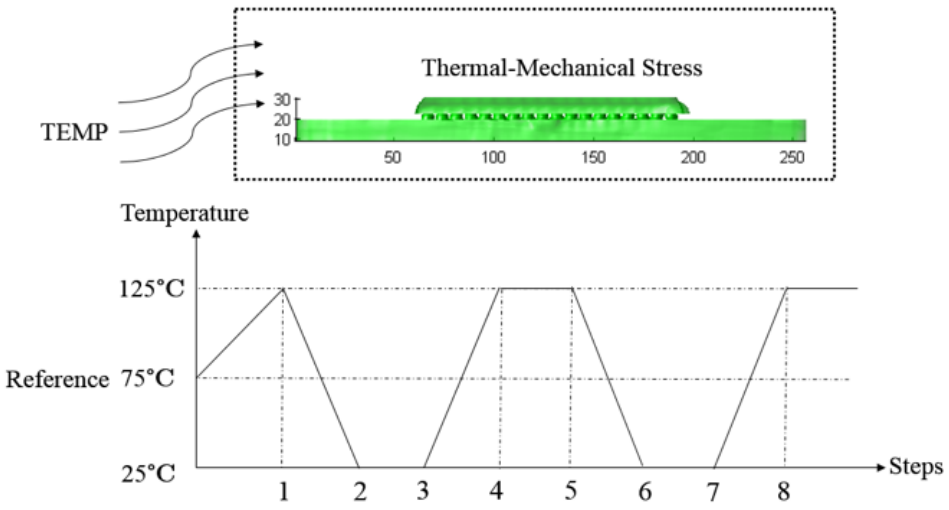


Figure 230 FEA Boundary Condition

The Figure 231 below shows the BGA324 global model with different property sections, and for the reason of modeling, four major sections will be used: (1) Solder Joint (2) Die (3) Molding Compound (4) PCB substrate. In the BGA324 local model, there are more sections to be used and assigned including the Solder Joint, the Die, the Molding Compound, the PCB substrate, the Adhesive, the BT substrate, and the Copper Pad. The section segmentations have been shown below in the Figure 232.

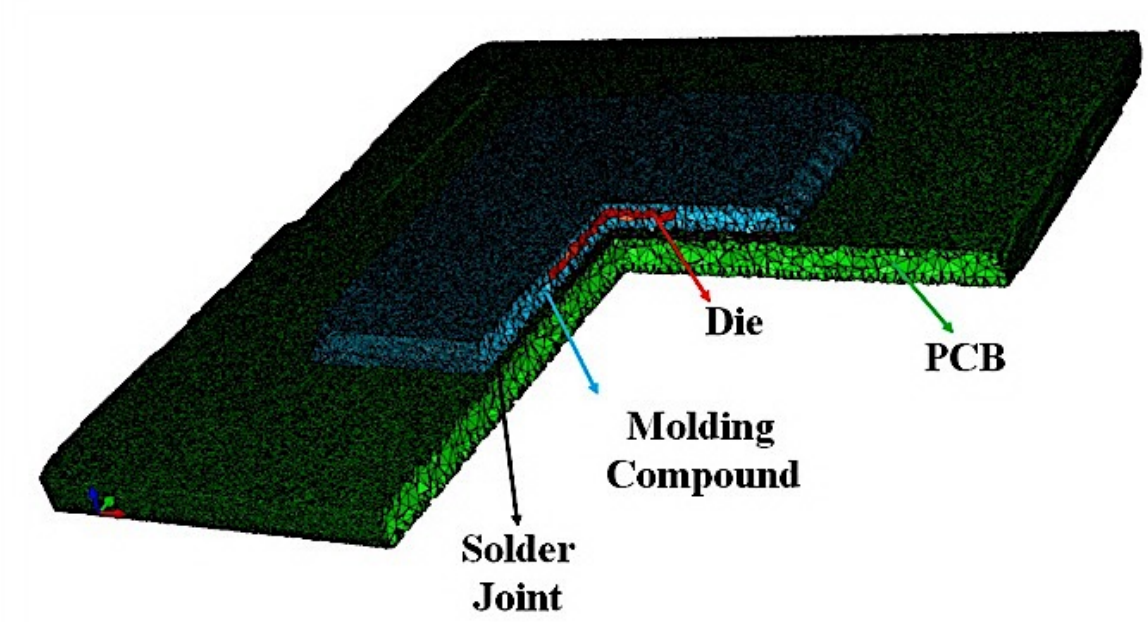


Figure 231 Global Model with Different Sections

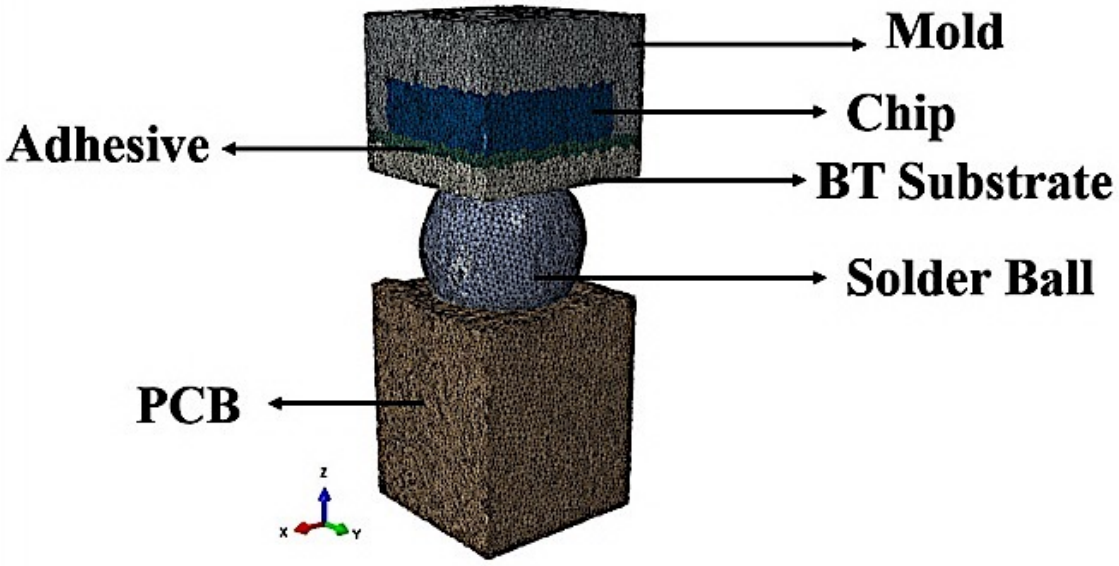


Figure 232 3D Local Mesh Model Section

6.8 Material ANNAND Model in the UMAT

In an advance Finite Element Analysis procedure, first of all, it uses a real 3D geometrical mesh model; secondly, it also utilizes users' material definition. In this section, we incorporate an Annand Visco-Plasticity model for a solder material SAC305. However, in the ABAQUS SIMULIA, there is not a direct built-in library of the Annand Visco-Plasticity model. Therefore, we define a material behavior using the UMAT, which is a Fortran77 based language library for the User Defined Material. The flowchart below illustrates a general procedure for this finite element method in the ABAQUS SIMULIA. The mesh model has been imported as an Orphan mesh that does not associate with a geometrical function. The Orphan mesh is imported from an '.INP' file with a node coordinator and element number information. Then, the sections with material definition will be assigned in the ABAQUS CAE. The ABAQUS standard implicit solver will be used, and the solution will be output and written in an '.ODB' file.

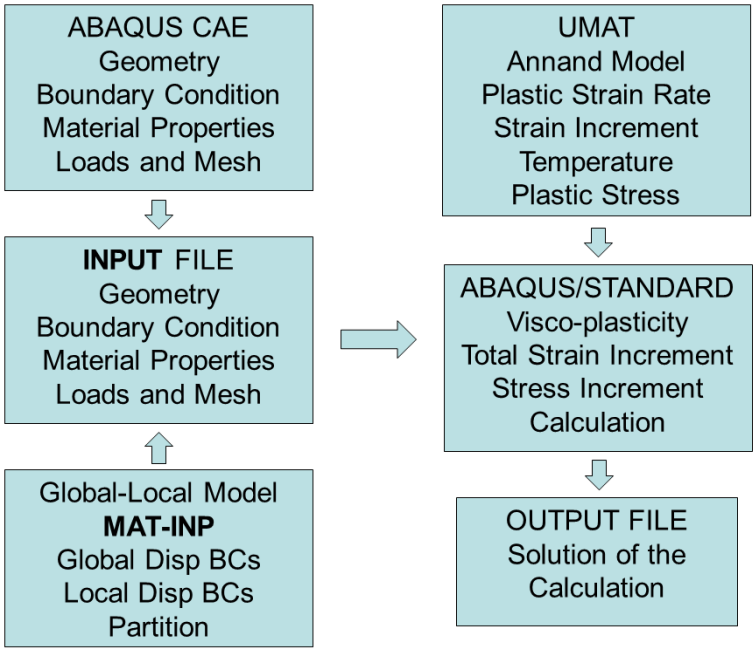


Figure 233 the advance FEM flow chart

In this chapter, the visco-plasticity constitutive model implements the Annand model, which assumes the visco-plasticity stress is a function of strain rate and temperature. The Annand model has nine constants that can be decided by curve fitting from tensile testing. The summarized Annand equation shows below:

$$\sigma = \frac{1}{\xi}\sinh^{-1}\left\{\left[\frac{\varepsilon_{pv}}{A}\exp\left(\frac{Q}{RT}\right)\right]^{m}\right\} \left\{\hat{s}\left[\frac{\varepsilon_{pv}}{A}\exp\left(\frac{Q}{RT}\right)\right]^{n} - \left[\left(\hat{s}\left[\frac{\varepsilon_{pv}}{A}\exp\left(\frac{Q}{RT}\right)\right] - s_0\right)^{1-a} + (a-1)\left[h_0\left(\hat{s}\left[\frac{\varepsilon_{pv}}{A}\exp\left(\frac{Q}{RT}\right)\right]^{n}\right)\right]^{-a}\varepsilon_p\right]^{1/1-a}\right\} \quad (170)$$

where ε_{pv} is Inelastic Strain Rate; T is temperature

The Annand Model Parameters has been chosen from a low strain rate constants generated by an experiment.

Constant Number	Annand Parameters	Units	SAC 305 Constant
1	S_0	MPa	21
2	Q/R	1/k	9320
3	A	Sec-1	3501
4	ξ	Dimensionless	4
5	m	Dimensionless	0.25
6	h_0	MPa	180000
7	ŝ	MPa	30.2
8	n	Dimensionless	0.01
9	a	Dimensionless	1.78

Table 16 Annad Low Strain Rate Parameters [Motlab, M., Cai,Z.2012]

Many metallic materials follow a Von Mises yield criterion, which interprets material yielding as a pure shear of deformation process that occurs when the effective shear stress reaches a critical value. The Von Mises criterion defines a yield function to be a form of:

$$f = \sigma_e - \sigma_Y \tag{171}$$

In the Annand model, it predicts for the yield stress (σ_Y) and the Ultimate Tensile Strength (i.e. UTS and Saturation Stress) depending on an effective plastic strain (ε_{pv}). The yield stress is predicted by the limit of ε_{pv} goes to 0:

$$\sigma_Y = \sigma|_{\varepsilon_{pv\to 0}} = cs_0 = \frac{1}{\xi}\sinh^{-1}\left\{\left[\frac{\dot{\varepsilon_{pv}}}{A}e^{\left(\frac{Q}{RT}\right)}\right]^m\right\}s_0 = \sigma_0 \tag{172}$$

The UTS is given by the limit as ε_{pv} goes to∞:

$$\sigma_{UTS} = \sigma|_{\varepsilon_{pv\to\infty}} = \frac{\hat{s}}{\xi}[\frac{\dot{\varepsilon_{pv}}}{A}e^{\left(\frac{Q}{RT}\right)}]^n\sinh^{-1}\left\{\left[\frac{\dot{\varepsilon_{pv}}}{A}e^{\left(\frac{Q}{RT}\right)}\right]^m\right\} = \sigma^* \tag{173}$$

The effective stress σ_e is defined in terms of principal stress components:

$$\sigma_e = (\frac{1}{2}[(\sigma_1-\sigma_2)^2+(\sigma_2-\sigma_3)^2+(\sigma_3-\sigma_1)^2])^{1/2} \tag{174}$$

$$= (\frac{3}{2}\sigma'_{ij}\sigma'_{ij})^{1/2}$$

σ_i is the principal stresses, σ'_{ij} are the components of the stress deviator tensor

The effective strain ε_e is defined similarly in terms of the effective stress:

$$d\varepsilon_e = (\frac{1}{2}[(d\epsilon_1-d\epsilon_2)^2+(d\epsilon_2-d\epsilon_3)^2+(d\epsilon_3-d\epsilon_1)^2])^{1/2} \tag{175}$$

$d\epsilon_i$: the principal strain increments

From the above, the flow curve can be expressed, for instance, in terms of the power law equation below:

$$\sigma_e = K\varepsilon_e^n \tag{176}$$

In the Levy-Mises equation, the Flow Rule, it describes the relationship between stress and strain for an ideal plastic solid where the elastic strain are negligible, and it relates stresses to plastic strain increments.

$$\frac{d\varepsilon_1}{\sigma'_1} = \frac{d\varepsilon_2}{\sigma'_2} = \frac{d\varepsilon_3}{\sigma'_3} = d\lambda \tag{177}$$

The above equation can be written in terms of actual stress:

$$d\varepsilon_1 = \frac{2}{3} d\lambda [\sigma_1 - \frac{1}{2}(\sigma_2 + \sigma_3)] \tag{178}$$

$$d\varepsilon_2 = \frac{2}{3} d\lambda [\sigma_2 - \frac{1}{2}(\sigma_1 + \sigma_3)] \tag{179}$$

$$d\varepsilon_3 = \frac{2}{3} d\lambda [\sigma_3 - \frac{1}{2}(\sigma_2 + \sigma_1)] \tag{180}$$

The term $d\lambda$ can be evaluated using the effective strain, which gives the following equation:

$$d\varepsilon_e = \frac{2}{3} d\lambda \, \sigma_e \tag{181}$$

$$d\lambda = \frac{3}{2} \frac{d\varepsilon_e}{\sigma_e} \tag{182}$$

Now, the relationship between the plastic strain increment and stress deviator tensor:

$$d\varepsilon_{ij}^{P} = \frac{3}{2} \frac{d\varepsilon_e}{\sigma_e} \sigma'_{ij} \tag{183}$$

Using relationship above, we can rewrite the Prandtl-Reuss Equation, proposed by the Elastic-Plastic Solid (Prandtl, 1925), (Ruess, 1930):

$$d\varepsilon_{ij}^{T} = d\varepsilon_{ij}^{E} + d\varepsilon_{ij}^{P} \tag{184}$$

In the ABAQUS UMAT Fortran Libarary, since the implicit solver has been used, therefore, it requires an implicit integration scheme with a Radial Return Method: A trail stress σ_t^{tr} increment is chosen, and then it takes updated stresses $\sigma_{t+\Delta t}^{tr}$, which is outside of the yield surface. Then, the stress is updated with a plastic correction to bring back the stress onto the yield surface at time

$t + \Delta t$. Also, the plastic correlation is always tangent to the center of yield surface at deviatoric stress space. The Hooke's law in multi-axial form in terms of stress and strain tensor can be written as following:

$$\sigma = 2G\varepsilon^{e} + \lambda\, \mathrm{Tr}(\varepsilon^{e})\mathrm{I} \tag{185}$$

From some derivations, we can obtain the relationship between the deviator stress and trial deviator stress:

$$\left(1 + 3G\frac{\Delta p}{\sigma_e}\right)\sigma' = \sigma^{tr\prime} \tag{186}$$

Then, by applying the differential operator δ, we can obtain:

$$\left(1 + 3G\frac{\Delta p}{\sigma_e}\right)\delta\sigma' + \frac{3G}{\sigma_e}\delta\Delta p\sigma' - \frac{3G\Delta p}{\sigma_e^2} = \delta\sigma^{tr\prime} \text{ and } \delta\sigma^{tr\prime} = 2G\delta\varepsilon^{tr\prime} \tag{187}$$

Used above form, we can obtain the relationship between δσ and δε, which leads us to derive the Material Jacobian.

$$\delta\sigma^{tr\prime} = 2G(\delta\varepsilon - \frac{1}{3}\mathbf{II}\colon\delta\varepsilon) \tag{188}$$

$$\delta\sigma = [2GQ\frac{\sigma^{tr\prime}}{\sigma_e^{tr}}\frac{\sigma^{tr\prime}}{\sigma_e^{tr}} + 2GRI + (K - \frac{2}{3}GR)\mathbf{II}]\colon\delta\varepsilon \tag{189}$$

where: I is fourth-order identity tensor, and I is identity matrix

$$Q = \frac{3}{2}\left(\frac{h}{h+3G} - \frac{\sigma_e}{\sigma_e^{tr}}\right);\ R = \frac{\sigma_e}{\sigma_e^{tr}} \tag{190}$$

Therefore, from above, we can derive the Material Jacobian matrix for updating Jacobian 'DDSDDE' in the UMAT.

$$\mathrm{DDSDDE} = \begin{pmatrix} D11 & \cdots & D16 \\ \vdots & \ddots & \vdots \\ D61 & \cdots & D66 \end{pmatrix}_{6\times 6} \tag{191}$$

The main diagonal direct tensor components of Material Jacobian can be calculated by taking the derivative of equation 185.

$$D_{ii} = \frac{\partial \delta \sigma_{ii}}{\partial \delta \varepsilon_{ii}} = 2GQ \frac{\sigma_{ii}^{tr\prime}}{\sigma_e^{tr}} \frac{\sigma_{ii}^{tr\prime}}{\sigma_e^{tr}} + 2GR + \left(K - \frac{2}{3}GR\right);\ (i = 1,2,3) \tag{192}$$

The off-diagonal direct tensor components of Material Jacobian can be calculated by taking the derivative of equation 185.

$$D_{ij} = \frac{\partial \delta \sigma_{ij}}{\partial \delta \varepsilon_{ij}} = 2GQ \frac{\sigma_{ij}^{tr\prime}}{\sigma_e^{tr}} \frac{\sigma_{ij}^{tr\prime}}{\sigma_e^{tr}} + \left(K - \frac{2}{3}GR\right); (i = 1,2,3) \tag{193}$$

The main diagonal shear components of Material Jacobian can be calculated by taking the derivative of equation 185.

$$D_{kk} = \frac{\partial \delta \sigma_{ij}}{\partial \delta \gamma_{ij}} = 2GQ \frac{\sigma_{ij}^{tr\prime}}{\sigma_e^{tr}} \frac{\sigma_{ij}^{tr\prime}}{\sigma_e^{tr}} + GR;\ (k = 4,5,6) \tag{194}$$

The off-diagonal shear tensor components of Material Jacobian can be calculated by taking the derivative of equation.

$$D_{ik} = \frac{\partial \delta \sigma_{ii}}{\partial \delta \gamma_{ij}} = 2GQ \frac{\sigma_{ii}^{tr\prime}}{\sigma_e^{tr}} \frac{\sigma_{ij}^{tr\prime}}{\sigma_e^{tr}}; (k = 4,5,6) \tag{195}$$

UMAT Pseudo Code:

1. Initialize the variables: E μ σ_y
2. Calculate the compliance matrix components: ***DDSDDE***
3. Calculate the Elastic Stress Components: $\boldsymbol{\sigma = E\varepsilon}$
4. Recover the Elastic and Plastic Strains:
    1). Calculate the σ_a From Anand Viscoplaticity Model
    2). Calculate Von Mises Stress σ_{mises}
    2). Calculate the Yield Stress using $\sigma_{Yield} = \sigma_{mises} - \sigma_a$
    if f = $\sigma_{mises} - \sigma_a$>0: (Actively Yielding)
        1) Calculate the Flow Direction.
        2) Solve the strain increment DE using Newton Method:
            1. $\sigma_{increment}$ = E$\Delta\varepsilon$
            2. g = σ_{mises} - $\sigma_{increment}$
            3. updating $\Delta\varepsilon$
        3) Updating the final stresses and strains
5. Update Jacobian Matrix from the Flow Theory.

6.9 Finite Element Analysis Results

We have finished a simulation of completed thermal cycle which ranges from 25ºC to 125ºC. In the simulation, since the Coefficients of Thermal Expansion (CTE) are different in materials of the package, it causes the significant warpage with a thermal load. It can be viewed from the contour that the middle part of the package has little displacement in vertical direction compared to the one at boundary. The local model, a critical solder joint region under the die shadow, experiencing a similar displacement gradient due to the CTE difference. This would lead to a consequence of fatigue of plasticity of the solder joint. The elastic shear strains have been shown in the following Figure 235. The contour has a similar pattern with the one in bending test. There is about the $4.5e^{-4}$ maximum shear strain, and it is from the E23. The total plastic strain energy dissipation (i.e. 10^{-3} N.m) by rate-dependent plastic deformation has been shown in the plot. The plastic strain energy describes how much internal energy has been consumed by the plasticity material, which makes it fragile to reach the threshold of failure. The plastic strain has been calculated by subtracting the elastic strain in the ANNAND model. The history of the cumulative plastic strain has been shown in the Figure 238, and it mainly distributes in front of the crack tip, and the green contour shows the plastic zone of the crack tip. In the plot, the crack would propagate along the highest plastic region that has been softened in each cycle.

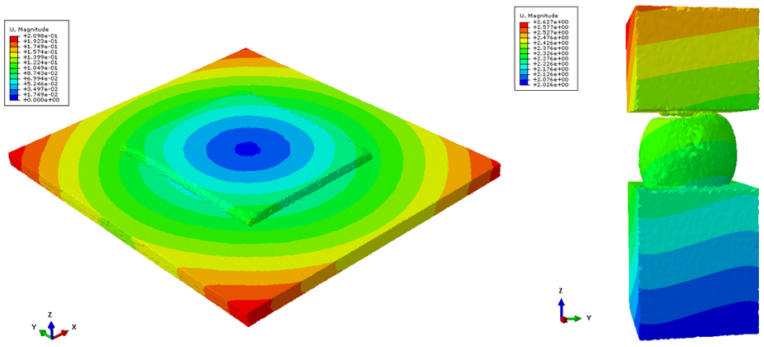


Figure 234 Global and Local Model Displacement Magnitude Example

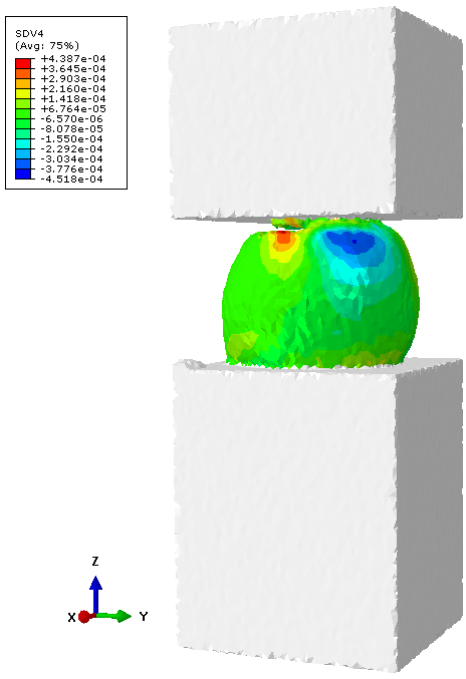


Figure 235 the elastic shear strain components S12

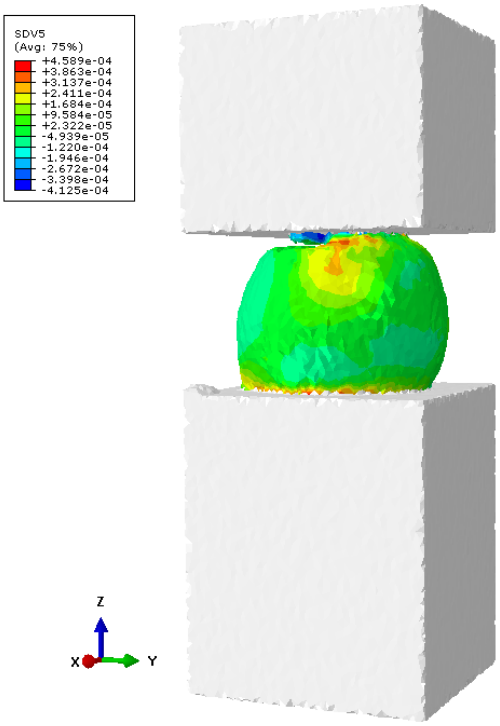


Figure 236 the elastic shear strain components S13

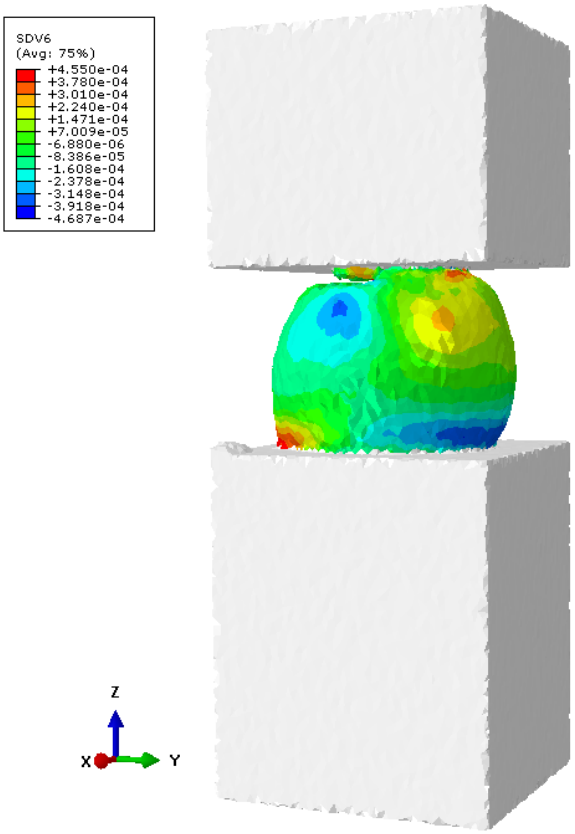


Figure 237 the elastic shear strain components S23

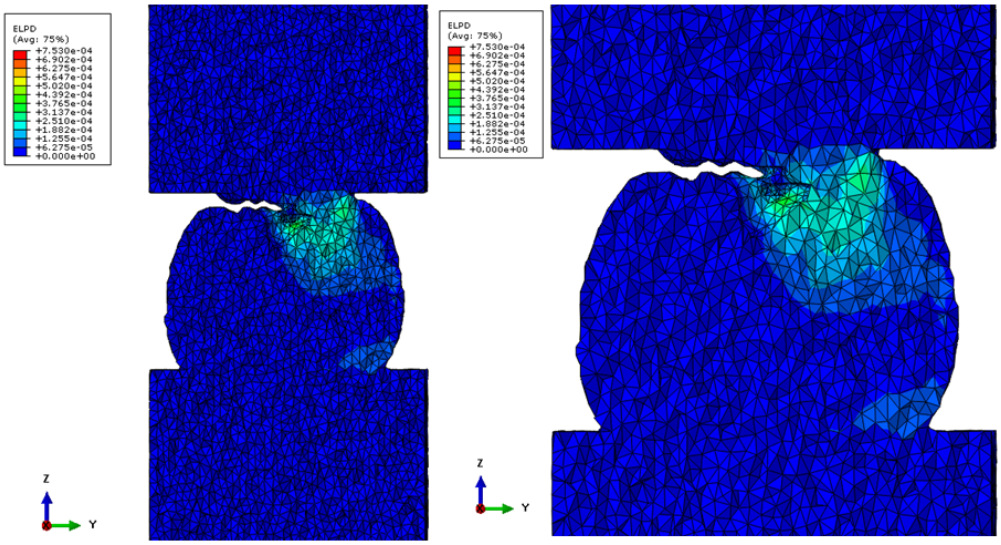


Figure 238 Total energy dissipated in the element by rate-dependent plastic deformation

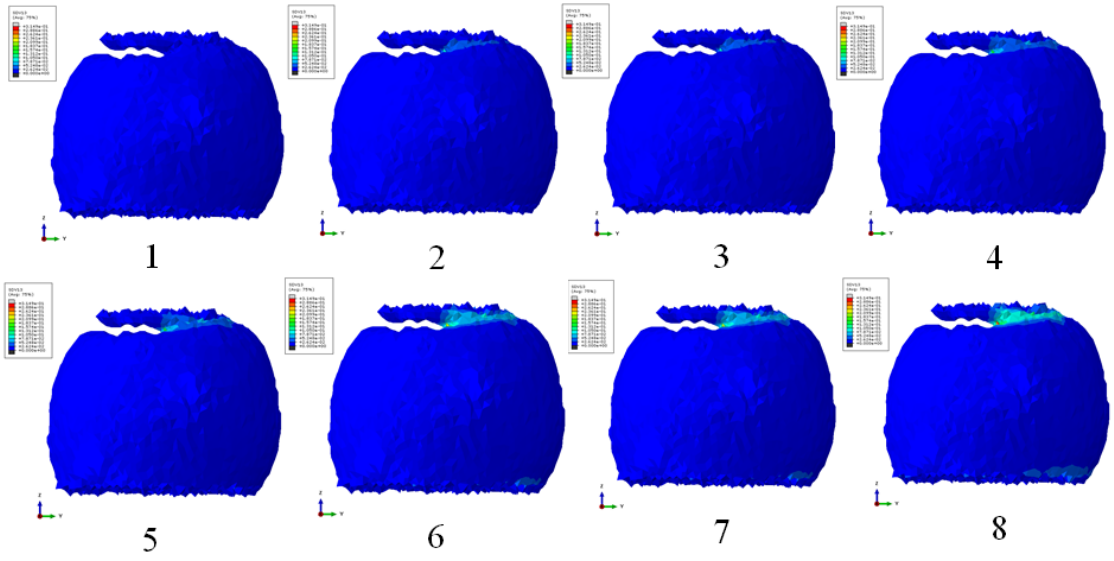


Figure 239 Effective Plastic Strain Increment of Time in a Thermal Cycle

In this chapter, we have utilized a global-local model, which has a consistency of displacement boundary. During a temperature cycling, there is a package warpage because the CTE difference exists in the different material sections. It will cause a mismatch between those sections and will generate a strain.

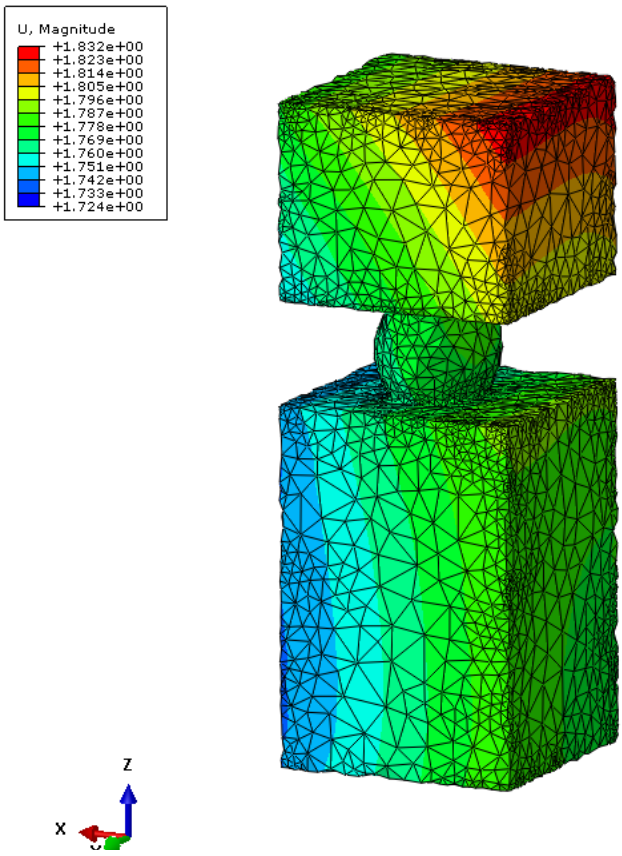


Figure 240 the Local Model Displacement Magnitude

We can see the stress concentration contour plot in the solder void during the deformation. The shear stress components: S_{12}, S_{13}, and S_{23} have been shown in the Figure 241. There is about 45° bending trend around the sold void during the deformation.

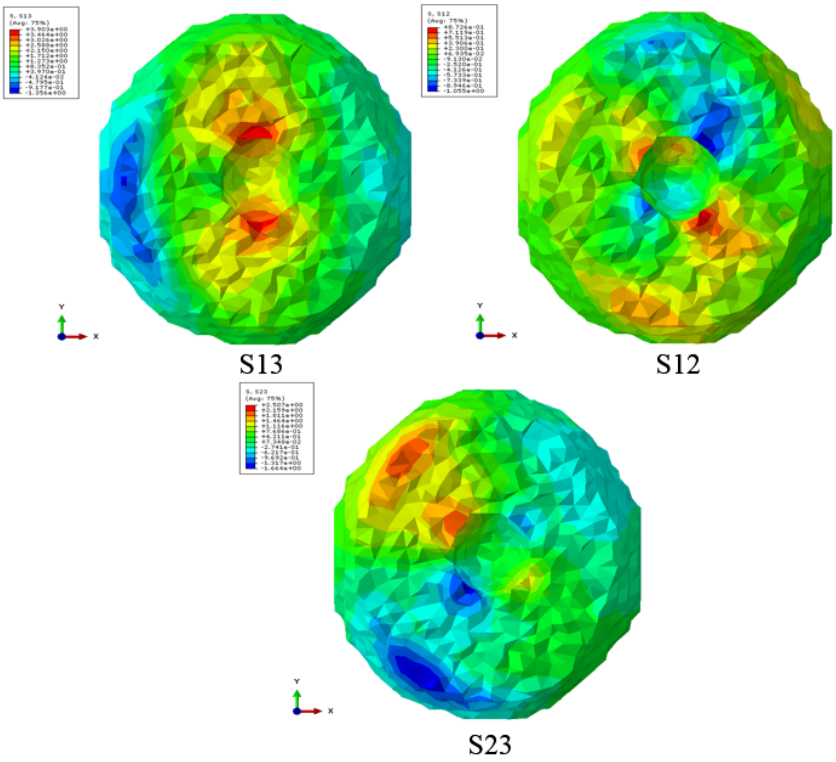


Figure 241 Shear Stress Concentration Contour in Solder Voids

The rate-dependent plastic deformation has been shown in following Figure 242, Figure 243, and Figure 244. We can see that there is a large plastic region in the solder. The plastic strain has been calculated by the Visco-plasticity model. The plastic region will increase as numbers of cycle increase.

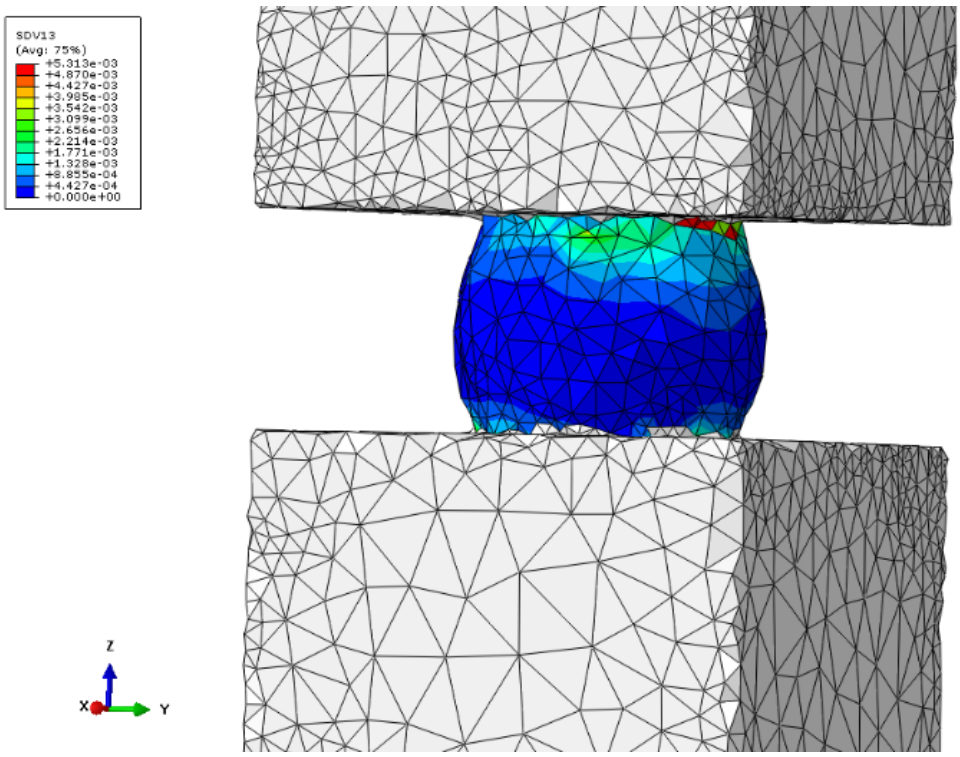


Figure 242 Plastic Strain S12

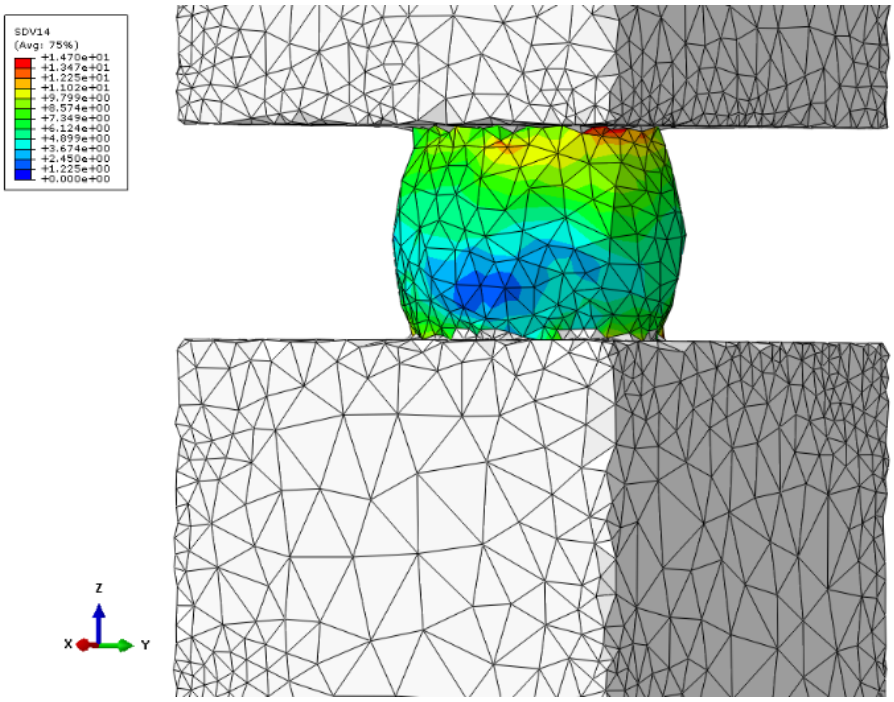


Figure 243 Plastic Strain S13

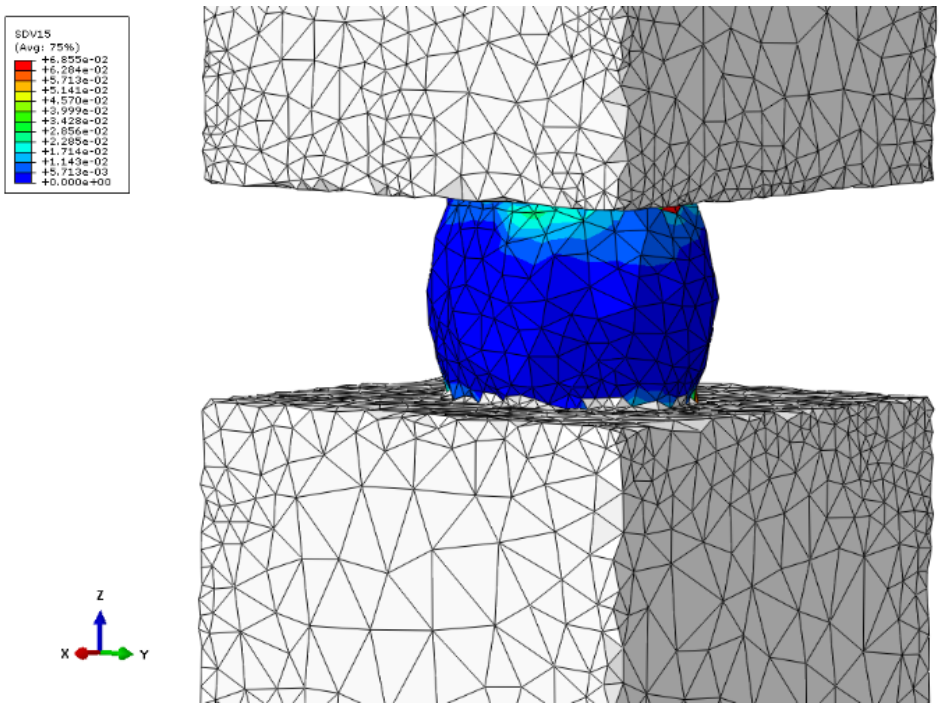


Figure 244 Plastic Strain S23

The following Figure 245 shows the dissipated plastic strain energy density in void elements as a function of time. The x-axis indicates the step in two complete cycles. A cycle has four stages including ramp up, soaking 1, ramp down, and soaking 2. The y-axis displays the value of strain energy density whose unit is J.

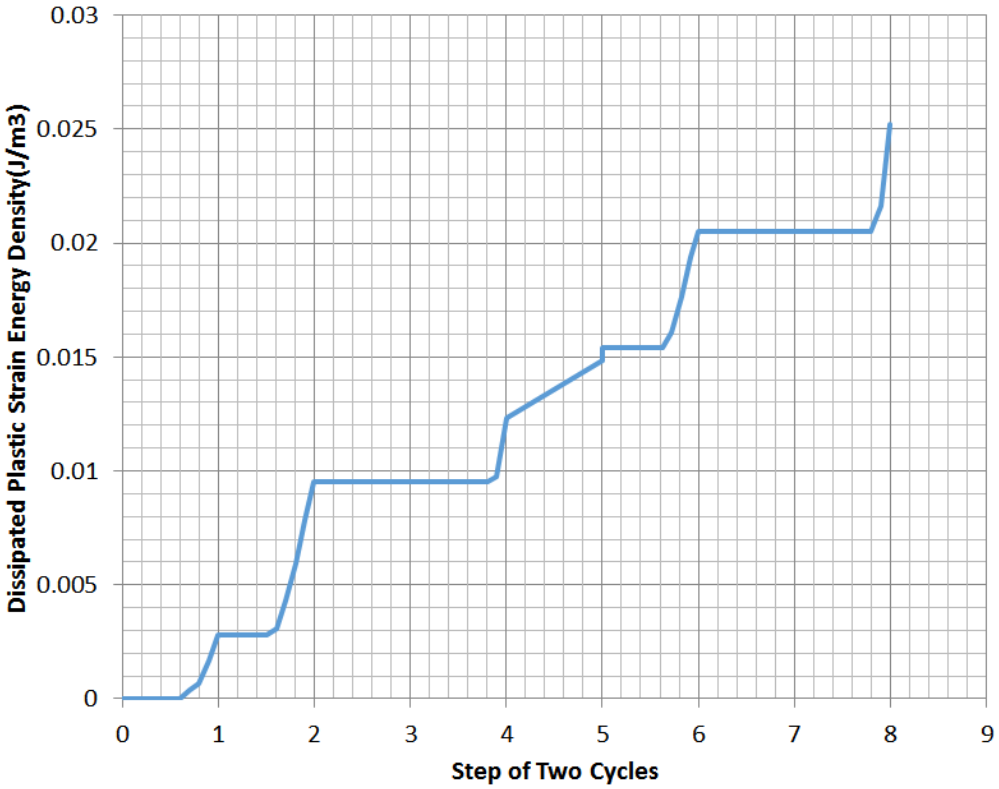


Figure 245 Dissipated Plastic Strain Energy Density

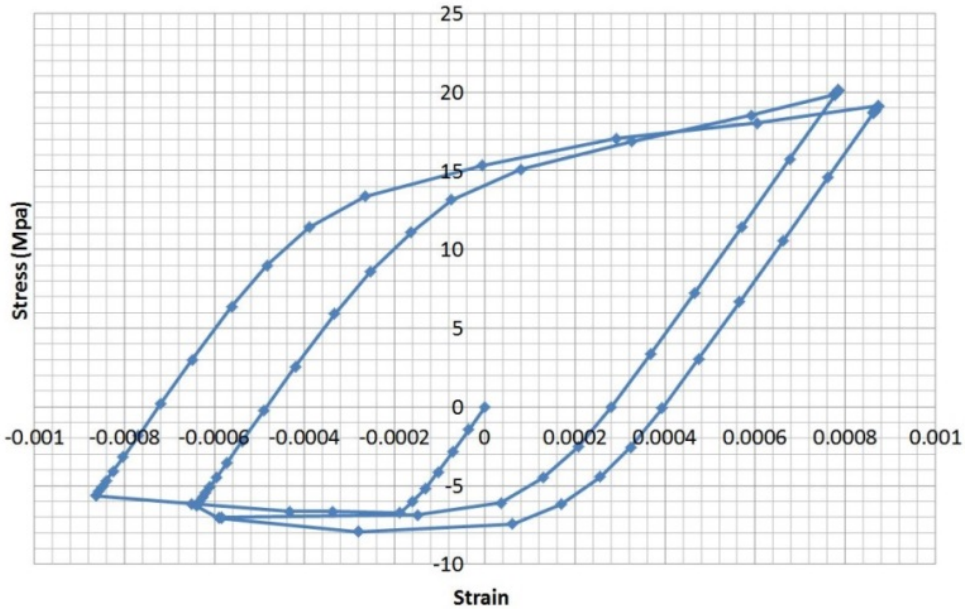


Figure 246 Crack Front Element Hysteresis Loop of Solder Joint

In this chapter, we develop a better scheme of the Finite Element Analysis to run a thermal-mechanical analysis, which simulates thermal cycling on a BGA324 package. A new method, using micro-CT 3D reconstruction geometry for electronics' Finite Element Analysis, has been shown. The micro-CT scan and 3D reconstruction have been utilized to generate a digital volume based on a grayscale value, and a mesh routine has also been demonstrated. In the mesh, this method produces a model from different ISO-surfaces, and the element, a C3D4 tetrahedron, has four vertices and four faces. To reduce simulation time, a global and local scheme has been developed, which applies a consistent boundary condition from displacement interpolation. Moreover, the Annand Visco-Plasticity material behavior model has been used and been incorporated into the ABAQUS CAE model. A presented void solder joint from the CT scan has been modeled and analyzed. Lastly, elastic shear strain components have been shown on the solder joint contour, and a plastic region has been shown in front of the void. Also, a cumulative plastic work has been shown on the void. The hysteresis loop of stress and strain has been shown in the history of thermal cycling.

Chapter 7 Data Driven Based Electronics Prognostic Health Management

7.1 Kalman Filter and Extended Kalman Filter in the BGA Package

7.1.1 Introduction to the Kalman Filter / Extended Kalman Filter Algorithm

The Kalman Filter (KF) is used for tracking the digital signal in the system, which has been widely used in the space navigation, electronics control, and economics. Previously, the Kalman Filter has been used in the guidance and tracking applications (Kalman 1960, Zarchan 2000). The basic concept about the KF is to use the recursive and error correction algorithm to keep track of a signal in a dynamic system.

The Extended Kalman Filter (EKF) is built upon the KF algorithm, in which the nonlinear fundamental state matrix has been implemented in order to adapt to advanced and nonlinear system. In its applications, system damage state estimation in the presence of measurement noise and process noise has been achieved using the Extended Kalman Filter (EKF). The following Figure 247 illustrates the basic recursive scheme in the KF and EKF.

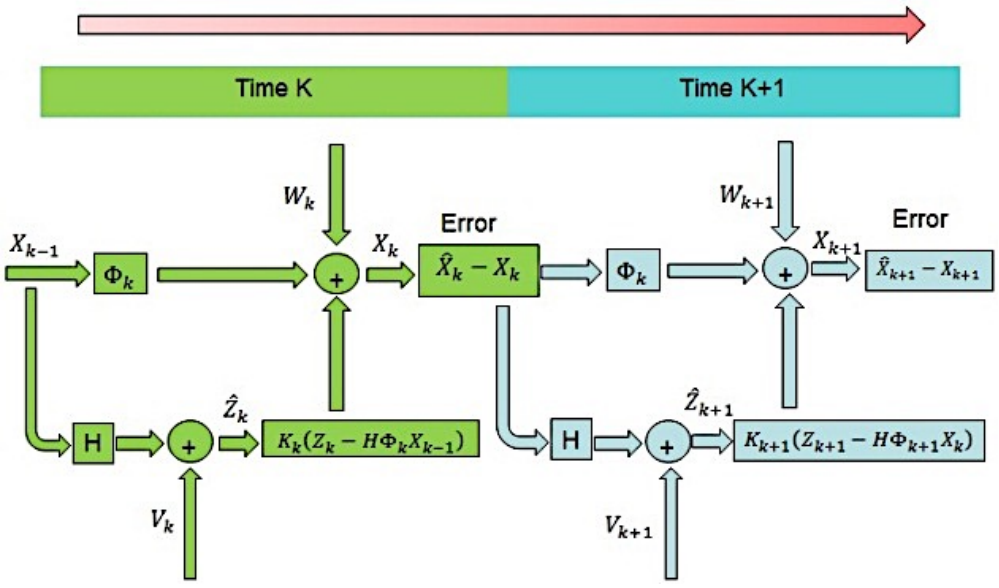


Figure 247 the Recursive Algorithm and Extended Kalman Filter

System state has been described in state space form using the measurement of the feature vector, the change of velocity in feature vector, and the change of acceleration in the feature vector. The equivalent Extended Kalman Filter equation for state space representation is in the presence of process noise and measurement noise is:

$$\dot{x} = Fx + w \tag{196}$$

$$\dot{x} = f(x) + w \tag{197}$$

F is the system linear dynamic matrix; f(x) is non-linear dynamic matrix; the G is measurement matrix; μ is measurement vector; and w is system white noise;

$$Z = H \cdot x + V \tag{198}$$

$$Z = h(x) + V \tag{199}$$

H is the measurement matrix, z is the measurement vector, h(x) is a measurement function, which is a nonlinear function of states, and v is zero-mean random process described by the measurement noise matrix. The process noise can be calculated by taking the expectation of white noise:

$$\mathrm{Q} = \mathrm{E}[\mathrm{ww}^{\mathrm{T}}] \tag{200}$$

Similarly, the measurement noise matrix is derived from the measurement noise as following:

$$\mathrm{R} = \mathrm{E}[\mathrm{vv}^{\mathrm{T}}] \tag{201}$$

Since the system-dynamics matrix 'F' and measurement equations are nonlinear, a first-order approximation is used in the continuous Riccati equations for the systems dynamics matrix F and the measurement matrix H. It is expected that the progression of interconnect damage is nonlinear, therefore, it needs to be linearized before it can be estimated. In the Extended Kalman Filter, the problem of linearization is addressed by calculating the Jacobian of the nonlinear function of states f(x) and the measurement function h(x) around the estimated state. System state at each future time has been computed based on the state space at predicting time point, system dynamics matrix, the

control vector, control matrix, measurement matrix, measured vector, process noise, and measurement noise. The matrices are related to the nonlinear system and measurement equations according to:

$$F = \left.\frac{\partial f(x)}{\partial x}\right|_{x=\hat{x}} \tag{202}$$

$$H = \left.\frac{\partial h(x)}{\partial x}\right|_{x=\hat{x}} \tag{203}$$

From the linear dynamic equations, we can clearly see that the world is linear as it assumed. All the problems can be solved linearly through the matrix approximation. Therefore, we can get the first and second derivatives from the linear matrix calculation;

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dddot{x} \end{bmatrix} = F \cdot \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} \tag{204}$$

The system dynamic matrix for the EKF is:

$$F_{EKF} = \begin{bmatrix} \frac{\partial \dot{x}}{\partial x} & \frac{\partial \dot{x}}{\partial \dot{x}} & \frac{\partial \dot{x}}{\partial b} \\ \frac{\partial \ddot{x}}{\partial x} & \frac{\partial \ddot{x}}{\partial \dot{x}} & \frac{\partial \ddot{x}}{\partial b} \\ \frac{\partial \dot{b}}{\partial x} & \frac{\partial \dot{b}}{\partial \dot{x}} & \frac{\partial \dot{b}}{\partial b} \end{bmatrix} \tag{205}$$

The system dynamic matrix for the KF is:

$$F_{KF} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tag{206}$$

We use this Jacobin Matrix to linearize the problem; therefore, it can use the similar KF updates. This is for the second order system, and thus we can find the transfer function F to describe certain system, and it is the key to find fundamental matrix $\Phi(t)$. In this chapter, for the EKF, we used the following exponential system model:

$$x = \alpha \cdot e^{\beta \cdot t} \tag{207}$$

The state vector is:

$$x_k = \lfloor x \quad \dot{x} \quad \beta \rfloor \tag{208}$$

This is an exponential function. The α and β are two coefficients that are decided by different systems. The first derivation ẋand the second derivation ẍ are derived from the following equation:

$$x = \alpha \cdot e^{\beta \cdot t} \tag{209}$$

Therefore, the elements in system dynamic matrix will be calculated as:

$$F_{EKF} = \begin{bmatrix} \beta & 1 & x \\ \beta^2 & \beta & 2\beta x \\ 0 & 0 & 0 \end{bmatrix} \tag{210}$$

Usually, the fundamental matrix Φ(t) can be obtained from two ways: the first way, we can get it from Laplace Transform, simply as the following:

$$\Phi(t) = \ell[(SI - F)^{-1}] \tag{211}$$

Where ℓ^{-1} is inverse of the Laplace Transform; The second way, known as the common way to find Φ(t), deriving from the Tylor Series expansion:

$$\Phi(t) = I + FT + \frac{(FT)^2}{2!} + \frac{(FT)^3}{3!} + \cdots \tag{212}$$

Normally, we use the first two terms for representing the fundamental matrix Φ(t), because simply adding more terms cannot improve the precision and convergence of filter. The Fundamental Matrix in the Extended Kalman Filter is:

$$\Phi_{EKF}(T) \approx I + F_{EKF}T = \begin{bmatrix} 1+\beta T & T & xT \\ \beta^2 T & 1+\beta T & 2\beta \cdot xT \\ 0 & 0 & 1 \end{bmatrix} \tag{213}$$

The Fundamental Matrix in the Extended Kalman Filter is:

$$\Phi_{KF}(T) \approx I + F_{KF}T = \begin{bmatrix} 1 & T & 0.5Ts^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \tag{214}$$

In the Kalman Filter, the Fundamental Matrix will be directly used to update the estimation from last estimation. Generally speaking, the process to find the optimal estimation can be expressed as following steps: First of all, we make the primary estimation, which should be approximate to the initiate value in the dataset, and secondly, we can find the projection using the fundamental matrix $\Phi(t)$ and simply calculate as:

$$\bar{x} = \Phi(t) \cdot \hat{x} \tag{215}$$

For the KF, the projection can be represented by:

$$\begin{bmatrix} \Delta\dot{x} \\ \Delta\ddot{x} \\ \Delta\dddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\dot{x} \\ \Delta\ddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ w \end{bmatrix} \tag{216}$$

For the EKF, the projection can be represented by:

$$\begin{bmatrix} \Delta\dot{x} \\ \Delta\ddot{x} \\ \Delta\dot{b} \end{bmatrix} = \begin{bmatrix} \frac{\Delta\dot{x}}{\Delta x} & \frac{\Delta\dot{x}}{\Delta\dot{x}} & \frac{\Delta\dot{x}}{\Delta b} \\ \frac{\Delta\ddot{x}}{\Delta x} & \frac{\Delta\ddot{x}}{\Delta\dot{x}} & \frac{\Delta\ddot{x}}{\Delta b} \\ \frac{\Delta\dot{b}}{\Delta x} & \frac{\Delta\dot{b}}{\Delta\dot{x}} & \frac{\Delta\dot{b}}{\Delta b} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\dot{x} \\ \Delta b \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ w \end{bmatrix} \tag{217}$$

The next estimate vector could be obtained from the following equation:

$$\hat{x} = \bar{x} + K(Z - H \cdot \bar{x}) \tag{218}$$

K: Kalman Gain
H: measurement matrix
Z: measurement.

Each time we update the Kalman Gain (KG) and Covariance Matrix (CM), which minimizes the errors and makes optimal approximation during each step. Thus, the Kalman Gain mainly deliver

the information about how our estimation is close to the observation. The way to obtain Kalman Gain (K) is from three Riccati recursive equations:

$$M_k = \Phi_k P_k \Phi_k^T + Q_K \tag{219}$$

$$K = M_k H_t (H M_k H_t + R_k)^{-1} \tag{220}$$

$$P_k = (I - KH) M_k \tag{221}$$

In the above equations, M_k is the covariance matrix; Φ is the fundamental matrix; Φ_t is the transpose of fundamental matrix; P_k is another covariance matrix that representing error according to the time; Q_k is the discrete process noise matrix, which is calculated as the following:

$$Q = \Phi_s \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{222}$$

$$Q_k = \int_0^{Ts} \Phi(\tau) \cdot Q \cdot \Phi(\tau)^T \, d\tau \tag{223}$$

H is the unit measurement matrix, and H_t is transpose of H; the K is the Kalman Gain; R_k represents the measurement noise of different system. We notice that those three equations run in the recursions: for the initial covariance error P_0, we can find variance matrix M_k that represents the current error in time. Then, we use it in the second equation to find the Kalman Gain K. After that, we substitute the estimated Kalman Gain K into the third equation to update covariance error P_k from the last time. Thus, we obtain the 'next' covariance error P_{k+1}. Iteratively, as we go back to the first equation, we can update covariance matrix M_k and Kalman Gain K. In the EKF, the Euler integration has been utilized instead of the KF's fundamental matrix. To get estimation by projection, it can be found the following two equations:

$$\bar{x}_k = \hat{x}_k + \bar{\dot{x}}_{k-1} T \tag{224}$$

$$\bar{\dot{x}}_k = \hat{\dot{x}}_k + \bar{\ddot{x}}_{k-1} T \tag{225}$$

We call the equation above is the update equations, $\hat{x}_k$ represents the projection from the last time k ; $\hat{\dot{x}}_{k-1}$ is the first derivative at time (k-1); $\hat{x}_k$ is the estimation at time k, $\hat{\dot{x}}_k$ is the first derivative at time k; T is the sample time.

$$\hat{x}_k = \bar{x}_k + K_1(Z - H \cdot \bar{x}_k) \tag{226}$$

$$\hat{\dot{x}}_k = \bar{\dot{x}}_k + K_2(Z - H \cdot \bar{x}_k) \tag{227}$$

The above equations are the basic Extended Kalman Filter equations, which are to find the estimation and its 'velocity'. Also, in those equations, it uses the same three Riccati Equations that expressed in the Kalman Filter to obtain the Kalman Gain K_1 and K_2.

From above, we can see that the Extended Kalman Filter have turned the non-linear problem into a linear solution through integration method. So at each step, the Extended Kalman Filter made a small integration, and if the integrate time is small, then the answer we get is becoming much closer to the real solution. However, the difficulty within the Extended Kalman Filter is to find the exact non-linear model to describe the system, which always contains many unknown coefficients. Therefore, the better we know about the test system, the better we can predict system model in the Extended Kalman Filter, for example, the theories and functions in the situation of Electronics' failure. Therefore, the prediction of Remaining Useful Life (RUL) would be close to the real RUL in our PHM network.

Algorithm: Filtering and RUL prediction:

1. Initiate $\hat{x}_0$
2. Make the projections:
   $\bar{x}_k = \hat{x}_k + \bar{\dot{x}}_{k-1} T$
   $\bar{\dot{x}}_k = \hat{\dot{x}}_k + \bar{\ddot{x}}_{k-1} T$
3. Calculate error covariance matrix before updata:

$$M_k = \Phi_k P_k \Phi_k^{\ T} + Q_K$$

4. Calculate the Kalman Gain:
$$K = M_k H_t (H M_k H_t + R_k)^{-1}$$
5. Update the estimation with measurement:
$$\hat{x}_k = \bar{x}_k + K_1 (Z - H \cdot \bar{x}_k)$$
$$\hat{\dot{x}}_k = \bar{\dot{x}}_k + K_2 (Z - H \cdot \bar{x}_k)$$
6. Calculate error covariance after measurement update:
$$P_k = (I - KH) M_k$$
7. Extrapolate feature vector to threshold value:
$$LM = \bar{x}_{k+n} \cdot e^{\bar{\beta}_{k+n}} + w_{k+n}$$
8. Report predicted RUL (and uncertainty);
9. Iterate to step 2 for next measurement (k = k +1);

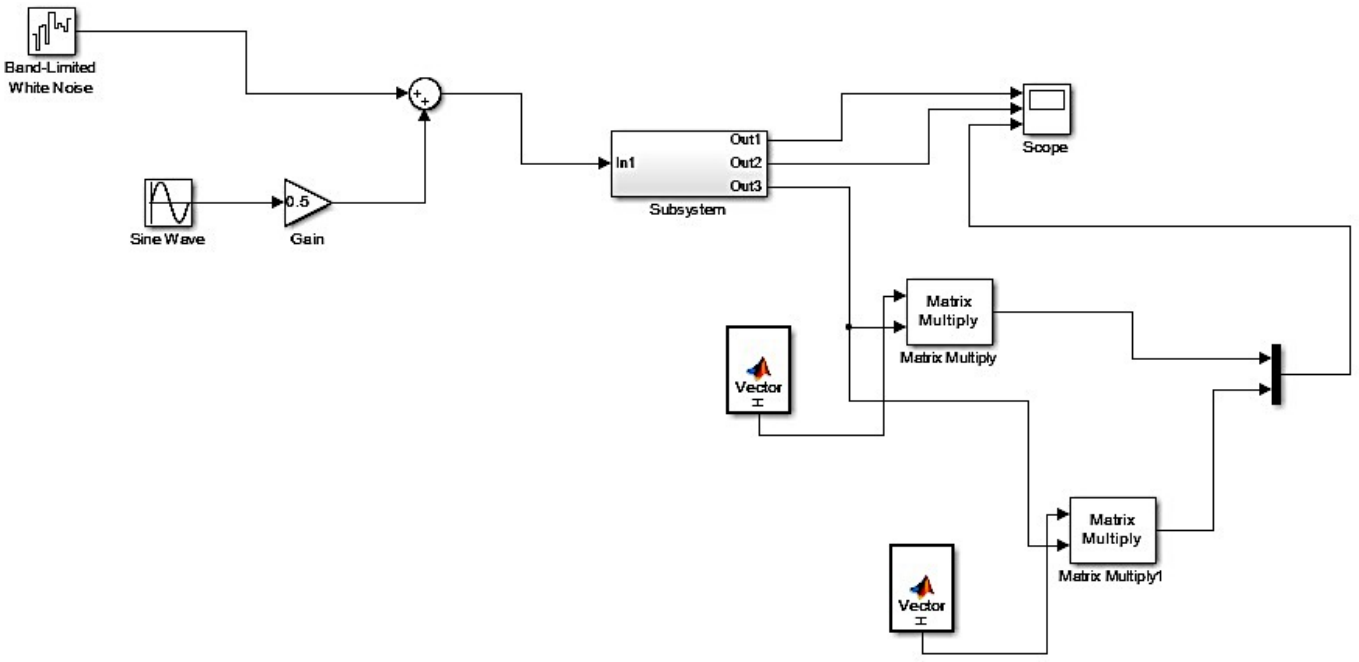


Figure 248 Simulink KF Simulation

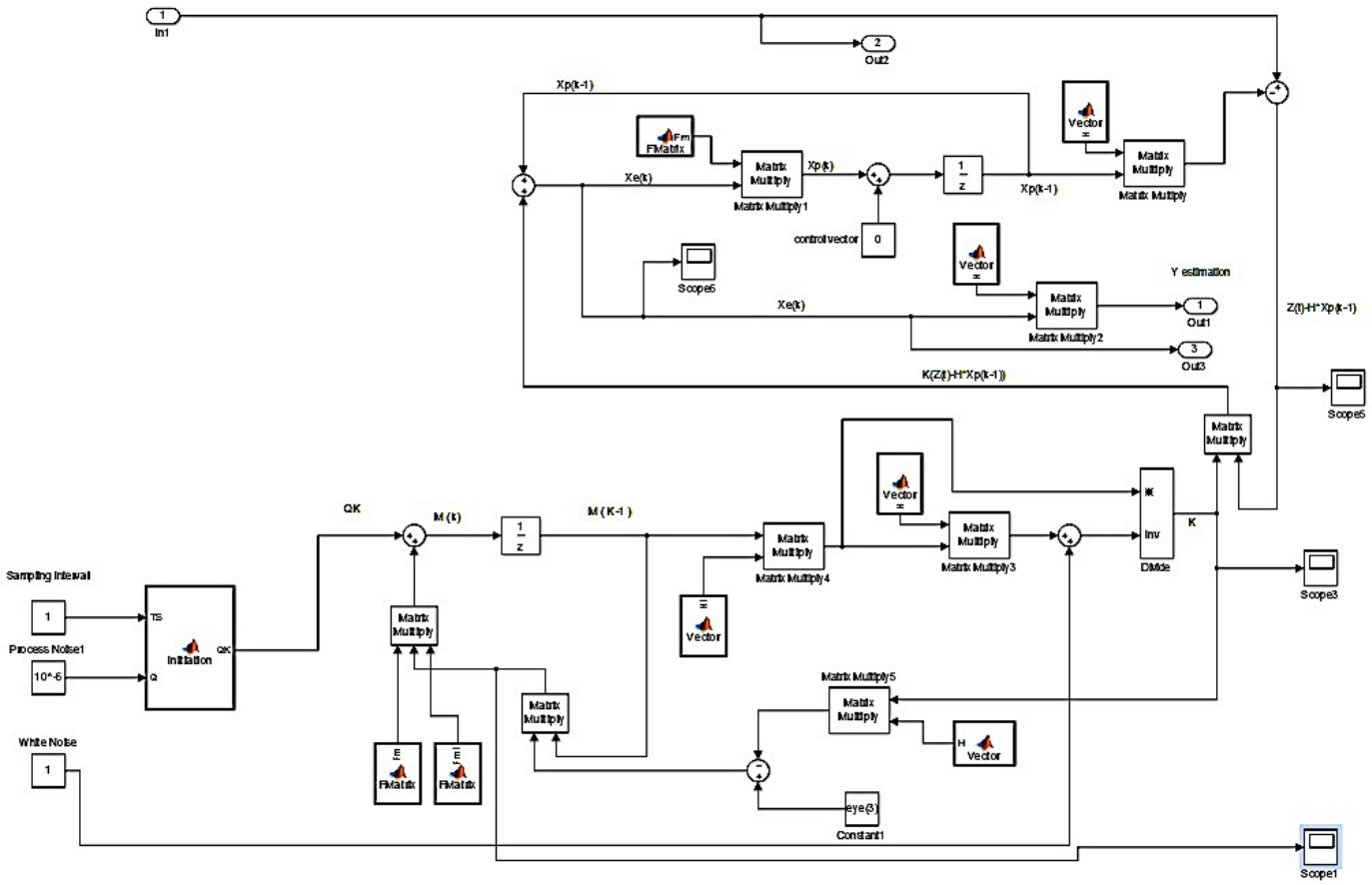


Figure 249 KF Simulink Diagram

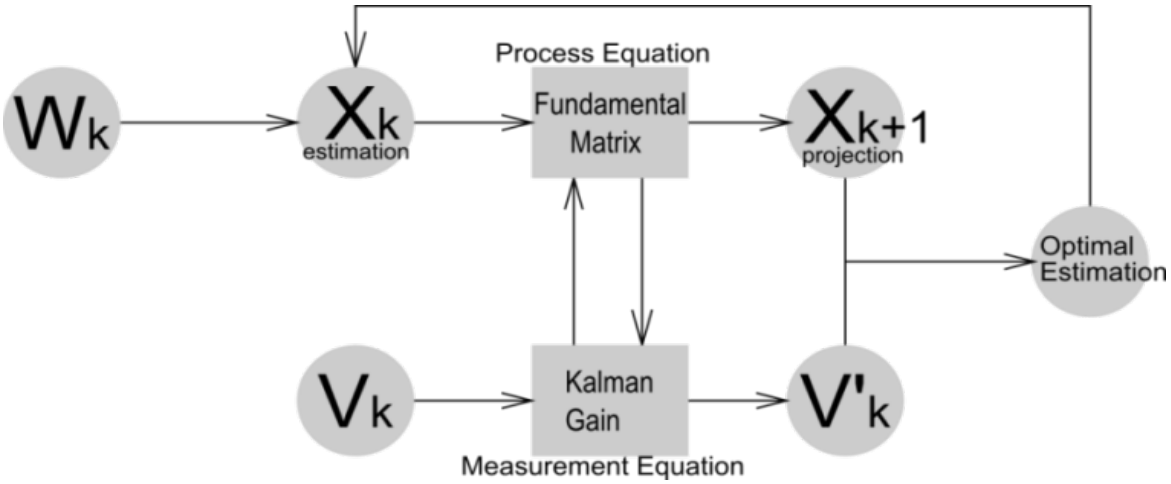


Figure 250 Kalman Filter Algorithm

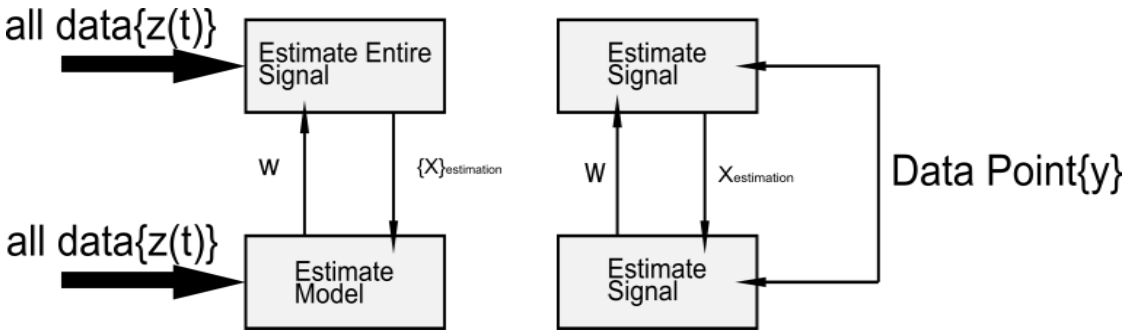


Figure 251 Extended Kalman Filter Algorithm

7.1.2 PCB Assembly Remaining Useful Life Prediction using KF/EKF

Electronic interconnects may have damage under exposure to vibration and mechanical shock. The damage may include an open circuit after a finite period of operation. The traditional methods for damage detection such as microscopy and surface polishing are destructive in nature and provide limited information offline. In this chapter, Resistance Spectroscopy (RS) and phase-sensitive detection based state space vectors have been used to monitor the health of interconnects and assess Remaining Useful Life. Two techniques including Kalman Filter and Extended Kalman Filter have been used to estimate the future state of the system. The measured state variable has been related to the underlying interconnect damage in a form of inelastic strain energy density. The performance of both the Prognostication Health Management (PHM) algorithms during vibration and mechanical shock has been quantified using performance evaluation metrics. This methodology has been demonstrated on lead-free area-array and advanced interconnect electronic assemblies. Model predictions have been correlated with experimental data. The presented approach applies to functional systems where corner interconnects in area-array packages may often be redundant.

In the experiment, the test board with several package architectures was used for generating the Dataset, the package architectures including the plastic ball-grid arrays, chip-array, ball-grid

arrays, tape arrays, ball-grid arrays, and flex-substrate ball-grid arrays are arranged on the test board. The ball counts in the range of 64 to 676 I/O are applied to the experimental matrix, and the pitch sizes are from 0.5 mm to 1 mm, and package sizes are from 6mm to 27 mm. So there are six different package architectures in this test board, each package measured uniquely with the single channel. There is a Table 17 showed the various sizes of those package architectures' parameters.

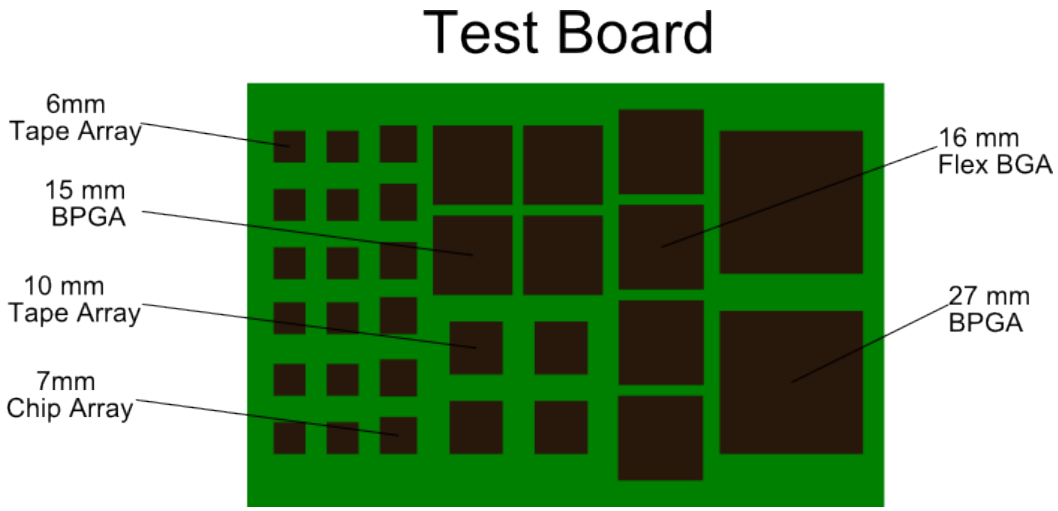


Figure 252 Test Board Architectures

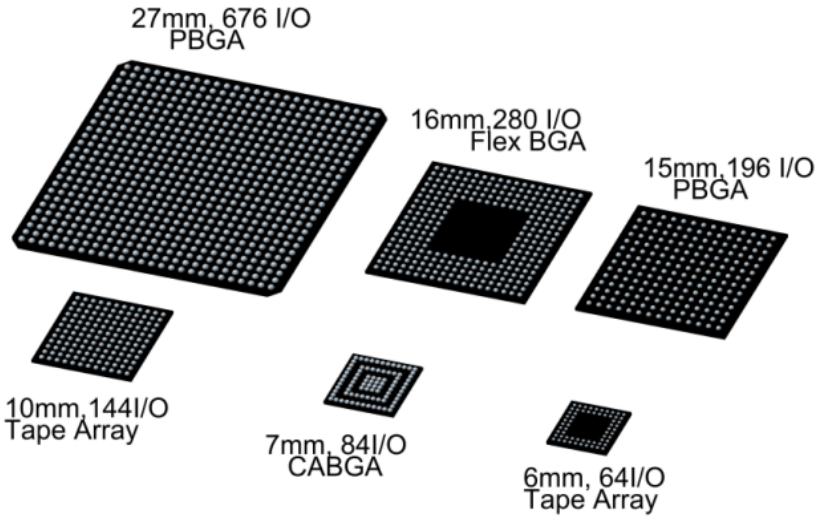


Figure 253 Interconnect array Configuration for Test Board

	6 mm Tape Array	7 mm Chip Array	10 mm Tape array	15 mm PBGA	16 mm Flex BGA	27 mm PBGA
I/O	64	84	144	196	280	676
Pitch (mm)	0.5	0.5	0.8	1	0.8	1
Die Size (mm)	4	5.4	7	6.35	10	6.35
Substrate Thick (mm)	0.36	0.36	0.36	0.36	0.36	0.36
Pad Dia. (mm)	0.28	0.28	0.30	0.38	0.30	0.38
Substrate Pad	NSMD	NSMD	NSMD	SMD	NSMD	SMD
Ball Dia. (mm)	0.32	0.48	0.48	0.5	0.48	0.63

Table 17 The Package Architecture' Parameters

The test board was set on a test machine, which is the Lansmont Model 23 drop tower in Figure 254. The 0.5 millisecond and 1500G impact force generated during the test to the test board by JEDEC standard JESD-B211. The continuity damage of the test board was measured by a sensor at each single drop. The continuous condition of electronics was detected with 10 million samples per second during the experiment. The board deformation was recorded by the High-Speed digital video during the shock, which contains the failure feature vectors. Later, it will be sent to a software, transferring the board deformation into the strain vectors to generate the experiment strains. In a word, the Resistance Spectroscopy will rely on the measurement from electronics combining with the actual physical model from the theory.

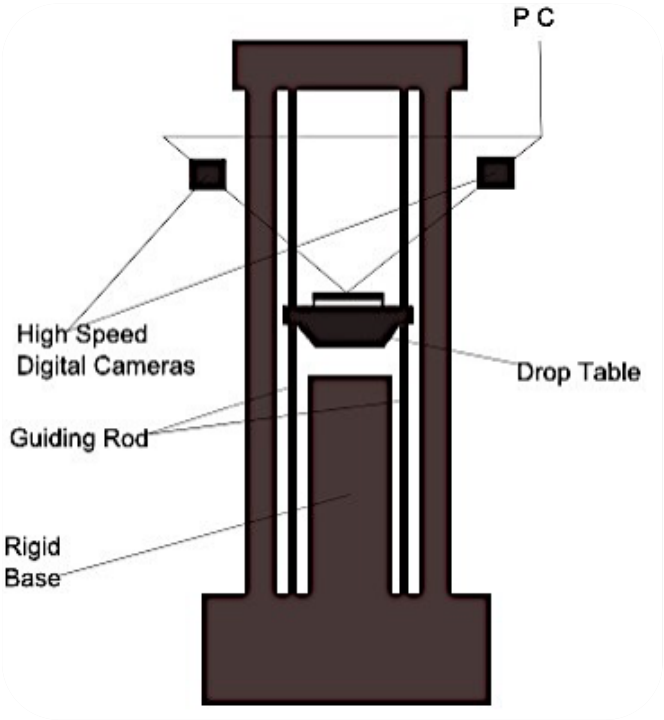


Figure 254 Drop Tower and High Speed Digital Cameras for Digital Image Correlation

The Phase-shift measurements had been continuing in the experiment until the entire package architectures failed. The packages displayed various degree damages before the continuous failures occur. An algorithm using the different metrics has been introduced to judge the damage of the packages. The number of open events multiplies by the event's severity to reach a critical value, which quantifies the severity of the failure. Therefore, the test procedure and damage detection were controlled and measured by this metric during the experiment, which is useful for us to get a general idea about the progression in the test. Each package is interrogated by the Resistance Spectroscopy technique individually. The switching has been used to cycle through all the packages on the test board.

In addition to the shock test, the test board with package architectures was also subjected to the vibration test on the LDS Model V772 vibration table. A step stress was applied to ramp up

the stress level to induce damage gradually, and the individual random stress profiles also used in the step stress. The performances of the Kalman Filter with the dataset are showed below, the blue line is the raw dataset, and the red line is Kalman Filter, which the noises, to some extent, have already been filtered out by the Kalman Filter.

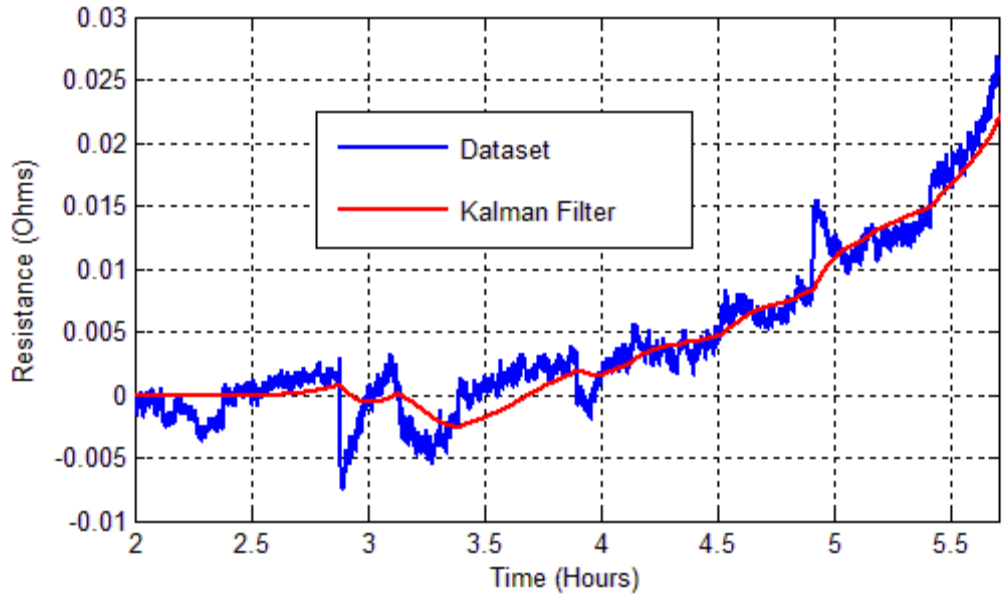


Figure 255 Kalman Filter Performance

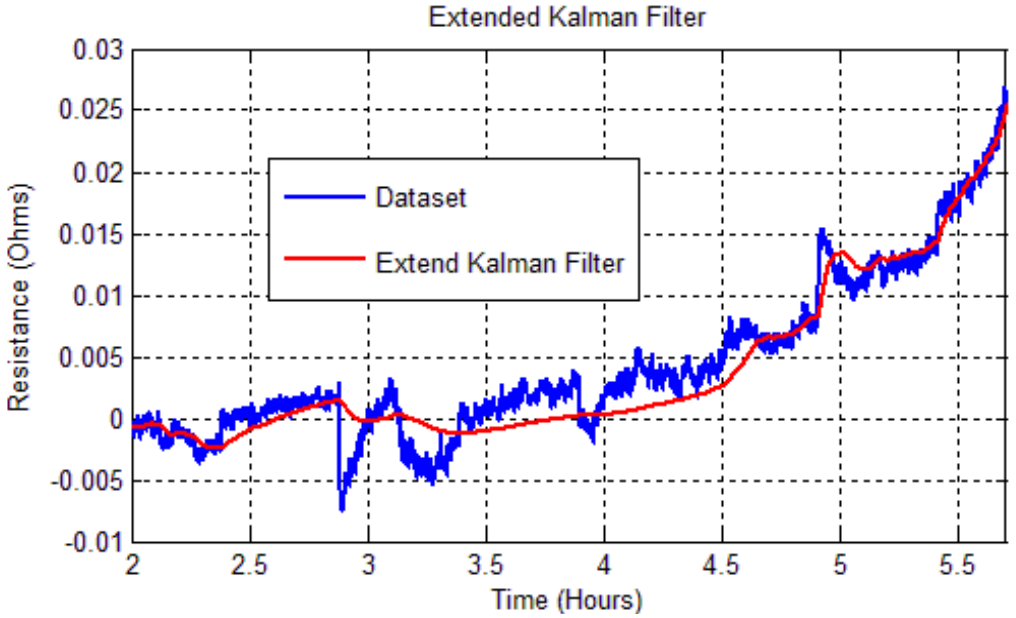


Figure 256 Extend Kalman Filter Performance

Figure 255 is the Kalman Filter, and Figure 256 is the Extended Kalman Filter, and from two plots, it is easy to find out that the EKF line is much lower than the KF line in the prediction. It can also be known that in this part the KF line is smoother than the EKF line. Actually, between the time 3.5 hours and 4.5 hours, the EKF is not performing very well, which there is little off from the dataset. However, when we take careful observations and comparisons, the EKF is much closer to the dataset when the experiment is approaching the End of Life (EOL), which could be better for us to make a prediction for the Remaining Useful Life (RUL) that is close to the failure. But even if the EKF was close to the raw dataset, we cannot make a promise that the EKF would be better than the KF at this simple comparison for the reason that we do not have much information concerning about the velocity and acceleration for the predicted point. So, there are some distinct results when applying them in the RUL prediction, which the predicted RUL depends on the estimations of state vectors. If the state vectors are predicted well the real failure close to the very end of time, then the method would be precise and better. As a matter of fact, the oscillation of velocity and acceleration line will make it difficult to predict, so the smooth lines would be better and less noisy.

The value of process noise decides the signal to be fluctuate or smooth, in the real sense, it tells us whether the model that we used in the Kalman Filter is precise or not. If the value of process noise is large, then it says the model is not exact. Thus, the Kalman Filter would adjust by itself discarding the model and adapting to the raw dataset in the measurement, which makes the line fluctuate.

The three Kalman Gains are represented by following two plots, the first plot is Kalman Filter Gains, and the second is Extended Kalman Filter Gains. The Kalman Gains expressed the time of convergence in the estimation; we can see that the three Kalman Gains behave similarly in

the Kalman Filter. Thus, those three Kalman Gains converged almost at the same time, but in the Extended Kalman Filter, the first and second gains converge quickly, and it took a long time for the third Gains to converge. The reason for a different convergent time in the EKF is that the third Kalman Gain represents the optimal estimation for the model coefficient 'b' in the non-linear model, which is unlike three Kalman Gains that in the Kalmen Filter that all presents the linear multiplier in the estimation. This different Kalman Gains result in the distinct performances in Remaining Useful Life Prediction.

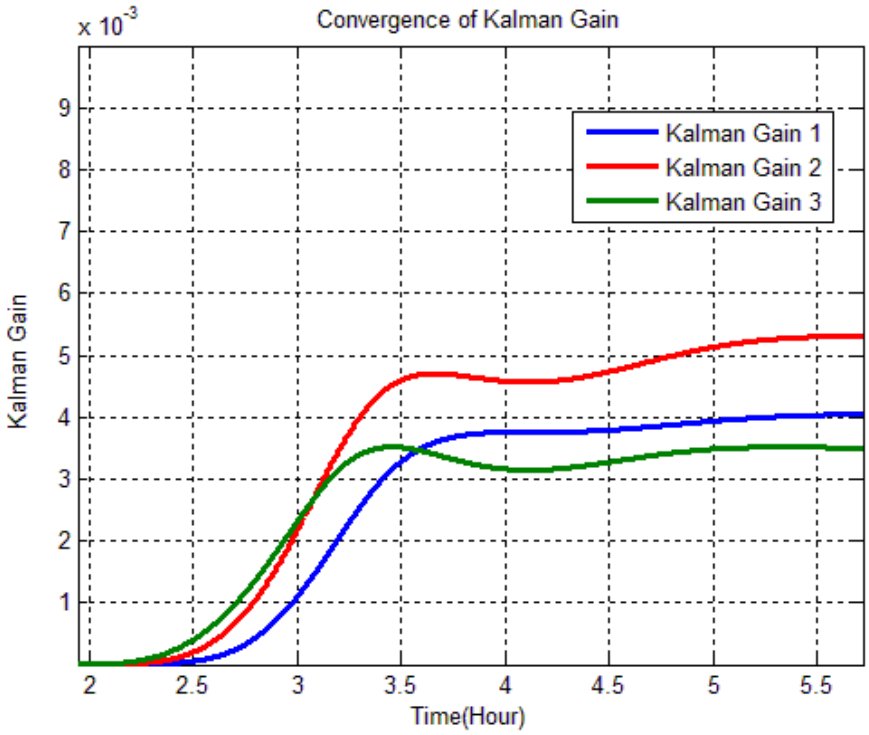


Figure 257 Kalman Gain in the KF

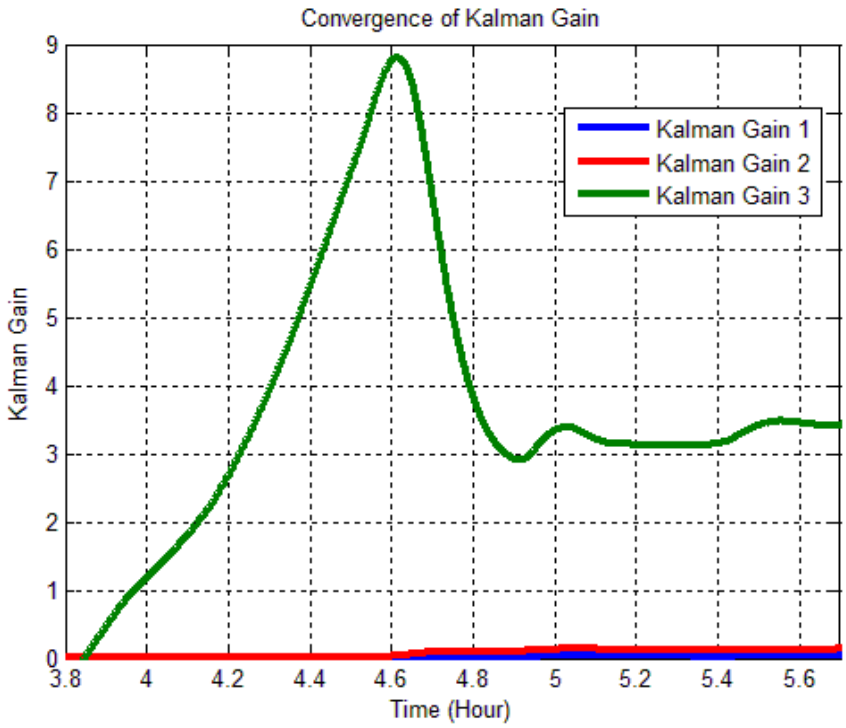


Figure 258 Kalman Gain in the EFK

In the following Figure 259 and Figure 260, the first part is the Kalman Filter prediction, and data is the resistance with experiment time. The second part is the first derivative (i.e. velocity) of the resistance in the data. The third part is the second derivative of resistance in the data. When the first derivative is negative, the Kalman Filter cannot make a prediction, because the negative velocity will cause the prediction of the resistance to drop, so there will be a decrement in the resistance. Also, the feature vector will make the predicted failure time tend to be infinite. We want the prognostic line to be accurate and less noisy, and this requires the KF to generate smooth velocity lines. In this plot, between 3 hours and 3.7 hours, the first derivative line drops under the zero line. Therefore, the prediction will be infinite and hard for us to understand.

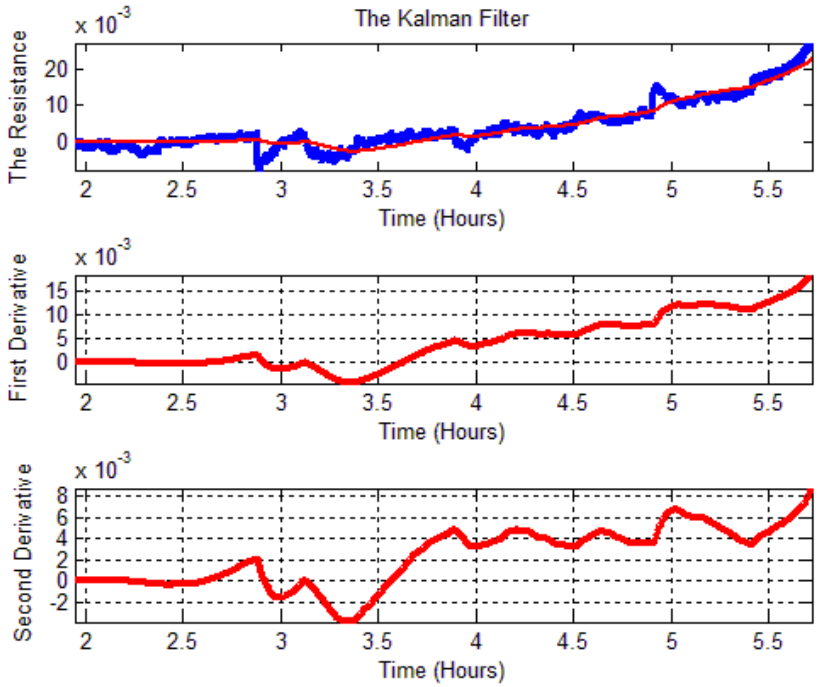


Figure 259 The Kalman Filter Performance

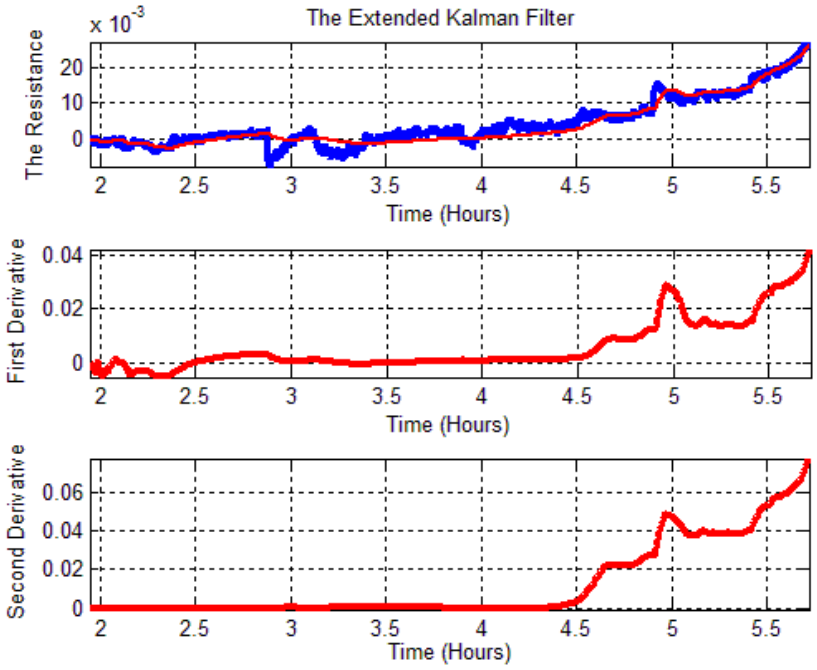


Figure 260 The Extended Kalman Filter Performance

The Newton Raphson's Method is utilized for the Prediction of RUL: when Kalman Filter made a prediction of the state vector from the dataset, and the remaining useful life could be estimated and calculated mathematically by solving the equation H(t) and find the time T-prediction:

$$H(t) = x_0 + \dot{x} \times t + \ddot{x} \times t^2 - f(EoL) \tag{228}$$

In order to find the root of above equation, we introduce the Newton Raphson's Method:

$$T(n+1) = T(n) - f(x) / f'(x) \tag{229}$$

T(n): estimated root at time n;

T(n+1): estimated root at time (n+1);

f(x)′: derivation of target equation;

The predicted RUL is known as the End of Life (EoL) minus the sampling time. So the algebra equation presents as following:

$$\mathrm{T}_{\text{predict RUL}} = \mathrm{T}_{\text{predict EoL}} - \mathrm{T}_{\text{sample}} \tag{230}$$

The real Remaining Useful Life is the time that when we finished the experiment and obtain the real End of Life, and looked back at each predicted time to check the predictions at each prediction. Similarly, the algebra expression of real RUL is:

$$\mathrm{T}_{\text{real RUL}} = \mathrm{T}_{\text{real EoL}} - \mathrm{T}_{\text{sample}} \tag{231}$$

Therefore, we can compare the predict RUL with the real RUL at each sampling time, and we can visualize two lines in the same plot, and if those lines were close, then the mode and the method we used for predicting the Remain Useful Life would be accurate.

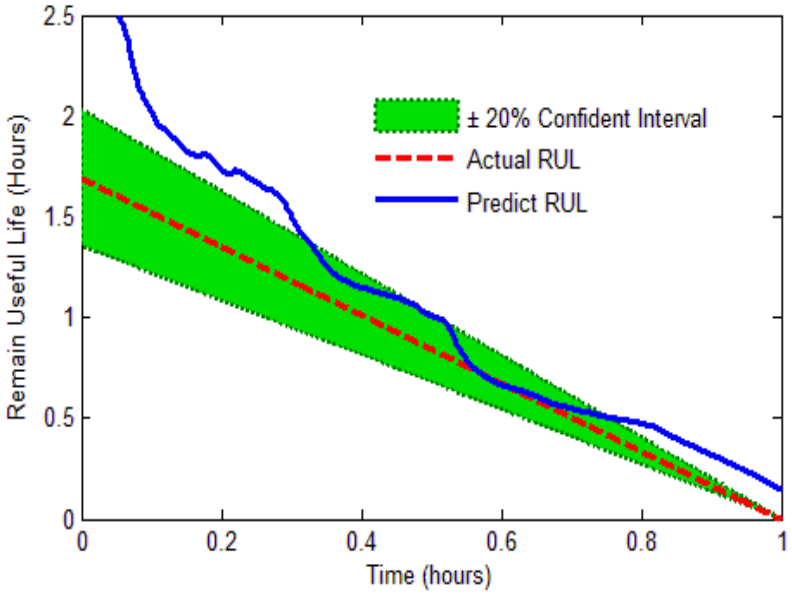


Figure 261 The RUL in Kalman Filter

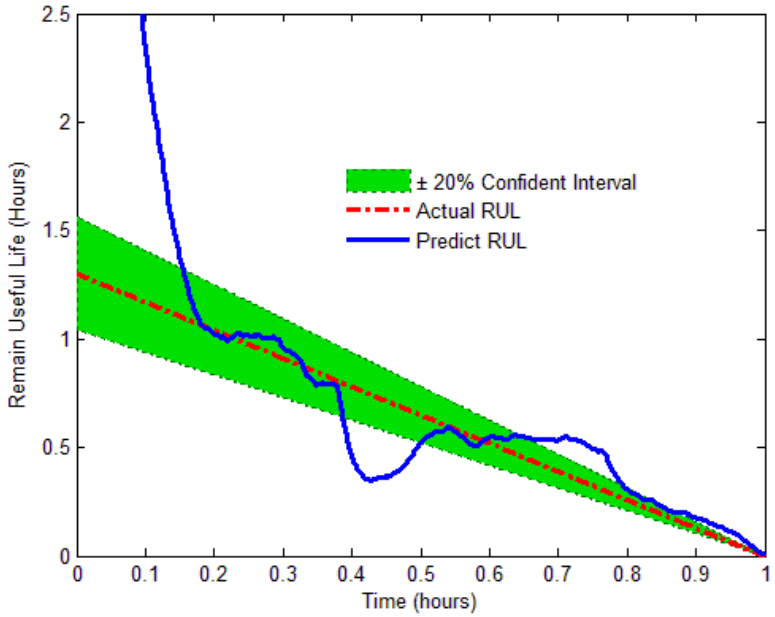


Figure 262 The RUL in the EKF

Those two plots show the prediction of Remaining Useful life using the projection vectors. The first one is from Kalman Filter, which has a smooth line. Between the normalized time 0.3

and 0.7, the predicted line is within the confident interval, and the predicted line is close and attached to the real RUL. Also, we can see that actual prediction starts earlier after almost 2 hours, which have a wide predicted band.

Compared with the KF, the EKF would be more precise on the prediction. We can clearly see in the second plot, the prediction is even close to actual RUL, which is between the normalized time 0.2 to 0.35 and 0.5 to the end. The EKF prediction is little divergent from the real RUL between 0.35 and 0.5. The prediction is down from the actual line, this because the projection vectors had sharply arisen and caused the time of predicted RUL becoming even shorter than the real RUL, and we can solve this problem by using less process noise, which make the projection vector line to become smoother. However, the prediction range will be much narrow compared with the KF, and we can see that the real prediction of the EKF starts at 1.2 hours. Therefore, we can conclude that the EKF is much better and accurate than KF, but the KF have wider predicted horizon, which enables us to make prediction earlier.

The factors that affect the performance of the KF/EKF algorithm are: (1) the basic Model; (2) the process noises; (3) the measurement noises; (4) the integrating time "Ts" in the differentiated equations. Those four elements are working together and adjustable in the algorithm for the Prognostic Health Management. The model decides our scheme to solve the problem is linear or non-linear. The process noise tells the accuracy of our model. The measurement noise contains the information about experimental noise in the dataset. The integrating time affects the solution precision in the Euler equations of the EKF.

In general, we mainly adjust the process and measurement noise to adapt the experiment. The process noise decides the information that we passed to the filter, which the higher process noise informs the filters that less precise of input model for this system. The measurement noise

tells the filters that how noisy is our dataset. In a word, the Kalman Filter and Extended Kalman Filter is a self-adapted algorithm with the implementation of process noise and measurement noise.

7.1.3 The KF and EKF Comparison

7.1.3.1 Prognostic Horizon (PH)

The definition of Prognostic Horizon(PH) is the difference between the time index j when the predictions first reach the performance criteria and the time of the End of Life (EoF).

$$\mathrm{PH} = \mathrm{EoL} - \mathrm{I} \tag{232}$$

The range of Prognostic Horizon indicates the length of RUL prediction, and the wider Prognostic Horizon is, the better performance our prediction. Also, the optimal score for Prognostic Horizon is that predictions are always within ±α bound, and the worst score is that its predictions are beyond the ±α bound. In this chapter, the Prognostic Horizon would be distinct because we use the different method to generate the RUL prediction, and one is the Kalman Filter, and the other is the Extended Kalman Filter. The better algorithm is the one has wider range of Prognostic Horizon. From the results, the Kalman Filter has a wide range of PH, which is from time 3.7 hours to the End of Life 5.7 hours. Compared with the Kalman Filter, the Extended Kalman Filter has relatively narrow PH band, which is from time 4.3 hours to the End of Life 5.7 hours. In the PH comparison, the Kalman Filter is better than the Extended Kalman Filter.

In the beta distribution comparison, the probability distributions are incorporated into the evaluation of performance metrics. Given that numerous uncertain and multiplied errors, the Confidence Interval (CI) could also be taken into consideration at each single point for the RUL. It uses the approximation of the distribution probability at each predicted point, r(k) at each point provides a set of discrete samples r(k) with their corresponding distribution probabilities. In this part, both the KF and EKF rely on the Gaussian distribution assumptions, describing uncertainty

by using the mean and variance of normally distributed r(k) prediction. However, in some cases, there are multimodal distributions, so an approximation with a mixture of Gaussians could be considered to get the distribution characteristics. The Kalman Filter uses the predicted point as the center to represent the mean of the normal distribution, and each updated value of covariance matrix is the variance of the normal distribution. From above, by incorporating enhanced visual representations, it improves the efficiency of these metrics in the performance comparison.

In addition to visual enhancements, distribution information can also be applied by calculating the probability mass of a prediction falling into the specified α-bounds using a point estimation. So this method may be a better judgment of the points that are close to the α-bound where the true prediction falls into. In the same time, we also introduce the β criterion, which states that a prediction is considered inside the α-bound only if the total probability mass of the corresponding distribution inside the bound is more than the predetermined threshold β. So this modified the previous definition of Prognostic Horizon by determining the index I as follows:

$$i = \min\left\{j \middle| (j \in l) \cap \left(\pi\left[r^l(j)\right]_{\alpha-}^{\alpha+} \geq \beta\right)\right\} \tag{233}$$

$$\pi\left[r^l(j)\right]_{\alpha-}^{\alpha+} = \int_{\alpha-}^{\alpha+} \Phi(x)dx, x \in R \tag{234}$$

$$\alpha^+ = r_* + \alpha \cdot EoL \tag{235}$$

$$\alpha^- = r_* - \alpha \cdot EoL \tag{236}$$

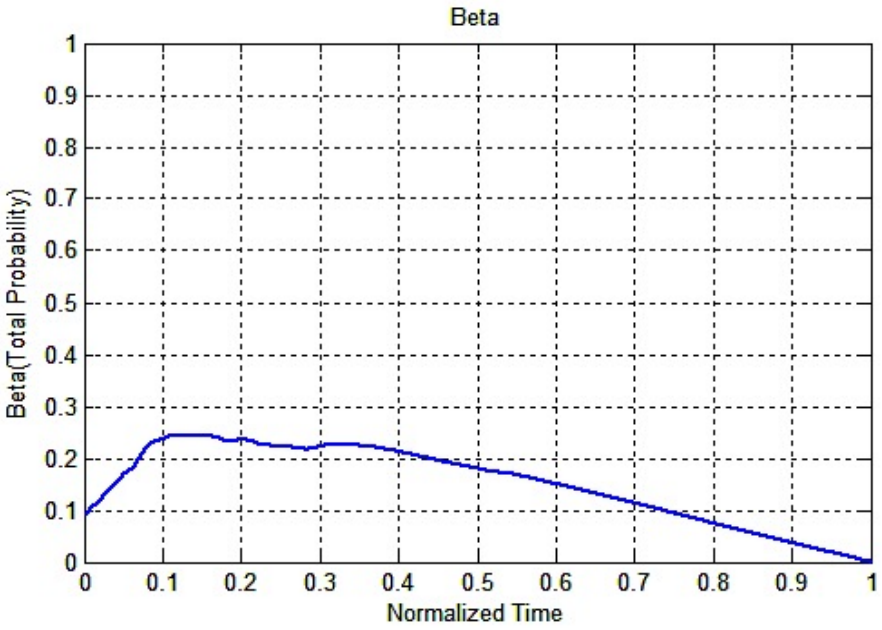


Figure 263 the value Beta of KF according to the Step Time

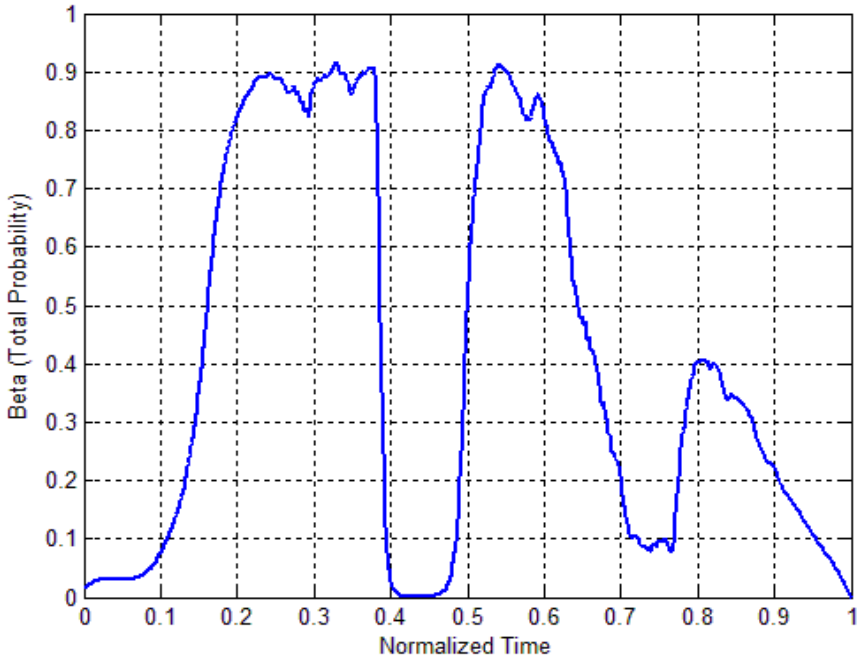


Figure 264 the value Beta of EKF according to the Step Time

The Beta Metric states the reliability and Confidence Interval of prediction. The higher value of beta, the higher reliable of the prediction is. Ideally, the value is '1', and in the KF, we can see that the beta value is low across the entire prediction horizon, and the top value is only about 2.5. However, in the EKF the, the beta value can reach as high as 0.9. Compared those two plots, the average of beta value is obviously higher in the EKF, and in the EKF beta plot, the concave between the normalized time 0.4 and 0.5 resulted from the hang-off line in the prediction by the EKF.

7.1.3.2 α+λ Performance

The α-λ performance quantifies prediction quality by determining whether the prediction falls within specified limits at particular times. Thus, it can be showed as the percentage of total RUL of the system. In general, we seek the answer whether the prediction is within 100% α of the actual RUL at specific time instance t_λ, which is expressed as a fraction of time between the point when an algorithm starts predicting and the actual failure. The following equation shows the definition of α+λ performance.

$$(1-\alpha)\cdot r_*(t) \le r^l(t_\lambda) \le (1+\alpha)\cdot r_*(t) \qquad (237)$$

Where: α is the accuracy modifier;

λ is a time window modifier such:

$$t_\lambda = t_p + \lambda\cdot(EoL - t_p);$$

The α-λ performance metrics provide us an intuition including several aspects: first of all, it visually summarizes the performance of an algorithm over a length of time in a very concise manner, and the α-λ plot could decide to combine outcomes of distinct algorithms in different stages of the system/component life. Secondly, the α-λ metric has an ability to establish performance limits for various algorithms. We can apply a fixed value β, and compute the best

value of α for each algorithm to assess what values may be more reliable for the performance requirements. It can be known that the requirements of large value β can be more restricted on precision of the prediction.

7.1.3.3 Relative Accuracy (RA):

The Relative Accuracy (RA) is the quantity describing the errors of the RUL prediction, at a given time, the error of the prediction can be calculated and normalized by the true value. The range of time is described as a fraction between 0 and 1. An algorithm with higher Relative Accuracy is always expected. Particularly, it gives us an idea about the accuracy of prediction before the End of Life. The mathematic formula for calculating the Relative Accuracy is listed below:

$$RA_{\lambda} = 1 - \frac{\left| r_*(t_{\lambda}) - r^{l}(t_{\lambda}) \right|}{r_*(t_{\lambda})} \tag{238}$$

$$t_{\lambda} = t_p + \lambda \cdot (EoL - t_p); \tag{239}$$

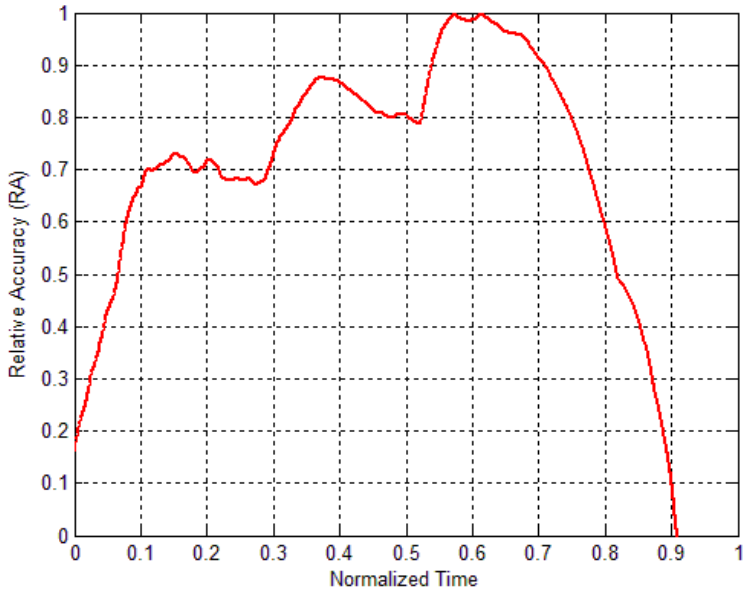


Figure 265 The Relative Accuracy in the KF

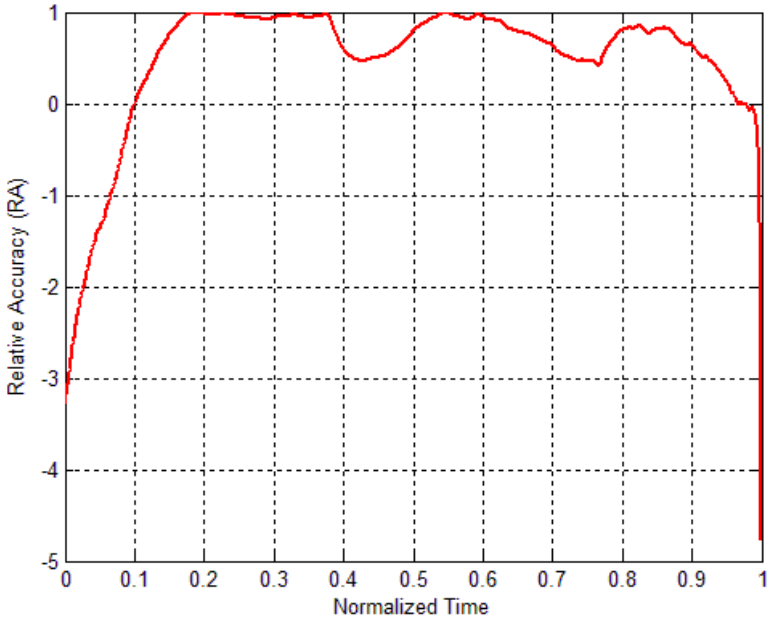


Figure 266 The Relative Accuracy in the EKF

In above Figure 255 and Figure 256, the optimal value of the Relative Accuracy reaches '1'. We can see that the EKF's Relative Accuracy is higher than the KF. Also at the end of the Prediction Horizon, the EKF still maintains at high value of the Relative Accuracy. In a word, the EKF provides more reliable and accuracy predictions.

7.1.3.4 Convergence

The Convergence is used to quantify a behavior, in which any metric like accuracy or precision improves with time to reach its perfect score. The Convergence is a meta-metric in a sense, and it is computed from other performance metrics in prognostic results. It provides even further information that how fast the algorithm learns and improves with respect to a specified metric as more and more values are available. The distance between the origin and the central area under the precision metric qualifies convergence. A low distance means fast convergence. Let (X_c, Y_c) be the center of mass of area under the curve $M(i)$, where $M(i)$ is non-negative prediction

error metric. The Convergence can be represented by C_M , the Euclidean Distance between the center of mass and$(t_p, 0)$. The equation to calculate the Convergence shows below:

$$C_M = \sqrt{(X_c - T_p)^2 + y_c} \tag{240}$$

$$X_c = \frac{\frac{1}{2}\sum_{i=p}^{EoP}(t_{i+1}^2 - t_i^2) \times M(i)}{\sum_{i=p}^{EoP}(t_{i+1} - t_i) \times M(i)} \tag{241}$$

$$Y_c = \frac{\frac{1}{2}\sum_{i=p}^{EoP}(t_{i+1}^2 - t_i^2) \times M(i)}{\sum_{i=p}^{EoP}(t_{i+1} - t_i) \times M(i)}^2 \tag{242}$$

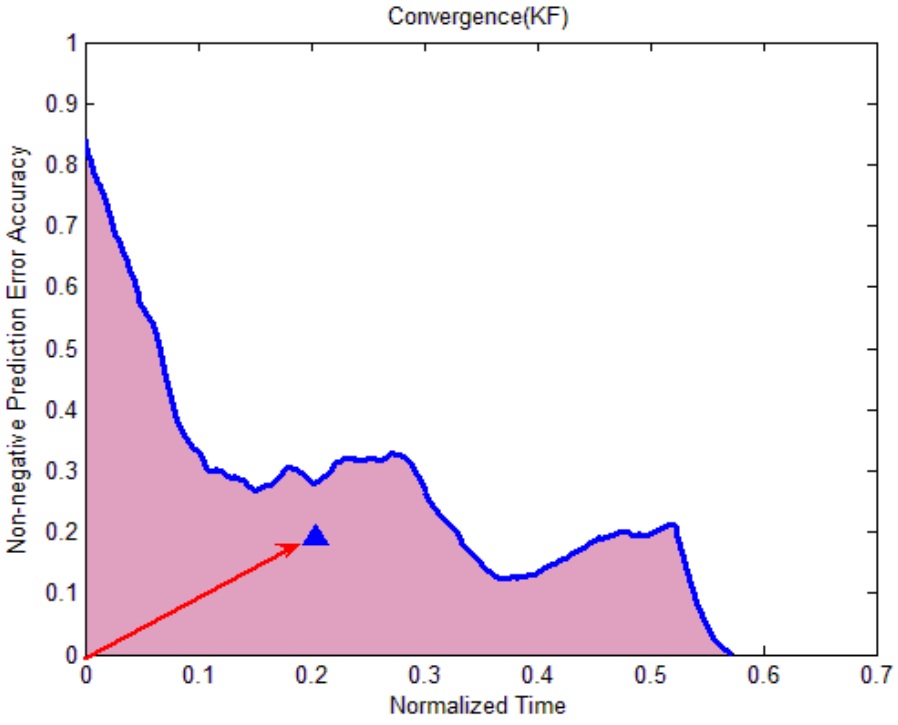


Figure 267 Kalman Filter Convergence

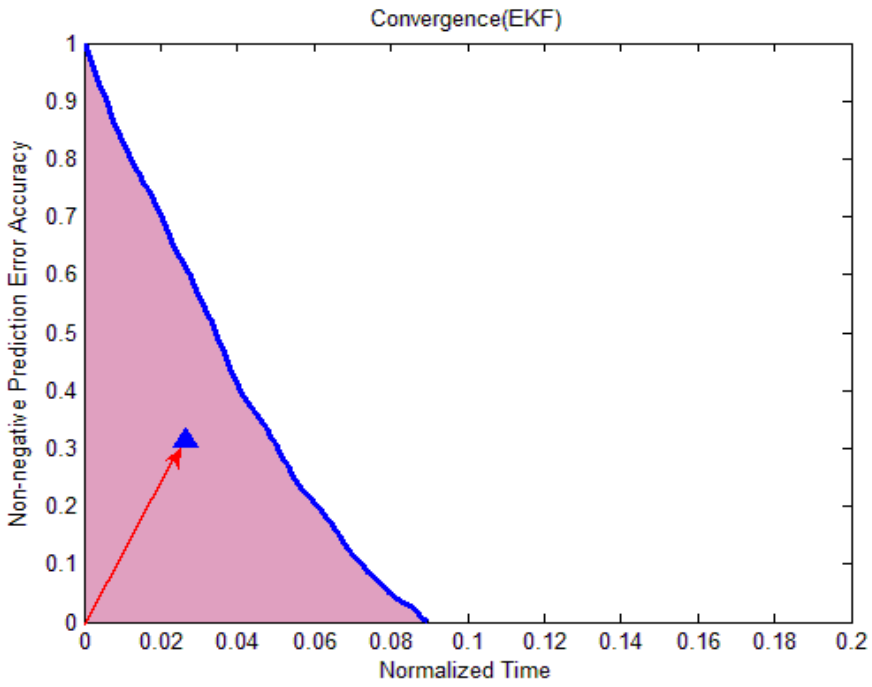


Figure 268 Extended Kalman Filter Convergence

Obviously, the Convergence value of the EKF is great than that in the KF, so the EKF converges faster than the KF. We can see it in the above plots, where the KF converges at normalized time '0.2', and the EKF converges almost at time '0.02'. Therefore, we can conclude that the EKF has an advantage to quickly find its optimal prediction.

7.1.4 Conclusions

A framework for Prognostic Health Management (PHM) of electronics has been based on the state-space vectors from the Resistance Spectroscopy (RS) measurements. The critical and threshold value of electronic resistance has been used to calculate and detect the Remaining Useful Life (RUL) of the test board. In both the Kalman Filter and the Extended Kalman Filter, the extrapolation of future feature vector has been used to predict the Time-to-Failure (TTF), in which the feature vector will cross the failure threshold. The Kalman Filter and the Extended Kalman

Filter will update the feature vector at every step using the recursive algorithm. Therefore, the better and the more precise feature vectors, the better predictions of filters will be, and in this chapter, the performances of the Kalman Filter and the Extended Kalman Filter were compared. It has already known that the Kalman Filter deals with the linear system problem, and the Extended Kalman Filter deals the non-linear problem. When applied those two methods in the RUL prediction in the BGA board failure test, we have got some accurate results. During the comparison, the Extended Kalman Filter is better than the Kalman filter in the RUL prediction. The Kalman Filter is inertia to the rapid change of the resistance, but the Extended Kalman Filter could adapt to it. However, the Kalman Filter has smoother prediction line than the Extended Kalman Filter, so the Kalman Filter deals with the system's noise better, and the Extended Kalman Filter is more precise to predict the value. The Average Relative Accuracy for the EKF is higher than the KF. Beyond that, the Kalman Filter also has some advantages in the implementation. The program could be much less expensive and runs efficiently. The Extended Kalman Filter is computational expensive compared to the KF. The Kalman Filter does not require us to know much about the system model, but in the Extended Kalman Filter if the system model is not true, the large compensation of process noise would be added to it to make a right prediction.

7.2 Bayesian Linear Regression

The Bayesian Regression has been widely used in the statistical regression analysis, and in this chapter, we introduce the Bayesian Regression algorithm for tracking and estimating the Remaining Useful Life of electronics. For the basic Bayesian Regression problem, we firstly introduce the posterior probability that is given by the condition probability:

$$p(C_k \mid X) = \frac{p(X \mid C_k)p(C_k)}{P(X)} \tag{243}$$

C_k: Category k

X: Samples

$p(C_k \mid X)$ is the normalized posterior that is same definition as it defined in the classification problem. $p(C_k)$ is the prior distribution. In the problem of regression, we must use the probability $P(X)$ by the summation over different groups. The summation definition shows as the following equation:

$$P(X) = \sum p(X \mid C_k)p(C_k) \tag{244}$$

Because the Bayesian Linear Regression applies quadratic model, the matrix presentation has been naturally introduced to simplify the problem. Suppose the real-valued input variable column vector is the form of following:

$$X = [x_1 \quad x_2 \quad \ldots \quad x_n]^T \tag{245}$$

x_i: input data

The real-valued predict target column vector is the following:

$$T = [t_1 \quad t_2 \quad \ldots \quad t_n]^T \tag{246}$$

In this chapter, we can fit the data with fourth order polynomial function with its synaptic weights. The basis function is a 1×M matrix, which can be shown as the following:

$$\Phi(x) = [1 \quad x \quad x^2 \quad ... \quad x^M]^T \tag{247}$$

The synaptic weights M×1column matrix is:

$$W = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ ... \\ w_M \end{bmatrix} \tag{248}$$

Therefore, the estimation matrix can be calculated as the following matrix multiplication:

$$y_i = W^T \cdot \Phi(x_i) \tag{249}$$

Then, the estimation column matrix can be represented as:

$$\mathrm{Y} = [y_1 \quad y_2 \quad ... \quad y_n]^T \tag{250}$$

In the Bayesian Curve-fitting problem, the likelihood function takes the Gaussian probability density equation, which is the condition probability over the input variables and synaptic weights as well as noise precision parameter β.

$$P(t \mid x, W, \beta) = N(t \mid W^T \cdot \Phi(x_i), \beta^{-1} I) \tag{251}$$

Then, we take the product of all the likelihood functions over the input variables:

$$P(T \mid X, W, \beta) = \prod_{i=1}^{n} N(t \mid W^T \cdot \Phi(x_i), \beta^{-1} I) \tag{252}$$

Now, we can maximize the above equation with respect to the distribution precision parameter β, which applies the Maximum Likelihood Estimation (MLE) for the distribution variance:

$$\frac{1}{\beta_{ML}} = \frac{1}{N} \sum_{i=1}^{\mathrm{N}} (y_i - t_i)^2 = \frac{1}{N} \cdot (Y - T) \cdot (Y - T)^T \tag{253}$$

N: Number of Samples

The above result gives the reciprocate variance that yields maximum probability of likelihood function. We can substitute the β_{ML} into $P(T \mid X, W, \beta)$, which gives:

$$P(T \mid X, W, \beta) = \prod_{i=1}^{n} N(t \mid W^T \cdot \Phi(x_i), \beta_{ML}^{-1}) \tag{254}$$

Then, we introduce the Bayesian conjugate prior Gaussian probability in the regression as following:

$$P(W \mid \alpha) = N(W \mid M_0, S_0) = N(W \mid 0, \alpha^{-1} I) \tag{255}$$

The α is the precision parameter of the distribution

M_0: Distribution Mean

S_0: Distribution Variance

Hence, we can write the Bayesian posterior function as:

$$\begin{aligned} &P(W \mid X, T, \alpha, \beta) \propto P(T \mid X, W, \beta) \cdot P(W \mid \alpha) \\ &= N(W \mid M_N, S_N) \\ &M_N = S_N(S_0^{-1} M_0 + \beta \Phi^T T) \\ &S_N^{-1} = S_0^{-1} + \beta \Phi^T \Phi \end{aligned} \tag{256}$$

M_N: Posterior Mean

S_N: Posterior Variance

Once, we can also apply the sum and product rule of probability, thus, it can allow us to write the predictive distribution to be:

$$P(t \mid \mathrm{T}, \alpha, \beta) = \int P(t \mid W, \beta) \cdot P(W \mid T, \alpha, \beta) dW \tag{257}$$

The condition distribution $P(t \mid \mathrm{T}, \alpha, \beta)$ can be calculated out as the distribution and probability with its mean and variance depending on the variable 'x'; Therefore, we can finally predict each output 't' at each time series input 'x', which iterates to the beginning until the convergence of posterior distribution.

$$P(t \mid x, \mathrm{T}, \alpha, \beta) = N(t \mid M_N^T \Phi(x), \beta^{-1} + \Phi^T(x) S_N \Phi(x)) \quad (258)$$

Algorithm Pseudo Code: Bayesian Linear Regression

(1) Initial the prior distribution $P(W \mid \alpha)$, mean M_0 and variance S_0, the likelihood function variance β.
(2) Calculate likelihood distribution $P(T \mid X, W, \beta)$.
(3) Calculate posterior distribution $P(W \mid X, T, \alpha, \beta)$, and its mean M_N and variance S_N^{-1}.
(4) Update the β using MLE method
(5) Calculate prediction distribution $P(t \mid \mathrm{T}, \alpha, \beta)$ and predictive mean $M_N^T \Phi(x)$ and predictive variance $\beta^{-1} + \Phi^T(x) S_N \Phi(x)$
(6) Iterate to Step 2 until the convergence of posterior distribution.

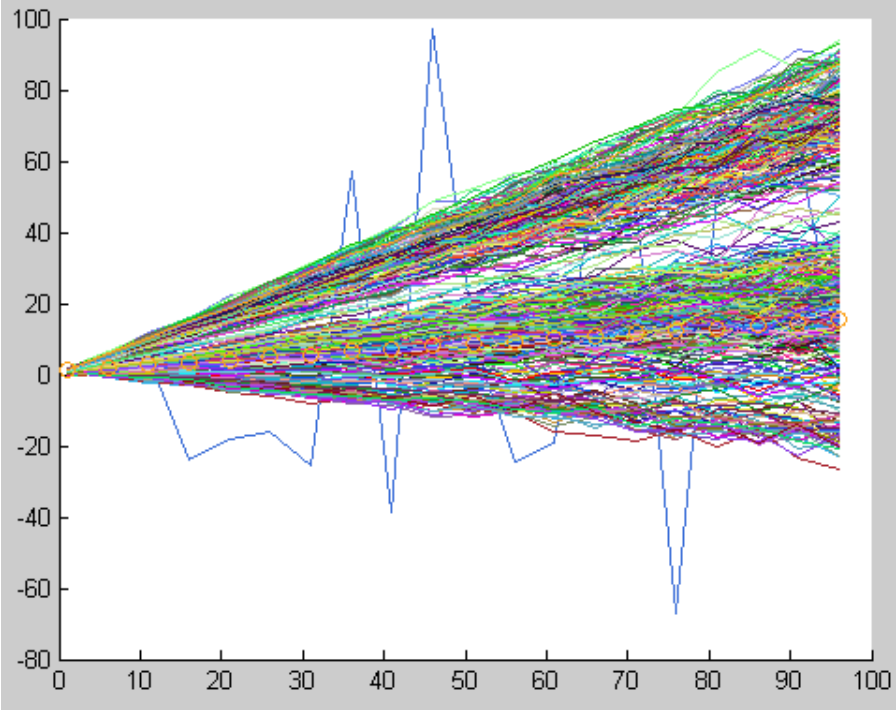


Figure 269 Posterior Estimation

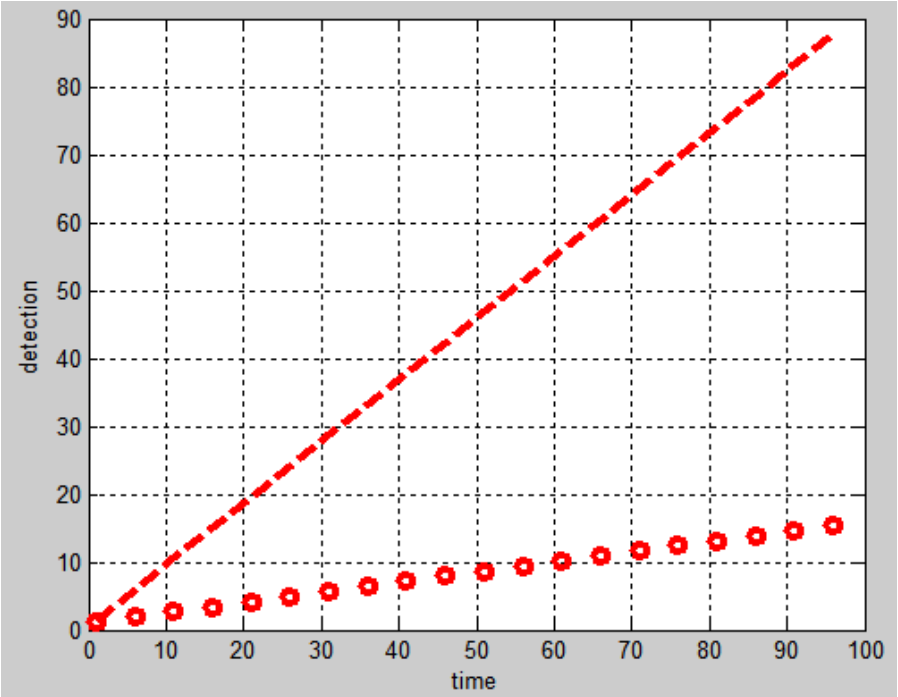


Figure 270 Bayesian Regression 1000 Iterations

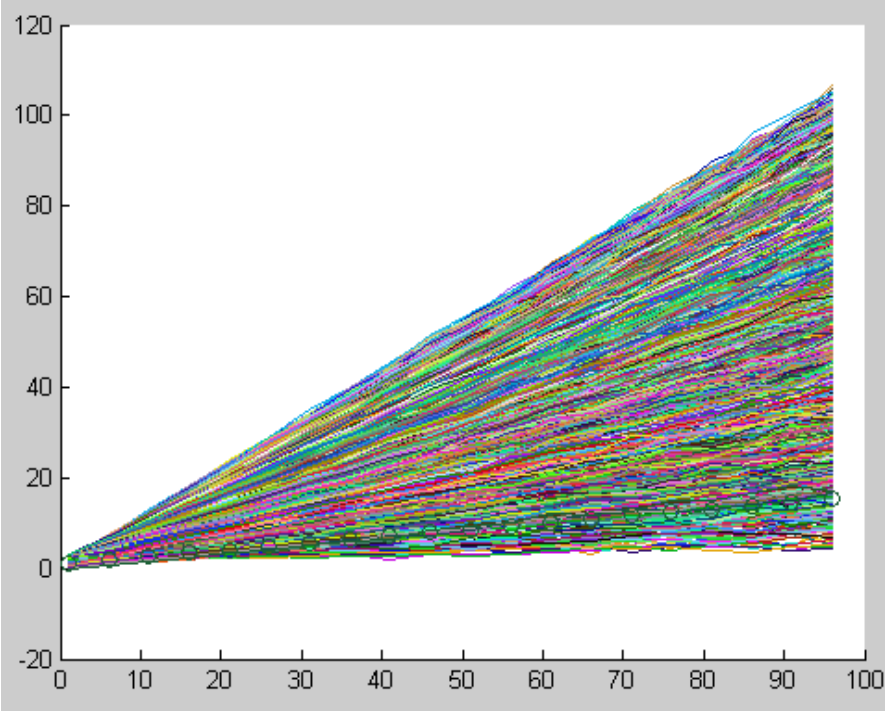


Figure 271 Bayesian Regression Posterior Estimation

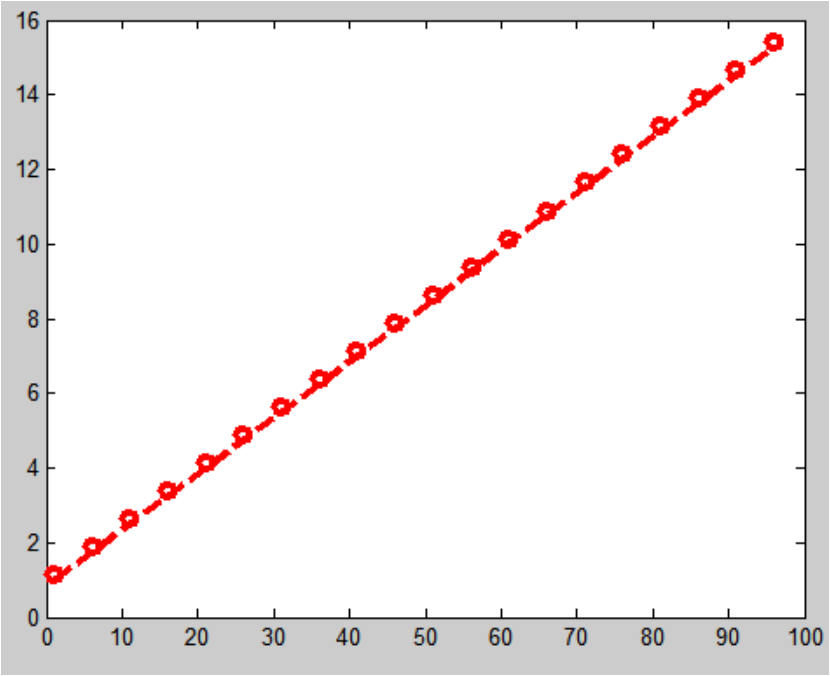


Figure 272 Bayesian Regression 10000 Iterations

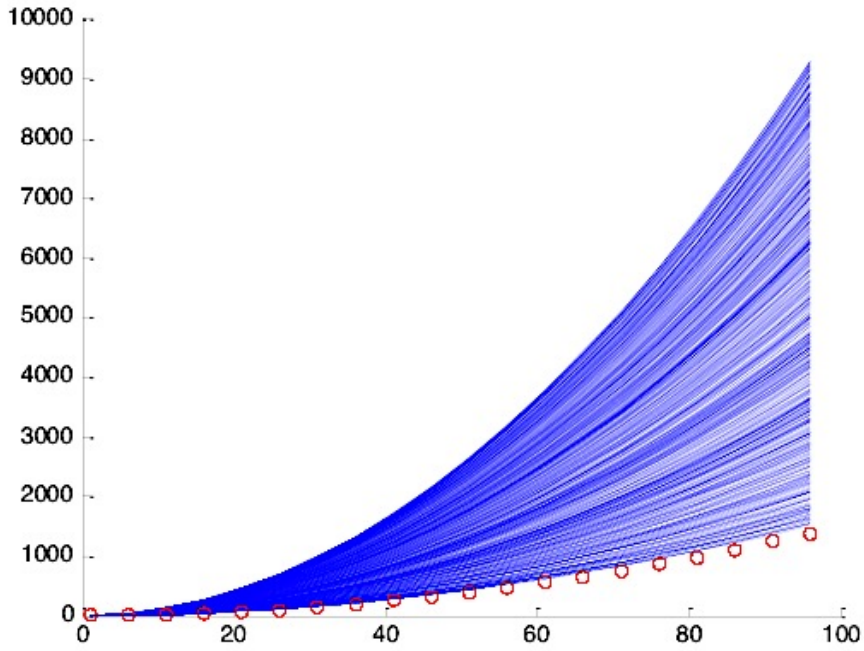


Figure 273 Bayesian Posterior Estimation

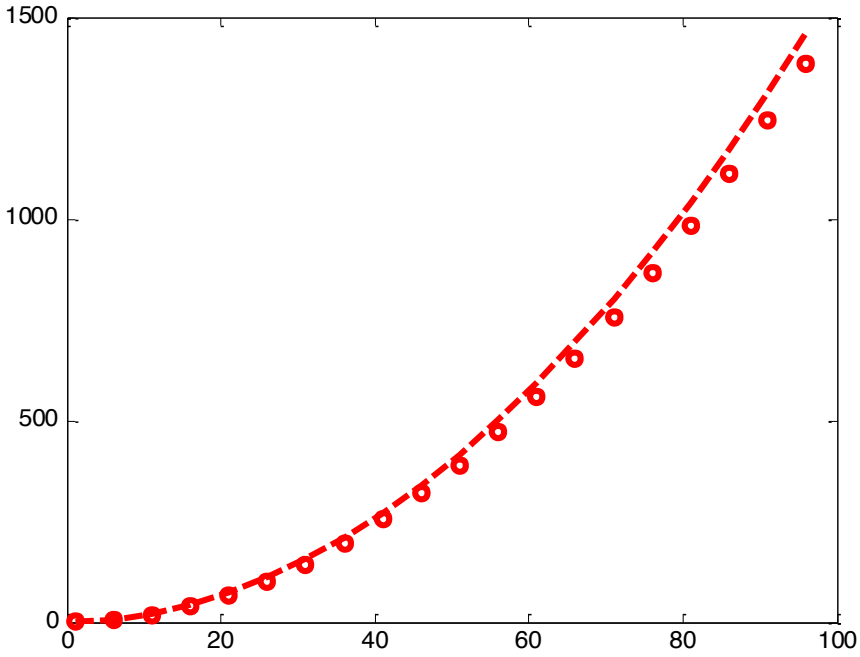


Figure 274 Bayesian Polynomial Fitting

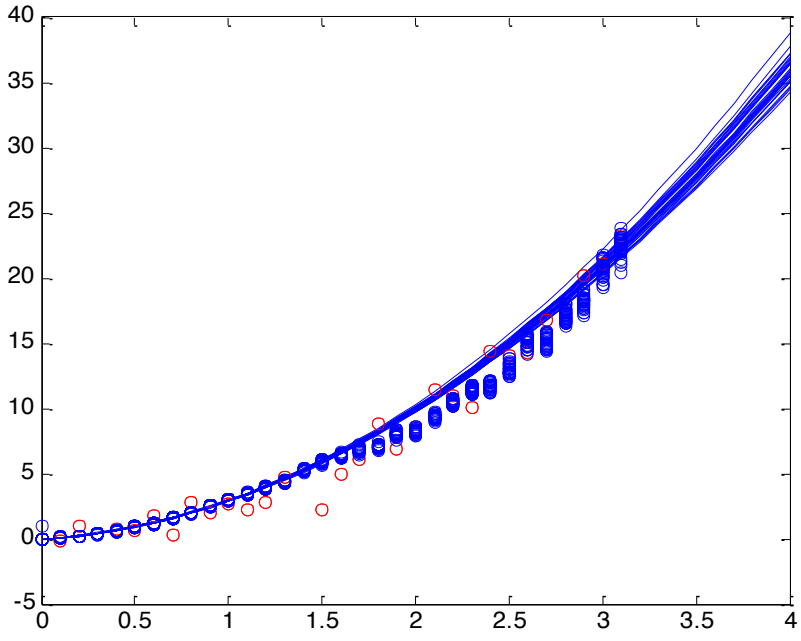


Figure 275 Bayesian Regression for Tracking Polynomial Curve

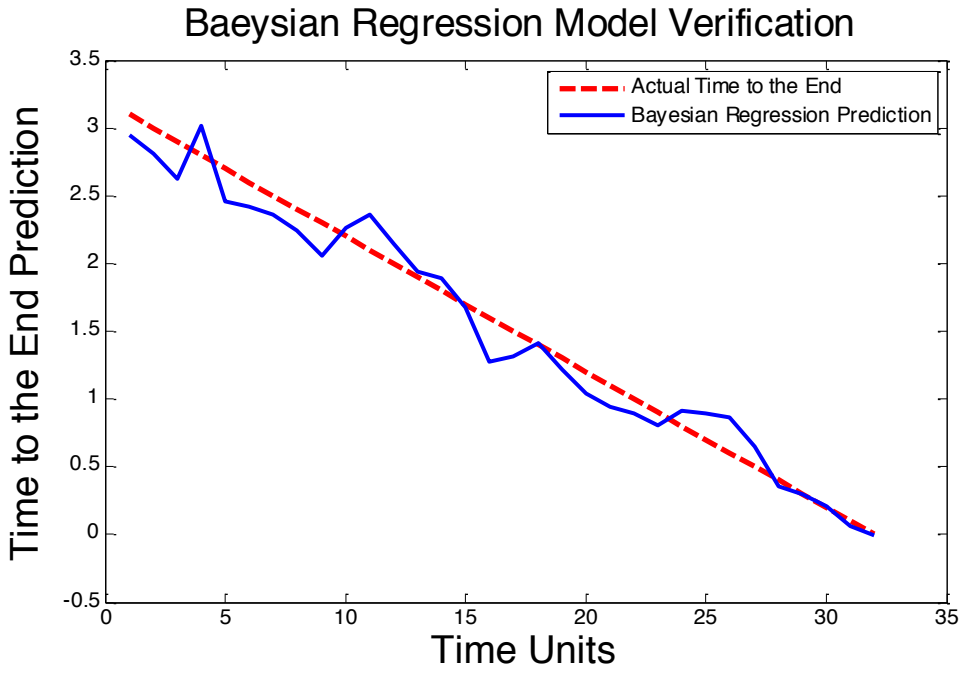


Figure 276 Bayesian PHM Framework Verification for Polynomial Curve

7.3 Classification Models

7.3.1 K-means Classification

The K-means Classification is the method that has been used to differentiate the failure group to the pristine group in the electronics failure analysis. The K-mean assumes we already know how many clusters in advance. Therefore, we can assign mean of each cluster and update the mean of cluster during the computation. In this chapter, we assume that weight variable a_{km}, which shows 'k' data belongs to the group 'm'. Thus, for the data k and group m, a_{km} is one, all remaining elements are zero.

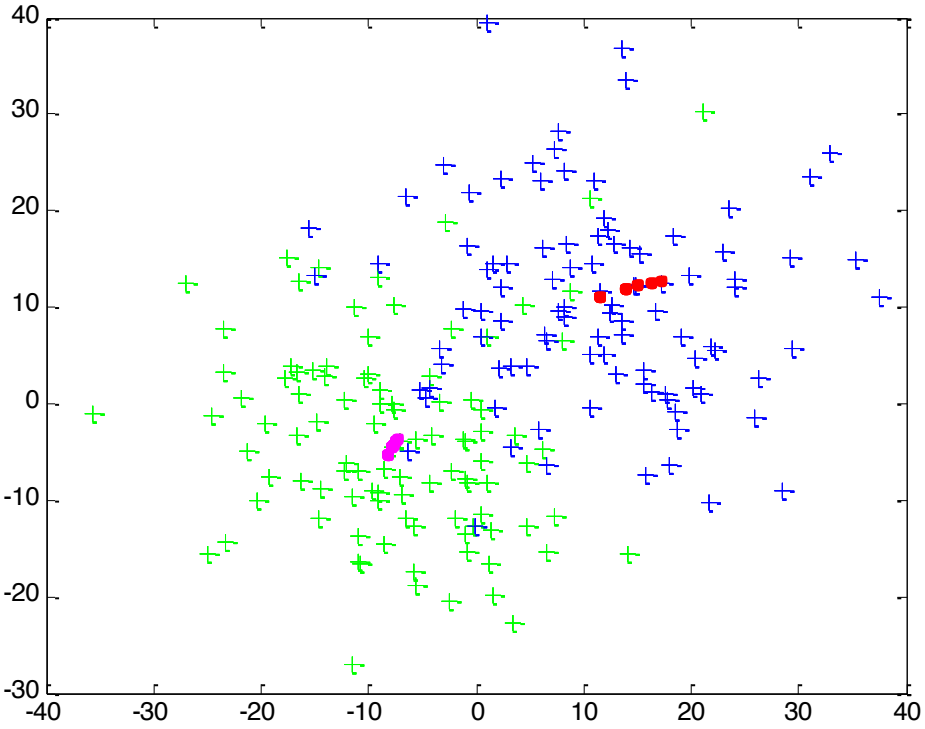


Figure 277 The K-mean Classification

The K-means clustering uses an iterative scheme to minimize the following object function:

$$J = \sum_{k=1}^{K} \sum_{m=1}^{M} a_{km} \|y_k - u_m\|^2 \tag{259}$$

y_k: input data k

u_m: mean of group m

We will iterate the following two steps until the means of two groups are converged: minimizing function J with respect to a_{km} , keeping u_m fixed by assigning the k-th data point to the closest cluster center:

$$a_{km} = \begin{cases} 1 \text{ if } m = \operatorname{argmin} \|y_k - u_m\|^2 \\ 0 \text{ otherwise} \end{cases} \tag{260}$$

The new mean of group has been updated using the weighted data in the group:

$$u_m = \frac{\sum_k a_{km} \mathbf{y_k}}{\sum_k a_{km}} \tag{261}$$

Each time, we are minimizing the J function. Therefore, the converged above terms are guaranteed.

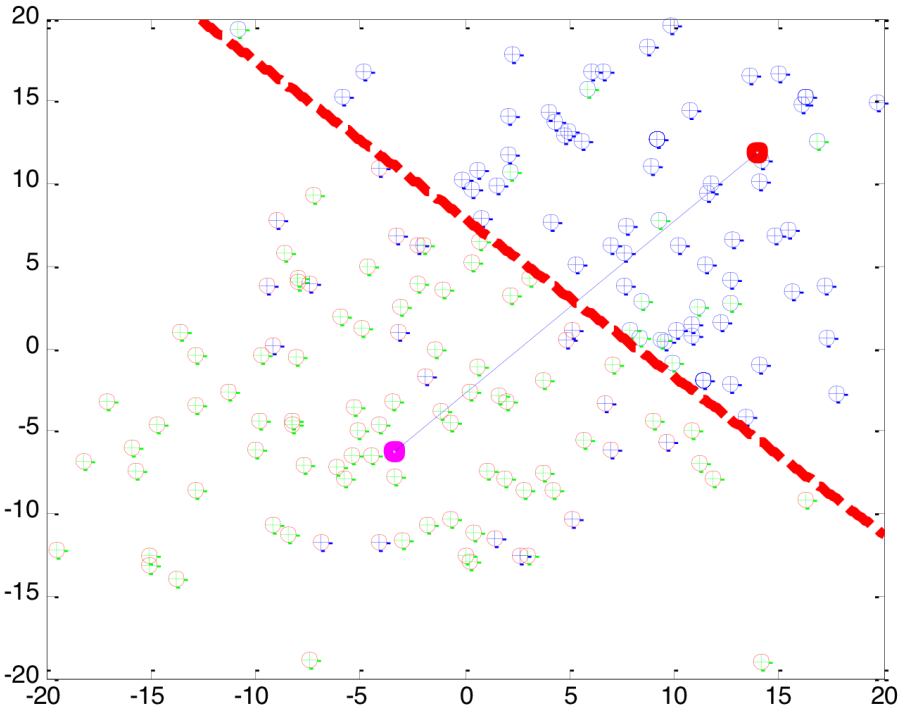


Figure 278 K-Mean Classification

7.3.2 Expectation Maximization Classification

The Expectation Maximization(EM) clustering formulates the clustering problem with respect to the mixed Gaussian mixture with M components:

$$P(y) = \sum_{m=1}^{M} A_m N(y|u_{m,} \sum m) \tag{262}$$

A_m: m group weight

$u_{m,}$: group means

m: number of groups

In the EM algorithm, the multinomial probability distribution:

$$P(a_k|A) = \prod_{m=1}^{M} N(y_k|u_m, \sum m) \tag{263}$$

The data likelihood function is:

$$P(y|u, \sum, A) = \prod_{k=1}^{K} \sum_{m=1}^{M} A_m N(y_k|u_{m,} \sum m) \tag{264}$$

In the E step:

$$w_k = \frac{A_m N(y_k|u_m, \sum m)}{\sum_{j=1}^{M} A_j N(y_k|u_{j,} \sum_j)} \tag{265}$$

In the M step:

$$u_m = \frac{\sum_{k=1}^{K} w_{km} y_k}{\sum_{k=1}^{K} w_{km}} \tag{266}$$

$$\sum m = \frac{\sum_{K=1}^{w} w_{km} (y_k - u_m)(y_k - u_m)^T}{\sum_{k=1}^{K} w_{km}} \tag{267}$$

$$A_m = \frac{\sum_{K=1}^{K} w_{km}}{K} \tag{268}$$

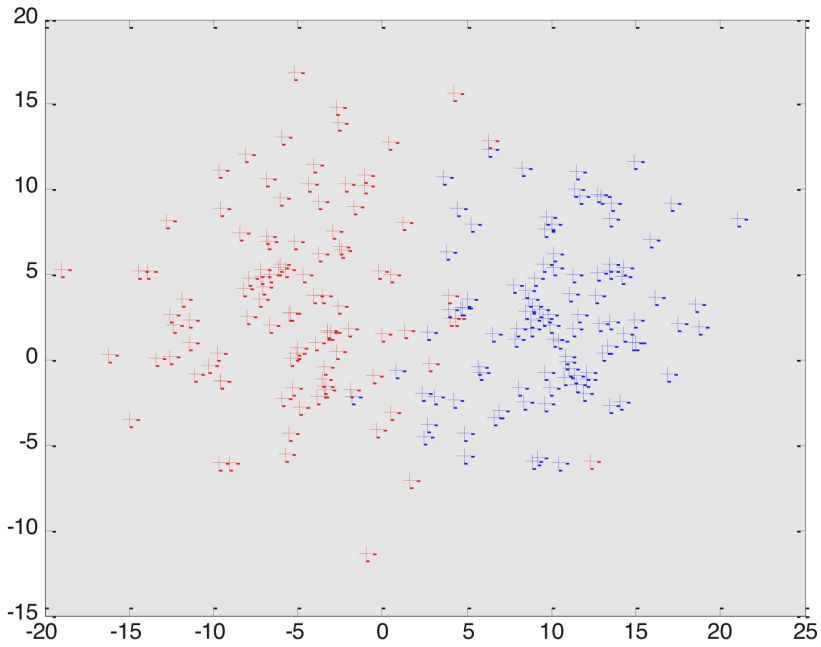


Figure 279 Simulated Two Groups

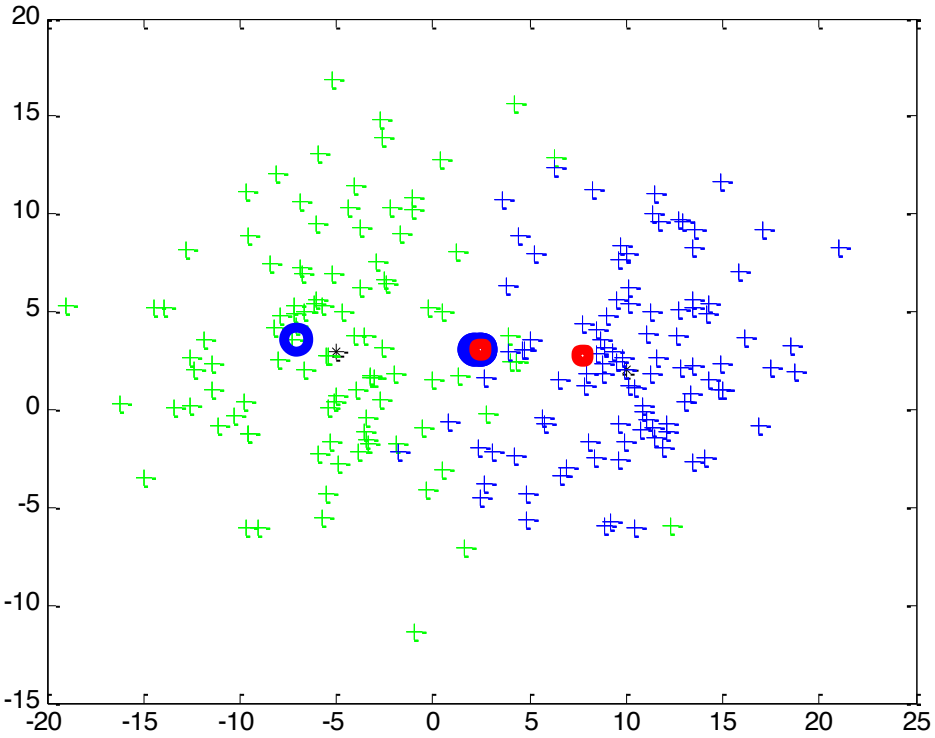


Figure 280 The EM Clustering Centers

7.3.3 Bayesian Classifier

In this chapter, we introduce the Bayesian Probabilistic Generative Models (Bishop, 2006) for the classification problem. The goal of the Bayesian Classification is to find the update posterior distribution for each group given the continuous input data. From the Bayes' Theorem, we need to model the class conditional density $p(X|C_k)$ which is a assumption from the Gaussian distribution and the class prior $p(C_k)$. Following by the Gaussian model assumption, the Gaussian likelihood function for continuous input data shows below:

$$p(X \mid C_k) = \frac{1}{(2\pi)^{1/2}} \cdot \frac{1}{|\Sigma_k|} \cdot \exp(-\frac{1}{2}(X - \mu_k)^T \Sigma_k^{-1}(X - \mu_k)) \tag{269}$$

The class prior $p(C_k)$ utilizes the weighted group form as:

$$p(C_k) = N_k / N \tag{270}$$

$$p(C_i) / p(C_j) = N_i / N_j \tag{271}$$

The N_k is the number of samples in the 'k' group. The N is the total numbers of sample. The weighted posterior for classifying the 'k' group is the ratio between its posterior and the sum posterior from all the groups:

$$p(C_k \mid X) = \frac{p(X \mid C_k)p(C_k)}{\sum_j p(X \mid C_j)p(C_j)} \tag{272}$$

The activate function for the multiple-classification is defined as:

$$a_k = \ln p(X \mid C_k)p(C_k) \tag{273}$$

Typically, for the two classes problem, we introduce the Logistic Sigmoid activate function defined by:

$$p(C_1 \mid X) = \frac{p(X \mid C_1)p(C_1)}{\sum_{j=1}^{2} p(X \mid C_j)p(C_j)} = \frac{1}{1 + e^{-a}} \tag{274}$$

Where the coefficient 'a' is given by:

$$a = \ln \frac{p(X \mid C_1)p(C_1)}{p(X \mid C_2)p(C_2)} \tag{275}$$

The decision boundary can be calculated through the equating the activate function or equating the posterior likelihood from above:

$$a_k = a_j \tag{276}$$

$$p(X \mid C_k)p(C_k) = p(X \mid C_j)p(C_j) \tag{277}$$

Then, we can take the logarithm for the Gaussian posterior distributions and find out the decision boundary as following quadratic form:

$$a_{kj} = X^T \cdot W_{kj}^1 \cdot X + W_{kj}^2 \cdot X + W_{kj}^3 \tag{278}$$

Where the quadratic decision boundary coefficients are solved as:

$$W_{kj}^1 = -\frac{1}{2}\left(\Sigma_i^{-1} - \Sigma_j^{-1}\right) \tag{279}$$

$$W_{kj}^2 = \left(\Sigma_k^{-1} \cdot \mu_k - \Sigma_j^{-1} \cdot \mu_j\right) \tag{280}$$

$$W_{kj}^3 = -\frac{1}{2}\left(\mu_k^T \Sigma_k^{-1} \mu_k - \mu_j^T \Sigma_j^{-1} \mu_j\right) - \frac{1}{2}\left(\log|\Sigma_k| - \log|\Sigma_j|\right) + \log\frac{p(C_k)}{p(C_j)} \tag{281}$$

μ_i: mean for group i

Σ_i^{-1}: varaince for group i

C_i: sample in the group i

W_{ij}: the weight of polynomial to separate group i and j

In order to solve the distribution coefficients, it is common and convenient to use the Maximum Likelihood Estimation (MLE) method to find out the parametric distribution that yields the maximum product of posterior. Therefore, we can solve this classification problem analytically and numerically. The product of group's posterior can be written as:

$$\prod p(X, C_i) = \prod p(X \mid C_i) p(C_i) = \prod \frac{N_i \cdot \mathrm{N}(X_i \mid \mu_i, \Sigma_i)}{N} \tag{282}$$

The overall group's posterior product can be thereby written as:

$$p(t, C_i, \mu_i, \Sigma_i) = \prod_{n=1}^{n} [\frac{N_i \cdot \mathrm{N}(X_i \mid \mu_i, \Sigma_i)}{N}]^{T_i} \tag{283}$$

Where $T_i = 1$ if $X_i \subset C_i$

Then, we can take derivative of the above equation with respect to C_i, μ_i, $\sum_i$, and then we can solve those equations to get the value for them. The expression solutions are given directly by the following equation:

$$C_i = \frac{N_i}{N} \tag{284}$$

Where N_i is the number of data in the i group, N is the summation of all the numbers in each group. The Gaussian group's mean is given by the following:

$$\mu_i = \frac{1}{N_i} \sum_{n=1}^{N} T_i \cdot X_i \tag{285}$$

The Gaussian Covariance is given below:

$$\Sigma_i = \frac{1}{N_i} \sum_{n \subset C_i} (X_n - \mu_i) \cdot (X_n - \mu_i)^T \tag{286}$$

The synaptic weights have been used to describe the distinguished classes. In the Bayesian classification method, the trained weights can be found as the mean value of posterior distribution. The Probability Density Functions(PDFs) of two groups can be derived from synaptic weights as listed form below:

$$F(x, y) = I^T W_2 I + W_1^T I + W_0 \tag{287}$$

$$I = \begin{bmatrix} x \\ y \end{bmatrix} \tag{288}$$

$$\begin{aligned} W_2 &= -\frac{1}{2}\Sigma_i^{-1} \\ W_1 &= \Sigma_i^{-1}\mu_i \\ W_0 &= -\frac{1}{2}\mu_i^T\Sigma_i^{-1}\mu_i - \frac{1}{2}\log|\Sigma_i| + \log(p(C_i)) \end{aligned} \tag{289}$$

We can expand the matrix expression of $F(x, y)$ as the quadratic area function:

$$\begin{aligned} &F(x,y) = W_2(1,1)x^2 + 2W_2(1,2)xy + W_2(2,2)y^2 \\ &+ W_1(1,1)x + W_1(2,1)y + W_0 \end{aligned} \tag{290}$$

7.4 Introduction to LED System RUL Prognostication Using KF/EKF

In this chapter, we cover some applications of the KF/EKF and Bayesian Method in the LEDs system, and we firstly focus on the application of Kalman Filter and Extended Kalman Filter Algorithm into the prediction of LED 70% Lumen Maintenance Life (L70), and the former prediction method, TM-21 calculator, can roughly make a prediction between average 20,000 and 36,000 hours for the bare LED's lifetime. The TM-21 calculator is Curve-Fitting method basing on the Arrhenius Equation, which needs to calculate the Activation Energy, Pre-decay factor, and Decay Rates. Also, the TM21 is also overall prediction related to the knowledge of upper and lower temperature cases. However, the Activation Energy may not be obvious and easily to be obtained. Thus, we can introduce substitutable prediction method: the KF/EKF prognostic method. It can dynamically update the prognostic L70 lifetime at each time of interest according to the system prognostic model. The KF/EKF algorithm bases on the identified system measurement noise and system process noise. The linear, polynomial, exponential stochastic models for predicting LEDs' life are discussed in this chapter, and each model has been incorporated into the KF/EKF algorithm to generate the L70 estimations. The Normal, Lognormal, and the Weibull

distributions have been used to fit those estimations. Then the F(t) and R(t) reliability functions have been generated. Also, the 95% confidence interval is shown.

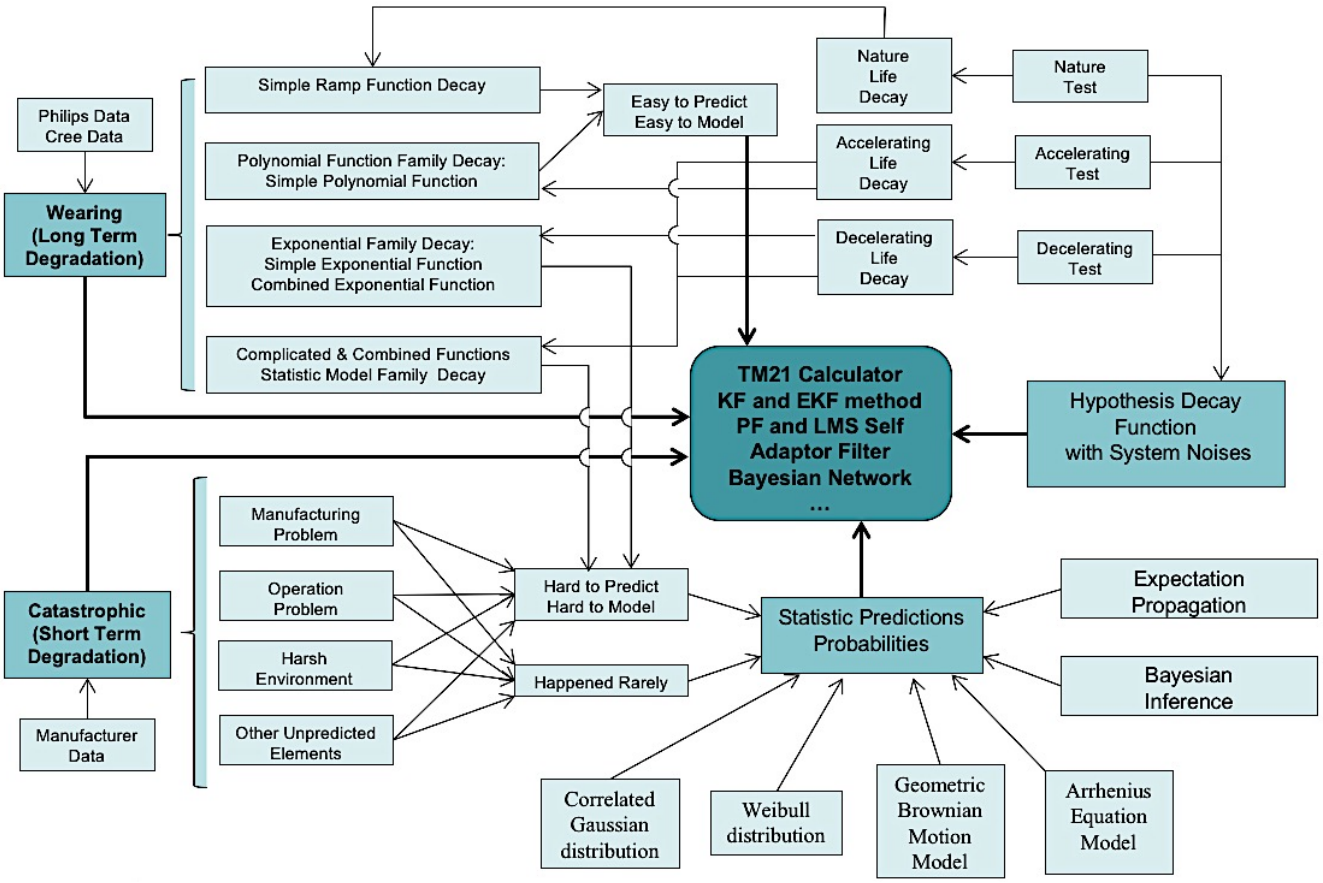


Figure 281 LED PHM Full Map

7.4.1 The TM-21 Calculator

The theory for the TM-21 includes four parts: the first part uses the Least Squares Fit (LSF) to determine the Projected Initial Constant α and Decay Rate Constant β. The governing function for the Lumen Maintenance according to time is the following:

$$\Phi(t) = \beta \cdot e^{-\alpha \cdot t} \tag{291}$$

The lumen maintenance Φ (t), is known for about a dozen datum points. In order to solve for α and β, the logarithm of both sides was taken to produce a linear function.

$$\log(\Phi(t)) = \log(\beta) + (-\alpha)\cdot t \qquad (292)$$
$$\overline{\Phi} = \overline{\beta} + \overline{\alpha}\cdot t$$

The second part is to get the coefficients of regression, which includes the optimal curve fit for the 'slope' and 'intercept'. The values of α and β were determined for the lower case temperature T1, and for the upper case temperature T2.

$$\overline{\alpha} = -\alpha = \frac{\sum_{i=1}^{N}\overline{\Phi(t)_i}\cdot\sum_{i=1}^{N}t_i^{\,2} - \sum_{i=1}^{N}t_i\cdot\sum_{i=1}^{N}(t_i\cdot\overline{\Phi(t)_i})}{N\cdot\sum_{i=1}^{N}t_i^{\,2} - (\sum_{i=1}^{N}t_i)^2} \qquad (293)$$

$$\overline{\beta} = \log(\beta) = \frac{N\cdot\sum_{i=1}^{N}t_i\cdot\overline{\Phi(t)_i} - \sum_{i=1}^{N}t_i\cdot\sum_{i=1}^{N}\overline{\Phi(t)_i}}{N\cdot\sum_{i=1}^{N}t_i^{\,2} - (\sum_{i=1}^{N}t_i)^2} \qquad (294)$$

N: number of points

t_i: time at point of interest

In the third part, the Activation Energy was calculated and the Pre-Exponential Factor A, In-Situ temperature coefficient α and In-Situ temperature coefficient β:

$$E_a = \frac{k\cdot\log(\frac{\alpha_1}{\alpha_2})}{\frac{1}{T_2} - \frac{1}{T_1}} \qquad (295)$$

T_i:Temperature Condition i

Using the Activation Energy E_a and the lower case α value and temperature, the Pre-Exponential Factor A, was determined.

$$A = \alpha_1 \cdot e^{(\frac{E_a}{k \cdot T_1})} \quad (296)$$

With the Pre-Exponential Factor known, the In-Situ temperature α value was determined.

$$\alpha_0 = A \cdot e^{-(\frac{E_a}{k \cdot T_1})} \quad (297)$$

Next, the In-Situ temperature β value was determined.

$$\beta_0 = \sqrt{\beta_1 \cdot \beta_2} \quad (298)$$

Lastly, with the values of α and β known for the In-Situ temperature, the L70 value in hours was determined.

$$L_{70} = \frac{\log(\frac{\beta_0}{0.7})}{\alpha_0} \quad (299)$$

7.4.2 The PHILIPS' Dataset Analysis

This experiment is conducted by following the IES LM-80 standards, and the CCT of the testing units are ranged from 2650K, 3000K, 3500K, 4000K, and 6000K. This analysis will focus on dataset that has three temperatures from 55°C, 85°C, and 105°C, the CCT is 2650K/3000K (i.e. Nominal CCT 2700K). The test product is LUXEON LXM3-PW series LEDs, which is shown in the following Figure 282.

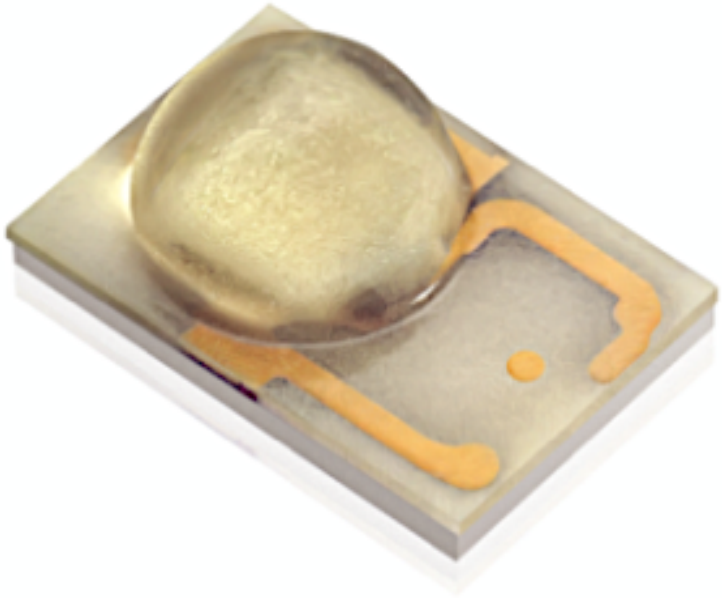


Figure 282 LED Test Product (Philips Ltd.)

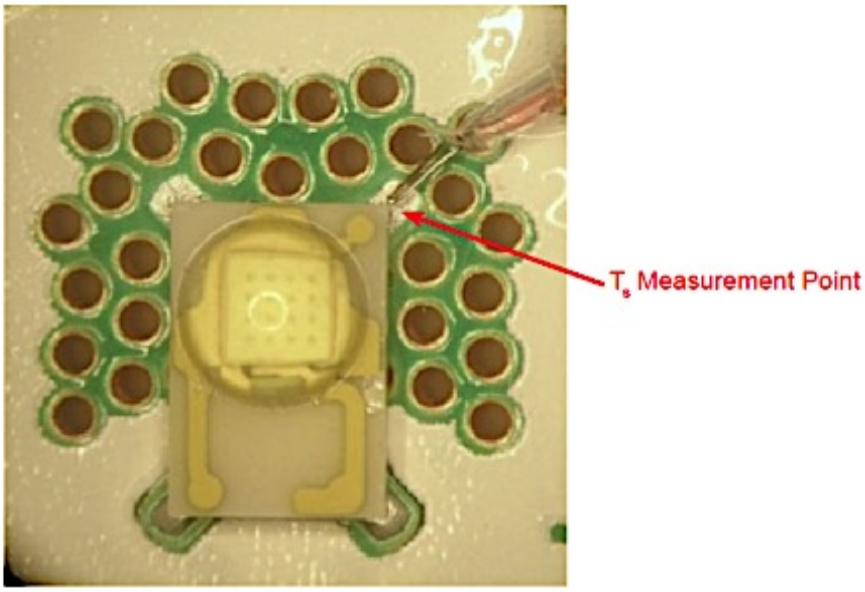


Figure 283 LED Temperature Measurement

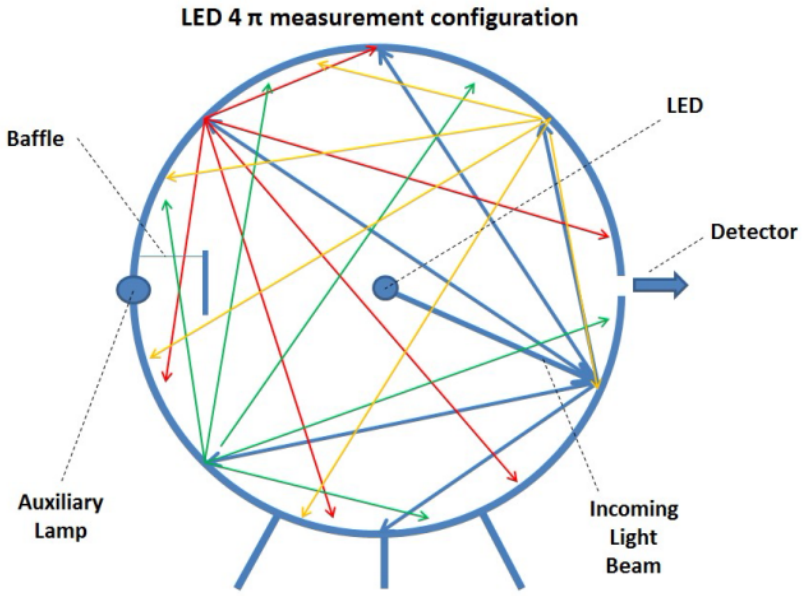


Figure 284 LED Light Gathering System

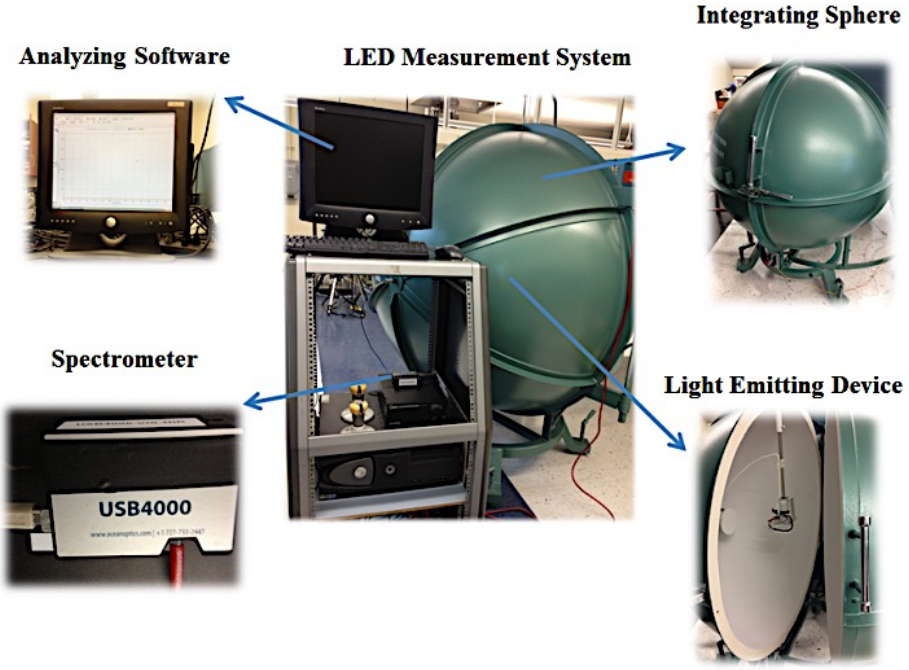


Figure 285 CAVE3 LED Data Collection System

In the dataset, there are many test conditions; each test condition includes 25 samples. In each sample, the lumen maintenance data has been recording from 0 to 9000 hours; the test matrix shows table below:

	Current	T s	CCT
Test1	0.35A	55˚C	3000K
Test2	0.35A	85˚C	3000K
Test3	0.35A	105˚C	3000K
Test4	0.35A	120˚C	3000K
Test5	0.5A	55˚C	3000K
Test6	0.5A	85˚C	3000K
Test7	0.5A	105˚C	3000K
Test8	0.5A	120˚C	3000K
Test9	0.7A	55˚C	3000K
Test10	0.7A	85˚C	3000K
Test11	0.7A	105˚C	3000K
Test12	1A	55˚C	3000K
Test13	1A	85˚C	3000K
Test14	1A	105˚C	3000K

Table 18 Experiment Setup

The LEDs failures rising from lumen degradation subject to the two categories, one is wearing down (i.e. Long-Term Degradation), and the other is a catastrophic failure (i.e. Short Tern Degradation). In this chapter, we mainly focus on the long term degradation of Lumen Maintenance, from which the wearing life of LEDs is predicted.

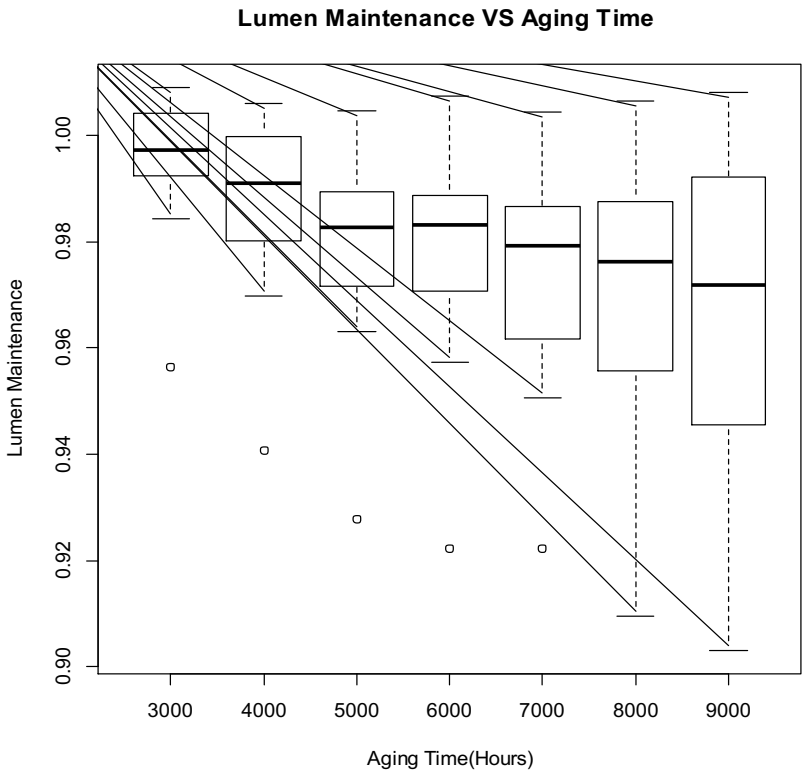


Figure 286 Distribution for the Philip's Dataset

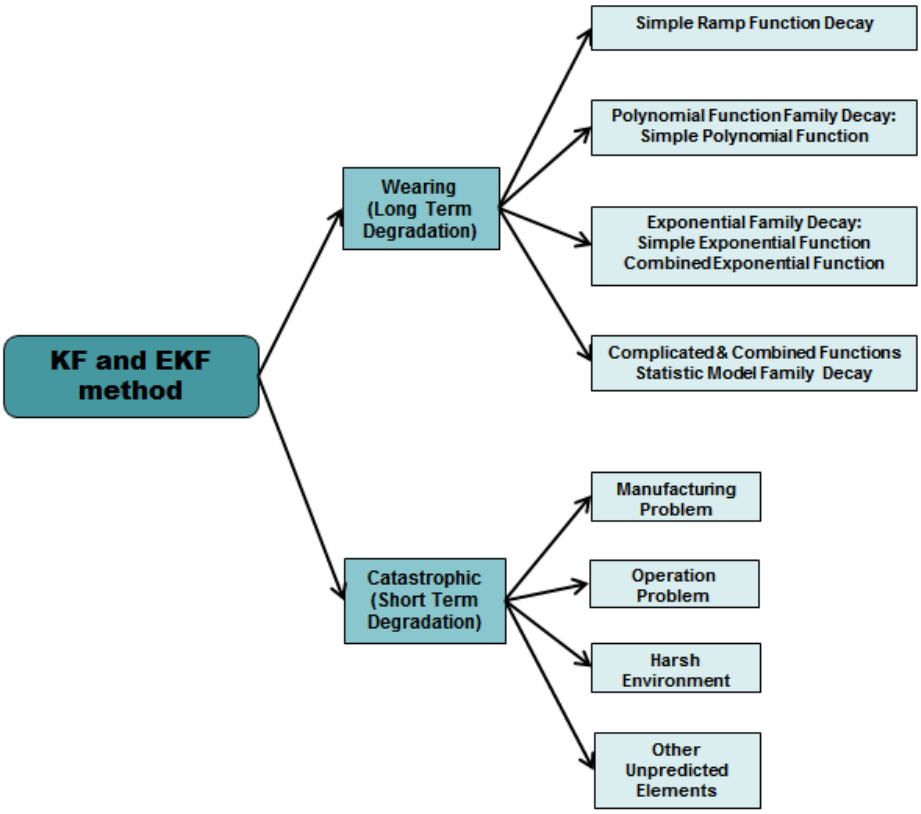


Figure 287 LEDs Categorical Failures

There is also possible for the LED failure derived from chromaticity, which causes the Correlated Color Temperature (CCT) change and delta u' and v' shifting. In the Philips Dataset DR05-1-LM80, the initial CCT of warm LUXEON LED value is 3000K whose the identified u, ' and v' in 1976 CIE (Janos Schanda, 2007) is shown in the following Figure 288.

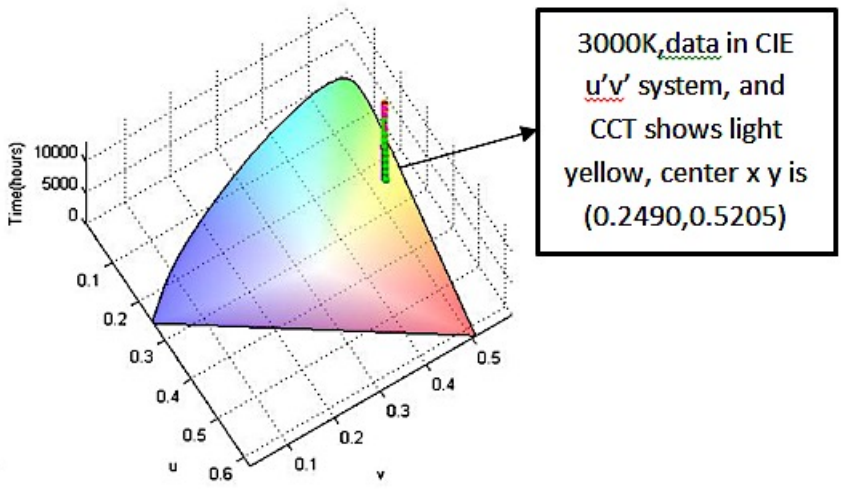


Figure 288 Visualized color of 3000K u' v' in 1976 CIE

The following Figure 289 shows the specified 1A 105°C test dataset, there are 26 samples, in this case, the mean CCT is 2952K. The black line shows in the figure is the Planckian Locus that is ideal black body radiation line.

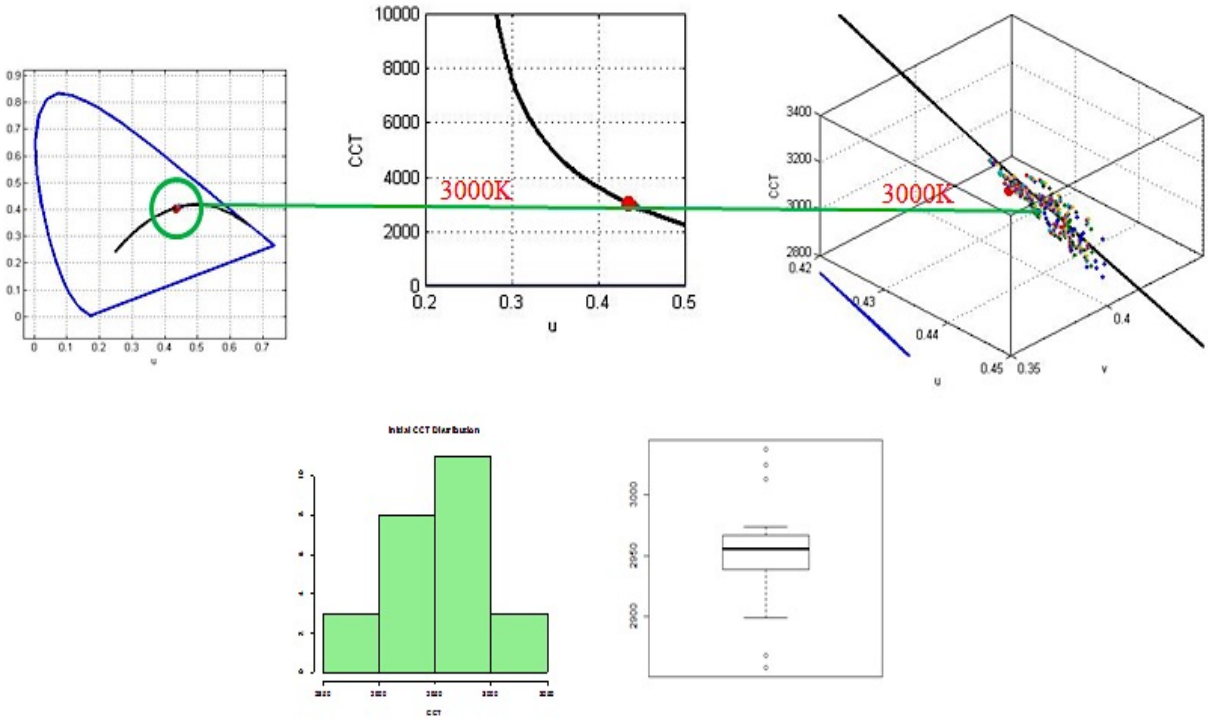


Figure 289 Specified 1A 105°C dataset and its distribution around Planckian Locus

It shows Figure 290, u' and v' value change according to the operating hours, the red line is the average of 55°C chromaticity, the blue line is average of 85°C chromaticity, the green line is average of 105°C chromaticity and the pink line is average of 120°C chromaticity. There is a tendency that u' and v' shifting according to the operating hours.

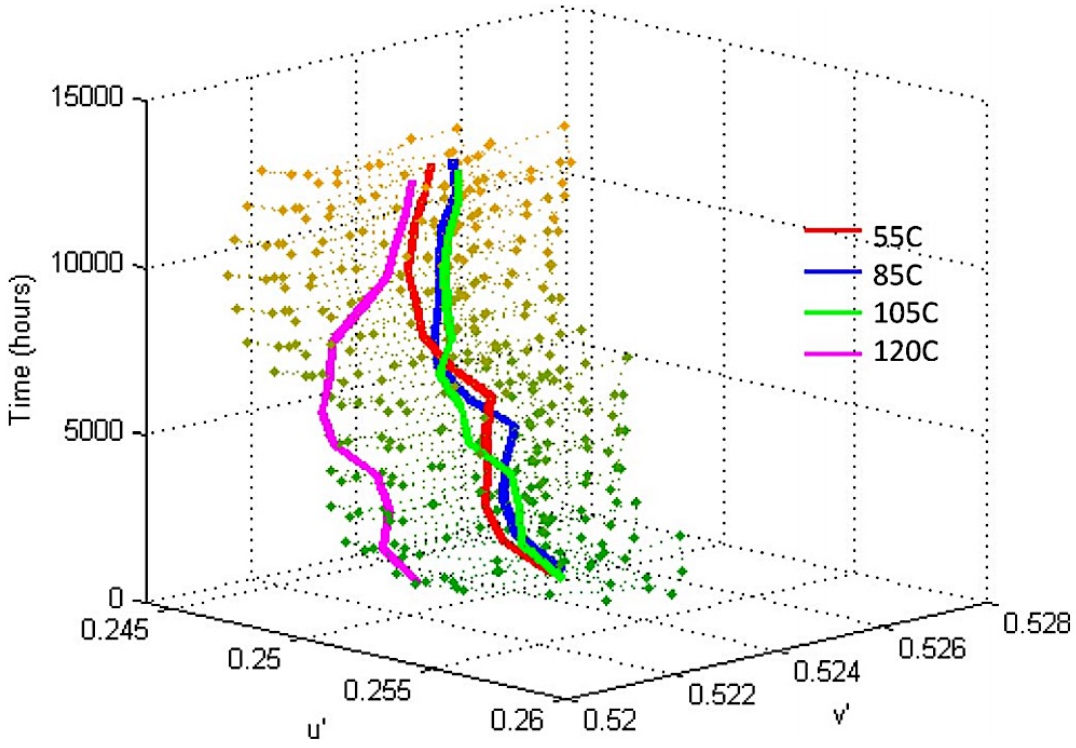


Figure 290 the slow u' v' shifting for the different temperatures

7.4.3 KF/EKF L70 Extrapolations

In the LEDs' Prognostic Health Management (PHM), we use the L70 life, which is the 70% Lumen Maintenance Output comparing with the 100% from the original as the failure criterion of LEDs. The RUL is estimated by the extrapolations at each evaluated point. The Figure 291 below shows the intuition of RUL for L70. In the plot the blue line is raw dataset from Philips LED

LM80-08 Data, the solid red line is the KF/EKF fitting lines, and the dash red line is the KF extrapolations for L70, the green line is L70 criterion, and the green arrow shows the Remaining Useful Life from the last evaluated point.

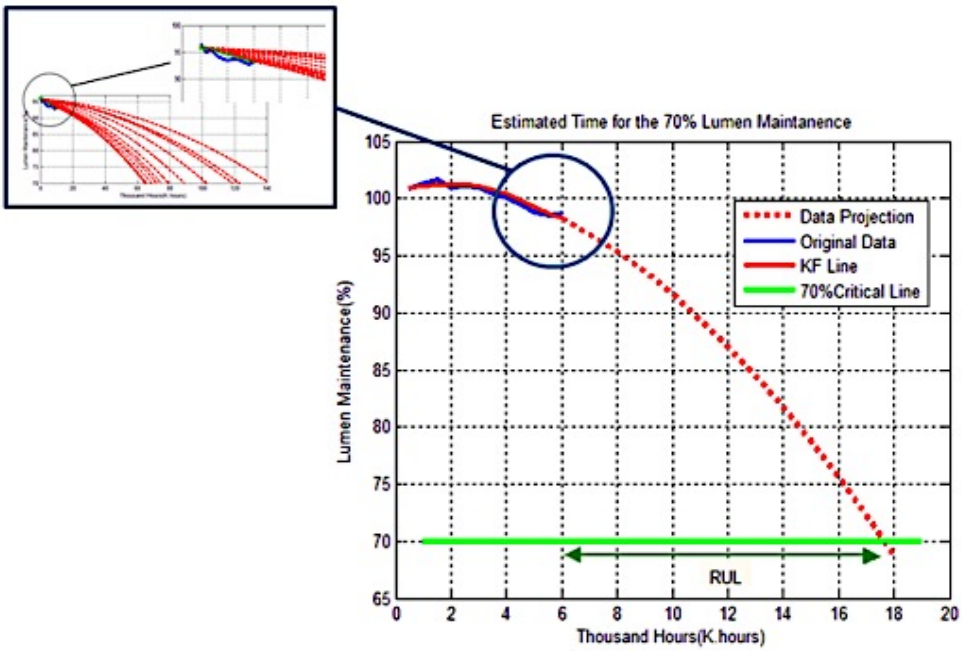


Figure 291 the Intuition of RUL for L70

7.4.4 TM-21 Estimations

The TM-21 extrapolation curve shows below Figure 292: the green dash lines are the extrapolations from the low temperature to the high temperature. The red dash line shows the overall prognostic degradation tendency. The prognostic L70 life of LED using TM-21 is about 36871.4 hours for long-term degradation. However, there is a limitation to use this TM-21 Calculator, we have to know at least three cases at different temperatures to correctly calculate the Activate Energy Ratio. Moreover, the TM-21 is single estimation value, which cannot provide any insight into the Probability and Distribution of L70 estimations.

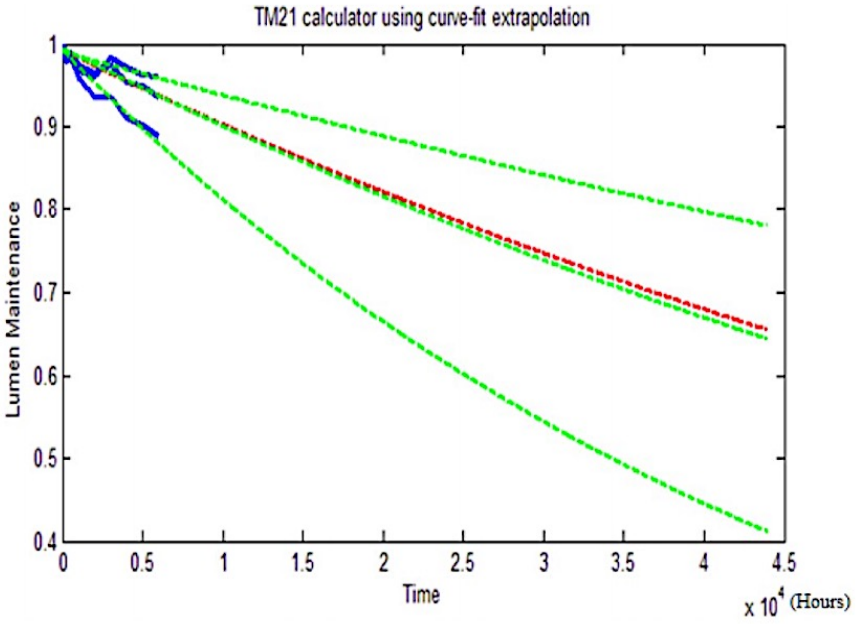


Figure 292 TM-21 Degradation Curve

7.4.5 KF/EKF Model Analysis

The general decay functions are summed in the Figure 293 below (Emil Radkov, 2011). Totally, there are seven models here we want to evaluate individually. However, there are only few failure mechanisms in the LED system: (1) Silicone Encapsulate Degradation; (2) Chip Degradation; (3) Phosphor Degradation; (4) Reflector Degradation; (5) Glass Degradation; Each failure mechanism has its unique function to describe it. The following plot shows the possible decay models existing in the LED systems.

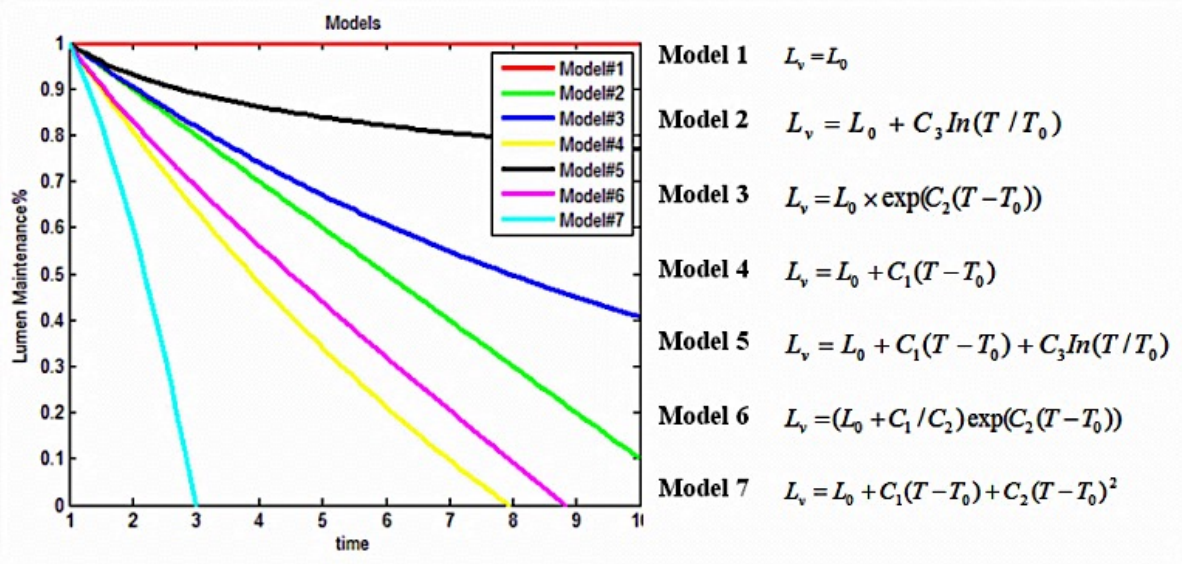


Figure 293 the Hypothesis Decay models

The overall degradation in Lumen Maintenance is resulted from several combined long term decays in the LED light. Thus, the failure mechanism is complicate.

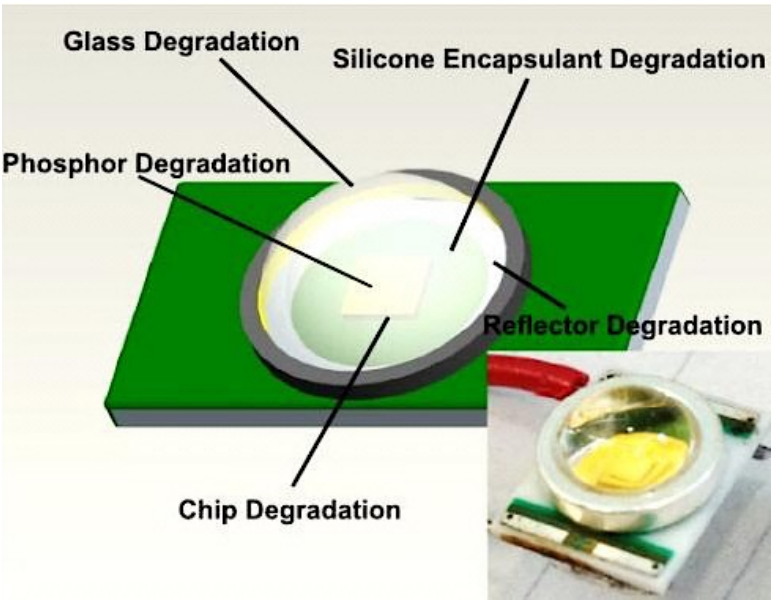


Figure 294 the LED Failures Types

In this chapter, we are mainly exploring the three degradation models: The first is the drift of charged defects in the chip, which is described as a ramp function. The second is Thermal Decomposition of Encapsulant, which is an exponential function, and the third one is the combination of those previous two models. The following Figure 295 shows the three models, and we can simply track those degradation lines to predict the Pseudo L70 life for the LEDs. The first model is the drift of charged defects in the chip, which is described by the function:

$$dL_o / dt = C_1 \tag{300}$$

$$L_o = L_I + C_1(T - T_0) \tag{301}$$

In this model, the decay rate is constant, which is presented by the straight lines. This could sometimes happen in the LED systems. The straight line does have its drawbacks and cannot truly describe the degradation because we know that the decay rate could not be maintained as constant throughout the LED Lifetime. Therefore, we also need to incorporate another decay function into the KF/EKF algorithm to evaluate lumen degradation.

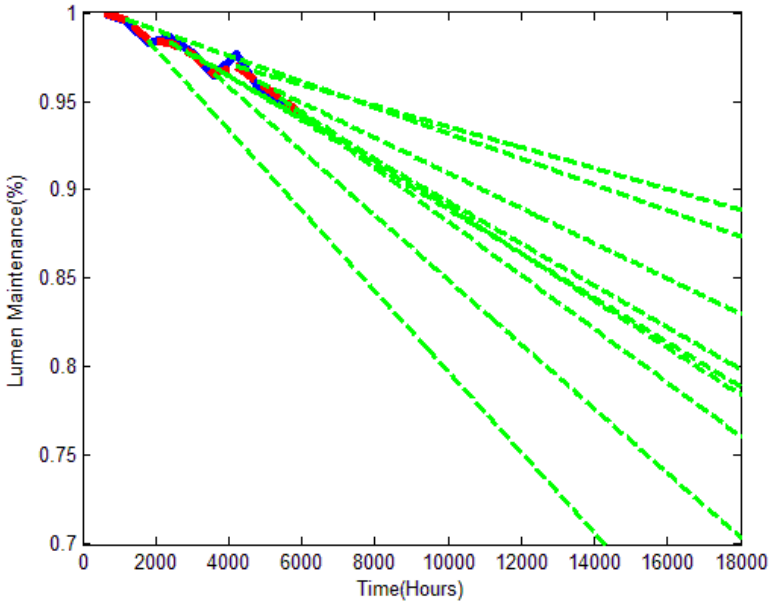


Figure 295 the Constant Decay Extrapolations

Now, we notice that the first stage for the Lumen Degradation probably results from silicone degradation (e.g. chip degradation, plastic degradation). However, as the time elapsing, the damages in the system would not be shown as single degradation trend. Several degradations are working together to accrue the damages and to accelerate the failure of the system. Therefore, we can use the polynomial function (e.g. Parabolic Function) embedded into the KF /EKF algorithm to prediction decay curve. In this model, it is accelerated degradation in the system that the Pseudo L70 life also would be shorter than other models. The Figure 296 below shows the accelerating degradation using the polynomial function and the decay rate is not constant, which expresses as:

$$dL_o / dt = C_1 + C_2 T \tag{302}$$

$$L_o = L_I + C_1(T - T_0) + 0.5C_2(T - T_0)^2 \tag{303}$$

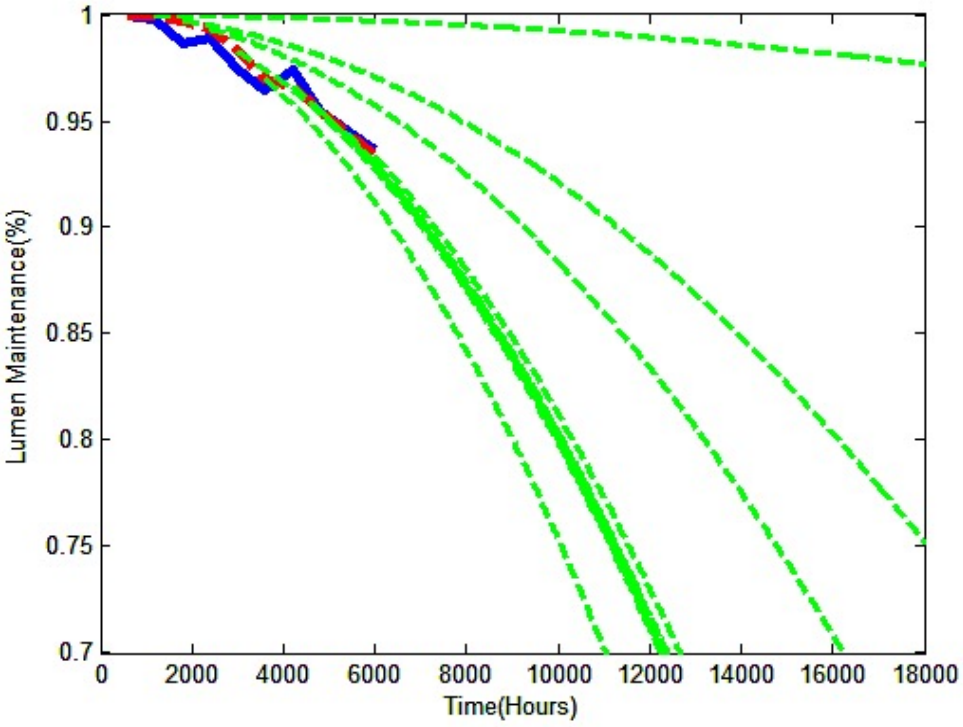


Figure 296 Accelerating Decay Extrapolations

Apart from the accelerating case that happened in a harsh environment, the LED system may experience long-term degradation in the tempered environment, in this case, the main degradation in the system results from Thermal decomposition of encapsulant in LEDs. Therefore, we can use the exponential function to present the decay curve in the KF/EKF algorithm. This extrapolation is presented in the Figure 297 below.

$$dL_o / dt = C_1 \times L_0 \tag{304}$$

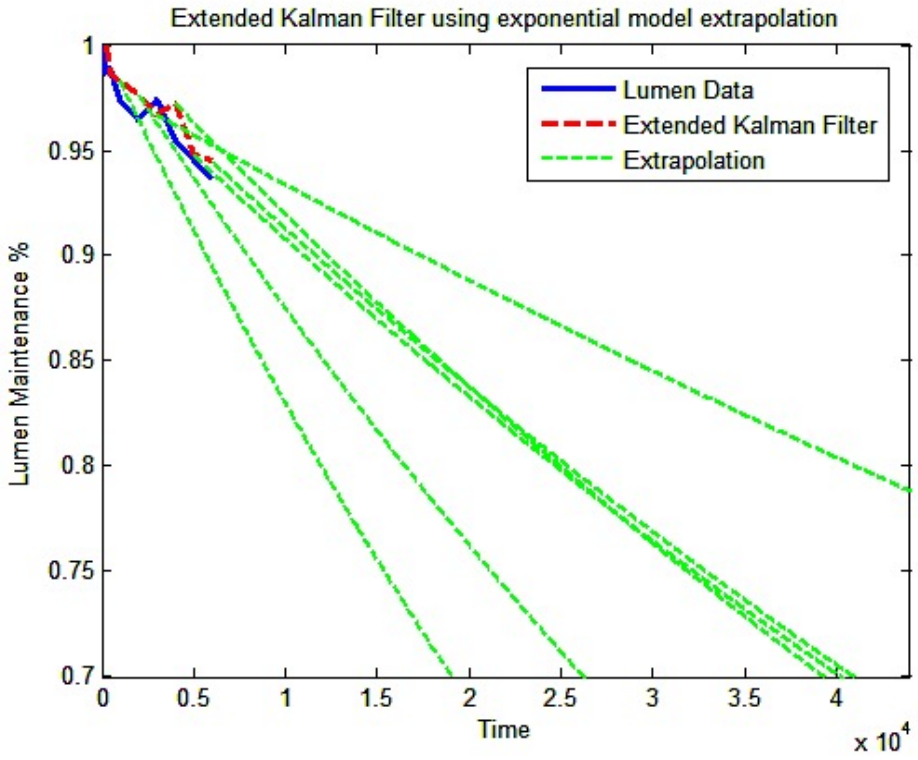


Figure 297 the Decelerating Decay Extrapolations

There is also a possibility that mixed model of decay function exists in the LED system. We combine the linear decay model (i.e. mode 1) and exponential decay model (i.e. mode 2) to construct complicate decay function, which shows below Figure 298:

$$dL_o / dt = C_1 + C_2 L_0; \tag{305}$$

$$L_o = (L_1 + C_1 / C_2)e^{C_2 \times (T - T_0)} - C_1 / C_2; \quad (306)$$

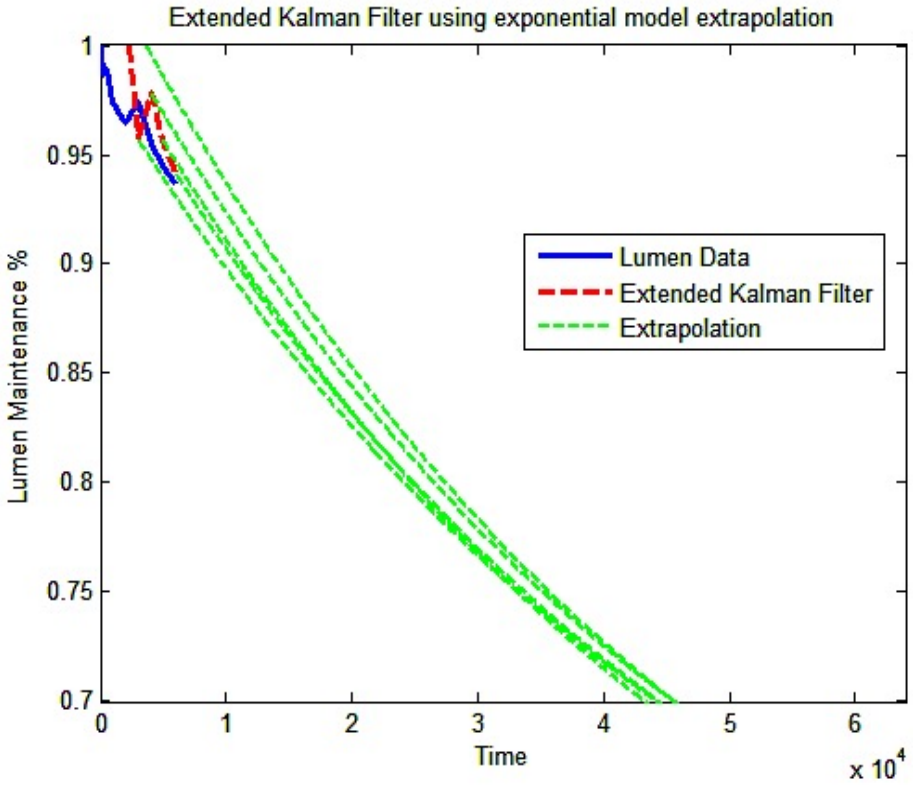


Figure 298 the Combined Model Extrapolations

7.4.6 L70 Life Distribution and Reliability Discussion

We can get the extrapolations and distribution from the Kalman Filter, which includes critical life information. In general, to make extrapolations simple, we apply the Decelerating Decay Extrapolation (i.e. Model 3) to predict the L70 life. The following picture Figure 299 depicts the KF Algorithm estimation, which shows on the red line. The original Philips LUXEON (1A, 105°C) dataset, which shows on blue. There are 25 samples in this dataset; each sample data is evaluated, and the life-estimation data is generated by the KF algorithm. In each sample, the pseudo L70 life has been estimated in KF using the Newton-Raphson Method that is illustrate as

following. When Kalman Filter made a prediction about the state vectors in each evaluated time, and the remaining useful life could be estimated and calculated mathematically by solving the equation H(t) and find the time T-prediction. In order to find the root of above equation, we applied the Newton Rapson's Method:

$$H(t) = x_0 + \dot{x} \times t + \ddot{x} \times t^2 - f(EoL) \tag{307}$$

$$T(n+1) = T(n) - f(x)/f'(x) \tag{308}$$

$T(n)$ is the estimated root at time n;

$T(n+1)$ is the estimated root at time (n+1);

$f'(x)$ is the derivation of target equation;

The Predicted RUL (i.e. T-predict) is known as the 'L70' End of Life (EoL) minus the sampling time (i.e. T-sample). So the algebra equations present as following:

$$L70_{EoL} = T_{predict} + T_{sample} \tag{309}$$

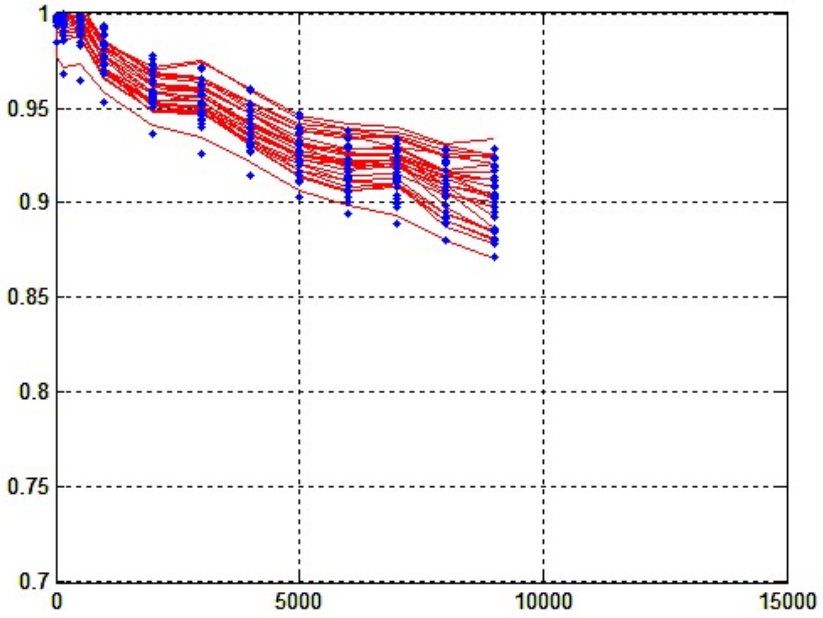


Figure 299 KF Estimations

Generally, the estimation would be more precise if we use the time after 2000 hours for predicting. Therefore, we abandon the data that less than 2000 hours and the Life distribution will be estimated from 2000 to 9000 hours. Each one thousand hours in between 2000 and 9000 hours, we made a prediction and extrapolation; The KF estimation has been keeping updated with the feeding Lumen Maintenance data, so we train the KF's gains to get the convergence of degradation rate and degradation accelerations. The following Figure 300 shows the pseudo L70 life predicted by the KF with an exponential function. The L70 estimation model:

$$\Phi = \alpha \cdot \exp(-\beta \cdot t) \qquad (310)$$

Φ : the Lumen Maintenance (%)

α : the initial degradation factor

β : is the degradation rate.

The estimation parameters α_i and β_i are shown in the following Figure 301and Figure 302. The figure shows the KF extrapolations, the green lines using exponential model. Those extrapolation lines have been drawn by the coefficients estimated by the KF in the table. The blue line in the Figure 300 shows the true estimation, which is the mean of the distribution; the red curve shows the variance of the distribution.

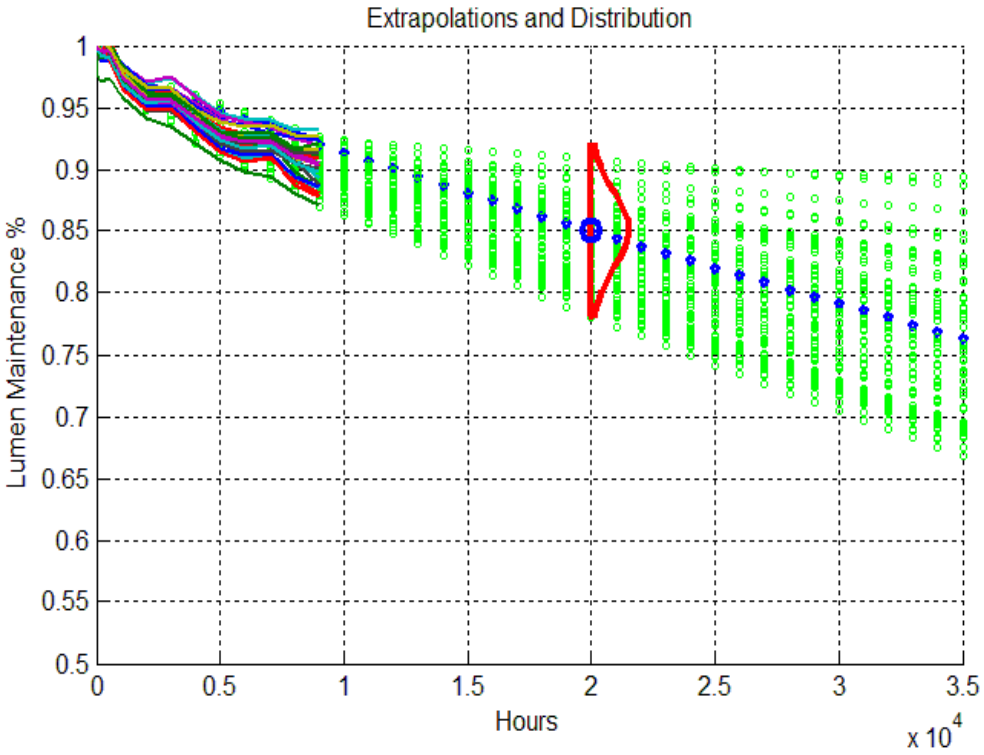


Figure 300 EKF Extrapolations and Mean Estimations

TIME(Hours)	2000	3000	4000	5000	6000	7000	8000	9000
Sample #1	0.97249775	0.96223057	0.95094965	0.93961329	0.93345292	0.93109569	0.9278904	0.9244353
Sample #2	0.93675318	0.9258443	0.91477558	0.90277471	0.89389849	0.88868318	0.88047323	0.87181775
Sample #3	0.97044543	0.95932826	0.94576012	0.93251007	0.92341457	0.92040252	0.91398656	0.90359362
Sample #4	0.96736985	0.95717009	0.94472647	0.93078358	0.92111065	0.91799286	0.90863972	0.89530531
Sample #5	0.97573055	0.96513312	0.95251816	0.93894826	0.93064645	0.92765329	0.92261038	0.91845788
Sample #6	0.96452917	0.95132119	0.9362477	0.92222237	0.91396327	0.91102709	0.9062907	0.90275804
Sample #7	0.95378058	0.94418653	0.93200491	0.92125832	0.91401041	0.91286264	0.90634122	0.89227446
Sample #8	0.95856379	0.94695341	0.92982794	0.91612394	0.90538307	0.90194346	0.89137171	0.88119472
Sample #9	0.96947121	0.95690609	0.94145512	0.92770557	0.91441239	0.91159903	0.8988392	0.88520733
Sample #10	0.95923287	0.94634436	0.93196974	0.92007433	0.91289298	0.91366773	0.91141624	0.90854344
Sample #11	0.97606984	0.97041376	0.960004	0.94663446	0.93775215	0.9340284	0.92873754	0.92869069
Sample #12	0.97728271	0.97142567	0.95962504	0.94496011	0.93506538	0.92859532	0.92124131	0.91943827
Sample #13	0.96424243	0.95290596	0.93991652	0.92738027	0.92022548	0.91755061	0.9134049	0.91320019
Sample #14	0.95003578	0.94029096	0.9279337	0.91622781	0.90891714	0.90811373	0.90330405	0.90019416
Sample #15	0.97547682	0.96529937	0.95099756	0.93709933	0.92908253	0.92374556	0.92084204	0.92009011
Sample #16	0.97335122	0.95907371	0.94177164	0.92695725	0.91775724	0.91589959	0.91191505	0.90927817
Sample #17	0.95631627	0.94377703	0.9276978	0.91150313	0.90030434	0.89811076	0.8898445	0.88058741
Sample #18	0.95670974	0.94593154	0.92972913	0.9142209	0.90292829	0.90043704	0.89245602	0.88448585
Sample #19	0.95749179	0.94794116	0.93383733	0.92167718	0.91476277	0.91317148	0.91019154	0.90504911
Sample #20	0.97229663	0.96310079	0.94908197	0.93621021	0.92819796	0.92702808	0.92409439	0.92324445
Sample #21	0.95454789	0.94170291	0.92788907	0.91645914	0.91168621	0.91380607	0.91186561	0.91198822
Sample #22	0.95312288	0.94378099	0.93099479	0.91583583	0.90615782	0.90437789	0.89504743	0.88633118
Sample #23	0.95825749	0.95341019	0.9438638	0.93020827	0.92339104	0.92240976	0.91671466	0.91708814
Sample #24	0.95201711	0.94162466	0.92713008	0.91118752	0.90003166	0.89965628	0.88859667	0.8788727
Sample #25	0.95589408	0.94875909	0.93625388	0.92186887	0.91223147	0.91089067	0.90423777	0.89786771
Sample #26	0.96246479	0.95218223	0.93868625	0.92519391	0.91645432	0.91415921	0.90798291	0.90236292

Table 19 Pre-decay Parameter Table

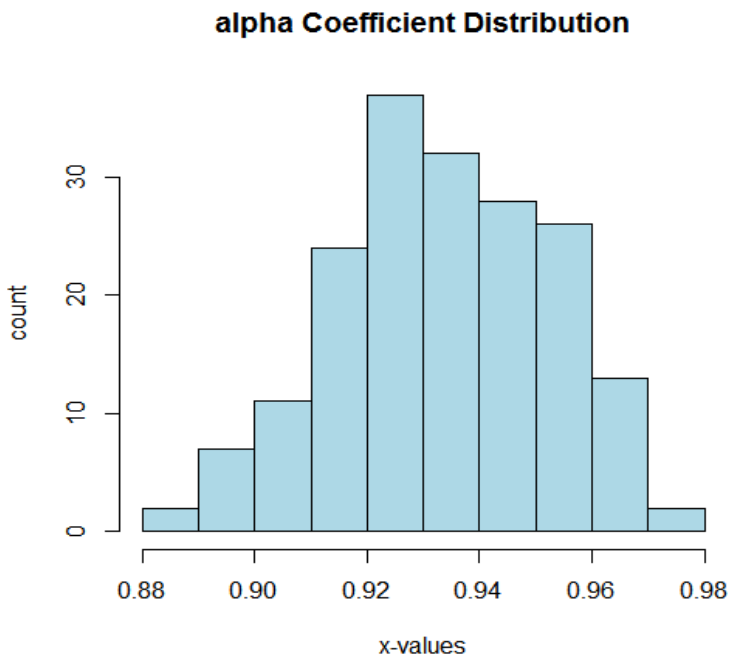


Figure 301 Pre-decay Parameter Distribution

TIME(Hours)	2000	3000	4000	5000	6000	7000	8000	9000
Sample #1	-0.0012899	-0.0014447	-0.0015564	-0.0015882	-0.0012675	-0.0007909	-0.0004987	-0.0003232
Sample #2	-0.0024896	-0.0022393	-0.0019665	-0.0017637	-0.0014063	-0.0009285	-0.0007885	-0.0007506
Sample #3	-0.0013955	-0.0015678	-0.0017678	-0.0018409	-0.0015991	-0.0010328	-0.0008179	-0.000911
Sample #4	-0.0015088	-0.001592	-0.0017089	-0.0018265	-0.0016162	-0.0010457	-0.0009943	-0.0012077
Sample #5	-0.0012001	-0.0014071	-0.0016154	-0.0017706	-0.0015271	-0.0010051	-0.0007395	-0.0005104
Sample #6	-0.0016219	-0.0018304	-0.0020217	-0.0020391	-0.0016533	-0.0010268	-0.0006844	-0.0003939
Sample #7	-0.0020727	-0.0019493	-0.0019071	-0.0017286	-0.0013514	-0.0007075	-0.0005756	-0.0009595
Sample #8	-0.0017934	-0.0018492	-0.0021257	-0.0020749	-0.0018062	-0.0011429	-0.001091	-0.0010519
Sample #9	-0.0014448	-0.0016805	-0.001953	-0.0019963	-0.0019443	-0.0012416	-0.0013107	-0.0014253
Sample #10	-0.0018965	-0.0020173	-0.0021058	-0.0019545	-0.0015096	-0.0006372	-0.0003079	-0.0001066
Sample #11	-0.0011571	-0.0011159	-0.0012816	-0.0015317	-0.0014183	-0.0010039	-0.0007822	-0.0003304
Sample #12	-0.0011448	-0.0011237	-0.0013678	-0.0016711	-0.0015745	-0.0012644	-0.0010711	-0.0006082
Sample #13	-0.0016262	-0.0017322	-0.001831	-0.0018178	-0.0014438	-0.0008859	-0.0005761	-0.0001529
Sample #14	-0.0021114	-0.0019567	-0.0018985	-0.0017602	-0.0013629	-0.000681	-0.0004471	-0.0002192
Sample #15	-0.0011773	-0.0013646	-0.0016752	-0.0018339	-0.0015531	-0.0011546	-0.0007158	-0.0002893
Sample #16	-0.0013515	-0.0017196	-0.0021036	-0.0021797	-0.001833	-0.0011028	-0.0006962	-0.0003491
Sample #17	-0.001967	-0.0020341	-0.0021989	-0.0022637	-0.0019549	-0.001152	-0.0009391	-0.0008734
Sample #18	-0.0019559	-0.001933	-0.0021331	-0.0021814	-0.0019101	-0.0011495	-0.0009298	-0.0007969
Sample #19	-0.0018701	-0.001796	-0.0019119	-0.0018294	-0.001414	-0.0007815	-0.000422	-0.0003239
Sample #20	-0.0012889	-0.0013847	-0.0016605	-0.0017526	-0.0014869	-0.0008637	-0.0005158	-0.0001699
Sample #21	-0.0021052	-0.0021603	-0.0021612	-0.0019458	-0.0013409	-0.0004868	-0.0001398	0.00017513
Sample #22	-0.0019877	-0.0018507	-0.0018565	-0.0019473	-0.0016494	-0.0009388	-0.0008829	-0.0008341
Sample #23	-0.0017761	-0.0014671	-0.0014147	-0.0015721	-0.0012622	-0.0006815	-0.0005464	-0.0001275
Sample #24	-0.0021699	-0.0020611	-0.0021129	-0.0021731	-0.0018815	-0.0009911	-0.0009955	-0.0009505
Sample #25	-0.0019213	-0.0016981	-0.0017428	-0.0018369	-0.0015881	-0.0008925	-0.0007099	-0.0005892
Sample #26	-0.0016891	-0.0017169	-0.0018403	-0.0018754	-0.0015735	-0.0009471	-0.0007265	-0.0005591

Table 20 Decay Rate Table

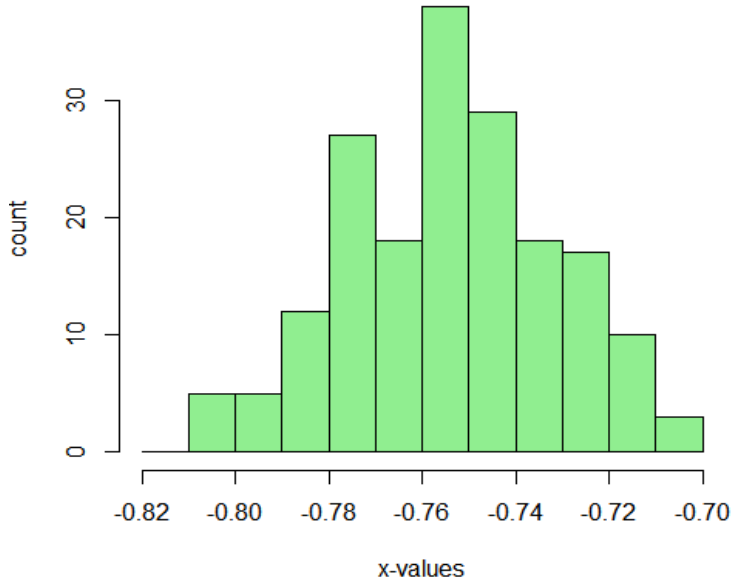


Figure 302 Decay Rate Distribution

7.4.7 KF Chromaticity Tracking

The Chromaticity u' and v' also could be tracked by the Kalman Filtering Algorithm. Therefore, we could gather the information into the reliability discussion of LEDs. Compared with lumen degradation, the chromaticity is very long term degradation, but the change of delta u' and delta v' would result in shortening LEDs useful life in some color restricted situations. The dash blue line is the KF chromaticity tracking for u' and v' in 9000 hours operating hours.

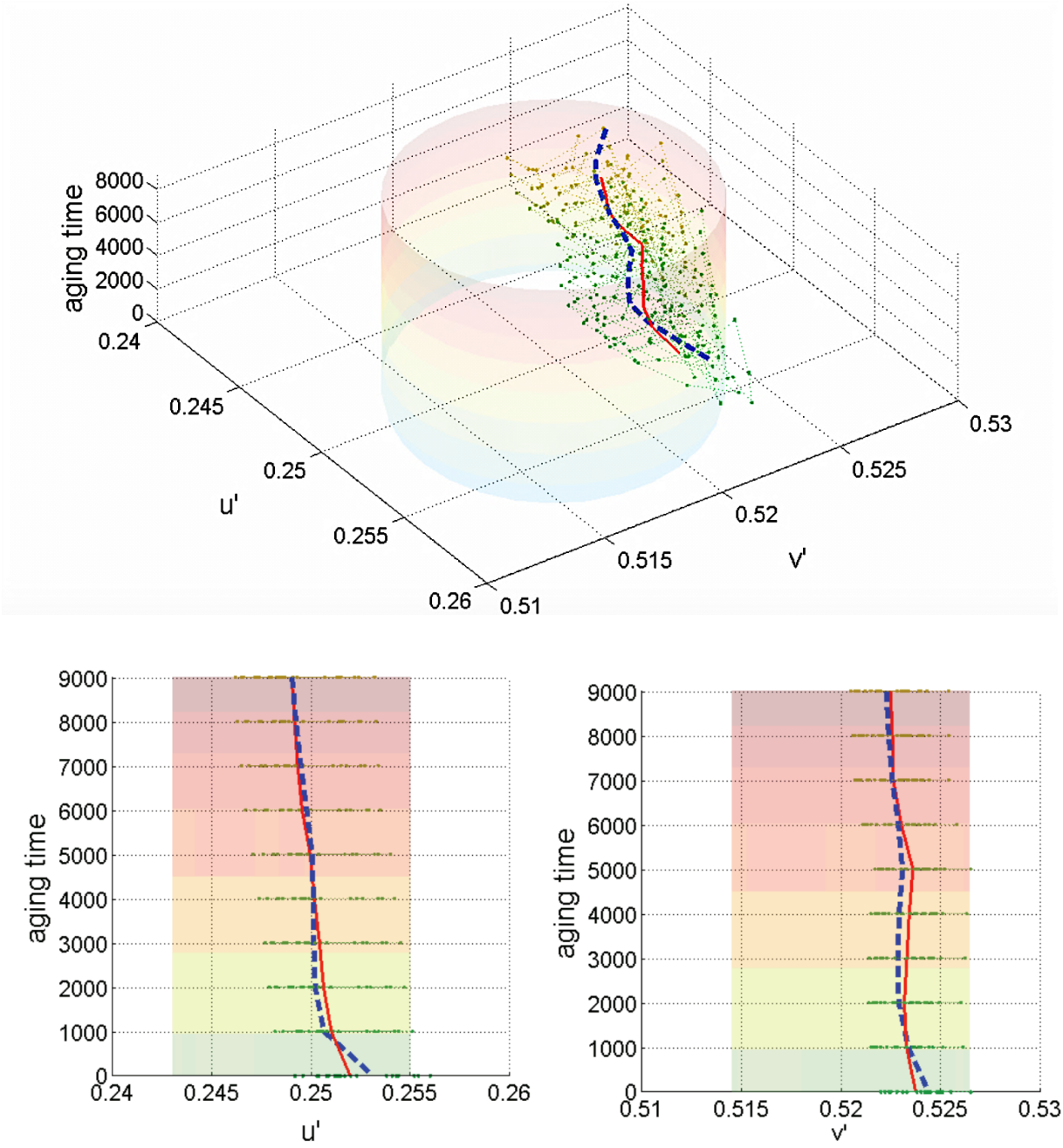


Figure 303 KF Chromaticity Tracking

The Figure 304 shows u' v' KF tracking in 2D 1976 CIE u' v' coordinate. The circle is ± 0.06 confident interval; the circle center is central point for 3000K.

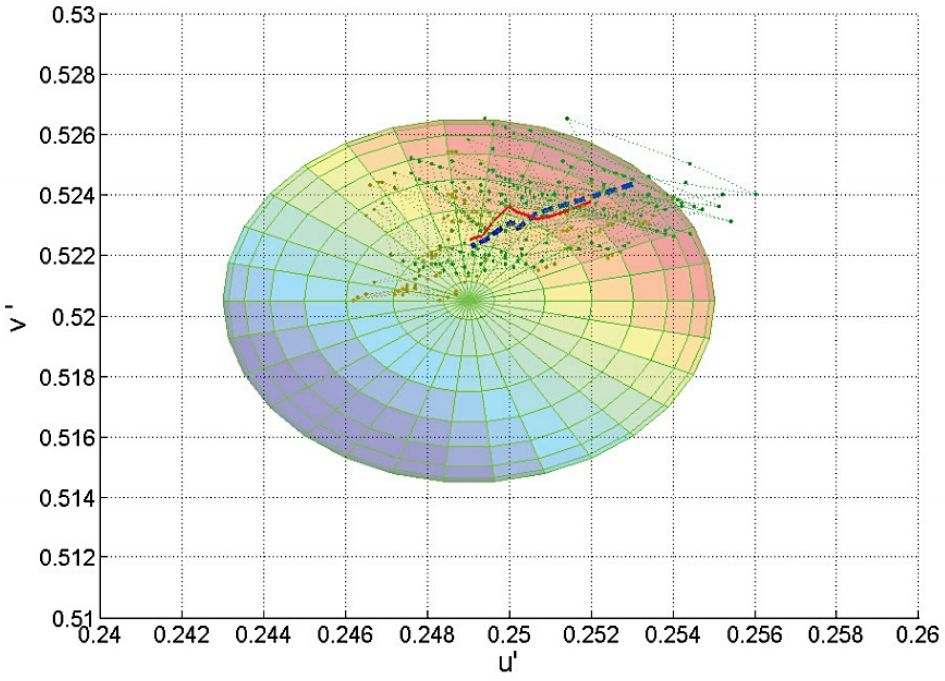


Figure 304 KF track with ± 0.06 confident interval

7.4.8 Prognostic Lumen Degradation Using EKF model

The lumen degradation could be prognosticated at given time, for example it can predict the Lumen Maintenance at 16,000 hours. We consider normal distribution for the estimation of training decay rates. The following Figure 305 and Figure 306 show the normal distribution beta factor and lumen estimation using previous 8,000 hours' degradation information.

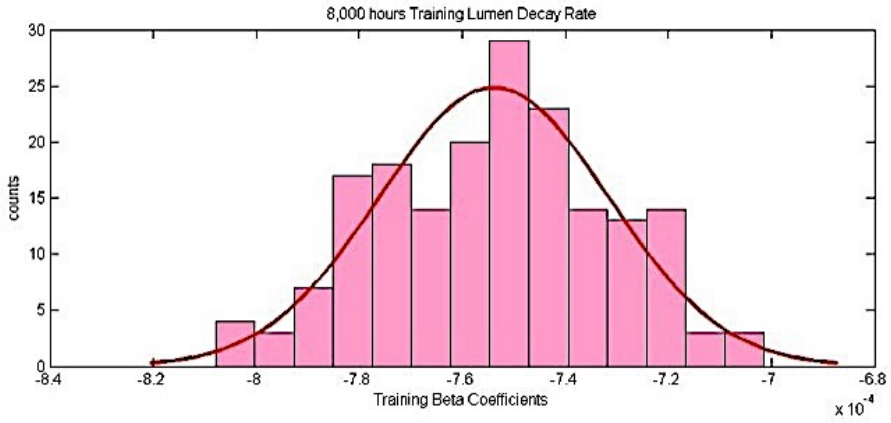


Figure 305 Fitted Distribution for Decay Rate β_i

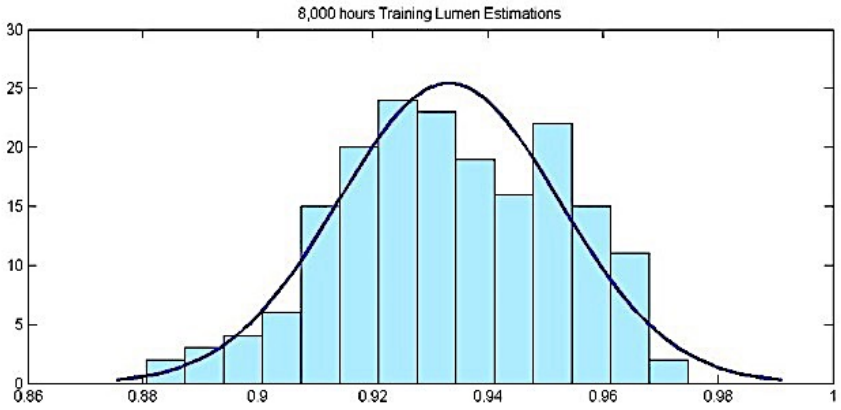


Figure 306 Fitted Distribution for Lumen Estimation α_i

We use the basic exponential degradation function to model the degradation path from 8,000 hours to 16,000 hours. In the Figure 307, we can figure out that the lumen estimation and decay rate estimation are also normally distributed around its mean value with identified variance. In the Figure 307, it shows the degradation path for estimating the 16,000 hours' life. The dots represent the lumen maintenance estimations, and the blue line shows the 16,000 hours' threshold.

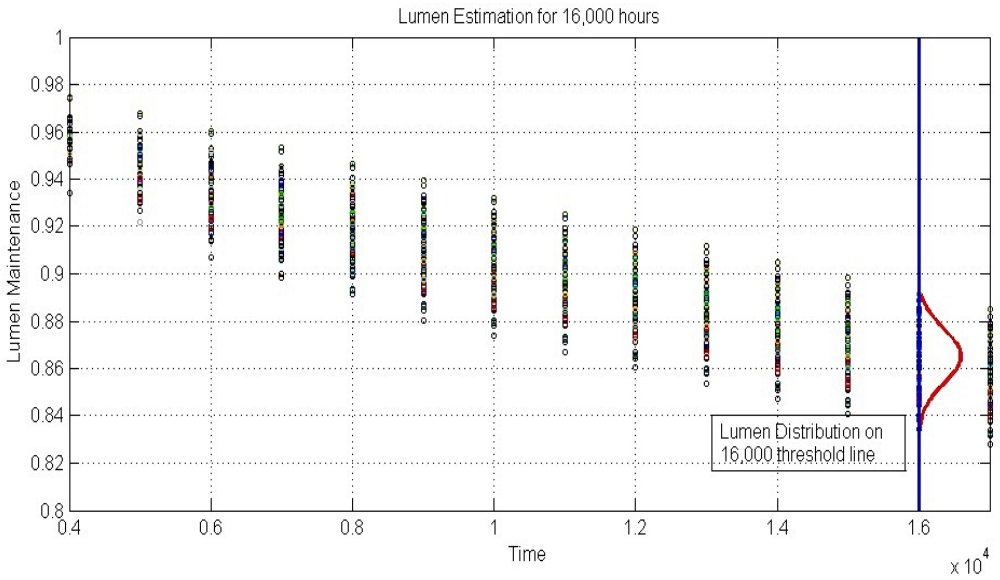


Figure 307 Lumen Degradation Path for Estimating Lumen at 16,000 hours

The histogram Figure 308 shown below is the lumen estimation at 16,000 hours, the blue bar shows the counts of estimated lumen value, the red line is fitted distribution. The mean value of 16,000 hours' lumen maintenance is 86.53%, and the mean value of 8000 hours is 92.21%. Therefore, the lumen maintenance degradation from 8000 hours to 16,000 hours is 5.68 %. The lumen maintenance variance at 8000 hours is 1.1196e-04, and the lumen maintenance variance at 16,000 hours is 1.2863e-04, so the variance increases 1.667e-5, which indicates the distribution shape is wider than our current's.

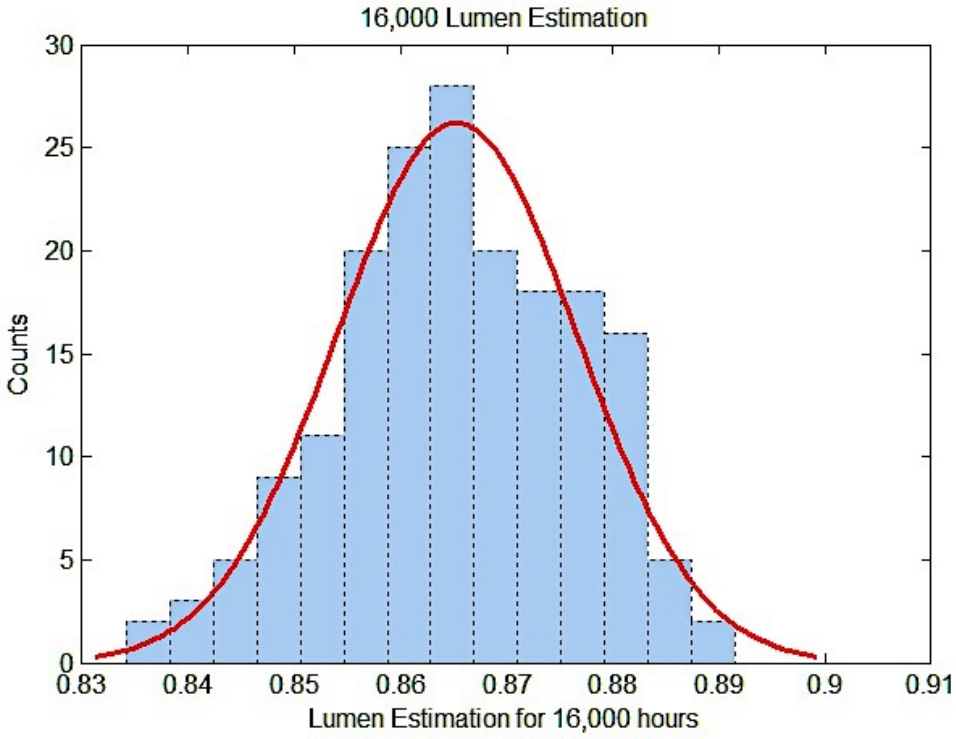


Figure 308 Lumen Estimation Summary for 16,000 hours

7.4.9 Life Distribution discussion

A cumulative distribution function F(t) represents the population fraction failing by age t. Any such continuous F(t) has the Mathematical properties. It is a continuous function for all t.

$$\lim_{t \to -\infty} F(t) = 0 \tag{311}$$

$$\lim_{t \to +\infty} F(t) = 1 \tag{312}$$

$$F(t) \leq F(t') \; \forall \; t < t' \tag{313}$$

The range of t for most life distribution is from 0 to +∞, but some useful distributions have a range from -∞ to +∞. The reliability function R(t) for a life distribution is the probability of survival beyond age t, namely, the survivor or survivorship function can be represented as:

$$R(t) = 1 - F(t) \tag{314}$$

We can use the method that fitting the probability for the distribution of pseudo L70 life over the sample population. Here, we only use Normal distribution, Lognormal distribution and Weibull distribution to fit the data distribution. The normal distribution has been used to describe the life of incandescent lamp filaments and electrical insulations. It is also used as the distribution for product properties. The population fraction failing by age y is shown in the Normal Cumulative Distribution Function below:

$$F(y) = \int_{-\infty}^{y} (2\pi\sigma^2)^{-1/2} \cdot e^{[-\frac{1}{2}(\frac{x-\mu}{\sigma})^2]} dx, -\infty < y < +\infty \tag{315}$$

The probability density shows in the Normal Probability Density below:

$$f(y) = (2\pi\sigma^2)^{-1/2} \cdot e^{[-(y-u)^2/(2\sigma^2)]}, -\infty < y < +\infty \tag{316}$$

The population fraction surviving age t shows in the Normal Reliability Function below:

$$R(t) = 1 - \Phi[(t - \mu)/\sigma] \tag{317}$$

The Lognormal distribution is widely used for fitting life data, including metal fatigue, solid state components and electrical insulation. The lognormal and normal distributions are related; the fact is used to analyze lognormal data with same methods for normal data. In the Lognormal Cumulative Distribution, the population fraction failing by age t is:

$$F(t) = \Phi\{[\log(t) - \mu]/\sigma\}, t > 0 \quad (318)$$

In the Lognormal Probability Density:

$$f(t) = \{0.4343/[(2\pi)^{1/2} t\sigma]\} \cdot e^{\{-[\log(t)-\mu]^2/(2\sigma^2)\}}, t > 0 \quad (319)$$

In the Lognormal Reliability Function, the population fraction surviving age t is:

$$R(t) = 1 - \Phi\{[\log(t) - \mu]/\sigma\} = \Phi\{-[\log(t) - \mu]/\sigma\} \quad (320)$$

The Weibull distribution is often used for product life, because it models either increasing or decreasing failure rates simply. It is also used as the distribution for products properties such as strength (e.g. electrical or mechanical), elongation resistance, and etc., in accelerated tests. It is used to describe the life of roller bearings, electronic components, ceramics, capacitors, and dielectrics in accelerated test. The Weibull Cumulative Distribution. The population fraction failing by age t is:

$$F(t) = 1 - e^{-(t/\alpha)^\beta}, t > 0 \quad (321)$$

In the Weibull Probability Density:

$$f(t) = (\beta/\alpha^\beta) t^{\beta-1} \cdot e^{-(t/\alpha)^\beta}, t > 0 \quad (322)$$

For the Weibull Reliability function, the population fraction surviving age t is:

$$R(t) = e^{-(t/\alpha)^\beta}, t > 0 \quad (323)$$

The following Table 21 and Table 22 respectively shows the fitting statistics for the Normal distribution, Lognormal distribution and Weibull distribution, the lower value of Cramer-von Mises Criterion (CMC-Minimum Distance), the better fitting of distribution it is. Here, the lognormal distribution fitting shows lowest criterion value, which says it has best fitting. The best fitted distribution for KF is lognormal distribution Figure 309 with CMC value 0.541, and normal distribution Figure 310 for EKF with CMC value 0.0968.

Goodness-Fit Tests for Three Distributions		
Distributions	Cramer-von Mises Criterion	P-Value
Normal	1.161	<0.010
Lognormal	0.541	<0.005
Weibull	1.050	<0.010

Table 21 KF Distribution Fitted Statistic

Goodness-Fit Tests for Three Distributions		
Distributions	Cramer-von Mises Criterion	P-Value
Normal	0.0968	<0.005
Lognormal	0.0981	<0.005
Weibull	0.295	<0.010

Table 22 EKF Distribution Fitted Statistic

Sample Number	2000 Hours	3000 Hours	4000 Hours	5000 Hours	6000 Hours	7000 Hours	8000 Hours	9000 Hours
#1	27489.54	25023.83	23685.12	23535.41	28707.6	43072	64517.22	95046.54
#2	13702.08	15487.11	17608.11	19423.68	23387.37	32705.07	37091.18	38245.12
#3	25409.11	23101.17	21021.76	20579.26	23322.44	33503.56	40612.04	37024.9
#4	23440.96	22654.39	21544.84	20600.83	22984.8	32927.03	34235.43	29376.25
#5	29672.5	25825.9	23068.5	21586.57	24650.18	35013.97	45340.62	62214.98
#6	21764.52	19759.54	18384.04	18520.96	22131.78	32660.54	45739.53	73582.97
#7	16924.8	18351.04	19010.09	20889.27	25739.77	44529.19	52883.09	34292.69
#8	19528.22	19340.87	17356.38	17967.61	20244.47	29178.12	30151.31	30883.49
#9	24540.55	21602.92	19174.2	19107.6	19742.47	28272.48	27074.95	25469.65
#10	18612.01	17947.1	17591.85	18987.18	23590.09	45206.93	93717.12	253566.9
#11	30730.98	32271.45	28645.38	24706.27	26616.15	35729.88	44148.15	94553.44
#12	31148.98	32162.24	27062.71	22955.62	24388.63	29349.74	33641.41	53830.7
#13	21694.15	20806.04	20095.8	20473.62	24945.27	37548.87	54190.14	182940
#14	16465.4	18082.01	18847.64	20292.97	25163.15	45218.92	65033.03	123760.6
#15	30187.47	26549.73	22292	20906.86	24229.46	31020.9	46309.75	103514
#16	26392.23	21311.29	18103.26	17883.93	20776.21	31377.4	45987.75	83918.28
#17	17861.8	17689.75	16807.37	16662.75	18873.12	28632.39	33551.98	35278.6
#18	17973.4	18575.96	17304.96	17239.57	19327.44	28905.09	34122.52	38354.42
#19	18750.01	19882.94	19075.09	20038.12	24924.18	41016.17	70220.76	88324.88
#20	27493.77	26042.62	22332.22	21590.18	24977.18	39522.88	61843.45	171885.7
#21	16733.07	16730.26	17040.82	18847.27	25704.68	61748.58	197134.1	-142059
#22	17528.88	19145.94	19360.56	18801.41	21650.49	34286.59	35840.47	37294.15
#23	19681.31	24059.58	25127.65	23085.77	27943.09	47483.96	57362.66	220908.9
#24	16171.43	17386.51	17299.83	17133.38	19359.35	32319.59	31963.49	32942.13
#25	18216.29	20906.95	20686.05	19988.39	22674.82	36506.39	44061	51250.55
#26	20851.54	20920.92	19942.71	19873.1	23122.82	35183.17	43808.51	54417.18

Table 23 EKF Estimation

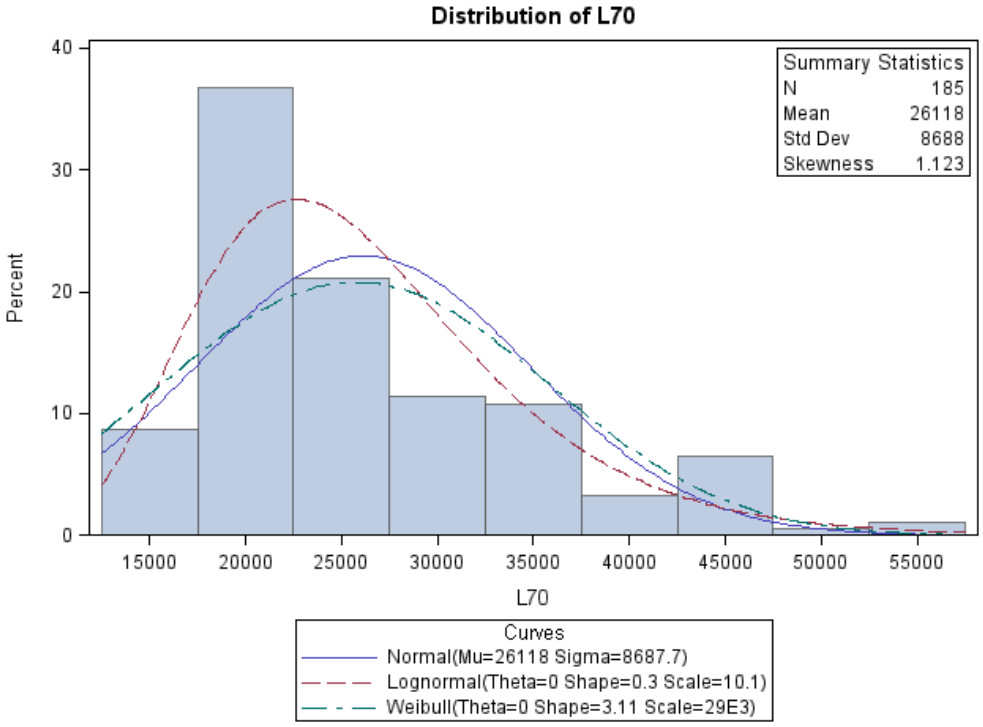


Figure 309 KF Distribution of L70

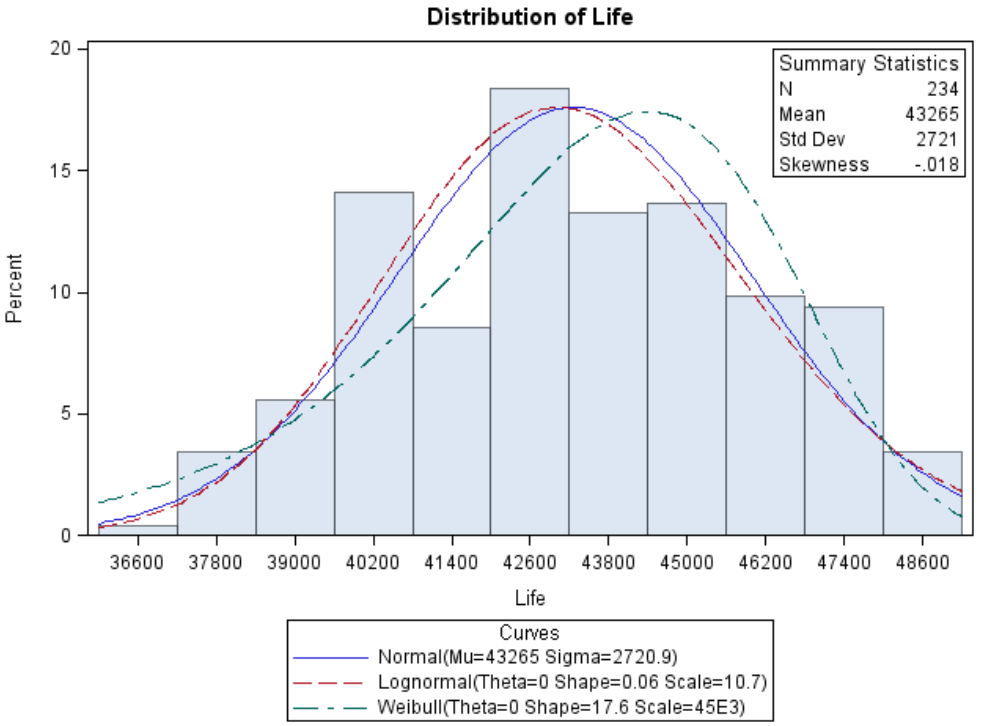


Figure 310 EKF Distribution of L70

7.4.10 KF Pseudo Life Distribution Approach

7.4.10.1. Normal Distribution

The KF prognostic pseudo L70 life follows the normal distribution with expectation μ (26118) and variance σ (8687.7).

$$F(t) = \Phi[(t - 26118)/8687.7] \quad (324)$$

The KF reliability function, R(t), can thereby be written by:

$$R(t) = 1 - F(t) = 1 - \Phi[(t - 26118)/8687.7] \quad (325)$$

7.4.10.2. Lognormal Distribution

The KF cumulative distribution of LEDs' L70 life would be written as the following function:

$$F(t) = \Phi\{[\log(t/10.122)]/0.3045\} \quad (326)$$

In the KF prognostic L70 distribution, we found that R(t) could be represented as:

$$\begin{aligned} R(t) &= 1 - \Phi\{[\log(t/10.122)]/0.3045\} \\ &= \Phi\{-[\log(t) - 24877]/0.3045\} \end{aligned} \quad (327)$$

7.4.10.3. Weibull Distribution

The KF L70 life follows the Weibull distribution with the shape parameter β (i.e. 3.11) and scale parameter α (i.e. 29000).

$$F(t) = 1 - e^{[-(t/29000)^{3.11}]}, t > 0 \quad (328)$$

The reliability function, R(t), can thereby be written by:

$$R(t) = 1 - F(t) = e^{[-(t/29000)^{3.11}]}, t > 0 \quad (329)$$

7.4.11 EKF Pseudo Distribution Approach

7.4.11.1. Normal Distribution

The EKF prognostic pseudo L70 life follows the normal distribution with expectation μ (i.e. 43265) and variance σ (i.e. 2720.9).

$$F(t) = \Phi[(t - 43265)/2720.9] \quad (330)$$

The EKF reliability function, R(t), can thereby be written by:

$$R(t) = 1 - F(t) = 1 - \Phi[(t - 43265/2720.9)] \quad (331)$$

7.4.11.2. Lognormal Distribution

The EKF cumulative distribution of LEDs' L70 life would be written as the following function:

$$F(t) = \Phi\{[\log(t/10.7)]/0.06\} \quad (332)$$

In the LEDs' L70 distribution, we found that R(t) could be represented as:

$$\begin{aligned} R(t) &= 1 - \Phi\{[\log(t/10.7)]/0.06\} \\ &= \Phi\{-[\log(t) - 44356]/0.06\} \end{aligned} \quad (333)$$

7.4.11.3. Weibull Distribution

The L70 life follows the Weibull distribution with the shape parameter β (i.e. 17.6) and scale parameter α (i.e. 45000).

$$F(t) = 1 - e^{[-(t/45000)^{17.6}]}, t > 0 \quad (334)$$

The reliability function, R(t), can thereby be written by:

$$R(t) = 1 - F(t) = e^{[-(t/45000)^{17.6}]}, t > 0 \quad (335)$$

The Normal Distribution, Lognormal Distribution and Weibull Distribution reliability function are shown below in the figures: the red line is the Log-normal reliability function of L70 life, the green line is the Normal reliability function of L70 life, and the blue one is Weibull reliability function of L70 life. Overall, in Kalman Filtering prognostic result, the lognormal distribution could be the best fitting and description of distribution for LEDs pseudo-L70 life. However, in the Extended

Kalman Filtering prognostic result, the normal distribution could be the best fitting and description of distribution for LEDs L70 life. Therefore, we apply the normal distribution in this situation to describe the LEDs' reliability. The figures below describe the 95% confidence intervals for the lognormal distribution and the 95% CI of the normal distribution. The parameters for the lognormal and normal distribution fitting are estimated by the Minimum Distance Estimation, which has been listed in the tables before.

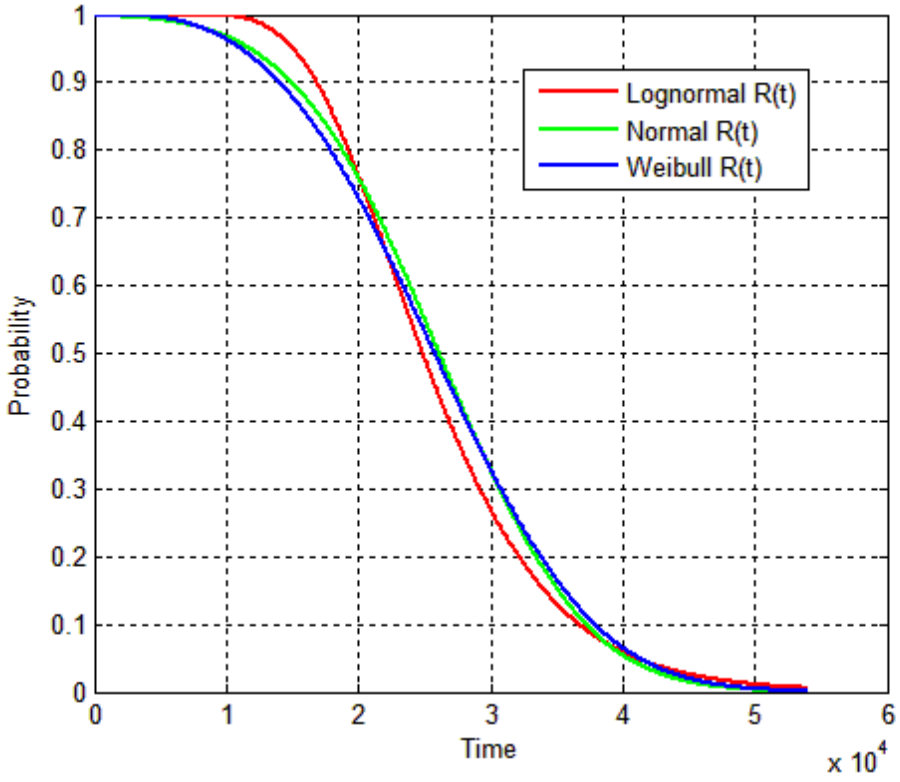


Figure 311 KF Three Distribution Reliability Functions

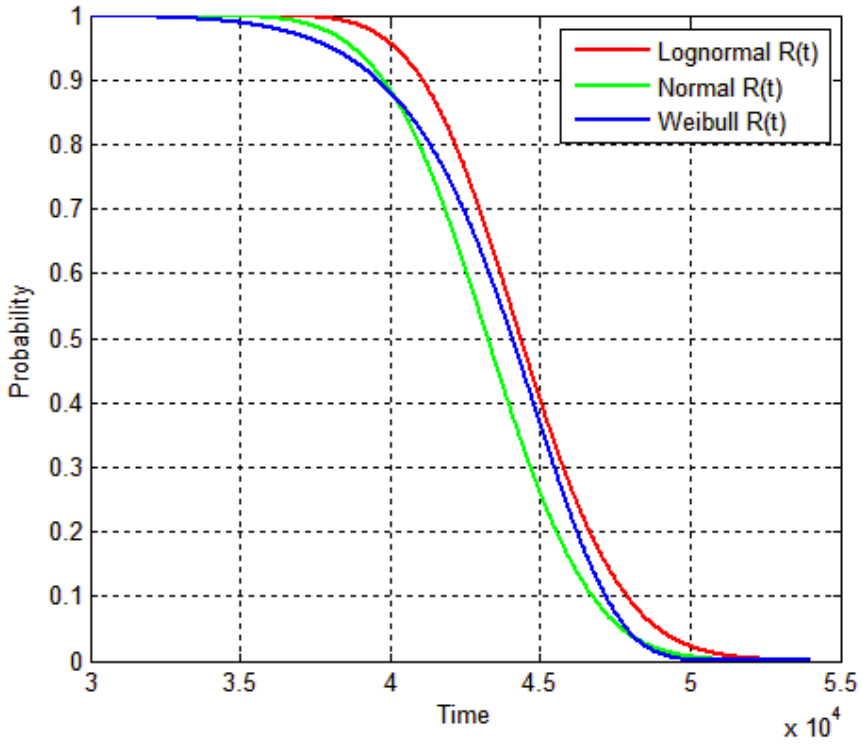


Figure 312 EKF Three Distribution Reliability Functions

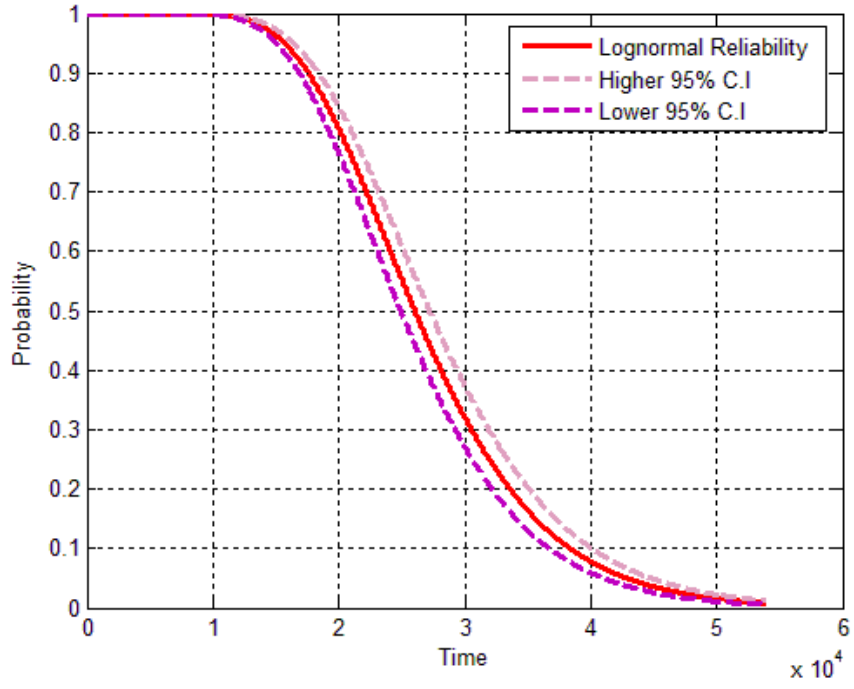


Figure 313 KF Lognormal Reliability Function with 95% CI

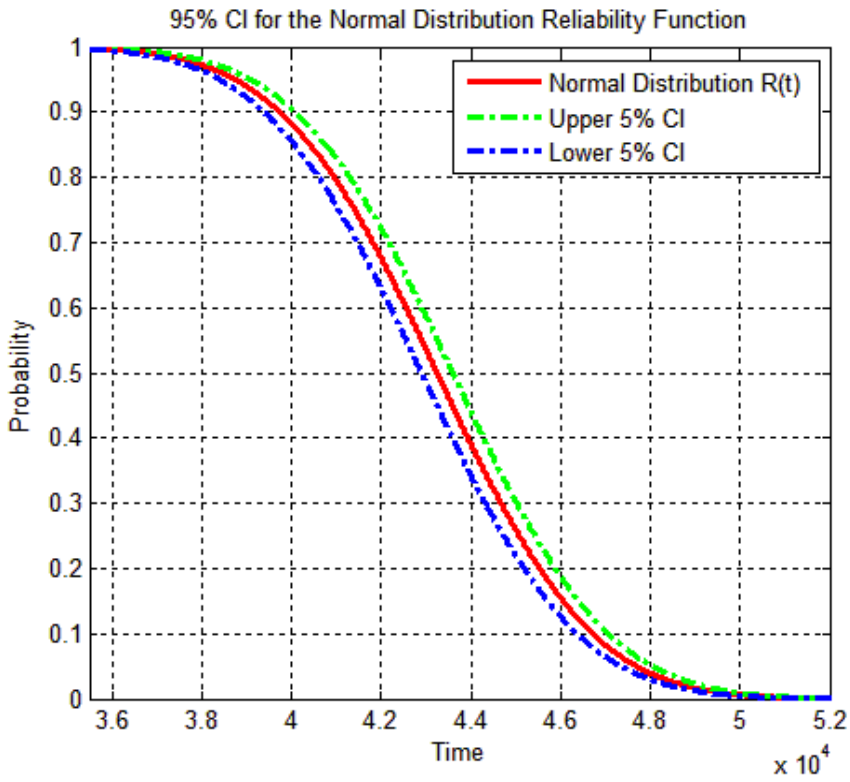


Figure 314 EKF Normal Reliability Function with 95% CI

7.4.12 CONCLUSIONS

In this section, the original method TM-21 has been introduced to calculate the L70 life for LEDs' system. Namely, the TM-21 is the L70 calculator that yields single estimation upon the different temperatures. It uses the exponential extrapolations basing on the LMS curve-fitting with the Activated Energy from the Arrhenius Equations. Then, we introduce the KF/EKF algorithm to estimate the pseudo L70 from different system models. The KF/EKF algorithm makes the projections and uses those projections to update the L70 estimations dynamically. When enough estimations having been updated, the L70 will converge at really small region and finally get the true estimations. Then, we can compare those two methods with Philips six thousand hours' data to make a corrective estimation on the pseudo L70 lifetime. The L70 Lifetime from TM-21 is about 36,000 hours; the KF/EKF estimations are range from 26,000 to 40,000 hours for the LEDs with

a different model. Overall, the KF/EKF can make the prediction according to the many identified system models, and it also can precisely yield the trustable estimations from converging L70 value.

7.5 LED Lumen Maintenance Tracking Using Bayesian Network System

The lighting industry is undergoing a change from the incandescent lamps and compact fluorescent lamps (CFL) to light emitting diodes (LED). Mercury widely used in CFLs has the potential of contaminating large amounts of drinking water to beyond drinkable levels even in trace amounts. Transition to LEDs can impact energy efficiency tremendously because nearly 17% of the annual energy consumption is used for lighting. LEDs are found applications in a wide variety of fields including automotive lighting, LED displays, street and home lighting. Traditional methods of failure detection often used for identification of failure in incandescent lamps may not apply to LEDs.

Currently, it is not possible to qualify SSL lifetime of 10-years and beyond often necessary for high-reliability applications, primarily because of lack of accelerated test techniques and comprehensive life prediction models. SSL comprises of several length scales with different failure modes at each level. Interactions between optics drive electronics, controls, and thermal design. Accelerated testing for one sub-system may be too harsh for another sub-system. New methods are needed for predicting SSL reliability for new and unknown failure modes. Presently, there is a scarcity of life distributions for LEDs and SSLs which are needed to assess the promised lifetimes.

In this chapter, a new method has been developed for assessment of the onset of degradation and classify failure mechanisms in luminaries by using metrics beyond lumen degradation that are currently used for identification of the failure. Luminous Flux, Chrominance-Luminance, Correlated Color Temperature Data on Philips L-Prize Lamps has been gathered under wet high temperature operating life (WHTOL) till lamp failure. Complete Weibull distribution has

been created for the test L-Prize lamps. The acquired data has been used in conjunction with Bayesian Probabilistic Models. The acquired data has been used to identify luminaires with the onset of degradation much before failure. Also, the lamp with different failure modes can be categorized from the pristine lamp. It is envisioned that the new experimental technique will enable the migration of failure distributions without continue experiment to the L70 life for the manifestation of failure.

7.15.1 Introduction

In the wet high-temperature experiment, the 85 degree Celsius and 85 Relative Humidity, the LED lamp testing data has been generated. Both the Lumen Maintenance and CCT are depreciating on time in the testing. At L-prize L70 hours, the percentage failure threshold of the Lumen Maintenance and CCT has been quantified, and the ranged critical failure boundary can be calculated through the Arrhenius Model (Sau Koh, 2013), and, this critical failure boundary is used for finding out and eliminating the poor quality product in the accelerating test in order to shorten the long term testing hours as well as the period of product development. In this chapter, we have been utilized the Bayesian Method (Bishop & Tipping, 2007), which is based on the learning process and unsupervised training algorithm. It can be used as binary or multiple classifications and linear and nonlinear regression with identified system models. Therefore, it can track the change of dynamic decay of system and predict the unknown space by the extrapolations from the previous information.

There are two vital variables quantifying the degradation of LED lamp system, one is Lumen Maintenance, which characterizes the emitting light intensity, and another is the Correlated Color Temperature (CCT), which depicts the correlated color of emitting light that can be related to the black body radiator. To quantify the system degradation, the numerical value of those two

variables can be tracked and followed to predict the failure time of LED lamps. After that, the prognostic metrics has been constructed and used to validate the prediction of the degradation system, which is similar to the Cross-validation. Finally, the Remaining Useful Life is thereby calculated using the identified system model.

Previously, that basing on the Kalman Filter (KF) and Extended Kalman Filter (EKF) have been developed and introduced into tracking the variables of LED system and monitoring the decaying system. (Lall, Wei, & Davis, 2013). The KF/EKF is a recursive algorithm, which is mainly established on the two noises such as process noise and measurement noise excitation iteration and is used for the noise cancellation (Lall, Wei, & Goebel, 2012). It also has been incorporated into specified system model to predict the RUL of an electronics system, which is a part of the Prognostic Health Management (PHM) of Electronic systems. (Lall 2004, 2005, 2006, 2007, 2008). The early failures will be warned by predicting and reporting the RUL of the system. Thus, appropriate maintaining action can be adopted. What is more, the lifetime (L70) can be predicted at a certain time to determine the sustainability of the products.

In this chapter, a new method, the supervised learning algorithm, is developed for the system prognostication, and the Naive Bayesian Classifier and Linear Bayesian regression algorithm has been adopted, which is founded on a method of parametric distribution propagation. Firstly, the Bayesian classification can elucidate the failure criterion in the accelerating testing, which judges whether the LED lamp matches the industrial standard, and indicates a possible failure mode it will pose by keeping the testing condition. Moreover, the Bayesian regression can follow and fit the tendency of system depreciation, then the prognostic Remaining Useful Life can thereby be calculated by trained regression model. At last, the PHM can be performed based on

this Bayesian network. The Weibull plot of L-prize life has been shown, and the reliability function is given by a fitted Weibull failure distribution.

7.15.2 Test Vehicle

The testing object is Philips Ambient-LED lamp (i.e. L-prize precursor), which is energy efficiency and high-quality LED lamp for the green and brighter future. Compared to incandescent light bulbs, the LED lamp can sustain more than 25,000 hours without interruption that lifetime is 25 times greater than it.

Figure 315 Ambient LED Lamp

For the structure of LED lamp, it is quite different from the traditional incandescent bulbs. The Ambient LED lamp has been made of several parts: (1) Optical part; (2) Luminaire part; (3) Lamp Driver; (4) Housing holder. The optical part contains the lamp lenses and covers that emitting lights can transmit, and it is in the yellow part of the Figure 315. The Luminaire part is

comprised of several Philips LUXEON LEDs. In this Ambient LED lamp, there are totally 18 LUXEON LEDs in the lamp and those have been divided into the three symmetric lamp housings. For the Lamp Driver, it converts the AC voltage into the DC voltage, which is required for LEDs' functionality. The Housing holder is the lamp structure that connects several previous parts.

In this experiment of an accelerating testing, a wet hot temperature operation life (WHTOL) has been established, which applies the Thermotron thermal humidity chamber that can create 85C/85H (the 85 degree Celsius and 85% relative humidity) harsh environment.

Photograph: Thermotron, Inc.

Figure 316 Thermotron thermal humidity chamber

The LED lamp measurement can be accomplished with the LEDs' measuring system. Typically, the lamp measurement system contains parts: (1) Light Emitting Device. (2) Light Gathering System. (3) Light Transmitting System. (4) Light Analyzing System. The light emitting device provides the AC voltage power connection to the LED lamp that is producing continuous

measurable light. The Light Gathering System redistributes and collects the entire light rays emitting from the LED lamp. In general, the Light Gathering System includes the integrating sphere and cosine diffuser, etc. The integrating sphere is an optical component that uniformly scatters the light, which has a special coating on its surface of inside sphere. With small output ports on the side of the sphere, the LED lights can be transmitted through the cosine diffuser, which is a detector, filtering and transferring the distributed light to the cable optical fiber. Then, the light will be carried into the Labsphere 'USB4000' Spectrometer, which is specialized Analog and Digital converter (ADC) for the light system. So, the digital data will be analyzed, and the Lumen Flux and CCT will be shown in the software.

7.15.2 Experimental Set-up

The LED lamp testing has been focusing on the both the Lumen degradation and Chromaticity degradation. As we mentioned before, the Lumen degradation characterizes the drop of light intensity in the history while the Chromaticity depicts that the shifting of Correlated Color Temperature (CCT). The LEDs' lamp CCT is the correlated temperature of emitting the hue that compared with the ideal black body radiator. In the experiment, the nominal CCT value for L-prize is 3000K.

Both the Lumen degradation and CCT shifting have been monitored to watch the lamp system degradation. Therefore, we can know that the failure time is and when measured value of any of those variables to cross the failure thresholds. We have been testing the LED lamp to L70 hours and extended the testing time to 2536.85 hours that beyond to the L70 life. The eight Ambient LED lamps' data has been generated every week. The testing matrix has been shown in the Table 24 below:

Testing Date	Testing Hours	Testing lamps
03/08/13	0.00	8
03/15/13	160.33	8
06/03/13	374.47	8
06/10/13	539.60	8
06/17/13	699.50	8
06/24/13	861.07	8
07/01/13	1022.73	8
07/08/13	1184.13	8
07/22/13	1520.52	8
07/29/13	1662.03	8
08/05/13	1822.23	8
08/12/13	1980.18	8
08/19/13	2144.87	8
08/26/13	2306.48	8
09/05/13	2536.85	8

Table 24 Test Matrix

7.15.3 Failure Rate Quantification

Generally, the LED lamps contain many luminaires in the phosphor chambers of the LED lamp, and the total of lumen flux emitting comes directly from those small luminaires. Also, those luminaire CCT change would affect the entire lamp CCT outputs, and luminaires failure directly accounts for lots of LED lamps failure. From above, we can relate the various luminaire testing with LED lamp testing and seek for the failure threshold from luminaire level to the whole lamp system.

Previously, the Philips company has released the IESNA LM-80 Test Report for the LUXEON luminaires, which has been used into the L-prize LED lamp. The IESNA LM-80 testing has been conducted at 55°C, 85°C, 105°C and 120°C. The 9000 thousand hours aging data has been reported, which the lumen output and u’ and v’ in the 1976 color space has been measured at the interval of 1000 hours after initial 500 hours warming up. From the former KF/EKF L70 analysis, there is a shifting about the u’ and v’ in color space. It is expected that increasing u’ and

v' would reversely cause the decrement of Correlated Color Temperature (CCT). So, the CCT would be dropping to certain level of percentage as the LED system goes to failure. Similarly, the lumen maintenance will also decrease in the temperature aging test. The lumen maintenance dropped about 7.79% after 8000 hours 105°C and 1A testing that is the most severe case in the experiment. We can see that the pure temperature socking test have only caused a small amount of decrement for the lumen maintenance.

However, compared to the Philips luminaire pure temperature sock testing, the L-prize LED lamp testing has the distinct Lumen Maintenance and CCT decreasing pattern that can be identified. In the L-prize testing data, we take a degradation data for analysis; we can see the percentage of CCT has been degraded to the 96% from the 100% at the 2500 hours testing. Significantly, the Lumen Maintenance has been degraded to the 68% after 2000 hours of accelerating test. There is a positive correlation of degradation pattern, joint depreciation, can be identified through the statistic method.

Typically, in the industrial application, the L70 (i.e. 70% Lumen Maintenance) life has been treated as the failure threshold for the luminaire intensity of light. Also, as far as for the CCT failure, the 7-step MacAdam ellipse fairly states that the target 'Δuv' and its tolerance is ±0.006, so the corresponding target failure CCT and tolerance is 3000±175K (ANSI, 2008). Therefore, for the nominal 3000K LED lamp, the CCT critical failure threshold is 94.17% as the CCT of 2825K. In the L-prize testing, the lumen decay is much more significant than the CCT decay, thus we mainly focus on the degradation of Lumen Maintenance for the Remaining Useful Life prognostication.

Since, it is possible that the early failure LEDs in the lamp could cause the Lumen and CCT decrease dramatically. Then, we can find out those 'poor quality' products of early failures

in the well-designed accelerating test. So, by utilizing the Arrhenius Equation model, we relate the pure temperature soak testing to the multi-physics accelerating test, for example, the 85C/85H test, one we used for testing the L-prize lamp.

To determine the Activation Energy (AE) is the key to the Arrhenius mathematic model. The Arrhenius Equation describes the formula that characterizing rate of chemical reaction. Since the LED Luminaire and lamp testing are designed under the various temperatures ranging from 55°C to 120°C. Therefore, the designed acceleration testing can be regarded as a temperature dependent reaction, which causes the system to decay such as lenses degradation, chip degradation, and phosphor degradation as well as LED driver degradation. We will find the Activation Energy for the accelerating testing for the L-prize Lamp, which is shown in the equation below:

$$\alpha = A \cdot e^{-\frac{E_{act}}{k_B \cdot T}} \tag{336}$$

α - Chemical Reaction rate
E_{act} - Activation Energy
A- Pre-Exponential Factor
k_B- Boltzmann Constant
T - Reaction Temperature

We use the Least Square (LS) fitting method to find out the decay rate for the CCT and Lumen Maintenance. For the fitting method, the curve fitting model applies the exponential model. Because in the basic exponential model $\Phi = \beta e^{\alpha t}$, the depreciation model only depends on the pre-decay factor 'beta' and decay rate 'alpha', and it is known that the TM-21 calculator also adopts such fundamental exponential model as well. The degradation line starts at 100%, thus the 'beta' takes '1' as the initial. Therefore, the decay rate would be the deterministic factor that determines the CCT and LM depreciation curve. So, in real problem, each sample has its distinct

decay rate that can be determined in the characteristic depreciation curve. We can see it in Figure 317 and Figure 318.

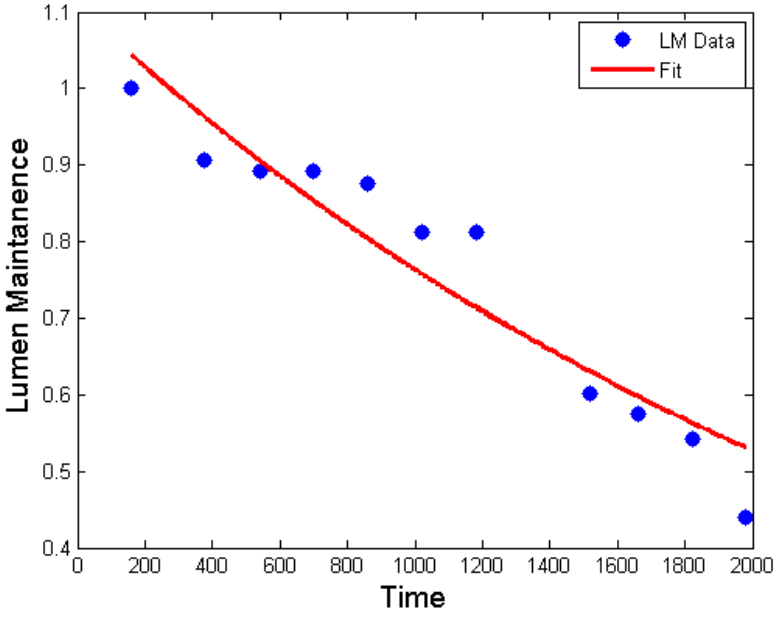


Figure 317 Lumen Maintenance Fitting

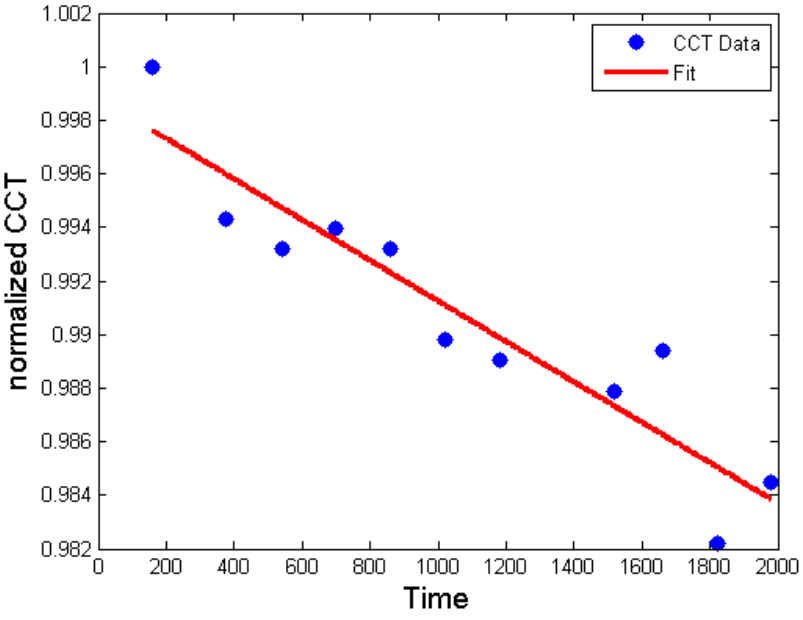


Figure 318 Correlated Color Temperature Fitting

The 55°C, 1A Philips dataset and L-prize lamp 85 °C/ 85H dataset have been used to construct the failure threshold in the high temperature and humidity situation for testing the LED lamp. The Philips 55°C, 1A dataset has 25 samples, after 9000 hours, the Lumen Maintenance degrades at the 94.08% from the pristine, and the CCT is degradation at 98.25% from 3000 hours to 9000 hours. Then, we will use curve-fitting to fit those depreciating data, and get its decay rate and the alpha of the exponential model for both LM and CCT. We also assume the pre-decay factors take the same constant. In the Philips dataset, the alpha mean for the Lumen Maintenance is -5.1452e^{-06} unit/hours, and for the CCT is -1.2812e^{-06} unit/hours, and the QQ plot (Figure 319) checks the normality of the alpha for both LM and CCT. The blue line shows the quantile of normal distribution, the red dots show the LM and CCT data. We can see the data is normally distributed except two or three outliers of noise data. Therefore, we will reasonably use the mean value of decay rate 'alpha' to represent the degradation rate of reaction under the designated

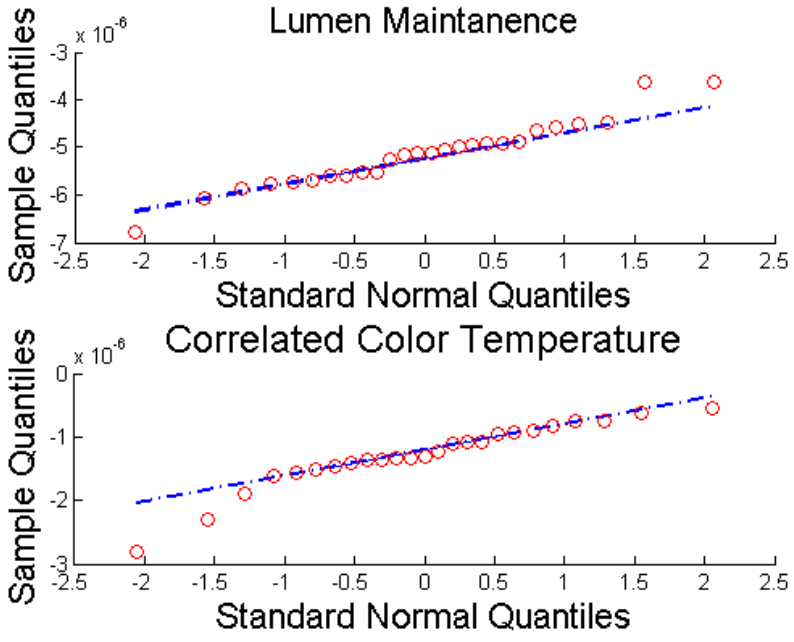


Figure 319 QQ plot

Similarly, for the L-prize dataset, the alpha of LM and CCT also follows the Gaussian distribution. The mean value of alpha for the Lumen Maintenance is -4.0544e^{-04} unit/hours, and for the CCT is -6.3055e^{-06} unit/hours. Since we assume that those decay rates from each sample will follow the Gaussian distribution with its sample mean and variance, then the mean value of decay rate will be regarded as the center and present the standard degradation curve in its specific testing condition. Using this ideal, we can find the center and standard degradation curve in both the lower temperature case and higher temperature case. Finally, we can use the Arrhenius model to calculate the Activation Energy.

It is the calculated Activation Energy that will be applied into calculating the higher temperature decay rate to derive the lower one. Therefore, we firstly choose the judgment and decide failure criterion of decay rate in the long term degradation (i.e. the luminaire testing). Then, we can decide the decay rate for the short term degradation. Therefore, the higher temperature failure criterion is proportional to the lower temperature case, which is the ratio of the mean value of two decay rate; it shows in the Equations:

$$\alpha_i = 1 \cdot \mathrm{e}^{-\frac{\mathrm{E}_{act}}{\mathrm{k_B}} \cdot \left(\frac{1}{\mathrm{T}_i}\right)} \tag{337}$$

$$\frac{\alpha_1}{\alpha_2} = 1 \cdot \mathrm{e}^{-\frac{\mathrm{E}_{act}}{\mathrm{k_B}} \cdot \left(\frac{1}{\mathrm{T}_1} - \frac{1}{\mathrm{T}_2}\right)} \tag{338}$$

The Activation Energy that between the 55°C (i.e. Luminaire Testing) and 85°C/85H data (i.e. Accelerating Life Testing) can be calculated using the Arrhenius Equation below:

$$\mathrm{E}_{act} = -\frac{\ln \frac{\alpha_1}{\alpha_2}}{\left(\frac{1}{\mathrm{T}_1} - \frac{1}{\mathrm{T}_2}\right) \cdot \mathrm{k_B}} \tag{339}$$

Substituting the two mean decay rates, α_1 = -5.1452e^{-06} and α_2=-4.0544e^{-04}, T1= 328.16K, T2= 358.15K, k_B= 8.617385e^{-05} eV/K, we can find the Activation Energy of Lumen Maintenance degradation is 1.4748. Likewise, the Activation Energy of CCT degradation is 0.5328. We will use those two standard Activation Energies for deciding the failure boundary of the CCT and the LM.

The failure decision boundary has initially been established in the long term case, and by utilizing the standard Activation Energy, the failure decision boundary in the accelerating life test can be thereby decided. Therefore, in the future, we can use this failure criterion in the acceleration test. Thus, the earlier failure LED lamps can be found and selected. This method generally reduces the testing time required for reaching long term L70 Life.

The failure criterion in the long term degradation case takes the curve of the maximum absolute decay rate, which will envelop all the degradation line in the case, we can see it in the green line from Figure 320. The mean value of maximum decay rate is -2.8060e^{-06}, and we apply the 95% Confident Interval (CI) for the maximum decay rate in order to decide the highest decay (i.e. lower boundary) that will characterize failure criterion.

$$\alpha_{LM}^{failure} = \alpha_{\max} - 1.96\frac{\sigma}{\sqrt{N}} = \text{-7.04e - 06 (lumen/hours)} \tag{340}$$

$$\alpha_{CCT}^{failure} = \alpha_{\max} - 1.96\frac{\sigma}{\sqrt{N}} = \text{-3.01e - 06 (cct/hours)} \tag{341}$$

Then, the failure criterion for the L-prize lamp can be calculated by Arrhenius model.

$$L\alpha_{LM}^{failure} = \alpha_{LM}^{failure} \cdot \mathrm{e}^{-\frac{E_{act}}{k_B}\left(\frac{1}{T_1}-\frac{1}{T_2}\right)} = \text{- 5.55e - 04 (lumen/hours)} \tag{342}$$

$$L\alpha_{CCT}^{failure} = \alpha_{CCT}^{failure} \cdot \mathrm{e}^{-\frac{E_{act}}{k_B}\left(\frac{1}{T_1}-\frac{1}{T_2}\right)} = \text{- 1.48e - 05(cct/hours)} \tag{343}$$

The time dependent decay curve for the L-prize shows below, the bold red line is the mean decay rate. It is obvious that the CCT decays almost linearly due to the small decay rate. However, the Lumen Maintenance exhibits much more exponential pattern. The green dash line in the Figure 320 shows the failure criterion for the L-prize Lamp. We can see that all of CCT degradation lines are within the requirement, and for the Lumen Maintenance there is one sample on the failure boundary.

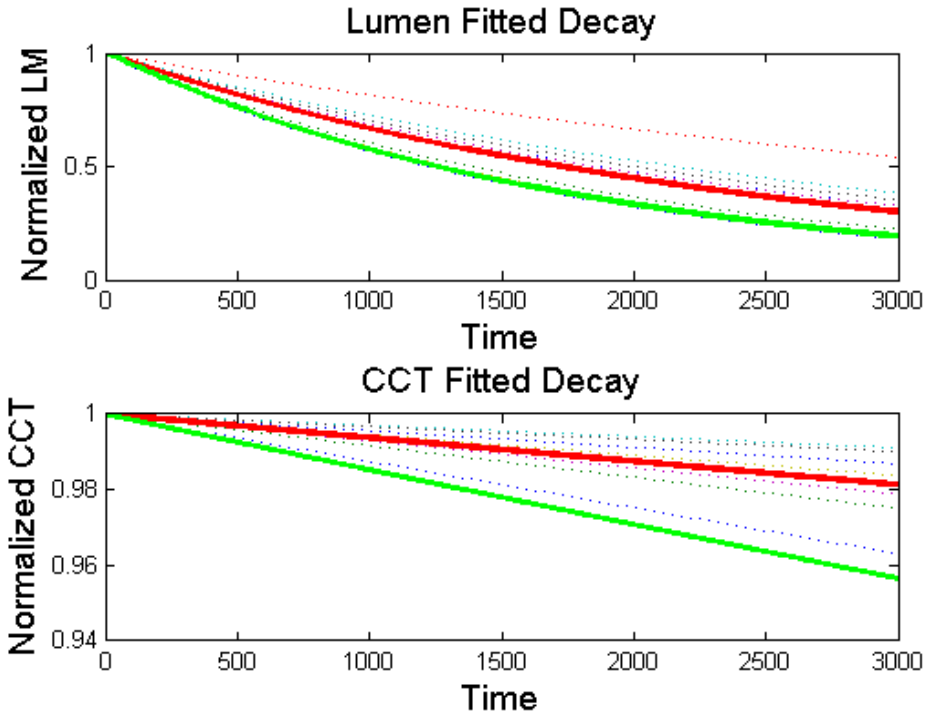


Figure 320 Characterized Decay Curve

7.15.4 Failure Detection 85C/85H Test

In the 85C/85H LED lamp testing, the failure criterion for the decay rate can be calculated using the Activation Energy from Arrhenius Model. Then, we can construct the theoretical failure boundary that both have the Lumen Maintenance and CCT decay feature to determine the failure

lamp. So, those poor quality lamps could be eliminated from life prediction and inspection in the long run. It is the Bayesian Classifier Algorithm that can help us to build such a deterministic boundary, which separates the health and failure lamps in the Lumen and CCT feature space.

First of all, the critical failure boundary, characterizing the maximum decay rate, decides the border between the high and low quality lamps in the accelerating L70 life test. In the designed 85C/85H testing, all the L-prize lamps have been running until the initial failure occurred, which can be either from the Lumen depreciation or color shifting. Then this failure time will be regarded as the Critical Failure time (T_{CF}). Then, from the previous, we already obtain the maximum decay rate, which we can calculate the critical value of Lumen Maintenance and CCT value at T_{CF} using the following formula:

$$LM_{\max failure} = 100\% \cdot e^{L\alpha_{LM}^{failure} \cdot T_{CF}} \tag{344}$$

$$CCT_{\max failure} = 100\% \cdot e^{L\alpha_{CCT}^{failure} \cdot T_{CF}} \tag{345}$$

As we know that the value of maximum Lumen Maintenance and CCT would be located in the LM/CCT coordinator, it is convenient to use the multivariate Gaussian method to generate the appropriate failure distribution with its mean failure value and specified variance for the ranged data of failure group. Also, it will be incorporated with the experimental L-prize data to seek out the appropriate and recognized boundary for determining the failure. What is more, the types of failure in the LED lamps can thereby be discerned by the points of the failure boundary it will probably across.

It applies the distribution fitting tool, the Maximum Likelihood Estimation (MLE) fitting method to find the parametric distributions. It matches the distribution variance between the 85C/85H L-prize data and maximum failure data, also it is convenient and comfortable to use the Bayesian classifier under the principal direction of variance that yields uncorrelated X and Y axis

variances, such as E[XY] =0. Then, after Bayesian analysis, we can find back the origin value using transform matrix. The original covariance matrix for the 85C/85H failure data shows below:

$$\sigma = \begin{bmatrix} \sigma_{11}^{'} & \sigma_{12}^{'} \\ \sigma_{12}^{'} & \sigma_{22}^{'} \end{bmatrix} \tag{346}$$

Then, the transformed variance matrix with uncorrelated variance data simply takes the Eigen values of fitted variance in the principal direction:

$$\sigma_S = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \tag{347}$$

The transformation matrix is the corresponding Eigen vectors:

$$R = \begin{bmatrix} m & -n \\ n & m \end{bmatrix} \tag{348}$$

Then, the mean value in the transformed coordinate can be shown as the following:

$$U_P = \mathrm{RU} = \begin{bmatrix} m & -n \\ n & m \end{bmatrix} \cdot \begin{bmatrix} LM_{Lprize} \\ CCT_{Lprize} \end{bmatrix} = \begin{bmatrix} LM_{\mathrm{Lprize}}^{P} \\ CCT_{Lprize}^{P} \end{bmatrix} \tag{349}$$

Similarly, we have the following equation:

$$U_P = \mathrm{RU} = \begin{bmatrix} LM_{\max\ failure}^{P} \\ CCT_{\max\ failure}^{P} \end{bmatrix} \tag{350}$$

Then, we can show the failure data migration plot, which indicates the trends from the pristine to the failure. In the Figure 321, the three fitted distributions have been generated using the Multivariate Gaussian distribution from the MLE method.

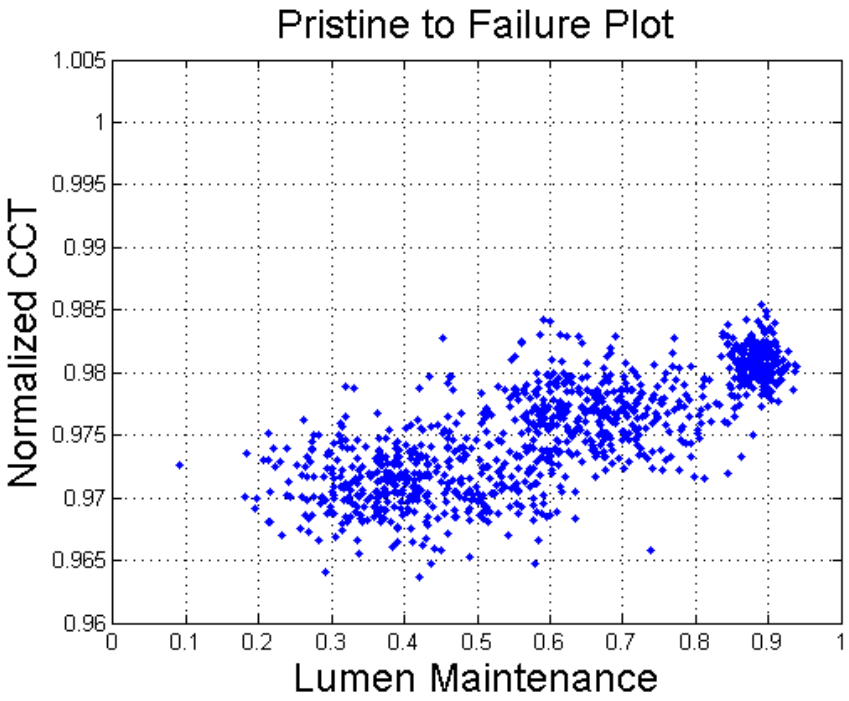


Figure 321 Fitted Depreciation Data

From the figure above, there is distinguished Lumen and the CCT depreciation ranging from the center at [0.9, 0.98] to [0.4, 0.97]. The data groups have L-prize pristine data, L-prize failure data and the maximum allowance distribution of LM/CCT depreciation generating from the critical failure boundary. We can classify the groups and decide and discipline of failure threshold using the Bayesian Classification Method, which is an unsupervised learning method, and it will learn and train the online data as long as there is continuous input into the training algorithm.

Moreover, other than finding the failure criterion boundary of accelerating test, the Bayesian learning algorithm can also be used for predicting and quantifying the Remaining Useful Life of LED lamp system. For example, it uses the quadratic parametric model and trains its coefficients to reach the convergence, then the trained model and its 'trajectory' will follow the

degradation curve. We will find that the Bayesian regression algorithm will track the decaying curve very well. It will be anticipated that the model of prediction can be validated through the cross-over validation. Overall, it is very intuitive learning algorithm that can be used for monitoring and supervising any decaying variable of the certain system.

7.15.5 Failure Analysis Results

Once the Bayesian classifier has finished the training process, we can process to check the classification results. In the Figure 323, it shows the failure data for the L-prize, and the initial degradation starts around the center at [1, 0.99], and migrates to [0.4, 0.98]. It is clearly to see that the degradation trend goes from the top right corner to the bottom left corner. Also, the red dash line shows the failure threshold between the healthy L-prize data and failure L-prize data.

For binary classification problem, there are two groups of data in this plot, and each of them has its own characterized means and variances. Since the points on the red line yields the equivalent posterior probability or the equal Mahalanobis distance from both the center, the minimum value of M-distance reveals the most likely failure point in the testing. Of course, this threshold failure line also helps us to judge whether the measured data belongs to the failure group.

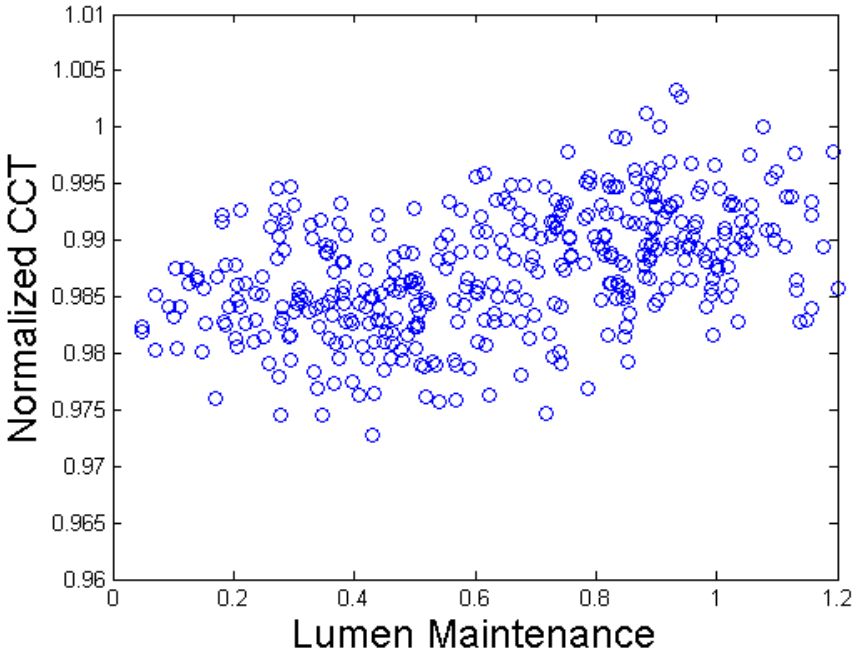


Figure 322 Pristine and Failure groups

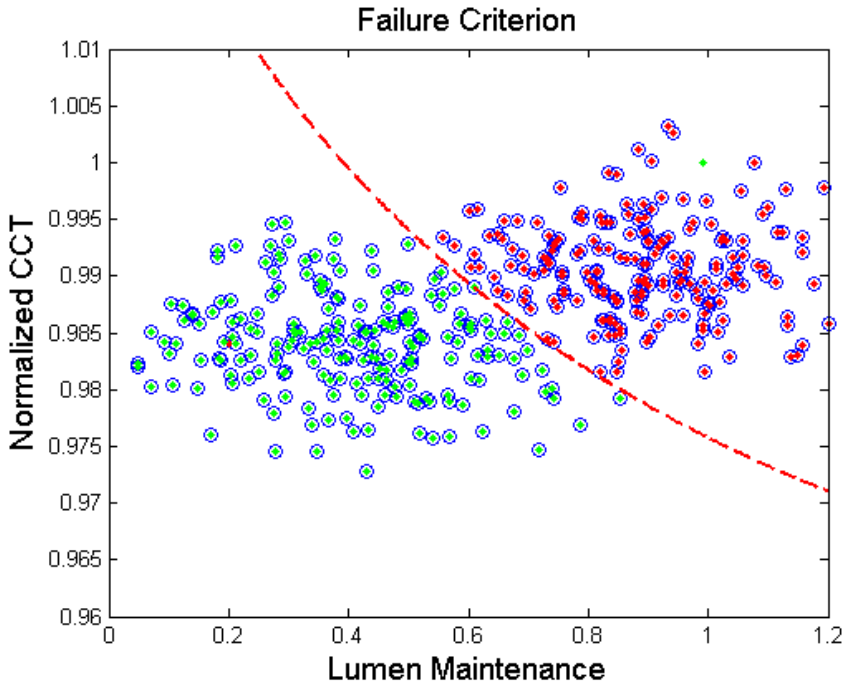


Figure 323 End of Life distribution and pristine

The classification decision boundary for the failure threshold can be calculated from previous analysis as equating the posterior Probability Density Functions (PDF):

$$\begin{aligned} \mathrm{G}(x,y) &= F_1(x,y) - F_2(x,y) = 0 \\ &= 3.72x^2 - 157.74xy + 131.11x + \\ &\quad 1823.78y^2 - 3984.71y + 2170.75 \end{aligned} \tag{351}$$

From the classification results, the trained weights for the failure group are:

$$W0 = -2.7019 \cdot 10^4 \tag{352}$$

$$W1 = \begin{bmatrix} 0.0305 \\ 5.4802 \end{bmatrix} \cdot 10^4 \tag{353}$$

$$W2 = \begin{bmatrix} -0.0016 & -0.0149 \\ -0.0149 & -2.7783 \end{bmatrix} \cdot 10^4 \tag{354}$$

Consequently, the following Figure 324 provides us an intuition about the class distributions: (1) Healthy Group (2) Failure Groups. The red PDF is corresponding to the failure L-prize data while the green PDF is the healthy L-prize data. The overlapping area displays the transition failure area, and if the generated testing data under this area, it is more likely that the system will decay quickly to pass into failure class, and then the maintenance and replacement should be takes care of. Typically, it requires this overlapping PDF region to be as smaller as possible in order to make timely decision, if so it requires the variances of two PDF to be smaller. This will be achieved through the training data to be specialized and concentrated at the location we interested in.

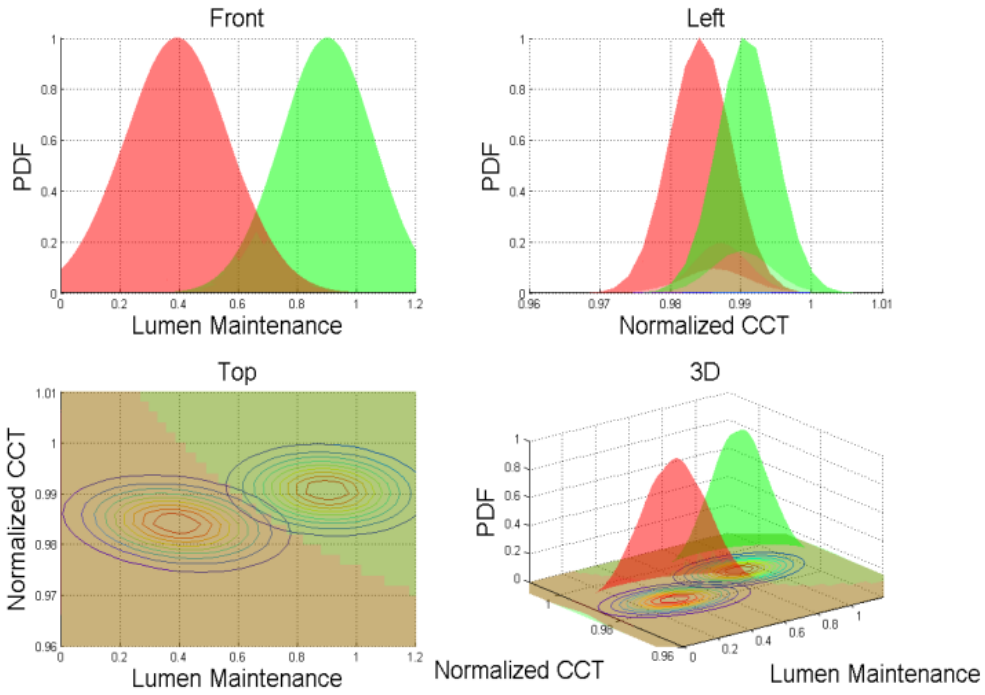


Figure 324 Failure and Pristine PDF distribution

Overall, the Bayesian unsupervised classifier is powerful classification tool, because it does not need to be systematically trained model and there is no necessary to create mapping function between the input space and the output space. Therefore, any online data and incoming data can join into the unsupervised training process. The algorithm will learn from the prior distribution and decide which class the data belongs to.

Apart from the binary classification, we can also classify multiple groups with the critical failure boundaries as we mentioned before. Similarly, in the multiple classification problems, we can specify the group number, which will allow us to track the degradation between the classes. As the data going into the different classes, the decision boundary will be updated, which allows us to modify and improve the failure criterions with respect to the testing time. The following Figure 325 shows the three groups classification, which we assign the initial parametric

distributions for the critical failure group, the pristine L-prize group and failure L-prize group. Lastly, the testing data have been grouped, and Bayesian Classifier calculates the mean and variance numerically.

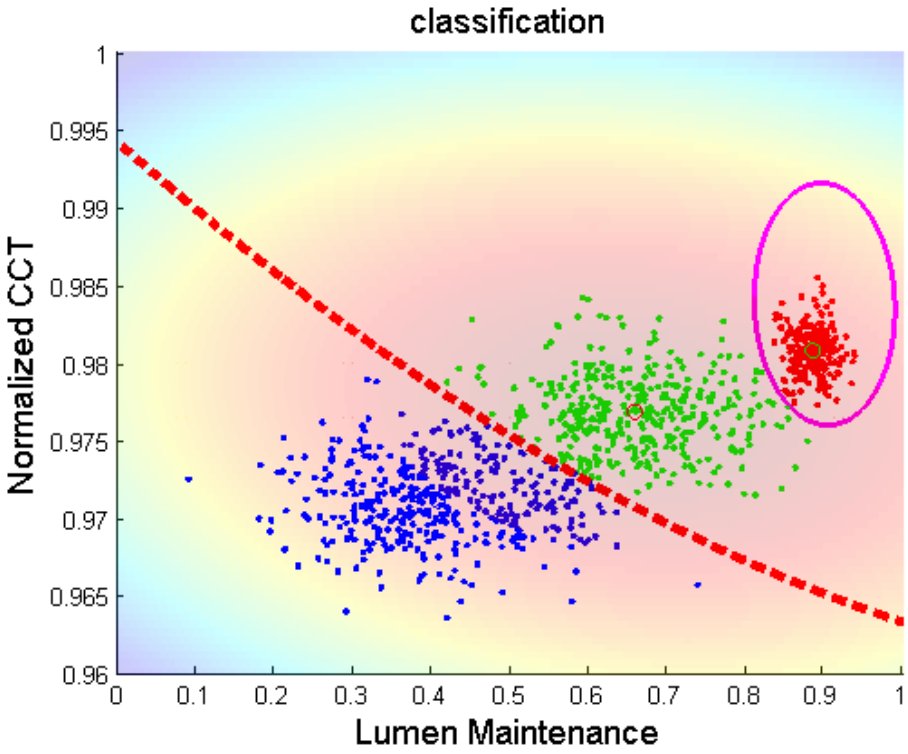


Figure 325 L70 Time Decision Boundary

In the Figure 326, it clearly demonstrates that the migration of decision boundary. The failure decision boundary between pristine L-prize group and failure L-prize group shows on the solid magenta line, and the critical failure boundary shows on the dash red line. The critical decision boundary implies that the Lumen Maintenance and CCT shifting for the good quality products should not exceed the critical decision boundary at the L70 time of accelerated life testing. It also elucidates that as the failure decision boundary tends to be wider and wider so the variance of data increasing. The Figure 326 demonstrates the three PDFs for the L-prize pristine, failure

group as well as critical failure group. As more failure data imports into the L-prize failure group, the decision boundary will be much more precise and deterministic.

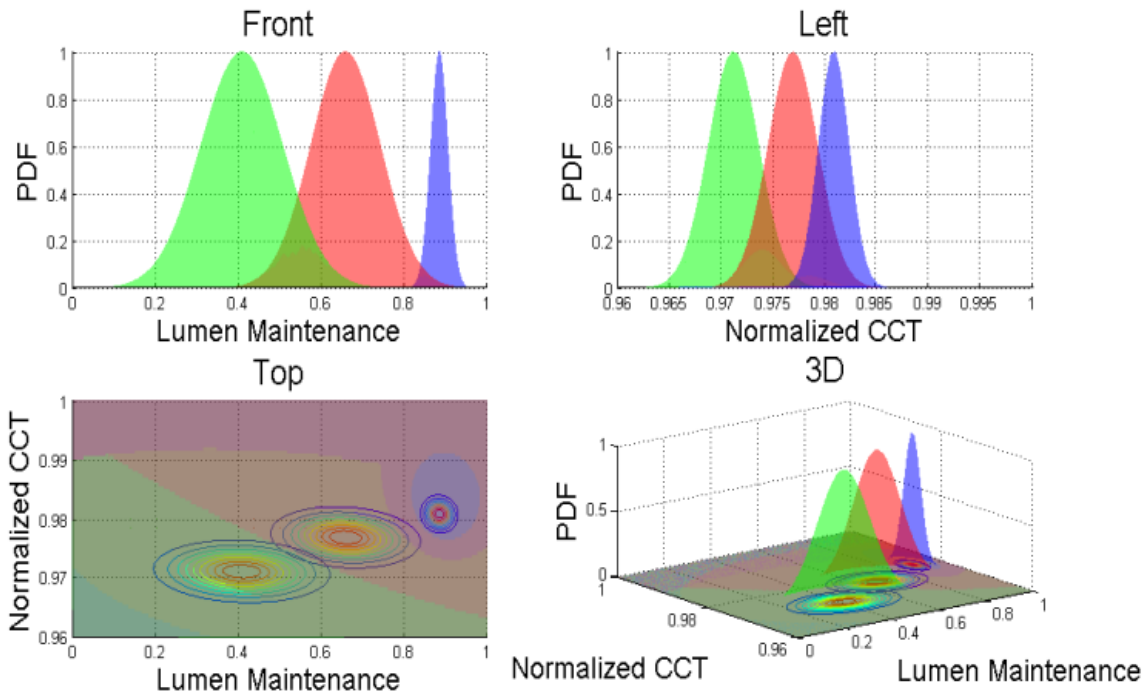


Figure 326 Critical Failure, L-prize Failure and Pristine PDF Distribution

7.15.6 Bayesian Regression for the system Remaining Useful Life

We can also use the Bayesian regression method to determine the Remaining Useful Life (RUL) for specified LED lamp. In this testing, because the lumen degradation is much more significant and quicker than the color shifting, we utilize the Lumen Maintenance (LM) degradation as the main indicator of system decay, which by fitting the Lumen Maintenance degradation curve, so the future trend can be predicted using this Bayesian curve fitting method.

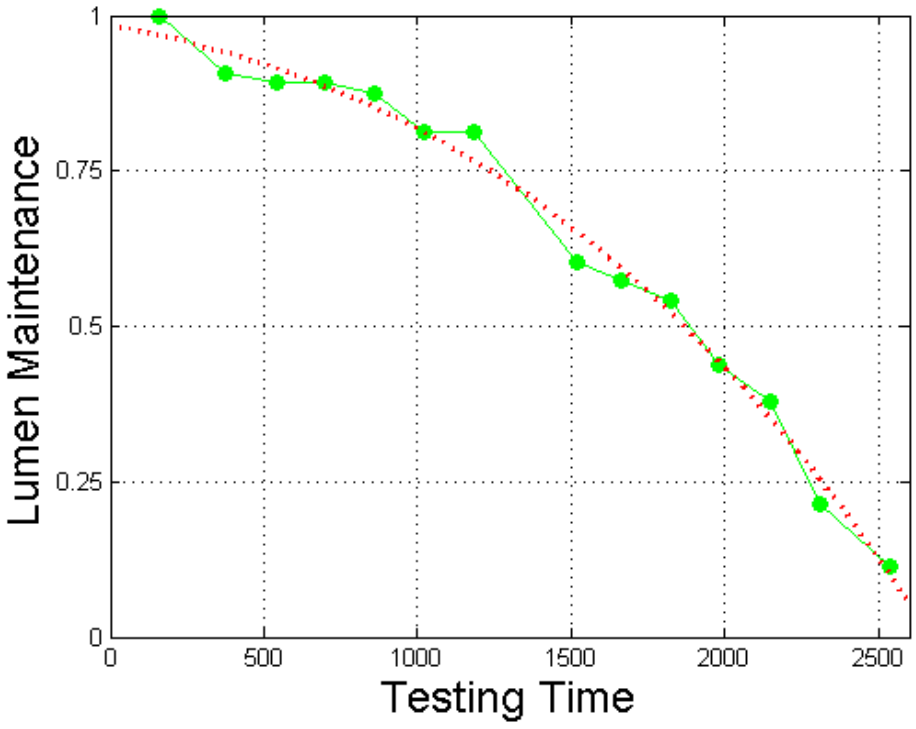


Figure 327 Lumen Maintenance Regression Lamp #2

First of all, it applies the linear Bayesian model (i.e. polynomial function) to track the degradation curve and predict the Remaining Useful Life (RUL) of the system. In the Bayesian linear regression method, the predicted Remaining Useful Life has been calculated at each measuring point. In addition, the true Remaining Useful Life can be checked and calculated at the time of system failure. Therefore, we can compare the predicted Remaining Useful Life with true Remaining Useful Life in the LED system to verify our predicting models.

The Bayesian Regression can be adapted to the dynamic system. The prior distribution provides the information that we already know about the system, and the posterior is affected by the combined PDFs from the prior distribution and likelihood function. Therefore, the old information would be viewed as the prior impacting the posterior. Also, the previous time posterior will become the next time prior. So, as the system time moving forward, the continuous incoming

new data would make the posterior distribution deterministic. The following Figure 328 presents the evolution of Lumen Maintenance degradation and predicting curve. Since more data has been utilized into modeling the feature of decay, the predicting curve fits the experimental data exactly.

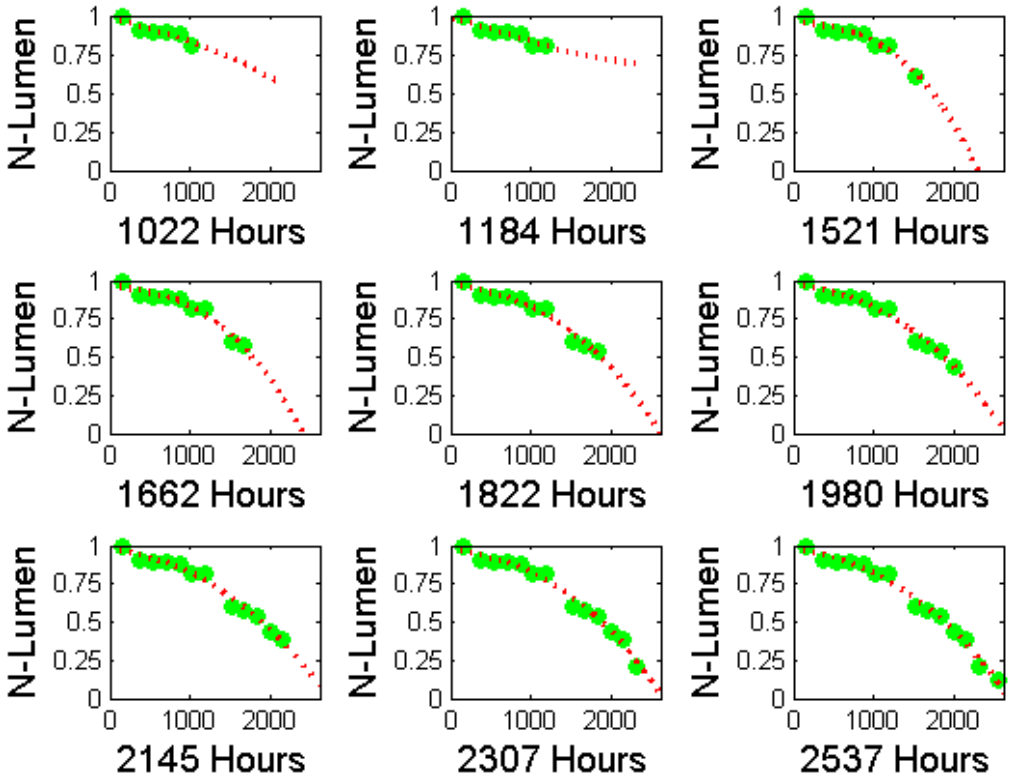


Figure 328 Bayesian Regression Unsupervised Learning Progresses

Quantitatively, given the identified value of system white noise (i.e. the variance) for the distribution that is adjustable and identified to distinct system, each weight for the model coefficients can be calculated. The following Figure 329 shows the Bayesian linear regression for the third order polynomial model. The green dots are the data points, and the red dots show the predicting decay curve. The testing time is up to 2537 hours. After testing, the percentage of Lumen Maintenance is dropping from 100% to 10%.

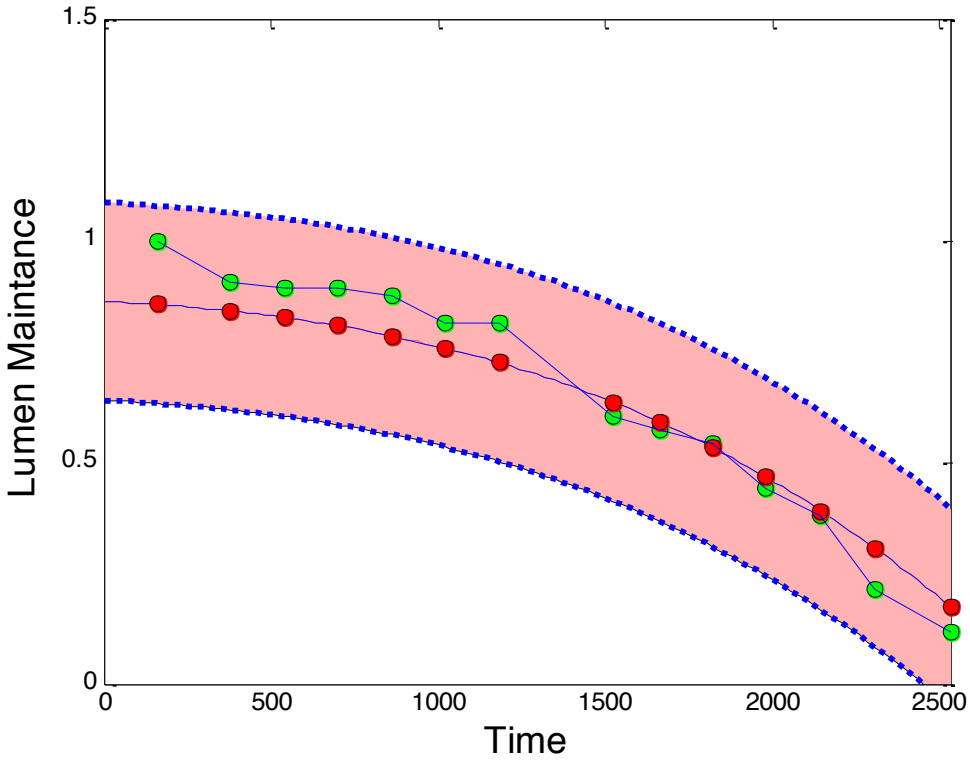


Figure 329 Bayesian Regression with Confident Interval

The Remaining Useful Life (RUL) can be calculated by the third order polynomial model. It is extrapolating the current prediction curve into the future space. There is a tendency of convergence for the coefficients of polynomial model, and the predicted RUL will finally catch up with the true RUL. The Figure 330 shows the predicting results. The red line is the actual Remaining Useful Life of L-prize system. The blue dash line is the prognostic Remaining Useful Life that predicted by extrapolation. The Green dash line is the ±20% Confident Interval, which the effective prediction should be within this boundary. As we can see, the actual prediction starts at 500 hours. The prognostic RUL line always matches the actual RUL line. At very end of the experiment, the RUL value is very small. So, the predicting result is very sensitive, and the

prediction will become very difficult. Therefore, in order to reduce the cost and replace the near failure components, we need to take an action before it falls into such an area.

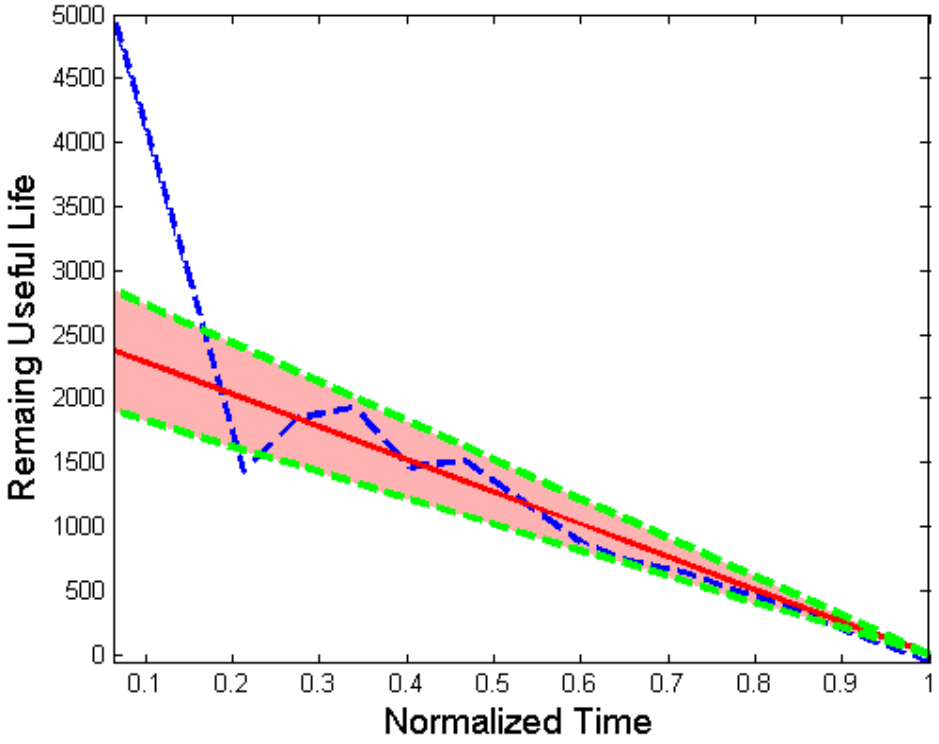


Figure 330 System Prognostic Remain Useful Life and Prognostic Verification

7.15.6 The Weibull Distribution of L70 Life

It introduces the Weibull distribution to describe the life of roller bearing, electronic components, ceramics, capacitors and dielectrics in the accelerated test (Abernethy,2000). Therefore, the Weibull probability plot of L-Prize L70 Life data has been generated. It has been demonstrated on the Figure 331 below. It uses the two parameters Weibull distribution to illustrate the failure time distribution, the Mean Time to Failure (MTTF). The two parameter Weibull distribution has been given as following PDF:

$$f(t) = \frac{\beta}{\eta}\left(\frac{t}{\eta}\right)^{\beta-1} e^{-\left(\frac{t}{\eta}\right)^{\beta}} \quad (355)$$

β is the shape (slope) parameter

η is the scale parameter

The characteristic life that is defined as the age at which 63.2% of the units will have failed. The estimated coefficient for the shape parameter and the scales parameter are: β = 7.1 and η = 1790.1. If β > 1.0, it indicates the wear out failures, and the MTTF and η are approximately equal.

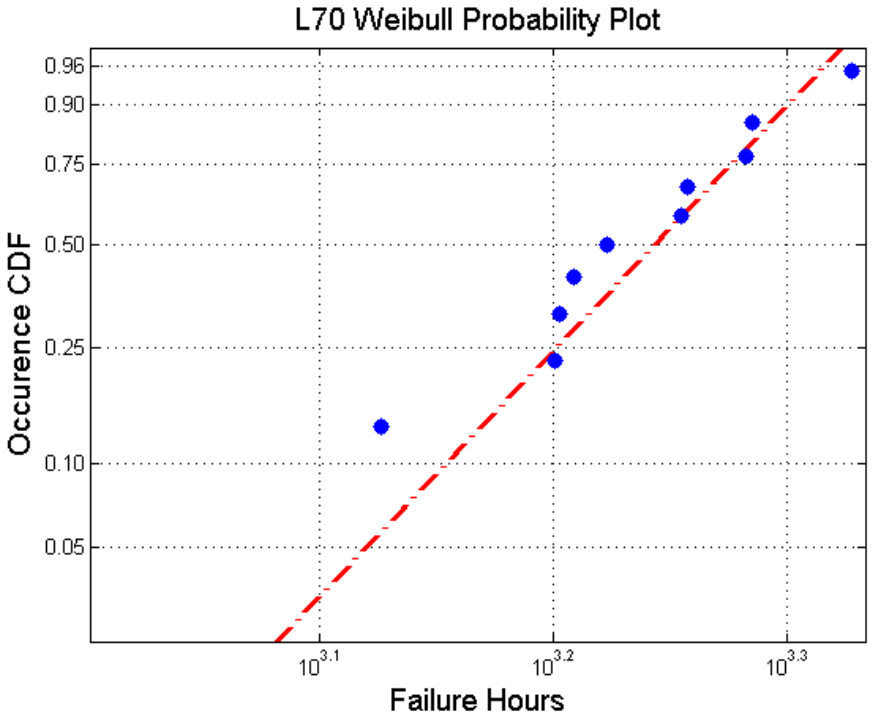


Figure 331 Weibull Distribution of L-prize L70 data

The Weibull cumulative distribution, the population fraction failing by age t is given as following CDF:

$$F(t)=1-e^{-(t/\eta)^{\beta}}=1-e^{-(t/1790.1)^{7.1}} \tag{356}$$

The reliability function is given by the 1-F(t), which shows in the Figure 332. Once time reaching 1700.2 hours the LED lamp reliability drops to 50%. The characteristic life (i.e. B63.2 life) is 1790.1 hours, which says 63.2% LED lamps have failed at this time in the accelerating 85H/85C testing.

$$R(t)=1-F(t)=e^{-(t/\eta)^{\beta}}=e^{-(t/1790.1)^{7.1}} \tag{357}$$

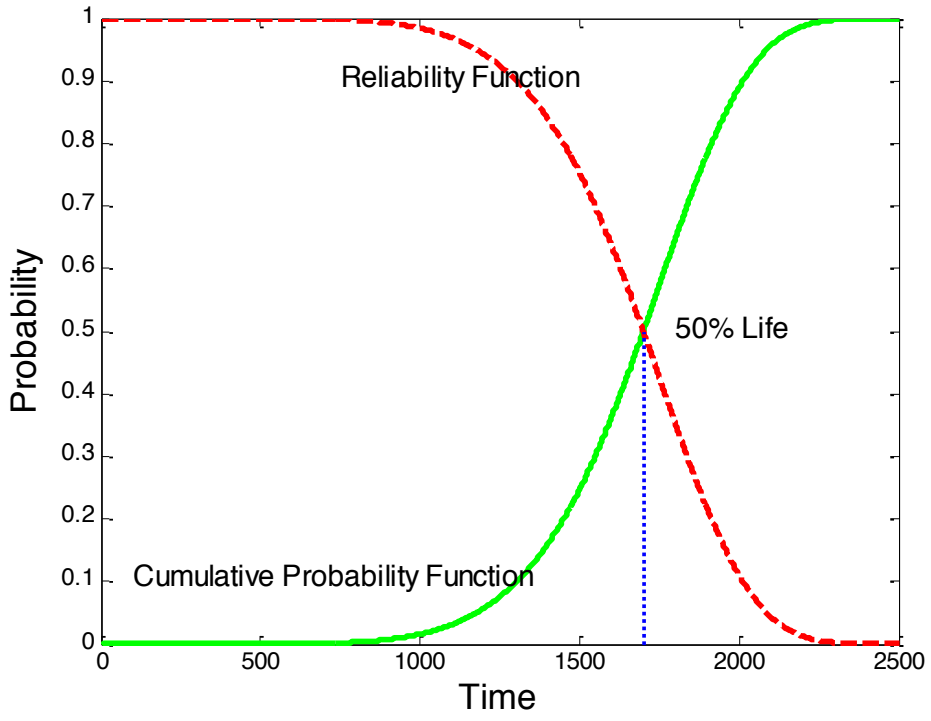


Figure 332 System Reliability Function and Cumulative Probability Function

7.15.7 Summary and Conclusion

This section, the under wet high temperature operating life (WHTOL) L-prize testing has been conducted, and the 2500 hours testing data has been generated. After almost 2500 hours testing, the Lumen drops from 923.51 lumens to the only 195.89 lumens. Compared with Lumen

Degradation, the L-prizes' CCT has only deprecated 2% from pristine. The L-prize L70 mean value is 1673.3 hours. Then, a framework for Prognostic Health Management of LED lamps has been established through the Bayesian classification and Bayesian regression algorithm, and the failure threshold has been found for the L-prize in the 85H/85C testing. Also, applying this framework of the PHM, the Lumen Degradation has been tracked and kept monitoring the Remaining Useful Life throughout the experimentation. In the experiment, the actual prediction can start at 500 hours, and the prognostic Remaining Useful Life is within ±20% Confidence Intervals. In the end, the Weibull distribution of L-prize L70 has been presented, the reliability function of L-prize lamp has been established. The characteristic life is 1790.1hours, and the 50% reliability time is 1700.2 hours. In the future, this failure analysis can be embedded into the Bayesian Network to improve the prognostic results and to increase the prognostic accuracy.

References

A , G., P, D., & J.C., D. (2007). 3D Strain Field Measurement by Correlation of Volume Images Using Scatter light: Recording of Images and Choice of Marks. Strain, 207-218.

Abernethy, R. B. (2000 Fourth Edition). Reliability& Statistical Analysis for Predicting Life, Safety, Survivability, Risk, Cost and Warranty Claims.

Antunes, P., & Dias, G. (2008). Nonlinear 3D Foot FEA Modelling from CT scan medical images. In book: true, pp.135-141.

Anuta, P. (1970, Oct). Spatial registration of multispectral and multitemporal digital imagery using fast Fourier transform techniques. IEEE Trans. Geosci. Electron. , 353-368.

Baker, S., & Matthews, I. (2001). Equivalence and Efficiency of Image Alignment Algorithms. Conference on Computer Vision and Pattern Recognition.

Baker, S., & Matthews, I. (2004). Lucas-Kanade 20 Years On: A Unifying Framework. International Journal of Computer Vision , 221–255.

Ballard, D., & Brown, C. (1982). Computer Vision. Prentice Hall.

Barthelat, F., Wu, Z., Prorok, b., & Espinosa, H. (2003). Dynamic Torsion Testing of Nanocrysalline Coatings Using High-Speed Photography and Digital Image Correlation. Experimental Mechanics.

Bay, B., Smith, T., Fyhrie, D., & Saad, M. (1999). Digital Volume Correlation: Three-dimensional Strain Mapping Using X-ray Tomography. Experimental Mechanics, 217.

Benoit, A., & Guerard, S. (2009). 3D analysis from micro-MRI during in situ compression on cancellous bone. Biomechanics.

Bing, P., Hui-min, X., & etc. (2006). Performance of sub-pixel registration algorithms in digital image correlation. Measurement Science and Technology.

Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer Science+Business Media,LLC.

Bishop, C., & Tipping, M. (n.d.). Bayesian Regression and Classification. Computer and Systems Sciences.

Blaber, J., Adair, B., & Antoniou, A. (2015). Ncorr: Open-Source 2D Digital Image Correlation Matlab Software. Experimental Mechanics.

Bower, A. (2016). EN2210 Continuum Mechanics. Retrieved from http://www.brown.edu/Departments/Engineering/Courses/En221/

Brault, B., Germaneau, A., & Dupre, J. (2013). In-situ Analysis of Laminated Composite Materials by X-ray Micro-Computed Tomography and Digital Volume Correlation. Experimental Mechanics.

Burton, A., & Radford, J. (1978). Thinking in Perspective: Critical Essays in the Study of Thought Processes. Routledge.

C, T., & T, K. (1992). Shape and motion from image streams under orthography: A factorization approach. International Journal of Computer Vision.

Cavalcanti, P., & Mello, U. (n.d.). Three Dimensional Constrained Delaunay Triangulation: A Minimalist Approach.

CGAL. (n.d.). Computational Geometry Algorithms Library. Retrieved from CGAL, Computational Geometry Algorithms Library: http://www.cgal.org

Cline, D., & Jeschke, S. (2009). Dart Throwing on Surfaces. Eurographics Symposium on Rendering.

Cunha, A., & Canann, S. (n.d.). Automatic Boundary Sizing for 2D and 3D Meshes.

D.Jovanovic, J., & Lj. Jovanovic, M. (2010). Finite Element Modeling of the Vertebra with Geometry and Material Properties Retreived From CT-Scan Data. Mechanical Engineering.

David P, R. (1985, June). Improvements in multiprocessor system design. ACM SIGARCH Computer Architecture News archive, 225-231.

Desbrun, M., & Meyer, M. (n.d.). Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow.

Dey, T., & Levine, J. (n.d.). Delaunay Meshing of Isosurfaces.

Donald, M. (1980). A New Technique for the Representation, Manipulation and Display of Arbitrary 3-D Objects by Computer. Rensselaer Polytechnic Institute.

Dunbar, D., & Humphreys, G. (2006). A spatial data structure for fast Poisson-disk sample generation. SIGGRAPH VIDEO.

Edelsbrunner, H., Kirchkpatrick, D., & Seidel, R. (n.d.). On the Shape of a Set of Points in the Plane.

Eric, W. (n.d.). Convolution Theorem.

FAA. (2004). Airplane Flying Handbook. Washington D.C.:U.S. Department of Transportation, Federal Aviation Administration. Retrieved from https://en.wikipedia.org/wiki/Aircraft_principal_axes

Felzenszwalb, P. (n.d.). Efficient Graph-Based Image Segmentation. Artificial Intelligence Lab, Massachusetts Institute of Technology. Massachusetts Institute of Technology.

FIGUEROA-O'FARRILL, J. (n.d.). BRIEF NOTES ON THE CALCULUS OF VARIATIONS.

Francesco, S., & Silva, A. (n.d.). Fourier Methods in CT:projection and reconstruction algorithms. Universidade de Aveiro, Campus Universitatrio, 3810 Aveiro, Portugal.

Frank, C., Hong, S., Maskarinec, S., & etc. (2007). Three-dimensional Full-field Measurements of Large Deformations in Soft Material Using Microscopy and Digital Volume Correlation. Experimental Mechanics.

Gates, M., & Lambros, J. (2011). Towards High Performance Digital Volume Correlation. Experimental Mechanics, 491–507.

Germaneau, A., & Doumalin, P. (n.d.). FULL 3D STRAIN MEASUREMENT BY DIGITAL VOLUME CORRELATION FROM X-RAY COMPUTED AND OPTICAL SCANNING TOMOGRAPHIC IMAGES. Laboratoire de Mécanique des Solides. Université de Poitiers.

Germaneau, A., Doumalin, P., & Dupre, J. (2007). 3D Strain Field Measurement by Correlation of Volume Images Using Scattered Light: Recording of Images and Choise of Marks. Strain, 207-218.

Germaneau, A., Doumalin, P., & Dupre, J.-C. (2008). Comparison between X-ray micro-computed tomography and optical scanning tomography for full 3d strain measurement by digital volume correlation. NDT&E International.

Grady, L. (2006). Random Walks for Image Segmentation. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.

H., Y., & A., A. (2016). Automated Failure Detection in Computer Vision Systems. Journal of Computational Vision and Imaging Systems.

H.S, P. (1999). Feature Based Methods for Structure and Motion Estimation. ICCV Workshop on Vision Algorithms.

Haase, S., & Noack, E. (2014). In-situ analysis of an industrial material compound package by means of X-ray micro tomography and digital volume correlation. iCT Conference.

Haldar, S., Gheewala, N., & Grande-Allen, K. (2011). Multi-scale Mechanical Characterization of Palmetto Wood using Digital Image Correlation to Develop a Template for Biologically-Inspired Polymer Composites. Experimental Mechanics.

Harry, N. (1928). Certain topics in telegraph transmission theory. Trans. AIEE, 617–644.

Houxun, M., & Alireza, P. (n.d.). Energy Minimization Methods in Computer Vision and Pattern Recognition. Springer.

Hu, Z., Luo, H., & etc. (2014). Internal Deformation Measurement of Polymer Bonded Sugar in Compression by Digital Volume Correlation of In-situ Tomography. Experimental Mechanics.

Huang, J., & Peng, X. (2011). a Digital Volume Correlation Technique for 3-D Deformation Measurements of Soft Gels. International Journal of Applied Mechanics.

Hussein, A., & Barbone, P. (2012). Digital Volume Correlation for Study of the Mechanics of Whole Bones. Procedia IUTAM.

Kazhdan, M., & Bolitho, M. (2006). Poisson Surface Reconstruction . Eurographics Symposium on Geometry Processing (2006).

Keating, T., & Wolf, P. (1975). An Improved Method of Digital Image Correlation. Photogrammetric Engineering and Remote Sensing, 993-1002.

Koch, C., & Georgieva, K. (2015). A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure . Infrastructure Computer Vision, 196-210.

Lall, P., & Wei, J. (2015). X-ray micro-CT and digital-volume correlation based three-dimensional measurements of deformation and strain in operational electronics. Electronic Components and Technology Conference. San Diego: IEEE.

Lall, P., & Wei, J. (2016). PBGA package Finite Element Analysis based on the physical geometry modeling using X-ray micro CT digital volume reconstruction. Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm). IEEE.

Lall, P., & Wei, J. (2016). Remaining Useful Life Assessment of Field Deployed Electronics Using X-Ray Micro-CT Based Digital Volume Correlation and Finite-Element Analysis. Electronic Components and Technology Conference (ECTC). IEEE.

Lall, P., Deshpande, S., Wei, J., & Suhling, J. (2014). Non-destructive crack and defect detection in SAC solder interconnects using cross-sectioning and X-ray micro-CT. Electronic Components and Technology Conference (ECTC). IEEE.

Lall, P., Wei, J., & Davis, L. (2013). L70 life prediction for solid state lighting using Kalman Filter and Extended Kalman Filter based models. 2013 IEEE 63rd Electronic Components and Technology Conference. IEEE.

Lall, P., Wei, J., & Davis, L. (2013). Prediction of lumen output and chromaticity shift in LEDs using Kalman Filter and Extended Kalman Filter based models. Prognostics and Health Management (PHM) (pp. 1-14). IEEE.

Lall, P., Wei, J., & Goebel, K. (2012). Comparison of Kalman-filter and extended Kalman-filter for prognostics health management of electronics. Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm). IEEE.

Lall, P., Wei, J., & Goebel, K. (2012). Comparison of Kalman-filter and extended Kalman-filter for the PHM of Electronics. Thermal and Thermomechanical Phenomena in Electronic Systems (pp. 1281-1291). San Diego: IEEE.

Lall, P., Wei, J., & Sakalaukus, P. (2015). Bayesian models for life prediction and fault-mode classification in solid state lamps. Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems. IEEE.

Larry, S. (1992). Basic Circuits.

Leclerc, H., Perie, J.-N., Roux, S., & Hild, F. (2010). Voxel-Scale Digital Volume Correlation. Experimental Mechanics.

Leclerc, H., Roux, S., & Hild, F. (2014). Projection Savings in CT-based Digital Volume Correlation. Experimental Mechanics.

Lecomte-Grosbras, P., Rethore, J., Limodin, N., & etc. (2014). Three-Dimensional Investigation of Free-Edge Effects in Laminate Composites Using X-ray Tomography and Digital Volume Correlation. Experimental Mechanics.

Lee, W., & Nguyena, L. (2000). Solder joint fatigue models: review and applicability to chip scale packages. Microelectronics Reliability.

Lenoir, N., Bornert, M., & etc,. (2007). Volumetric Digital Image Correlation Applied to X-ray Microtomography Images from Triaxial Compression Tests on Argillaceous Rock. Strain.

Leplay, P., Rethore, J., & etc. (2013). Three-dimensional Analysis of an In-situ Double-torsion Test by X-ray Computed Tomography and Digital Volume Correlation. Experimental Mechanics.

LEWINER, T., & LOPES, H. (2003). Efficient implementation of Marching Cubes' cases with topological guarantees. ACM Press.

Lewis, J. (1995). Fast Template Matching. Canadian Image Processing and Pattern Recognition Society (pp. 120-123). Quebec City: Canadian Image Processing and Pattern Recognition Society.

Limodin, N., Rethore, J., & etc. (2011). Analysis and Artifact Correction for Volume Correlation Measurements Using Tomographic Images from Laboratory X-ray Source. Experimental Mechanics.

Liu, D., & Soran, B. (n.d.). A Review of Computer Vision Segmentation Algorithms. University of Washington, Department of Computer Science Engineering.

liu, L., & Morgan, E. F. (2007). Accuracy and precision of digital volume correlation in quantifying displacements and strains in trabecular bone. biomechanics, 3516-3520.

Liu, L., Zhang, J., & Ju, S. (n.d.). A Novel Method of Creating Models for Finite Element Analysis Based on CT Scanning Images.

Lu, H., & Cary, P. (1999). Deformation Measurements by Digital Image Correlation: Implementation of a Second-order Displacement Gradient. Experimental Mechanics.

Lucas, B., & Kanade, T. (1981). An Iterative Image Registration Technique with an Application to Stereo Vision. International Joint Conference on Artificial Intelligence (pp. 674-679). International Joint Conference on Artificial Intelligence.

M.G.L., G. (2000). Surpassing the lateral resolution limit by a factor of two using structured illumination microscopy. Microscopy. Retrieved from http://onlinelibrary.wiley.com/doi/10.1046/j.1365-2818.2000.00710.x/abstract

McGinnis, M., Pessiki, S., & Turker, H. (2004). Application of Three-dimensional Digital Image Correlation to the Core-drilling Method. Experimental Mechanics.

McGraw-Hill Companies, I. (2003). Computer Vision. Retrieved from The Free Dictionary By Farlex: http://encyclopedia2.thefreedictionary.com/computer+vision

Moegeneyer, T., & Helfen, L. (2013). 3D Digital Volume Correlation of Synchrotron Radiation Laminography Images of Ductile Crack Initiation: An Initial Feasibility Study. Experimental Mechanics.

Motwani, M., & Gadiya, M. (n.d.). Survey of Image Denoising Techniques. University of Nevada, Reno. University of Nevada, Reno.

N, L., M, B., & J, D. (2007). Volumetric Digital Image Correlation Applied to X-ray Microtomography Images from Triaxial Compression Tests on Argillaceous Rock. Strain, 93-205.

Pan, B. (2014). An Evaluation of Convergence Criteria for Digital Image Correlation Using Inverse Compositional Gauss–Newton Algorithm. An International Journal for Experimental Mechanics.

Pan, B., & Xie, H. (2008). Study on subset size selection in digital image correlation for speckle patterns . OPTICS EXPRESS.

Pan, B., Li, K., & Tong, W. (2012). Fast, Robust and Accurate Digital Image Correlation Calculation Without Redundant Computations. Experimental Mechanics, 1277–1289.

Pan, B., Qian, K., Xie , H., & Asundi, A. (2009). Two-dimensional digital image correlation for in-plane displacement and strain measurement: a review. Measurement Science and Technology.

R.L, H., & S.E, R. (1992). Strain Gage Users' Manual. Society for Experimental Mechanics.

Radon, J. (1917). Retrieved from https://en.wikipedia.org/wiki/Radon_transform

Rae, P., Williamson, D., & Addiss, J. (2010). A Comparison of 3 Digital Image Correlation Techniques on Necessarily Suboptimal Random Patterns Recorded By X-ray. Experimental Mechanics.

Rethore, J., & Tinnes, J.-P. (2008). Extended Three-dimensional Digital Image Correlation(X3D-DIC).

Roux, S., Hild , F., Viot, P., & Bernard, D. (2008). Three-dimensional image correlation from X-ray computed tomography of solid form. Composites, 1253-1265.

Rueckert, D., & Sonoda, L. (n.d.). Comparison and evaluation of rigid and non rigid registration of breast MR images. Division of Radiological Sciences and Medical Engineering, Guy's Hospital London UK.

Sau Koh, G. S. (2013). Product Level Accelerated Lifetime Test for Indoor LED Luminaires. IEEE, (pp. 978-1-4673-6139-2/13).

Saxena, A., Celaya, J., & etc. (2009). On Applying the Prognostic Performance Metrics. Annual Conference of the Prognostics Health Management Society.

Schreier, H., & Sutton, M. (n.d.). Systematic Errors in Digital Image Correlation Due to Undermatched Subset Shape Functions.

Schwemmer, C., & Prummer, M. (n.d.). High-Density Object Removal from Projection Images using Low-Frequency-Based Object Masking. Pattern Recognition Lab, University of Erlangen-Nuremberg.

Shao, X., Dai, X., & He, X. (2014). Noise robustness and parallel computation of the inverse compositional Gauss–Newton algorithm in digital image correlation. Optics and Lasers in Engineering.

Silva, T. (2014). Image-based registration methods for quanti cation and compensation of prostate motion during trans-rectal ultrasound (TRUS)- guided biopsy.

Smith, T., Bay, B., & Rashid, M. (2002). Digital Volume Correlation Including Rotational Degrees of Freedom during Minimization. Experimental Mechanics.

Subhash, G., Liu, Q., Moore, D., Ifju, P., & haile, M. (2011). Concentration Dependance of Tensile Behavior in Agarose Gel Using Digital Image Correlation. Experimental Mechanics.

Sutton, E., & Orteu, J. (n.d.). Image Correlation for Shape, Motion and Deformation Measurements.

Sykora, D., & Dingliana, J. (n.d.). As-Rigid-As-Possible Image Registration for Hand-drawn Cartoon Animations.

Szeliski, R. (n.d.). Computer Vision: Algorithms and Applications. Springer Science & Business Media.

Terlizzi, T. (2011). Short Seminar on Electronic Semiconductor.

Toyjanova, J., Bar-Kochba, E., Lopez-Fagundo, C., Jonathan, R., Hoffman-Kim, D., & Franck, C. (2014). High Resolution, Large Deformation 3D Traction Force Microscopy. Pone, 10.1371.

Tzimiropoulos, G. (n.d.). AN AFFINE INVARIANT FUNCTION USING PCA BASES WITH AN APPLICATION TO WITHIN-CLASS OBJECT RECOGNITION.

Vulovic, S., Korunovic, N., & Trajanovic, M. (2011). Finite Element Analysis of CT based femur model using finite element program PAK. Journal of the Serbian Society for Computational Mechanics, 160-166.

Wang, T., & Jiang, Z. (2016). GPU Accelerated Digital Volume Correlation. Experimental Mechanics, 297–309.

Wang, W.-C. (n.d.). Moire Method, University of Washington ME557. Retrieved from http://depts.washington.edu/mictech/optics/me557/moire_a.pdf

Warren, D., & Strelow, E. (1985). Electronic Spatial Sensing for the Blind: Contributions from Perception. Springer.

Wattrisse, B., Chrysochoos, A., & etc,. (2000). Analysis of Strain Localization during Tensile Tests by Digital Image Correlation. Experimental Mechanics .

Wgsimon. (2011, 5). A plot of CPU transistor counts against dates of introduction; note the logarithmic vertical scale; the line corresponds to exponential growth with transistor count doubling every two years. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Moore's_law

Wikipedia. (2014). Computer Vision. Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Computer_vision

Xu, J., Moussawi, A., & Lubineau, G. (2015). Using Image Gradients to Improve Robustness of Digital Image Correlation to Non-uniform Illumination: Effects of Weighting and Normalization Choices. Experimental Mechanics, 963–979.

Ying, X., & Xin, S.-Q. (2011). An Intrinsic Algorithm for Parallel Poisson Disk Sampling on Arbitrary Surfaces.

Yitzhak, K. (1976). An Introduction to Harmonic Analysis.

Zhang, D., Eggleton, C., & Arola, D. (2002). Evaluating the Mechanical Behavior of Arterial Tissue using Digital Image Correlation. Experimental Mechanics.

Zhou, K., & Gong, M. (n.d.). Data-Parallel Octrees for Surface Reconstruction. IEEE Transaction on visualization and computer graphics.