The Pennsylvania State University

The Graduate School

College of Engineering

A MODIFIED STRUCTURE FROM MOTION FRAMEWORK BASED ON SMARTPHONE IMAGING FOR 3D RECONSTRUCTION OF ARBITRARY SHAPES

A Thesis in

Industrial Engineering

by

Amey Narayan Vidvans

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2017

The thesis of Amey Narayan Vidvans was reviewed and approved* by the following:

Saurabh Basu

Assistant Professor of Industrial Engineering

Thesis Adviser

Edward De Meter

Professor of Industrial Engineering

Janis Terpenny

Peter and Angela Del Pezzo Department Chair and Professor of Industrial Engineering

*Signatures are on file in the Graduate School

Abstract

There is a need for scalable open source 3D reconstruction systems for reverse engineering and in other applications involving additive manufacturing. This is because most commercially available 3D reconstruction systems are capital and resource intensive. To address this, a novel reconstruction technique is presented in this thesis. The technique involves digital image correlation based characterization of surface speeds during rigid body rotational motion of the object of interest. Proof of concept of the same is shown using simulation and empirical characterization. Towards this, a stage is designed and fabricated in-house using poly-lactic acid (PLA) polymer with a Makerbot 3D printer. The stage is computer controlled using an Arduino controller. Dimensional errors are characterized between the reconstructed model and the original geometry. Herein, some commonly occurring shapes are characterized. Effect of Gaussian curvature on the accuracy of reconstruction are briefly explored. It is shown that highly accurate 3D reconstructions can be performed using scalable hardware involving smart phone based optical photography.

Table of Contents

List of Tables

Chapter 1

Chapter 2

Chapter 3

List of figures

Chapter 2

Chapter 3

Chapter 4

PREFACE

A novel three-dimensional reconstruction tool is developed in this thesis. This technique is extensively explored and presented.

A manuscript for submission for publication as first author was derived from the work in this thesis.

Acknowledgement

I would like to acknowledge the support of my adviser Dr. Saurabh Basu and his exemplary guidance regarding the work presented in this thesis.

I would also like to thank Mr. Travis Richner and Mr. Daniel (Dan) Supko for their help in the fabrication of the experimental setup.

This work would not have been possible without the support of my parents, friends, and Almighty God.

Chapter 1: Introduction

1.1 Introduction

Additive manufacturing is built on the concept that products can be fabricated by depositing material in a layer by layer manner to generate the desired geometry [1]. Since this family of processes is flexible, automated and results in lower scrap, they are naturally attractive to manufacturers [2]. Hence, the advent of additive manufacturing has the potential to revolutionize the manufacturing industry. Since manufacturing has become a highly competitive business, there is also a need to place products on the market faster to maximize profit and sales [3]. It is also expected to result in an explosion of innovation as more complex designs will be manufactured relatively faster [4]. Additive manufacturing, hereafter referred to as AM has applications in medicine, aerospace/automobiles, and construction [5]. For example, a custom knee joint or dental implant can be manufactured according to the needs of the patient [6][7]. Another application is the manufacture of complex aerospace components unmatched by conventional techniques [8]. Architectural applications of AM are also emerging that permit creation of columns, walls, and facades [9].

Frequently, one of the requirements of reverse engineering through AM is a point cloud that can be converted into an STereoLithography(STL) file format which in turn is transferred to an additive manufacturing machine [10]. Hence, a scanning system that can recreate three-dimensional structure is necessary. This can lead to digitization of a vast range of products that could be used to generate a multitude of custom designs [11]. One potential application of such scanning systems, that is of interest to the author is the use of three-dimensional scanning tools in the repair and refurbishing of large scale industrial components when original designs are not available. Additive manufacturing is already being used for such applications in the aerospace industry [12]. An important allied application of such scanning systems is in the verification of theoretical solid mechanics models. Since, commercial scanning systems due to their cost inhibit the scalability of the applications, an open source scanning system with customizable software and hardware is needed. As a comparative study, a table with the cost in dollars ($) of some commercial scanning systems, including software is shown below in Table 1.1.

Table 1.1: Costs of Commercial 3D scanners

Manufacturer	**Product Name**	**Price in US Dollars ($)**
MakerBot®	Digitizer™	799[13]
Fuel 3D	SCANIFY	799[14]
Matter & Form	Matter & Form 3D Scanner	519.99[15]
Next Engine	Scanner Ultra	2995[16]

Thus, there is a need for an open source three-dimensional scanning system, including both hardware and software, that can give comparable results at a lower cost. This technique also needs to be replicable, easy to use so that it can be applied to a wider range of subjects namely Biology, Solid Mechanics, and Archaeology. Towards this, the purpose of the thesis is to formulate a technique for object reconstruction. A novel reconstruction technique with customizable hardware and software has been developed and presented in this thesis. Further, a proof of concept for the same is shown using simulated and empirical characterization. Finally, a comparison with a commercial scanning system is carried out.

A thorough understanding of present reconstruction techniques is necessary for development of a reconstruction technique that meets the requirements described above. In the next chapter, we take a look at a wide range of reconstruction techniques.

1.2 References:

[1] Additive manufacturing of metals, D Herzog, V Seyda, E Wysick, C Emmelmann , Acta Materialia , 2016

[2] Three dimensional Printing: The Physics and Implications of Additive Manufacturing, E Sachs, M Cima, J Cornie , D Brancazio , J Bredt , A Corodeau , T Fan , S Khanuja , A Lauder , J Lee , S Michaels , 1993

[3] Factory Physics, 3rd Edition, W Hopp, M Spearman, Waveland Press Inc.

[4] The rise of 3-D printing: The advantages of additive manufacturing over traditional manufacturing, M Attaran, Business Horizons, 2017

[5] Additive Manufacturing: An Overview, R Singh, S Singh, Reference Module in Materials Science and Materials Engineering, 2017

[6] 3D Printing/Additive Manufacturing Single Titanium Dental Implants: A Prospective Multicenter Study with 3 Years of Follow-Up, S Tunchel, A Blay, R Kolerman, E Mijiritsky, J Shibli, International Journal of Dentistry, 2016

[7] Additive manufacturing in prosthesis development – a case study, P Maji, A Banerjee, P Banerjee, S Karmakar, Rapid Prototyping Journal, 2014

[8] Additive Manufacturing for Aerospace Application, M Chu, L Wang, H Ding, Z Sun, Applied Mechanics and Materials ,2015

[9] Developments in construction-scale additive manufacturing processes, S Lim, R Buswell , T Le , S Austin , A Gibb , T Thorpe ,Automation in Construction , 2012

[10] https://www.stratasysdirect.com/resources/how-to-prepare-stl-files/

[11] Custom design of furniture elements by fused filament fabrication, N Grujovic, F Zivic, M Zivkovic , Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2016

[12] Additive Manufacturing and Characterization of Rene 80 Superalloy Processed Through Scanning Laser Epitaxy for Turbine Engine Hot-Section Component Repair, R Acharya, R Bansal, J Gambone , M Kaplan , G Fuchs , N Rudawski , S Das , Advanced Engineering Materials , 2015

[13] https://www.staples.com/Makerbot-Digitizer-3D-Desktop-Scanner-MP03955/product_1439052

[14] https://scanify.fuel-3d.com/meet-scanify/

[15] https://matterandform.net/scanner

[16] http://www.nextengine.com/

Chapter 2: Literature Survey of Reconstruction Techniques and Theory of Digital Image Correlation

Note: Sections of this chapter are derived from the working manuscript mentioned in the Preface

2.1 Literature Survey

There has been a great interest in developing economic methods and tools to create 3 dimensional shapes from images. Some techniques require multiple views to extract shape information viz Structure from Motion (SfM) whereas some can extract information from just one view viz shape from shading [1].

The range of techniques for reconstruction is broadly classified as active and passive [1]. Active 3D reconstruction techniques utilize either some sort of radiometric or mechanical means to determine depth which is used in creation of a depth map which can be used in creation of the 3D model. On the other hand, passive reconstruction techniques utilize radiance based visual cues from sensors that may include tracking features viz SfM or using the silhouettes of objects viz shape from silhouette.

There is an abundance of techniques that make use of structured light in reconstruction. However as shall be seen all these techniques require specialized hardware that limits their scalability. An example of an active reconstruction technique utilizing structured light is the work done by Izadi et al. [2]. In this study, the authors use a Kinect™ camera to generate depth maps which in turn can be used to generate point clouds. Kinect is a range of motion sensing imaging systems, developed by a leading multimedia company for their line of visual entertainment systems. It uses a structured lighting technique (depth mapping using projected patterns) to generate depth patterns and assign three dimensional coordinates to points. It can generate complex surface geometry and not just “noisy” point clouds that are then integrated across multiple images to recreate scene geometry. Moreover, it also allows for interactive reconstruction and segmentation of objects of interest.

An additional example of reconstruction using structured light is found in the work of Levoy et al. [3]. The authors use an algorithm that integrates range finding, merging, and rendering to extract shape information in real time using a Digital Light Processing projector and camera. The range finding technique used is stripe based triangulation, which determines three-dimensional coordinates from two-dimensional contours and Iterative Closest Point (ICP) based image registration. The distinguishing feature of the technique is the ability to generate complete scans that are needed for engineering applications.

Another example of an active reconstruction technique is the use of photometric stereo in reconstruction. An example of photometric stereo reconstruction is the work by Zhang et al. [4]. Photometric stereo utilizes the parallax between two non-parallel cameras to recreate objects. In this work, a series of four images illuminated by LEDs from different angles is captured and used to reconstruct the object. The error observed in this study is approximately 6 % for a 50-mm object.

A unique type of reconstruction, using digital Moire topographic technique to extract three dimensional scans from objects was utilized by Li et al. [5]. A white light digitizer was used for image acquisition and transferred to the surface reconstruction module. An interesting aspect of this paper was the registration of different range images to produce a single object that will aid in the generation of complete scans as opposed to partial scans [4]. Thus, as mentioned before, we note the requirement of specialized hardware, either a projector or camera in each of these cases that use structured lighting.

Another approach to 3-dimensional reconstruction is shape from shading, which is also an active reconstruction technique. The fact that artists very skillfully use shading and reflection cues in paintings points to the existence of a mathematical framework in shape extraction. In the human brain, the depth perception in part is aided by shading cues [6]. According to Pentland [7], there exist two kinds of algorithms for extracting shape from shading viz Global algorithms and Local algorithms. Global algorithms developed by Horn et al. [8] utilize an iterative scheme to determine shape in the form of characteristic strips. The initial conditions, like surface depth and orientation at the beginning of the strip are known. Spherical approximations regarding surfaces are carried out at points of maximum intensity on the characteristic strips and these solutions are then “propagated” along the characteristic equation.

The local algorithm developed by Pentland [7], extensively uses Fourier Transforms to generate depth maps from image intensities. A Taylor series expansion of the reflectance function at a point is converted into the frequency domain and with suitable substitutions from the partial derivative representation of the slope and orientation, reordered and converted into an expression for the surface topology.

There is further classification in reconstruction techniques by the type of information that is revealed by them i.e. surficial and volumetric. Although not classified in literature as active, it is obvious that since external energy is involved in volumetric reconstruction, they fall under the category of active techniques.

One of the inherent advantages of volumetric reconstruction techniques as opposed to techniques that reveal only surficial features, is that internal features of a component can also be recreated. We look at a range of such techniques that can recreate internal structure. These techniques make use of a mathematical tool known as Radon Transform to generate point clouds.

This is one of the earliest known reconstruction techniques and has been widely researched since the 1970's [9].

The creation of this point cloud, can be accomplished by a range of techniques including computed tomography, neutron tomography or ultrasound computer tomography [10] and then using a reconstruction model to stitch the slices obtained from each projection and create a 3-dimensional representation [11]. The radon transform is a mathematical tool developed by J Radon [12]. It is defined as follows through Eq.1

$$R(\rho, \tau) = \int_{-\infty}^{+\infty} f(x, \rho x + \tau) dx \tag{1}$$

Where ρ and τ are the slope and intercepts of the line in red, along which the integral is taken, shown in the Fig. 1. Thus, essentially the radon transform is an accumulation of integrals along a line across the shape of interest, represented as a function.

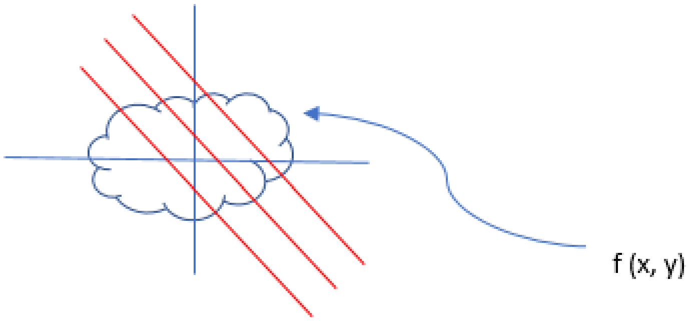

Fig 1 Radon Transform Schematic

Recording the intensity of the line integrals along the function in the form of a projection can be used to reconstruct the function. This is an inverse problem, as are most reconstruction problems. When used for reconstruction, the phenomenon of attenuation of energy is used to determine depth. As can be inferred that, a wide range of energy sources that can penetrate surfaces can be used. Hence X rays, Neutron rays, Ultrasound waves are used in volumetric reconstruction. A number of numerical techniques have been developed to solve these inverse problems. An industrial application of this technique is for generating three dimensional scans of objects for manufacturing applications [13].

As an example of passive reconstruction, we look at structure from motion, a widely used passive reconstruction technique.

Conversion of images into 3 dimensional models is a widely researched subject in computer vision and often called structure from motion. One of the seminal works on this topic was by Tomasi and Kanade who came up with a factorization approach to extract shape and motion from a measurement matrix, constructed from coordinates of tracked points utilizing singular value decomposition [14]. This is based on the Kanade-Lucas-Tomasi feature tracker that can map the displacement between features through the minimization of normalized cross correlation metric [15]. This technique even accounts for occlusions that might occur due to the feature points being hidden in some views. Subsequently, these tracked points across frames have to be triangulated to determine their location in space. The algorithm starts off by tracking P points through F frames and generating a matrix known as a measurement matrix, W which is 2F*P in size. Consequently, it is shown that by Singular Value Decomposition (SVD), this measurement matrix can be split into matrices, one representing camera rotation M(2F*3) and the other representing shape of the object S(3*P) by virtue of the rank theorem. The measurement matrix can be decomposed as shown in Eq. (2), (3) and (4). Here L, Σ and R are intermediate SVD products.

$$W = L\Sigma R \qquad (2)$$

$$M = L'(\Sigma)^{0.5} \qquad (3)$$

$$S = (\Sigma')^{0.5}R' \qquad (4)$$

The primes L’, Σ’, R’ are defined as follows in Eq. (5), (6) and (7) by block partitioning,

$$L = [L' \quad L''] \quad \text{, L' is 2F*3, L'' is 2F*(P-3)} \qquad (5)$$

$$\Sigma = [\begin{matrix} \Sigma' & 0 \\ 0 & \Sigma'' \end{matrix}] \quad \text{, Σ' is 3*3, Σ'' is (P-3) *(P-3)} \qquad (6)$$

$$R = \begin{bmatrix} R' \\ R'' \end{bmatrix} \quad \text{, R' is 3*P, R'' is (P-3) * P} \qquad (7)$$

The rank theorem states that the image stream used for the analysis is “highly rank deficient “and that 2F*P points are redundant. Hence, an SVD based factorization approach works in isolating the shape and camera rotation matrices described earlier.

We also look at another family of passive reconstruction techniques i.e. shape from silhouette by Laurentini et al. [16]. This, essentially is volume intersection obtained from projections of visible points. Each projection constrains the object inside a visual hull that comes closest to the real shape of the object.

Shape carving techniques can also be used to reconstruct shapes and scenes from arbitrarily located camera views that are known. A theory of shape reconstruction by satisfaction of constraints is proposed in the work of Kutulakos et al. [17]. The shape carving algorithm is unique because it is truly a universal reconstruction routine and imposes no restrictions on the camera viewpoints. Also, the necessity of developing merging and rendering algorithms is eliminated as shapes can be reconstructed in a single step. It is in part a progression of the work by Laurentini et al. that utilizes the concept of *photo consistency*. The work begins to diverge as radiance constraints are added to provided information beyond the visual hull. The volume is carved from an arbitrary volume till no photo- inconsistent voxels can be found. This converges to the photo hull of the object under consideration.

A simple passive reconstruction technique is known as the shape from focus routine [18]. It measures the depth of an object by varying the regions in focus which is done by translation of the object. An automated system for detection of focus is also implemented which is based on the concept that as details in the image increase (which means the object is being brought to focus), so does the high frequency content in the Fourier transform of the image.

Thus, there is an abundance of techniques that can reconstruct objects for conversion to point clouds. Unfortunately, barring a few notable works [4] [5], the majority of the papers do not discuss the accuracy of their approaches. The scalability in these approaches is also a factor that is not addressed.

Here in this thesis, we propose a new approach based on digital image correlation, hereafter referred to as DIC for reconstruction. Suitable improvements to original proposed techniques are also presented, along with an experimental validation.

2.2 Introduction to Digital Image Correlation

Digital image correlation is defined as follows in the book by Sutton et al. [19]

"The term "digital image correlation" refers to the class of non-contacting methods that acquire images of an object, store images in digital form and perform image analysis to extract full-field shape, deformation and/or motion measurements."

Digital image correlation is primarily used in context of image registration. Image registration is the process of fusing similar patterns across several images irrespective of scale, rotation or translation.

'Image registration is the process of overlaying images (two or more) of the same scene taken at different times, from different viewpoints, and/or by different sensors. The registration geometrically aligns two images (the reference and sensed images)' [20].

A fundamental requirement of image registration is the identification of 'regions of local similarity' [21]. According to this paper, the process of determining similarity is termed 'correlation'. One of the earliest attempts at correlating templates among images was made by G Hobrough who invented a technique to "*correlate high-resolution reconnaissance photography with high precision survey photography in order to enable more precise measurement of changeable ground conditions"* [19] [22]. Over time the technique evolved to processing digital images as the prevalence of digital images became widespread. A variety of commercial as well as open source implementations of deformation based measurements are now available [23] [24] [25] [26] [27].

2.3 Applications of Image correlation techniques

A brief overview of the applications of digital image correlation is necessary as it useful to know how digital image correlation is being used. It can also help in our application of extracting shape information from 2D images.

Deformation measurement: A widely used application of Digital image correlation is in experimental mechanics research as mentioned before to extract strain, stress and deformation fields. There are also several works in literature expounding advances made in fracture mechanics using digital image correlation [28]. Thus, digital image correlation began to be applied for measuring in plane displacements, a point to be noted and used further in the thesis. It is also used for geological applications to map deformation in rocks and other geo-structures [29]. As an extension of this, DIC is also used in 3 D deformation measurement in conjunction with X-ray tomography, also termed digital volume correlation(DVC) [30].

Fluid mechanics applications: A variant of digital image correlation is incorporated in Particle Image Velocimetry (PIV) that uses tracer particles to measure fluid properties like velocity and other related properties not only in two dimensions but in three dimensions using multiple camera systems [31]. These systems are frequently used in engineering and biological research [32], [33].

Reconstruction of surfaces: There have been several works in the literature of the use of digital image correlation technique to reconstruct microscopic surfaces [19]. Coupled with stereo cameras, digital image correlation can also be used to reconstruct planar surfaces [34].

2.4 The correspondence problem

A major problem is the correspondence problem in pixel tracking which arises from the inability of a technique for motion detection to determine direction of motion and matching pixel locations accurately. A very essential requirement for motion detection using Digital Image correlation is the presence of a non-repeating random pattern that provides a rich information base for pixel tracking. An example of such a random pattern is a speckle pattern. A speckle pattern is overlaid on the shapes in the code that allows for tracking of pixels and shape reconstruction. This solves the correspondence problem with respect to the present research. An example of computer generated speckle pattern is shown below in Fig.2.

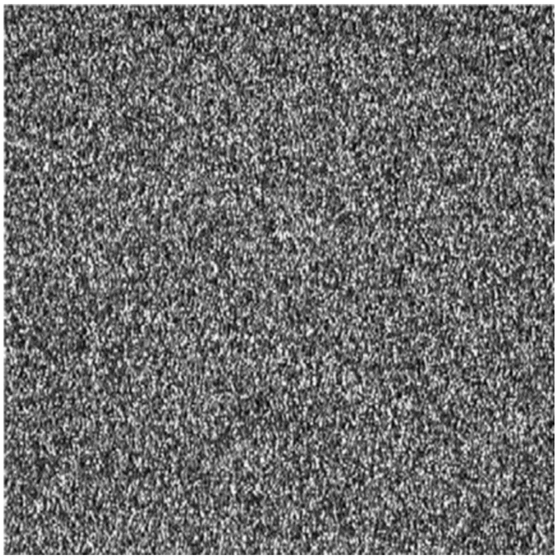

Fig 2: Example of Speckle pattern generated [35].

It will be seen that this speckle pattern is present on natural specimens and an artificial pattern is not necessary however for metal and such surfaces, an artificial pattern must be overlaid. There is presence in the literature of a parameter ‘mean intensity gradient’ that can be used to evaluate the effectiveness of the speckle pattern [36]. It is shown that a higher mean intensity gradient is indicative of a better speckle pattern. This also means that a speckle pattern with more high frequency pixels will be tracked better, which is a very useful insight to be used later in the experimental phase.

2.5 The matching problem

We assume the reader is familiar with the one-dimensional matching problem. The two-dimensional matching problem, as it is applied to images is described below. This is also known as the Kanade-Lucas tracker. The following is derived from Sutton [19].

The problem is to minimize the difference between a template and reference image. The sum of squared deviations (SSD) in their gray values is shown below in Eq. (8).

$$\bar{d}(opt) = argmin\, \Sigma \,|\, \mathrm{G}(\mathrm{x}+\bar{d})\text{-}\mathrm{F}(\mathrm{x})|\text{^}2 \tag{8}$$

Where G(x) represents the displaced image while F(x) is the template.

Expanding this cost function in Eq. (9) (which is to be minimized) with the Taylor's series and ignoring higher orders we get,

$$\chi^2(\bar{d}x + \Delta x, \bar{d}y + \Delta y) = \Sigma(|G(x+\bar{d}) - \frac{\partial G}{\partial x}\Delta x - \frac{\partial G}{\partial y}\Delta y - F(x)|)\text{^}2 \tag{9}$$

This equation can be solved iteratively with current estimates being $\bar{d}x$ and $\bar{d}y$ in the x and y directions respectively. Differentiating the above equation w.r.t Δx and Δy and equating the result to zero since a minimum is sought, we get the following expression as shown in Eq. (10).

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \sum\left(\frac{\partial G}{\partial x}\right)^2 & \sum\frac{\partial G}{\partial x}\frac{\partial G}{\partial y} \\ \sum\frac{\partial G}{\partial x}\frac{\partial G}{\partial y} & \sum\left(\frac{\partial G}{\partial y}\right)^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum\frac{\partial G}{\partial x}(F-G) \\ \sum\frac{\partial G}{\partial y}(F-G) \end{bmatrix} \tag{10}$$

In this example, we have discussed only translation displacement between the template and reference image and only square template shapes. However, the theory can easily be extrapolated for different subset shapes and types of displacements. In our case, it is not necessary, since our displacement is assumed to be translational only. As will be seen later, this is a valid assumption and considerably simplifies the problem of reconstruction.

In the modern method of correlation, we require a metric and computational technique to compute this metric in order to successfully correlate a template and image.

2.6 Correlation metrics

There are several metrics that have been devised for image correlation over the years. The section above describes only one of the many criterion namely the sum of squared deviations(SSD). We look at some more of these criterion in this section. We also look at the question of which one of these criterion is the most accurate for our problem. The following Table 2.1 lists some formulae mentioned in Sutton [19].

Table 2.1 Optimization criterion and their formulae

Optimization criterion	Formula
SSD (Sum of squared deviations)	$\sum(G-F)^2$
ZSSD (Zero sum of squared deviations)	$\sum\big((G-E(G))-(F-E(F))\big)^2$
NSSD (Normalized sum of squares deviations)	$\sum\left(\frac{\sum FG}{\sum G^2}G-F\right)^2$
ZNSSD (Zero normalized sum of squares deviation)	$\sum(\left(\frac{\sum FE(G)}{\sum E(G)^2}G-E(G)\frac{\sum FE(G)}{\sum E(G)^2}\right)-(F-E(F)^2))$
NCC (Normalized correlation coefficient)	$1-\frac{\sum FG}{\sqrt{\sum F^2\sum G^2}}$
SAD (Sum of absolute deviation)	$\sum\lvert F-G\rvert$

E(J) represents the mean intensity of J, J being an image represented as a two-dimensional array

However, according to Pan [37], the most widely used criteria are ZNCC (zero mean, normalized correlation coefficient), ZNSSD (zero mean normalized sum of squared deviations) and PSSD (Parametric sum of squared deviations). According to Pan, there is no difference in choosing any of these criterion over the other since they are all equivalent and reduce to each other mathematically.

The technique we adopted was based on the maximization of the Zero Normalized Cross Correlation (ZNCC) coefficient because Fast Fourier based computational techniques have been developed and are easy to implement in MATLAB [21]. The formula can be seen in Eq.11.

$$\gamma(x,y) = \frac{\sum_{u,v}\left[f(x+u,y+v)-\overline{f_{x,y}}\right](t(u,v)-\bar{t})}{\sum_{u,v}\left[f(x+u,y+v)-\overline{f_{x,y}}\right]^2 \sum_{u,v}[t(u,v)-\bar{t}]^{0.5}} \tag{11}$$

As can be seen that the formula for ZNCC has a lot of similarity with the NCC. The mean intensity of each two-dimensional array is subtracted from each sum in the numerator and denominator to make it more robust against intensity changes across images.

A number of approaches have been formulated in succession to the spatial domain implementation such as SSDA and Gradient Descent [21] [38].

However, a frequency domain representation is shown to be superior in terms of speed by Lewis [38]. Since the template is always smaller than the reference image for all applications in the real world, the transform domain implementation is faster than the direct spatial computation. A novel method involving precomputing the integral of the image and image square over the search area from tables is also shown. The maxima of the ZNCC coefficient is extracted and is determined to be position of the template. For example, in Fig 3(a) and 3(b), the displacement of the template shown in the box outline is determined to be Δx=15 and Δy=0 from the correlation coefficient array represented in Fig 3(c).

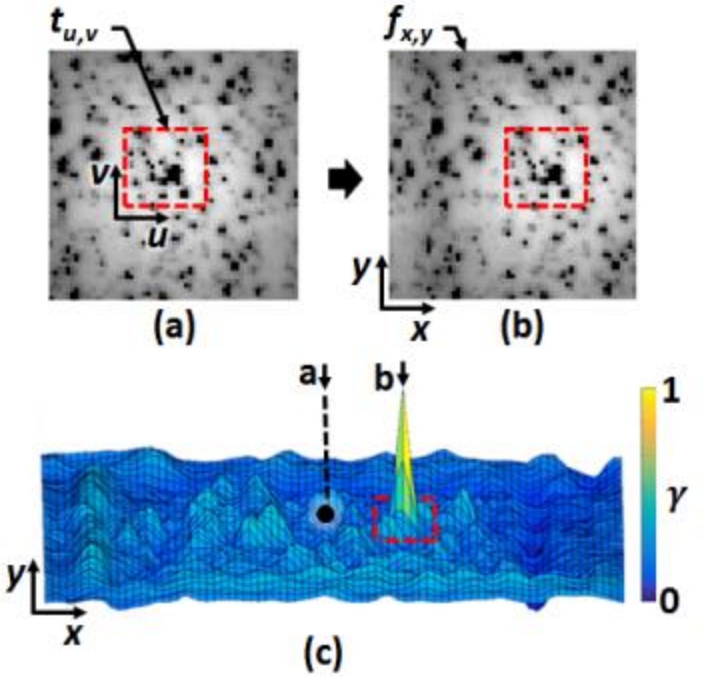

Fig 3a.) Initial Position of Pixels b.) Final Position of Pixels c.) Cross Correlation Field: Calculation of displacement using ZNCC

2.7 Interpolation for subpixel accuracy

The output from the algorithm described in section 2.6 is an array of ZNCC coefficients. The location of the maxima of these ZNCC coefficients is the new integer location of the template as explained above. However, displacements are not integers and must be resolved by interpolation. We look at the interpolation between grid points. We also discuss the introduction of systematic errors in the displacement determination as a result of this.

The interpolation is represented by a polynomial between the integer values of the ZNCC. In order to recreate the original function, a convolution with a function whose pulse extends over an infinite time interval e.g. sinc function, also represented as $\frac{\sin(x)}{x}$. An approximation involves estimating an interpolation function and using polynomials as explained above.

Seminal work by Unser [39] explains the interpolation carried out with a basis function such as a cubic spline kernel in 2 dimensions. It can be simply explained if we think of the original function as a linear combination of a certain number of the basis functions. This can be expressed mathematically as a convolution of the vector of coefficients and the basis function. The problem then reduces to finding the optimum coefficients, c(k) of each of these pieces by inverse convolution. The coefficients are found by inverse filtering as shown in the Eq. (12) and (13).

$$s(k) = b * c(k) \text{ (12)}$$

$$c(k) = b^{-1} * s(k) \text{ (13),}$$

Where $b^{-1}$is the inverse convolution operator of the basis function and s(k) is the discrete signal.

The interpolation introduces systematic error in the routine because it produces both phase and amplitude error. It can be shown that the error is in fact dependent on position. The systematic error, as a function of true subpixel displacement can be simulated by a translation test for images with speckle pattern having a continuous gray value distribution and a bimodal distribution. It is plotted below in Fig 4(a) and 4(b) respectively. As can be seen and observed independently by the authors as well that increasing the higher order terms in the interpolation function increases the accuracy by reducing the systematic error [40].

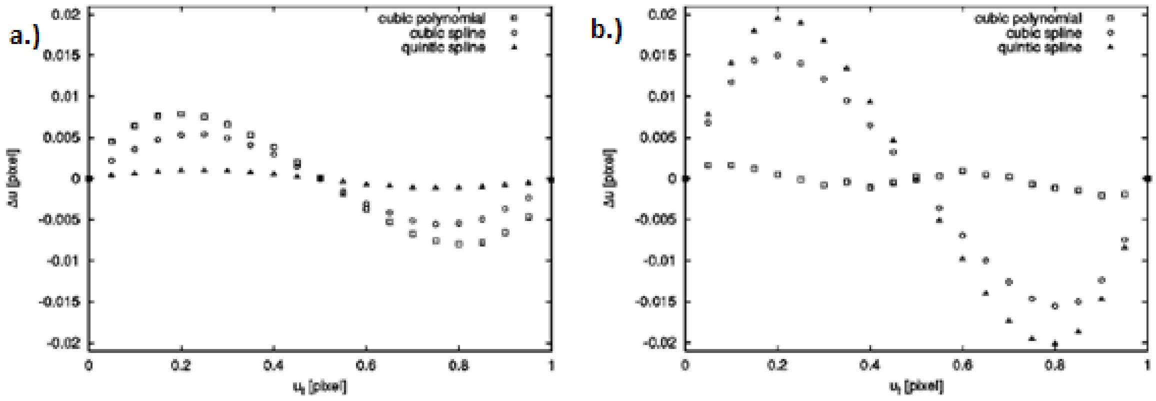

Fig 4 Interpolation kernels and their performance: Errors for speckle patterns with a.) continuous gray value distribution b.) bimodal distribution [41]

A detailed derivation of these errors can be found in [41]. Hence a technique that can increase the accuracy of the interpolation is expounded. A similar proof by simulation was discovered independently by the authors which will be discussed in the next chapter. This shall be used in the simulation and experimental stage to improve performance of the reconstruction technique.

2.8 Demonstration of deformation measurement

An illustration of the code that can be developed for deformation measurement was developed by the author. The code can be seen in the Appendix. The data set was synthetically generated by displacing the original array represented as an image by a unit distance in the y-direction. The results of the code are shown below in Fig 5.

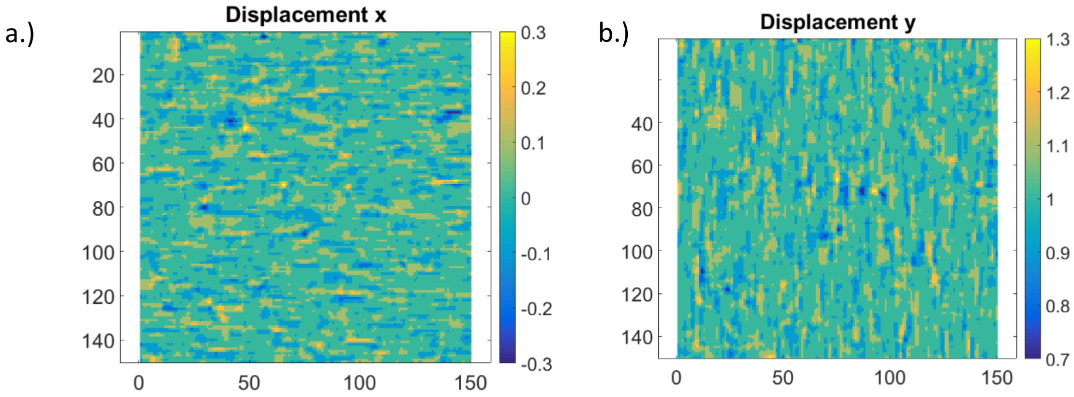

Fig 5 a.) Displacement in y direction in the dataset b.) Displacement in x direction in the dataset

Such displacement fields can be used to calculate the in-plane strain and stress using constitutive relations. As mentioned, they can be useful in validating finite element solid mechanics models and experimental mechanics.

2.9 Summary

In this chapter, we have introduced digital image correlation and the theory behind it as applicable in the succeeding chapters. Applications of digital image correlation were detailed. Generating deformation maps using Digital image correlation was demonstrated. The process can be summarized in the Fig 6 below.

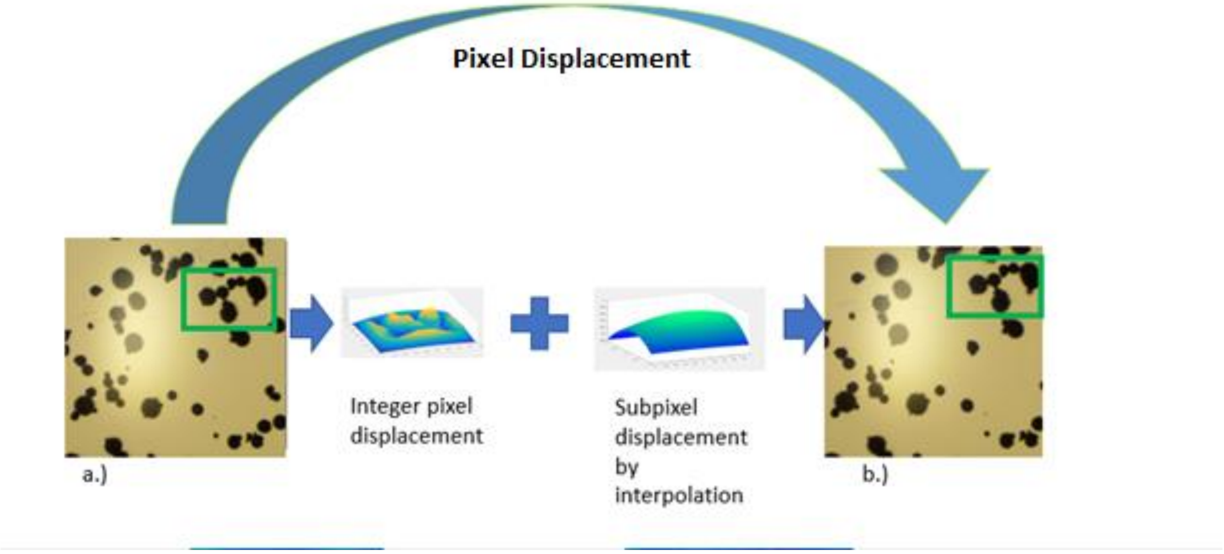
Pixel Displacement
Integer pixel displacement
Subpixel displacement by interpolation
a.)
b.)

Fig 6 a.) Old position of pixel group b.) New Location of pixel group

2.10 References:

[1] 3D Reconstruction from Multiple Images Part 1: Principles, T Moons, L Gool, M Vergauwen , Foundations and Trends in Computer Graphics and Vision, 2009

[2] KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera, S Izadi, D Kim, O Hilliges, D Molyneaux, R Newcombe, P Kohli, J Shotton , S Hodges , D Freeman , A Davison , A Fitzgibbon , ACM , 2011

[3] Real-time 3D Model Acquisition, S Rusinkiewicz, O Hall-Holt, M Levoy , ACM , 2002

[4] A fast 3D reconstruction system with a low-cost camera accessory, Y Zhang, G Gibson, R Hay, R Bowman, M Padgett, M Edgar, Scientific Reports, 2015

[5] A reverse engineering system for rapid manufacturing of complex objects, L Li, N Schemenauer, X Peng, Y Zeng, P Gu, Robotics and Computer Integrated Manufacturing, 2002

[6] Perception of Shape from Shading, V Ramachandran, Nature, 1988

[7] Linear Shape from Shading, A Pentland, International Journal of Computer Vision, 1990

[8] Height and Gradient from Shading, B Horn, International Journal of Computer Vision, 1990

[9] Fast Image Reconstruction Based on a Radon Inversion Formula Appropriate for Rapidly Collected Data, G Herman, A Naperstek, SIAM Journal on Applied Mathematics, 1977

[10] Image reconstruction technique using projection data from neutron tomography system, W el Bar, I Mahmoud, H Konber, T Mongy, Alexandria Engineering Journal, 2015

[11] Digital reconstruction of multidimensional signals from their projections, R Mersereau, A Oppenheim, Proceedings of the IEEE, 1974

[12] Peter Toft, PhD thesis- The radon transforms- theory and Implementation.

[13] Applied X-ray computed tomography, C Buynak , R Bossi , Nuclear Instruments and Methods in Physics Research B , 1995

[14] Shape and Motion from Image Streams under Orthography: a Factorization Method, C Tomasi, T Kanade, International Journal of Computer Vison ,1992

[15] Detection and Tracking of Point Features, CMU Technical Report, C Tomasi, T Kanade ,1991

[16] How Far 3D shapes can be understood from 2D silhouettes, A Laurentini, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995

[17] A Theory of Shape by Space Carving, K Kutulakos, S Seitz, International Journal of Computer Vision, 2000

[18] Shape from focus, S Nayar, Y Nakagawa, IEEE transactions on pattern analysis and machine intelligence ,1995

[19] Image Correlation for Shape, Motion and Deformation Measurements Basic Concepts, Theory and applications, M Sutton, J Orteu, H Schreier, Springer, 2009

[20] Image registration methods: a survey, B Zitova, J Flusser, Image and Vision Computing, 2003

[21] A Class of Algorithms for Fast Digital Image Registration, D Barnea, H Silverman, IEEE Transactions on Computers, 1972

[22] G Holbrough, Methods and Apparatus for correlating corresponding points in two images,2964642, US Patent Office, 1960

[23] www.ncorr.com

[24] http://correlatedsolutions.com/vic-3d/

[25] https://www.dantecdynamics.com/products-and-services

[26] http://correlatedsolutions.com/digital-image-correlation/

[27] http://www.instron.us/en-us/our-company/press-room/blog/2015/june/amandp-on-dic

[28] Quantitative in situ study of short crack propagation in polygranular graphite by digital image correlation, M Mostafavi, T Marrow, Fatigue and Fracture of Engineering Materials and structures, 2011

[29] Investigation of the rate dependence of fracture propagation in rocks using digital image correlation (DIC) method, G Cao, W Yao, K Xia, Z Li, Engineering Fracture Mechanics, 2015

[30] Subset Refinement for Digital Volume Correlation: Numerical and Experimental Applications, M Gates, J Gonzalez, J Lambros, M Heath, Experimental Mechanics, 2015

[31] Stereo particle image velocimetry measurement of 3D soil deformation around laterally loaded pile in sand, Y Bing-xiang , C Wen-wu , J Tong , W Yi-xian , C Ke-ping, Journal of Central South University , 2013

[32] Characterization of YAG:Dy,Er for thermographic particle image velocimetry in a calibration cell , E Hertle , S Will , L Zigan ,Measurement Science and Technology , 2017

[33] Digital particle image velocimetry studies of nasal airflow, S Chung, S Kim, Respiratory Physiology and Neurobiology, 2008

[34] Automated 3D reconstruction using a Scanning Electron Microscope, N Cornille , D Garcia ,M Sutton , S McNeill , J Orteu

[35] Optical characterization of frame grabbers, Optics and lasers in Engineering, 2013

[36] Mean intensity gradient: An effective global parameter for quality assessment of the speckle patterns used in digital image correlation, B Pan, Z Lu, H Xie , Optics and lasers in Engineering, 2010

[37] Recent Progress in Digital Image Correlation, B Pan, Experimental Mechanics, 2011

[38] Fast normalized cross correlation, J Lewis, Vision Interface,1995

[39] Splines, A perfect fit for Signal and Image Processing, M Unser, IEEE Signal Processing Magazine, 1999

[40] Systematic errors in digital image correlation caused by intensity interpolation, H Schreier, J Braasch, M Sutton, Optical Engineering, 2000

Chapter 3: Simulation of Reconstruction and Error Derivation

Note: Sections of this chapter are derived from the working manuscript mentioned in the Preface

3.1 Introduction

The chapter describes the reconstruction scheme adopted and simulations carried out with the proposed technique. The idea of using relative velocity to determine the position of an object is not a new concept and has been described in different eras [1]. The present reconstruction routine is a modern attempt to use velocity measurements, carried out via DIC as a means to describe the shape of an object. A number of reconstructions carried out by simulation are shown as proof of concept. A methodology for error calculation is also presented, which is important in quantifying the deviation from the actual object. The effect of curvature and radius on error is also explored and shown. A more accurate interpolation subroutine using B-splines is used to improve accuracy. This is the basis on which subsequent experimental reconstruction has been carried out.

3.2 Reconstruction by velocity measurements

As mentioned earlier, the concept of determining shape with velocity measurements has been around since the 1910's [1]. However, a complete reconstruction technique based on this concept was missing from the literature. In the human brain, depth perception occurs due to motion of an object by virtue of the kinetic depth effect [2].

The concept underlying our proposed methodology for 3D-reconstruction is illustrated in Fig. 1. Fig.1 shows an arbitrary non-spherical shape that is rotated about an axis. During rotation at angular speed ω, points further away from this axis will feature a larger surface speed as governed by V = r*ω. Consequently, these points will exhibit a larger displacement $\Delta s = r\Delta$⍰

for angular displacement ΔΘ. This feature is evident upon comparison of points 1 and 2 in Fig. 1, these exhibiting larger, i.e. r_1, and smaller i.e. r_2, radii with respect to the axis of rotation, respectively. Upon rotation through an angle ΔΘ, the point 1 moves a larger distance $\Delta s_1 = r_1 \Delta\theta$, in comparison with the corresponding distance $\Delta s_2 = r_2 \Delta\theta$. In this regard, the crux of the proposed approach to 3D-reconstruction lies in the delineation of the aforementioned displacement Δs from the sequence of digital images of the rigid body rotation process.

Consequently, these displacement measurements are converted into coordinates. The shapes used in this simulation were rotated at a known rate. This concept was coded in MATLAB. A few examples of reconstruction are presented below.

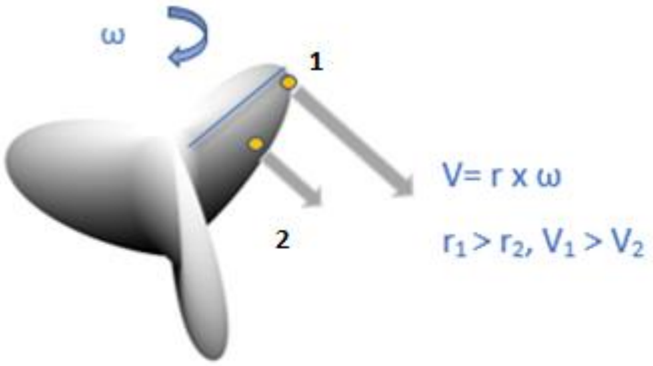

Fig 1 Schematic of the reconstruction scheme

3.3 Examples of reconstruction

As can be seen, the reproduction of complex shapes shown in Fig. 2a is carried out and displayed in Fig. 2b with the code developed using the concept described above.

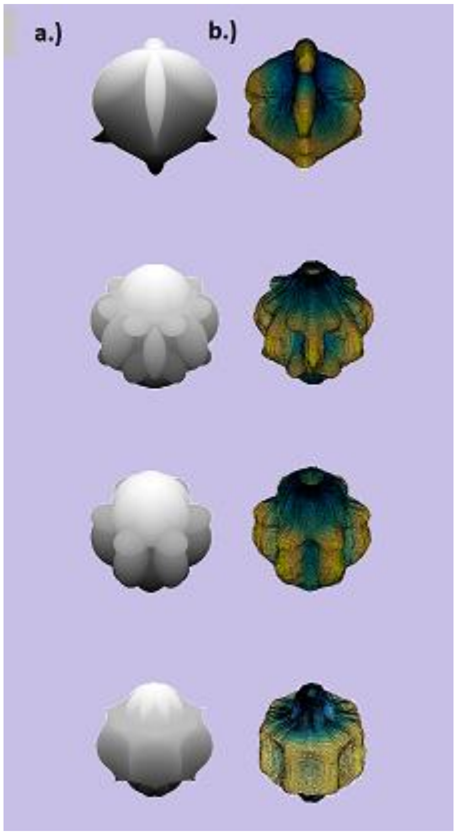

Fig 2 a.) Original Shapes b.) Reconstructed shapes

3.4 Description of the code and methodology

The shapes for reconstruction are generated with a shape generator module. These shapes were generated by distorting the surface of a sphere by a sinusoidal function which could be modified by changing the parameters defining the sinusoidal function. The view command effectively rotates the shape as a rotation tensor would about the Z axis [3]. An image stream generated by capturing the orthographic projections of the shapes as they are rotated about the Z axis in MATLAB using the saveas commands [4]. A speckle pattern was overlaid on the shapes. As mentioned in Chapter 3, this is required for tracking pixels. A list with all image file names is necessary for the cross-correlation engine that is described later. A grid is created coinciding with the axis of rotation to aid in correlation, which means that the grid should be a series of points that are collinear. A cross correlation engine correlates pixels of a defined length and width across images and generates an array containing their new positions from the grid. This part of the code was adapted from the work of Eberl et al. [5]. The reconstructed shape is generated by deriving the velocity of each point and using these values to fix the coordinates of the point. The X and Y coordinates are characterized from the magnitude of velocity and the angle to which the object has rotated at that point. The code can be succinctly explained in the form of a flowchart shown below in Fig. 3.

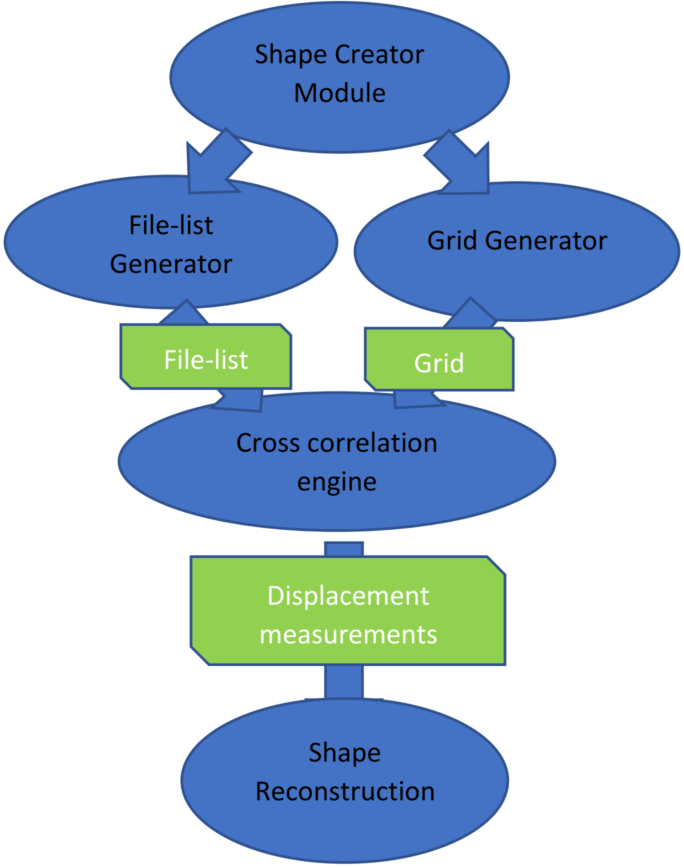

Fig 3 Flowchart of reconstruction

This differs from the structure from motion approach that utilizes triangulation to determine location of points. Further, structure from motion algorithms extract the pose of the camera by comparing the images, in comparison with DIC based reconstruction, wherein this is defined a-priori. In fact, the current technique can be thought of as a modified structure from motion as the image correlation technique used is from the same family as structure from motion.

3.5 Error Calculation and Effect of curvature

The problem of calculating the error between the extracted shape and true shape is a difficult one as there is no one-to-one correspondence between the points of the original shape and the points of the reconstructed shape. The reconstructed shape in this case is less dense than the original shape and is dependent on the grid size and hence there is a mismatch in the correspondence. To overcome the problem, both the reconstructed coordinates and the original set of coordinates were scaled to the same size by maximum value normalization and converted into polar form. Thereafter, for each point on the reconstructed surface the closest four points on the original surface were identified using the metric of smallest Euclidean distance. The average radial distance of these points from the origin was used to calculate the error between the original and reconstructed shape. To aid in the calculating the Euclidean distance, a routine to calculate Euclidean distance between points developed by MATLAB was used [6].

Five standard shapes characterized by a variety of Gaussian curvatures were selected for describing the error between the original shape and reconstructed shape. Gaussian curvature, as is widely known is an intrinsic measure of curvature. The Gaussian Curvature is defined simply as the product of the principle curvatures which define the surface [7]. If the hypothesis, that the curvature affects error is true, then there should be a correlation between the error and the curvature metric. The shapes and their reconstructions are shown in Fig 4,5,6,7 and 8. The errors calculated using the technique described above are shown in Figs. 9,10,11,12 and 13.

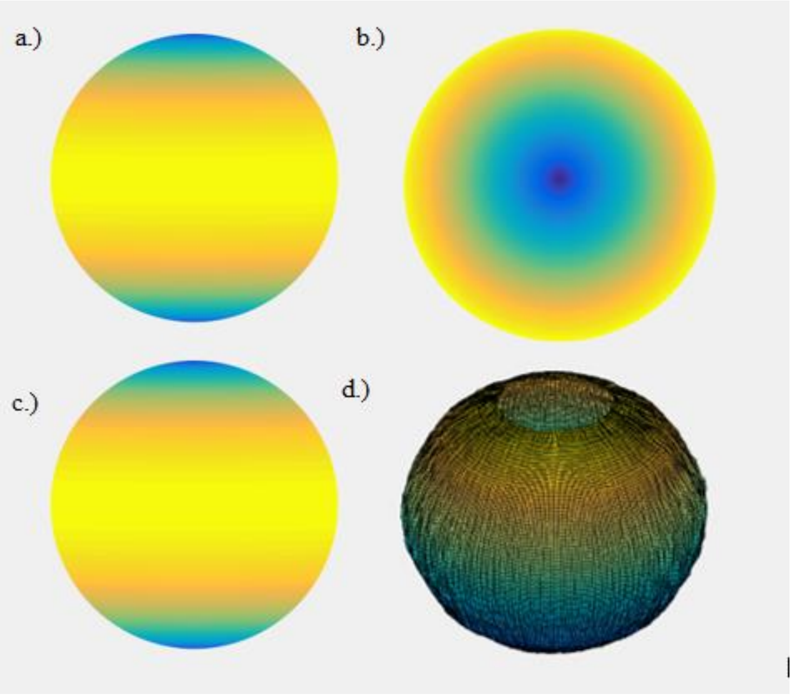

Fig 4 Views of sphere used for reconstruction a.)Front View b.) Top View c.)Side View d.) Reconstructed Shape

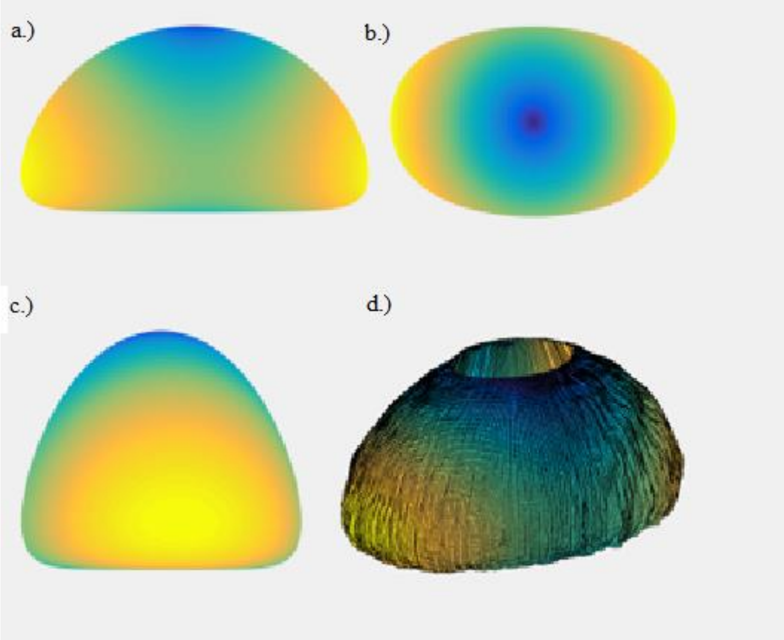

Fig 5 Views of shape 1 used for reconstruction a.)Top View b.) Side view c.)Front view d.) Reconstructed Shape

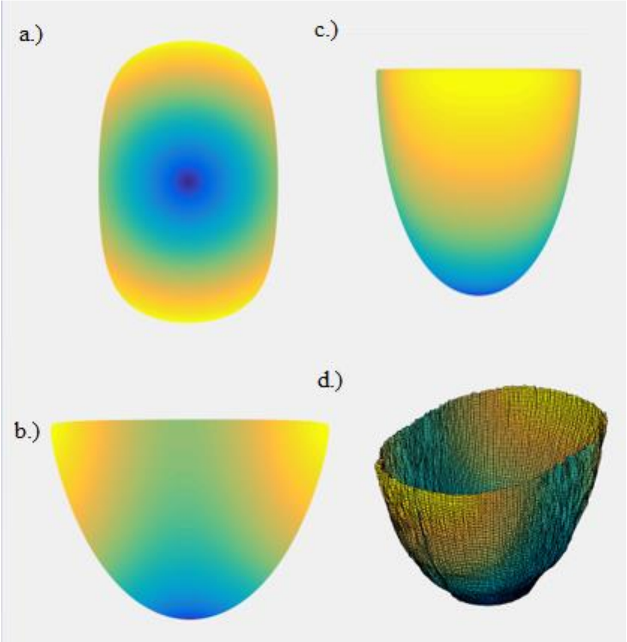

Fig 6 Views of shape 2 used for reconstruction a.)Top View b.) Side view c.)Front view d.) Reconstructed Shape

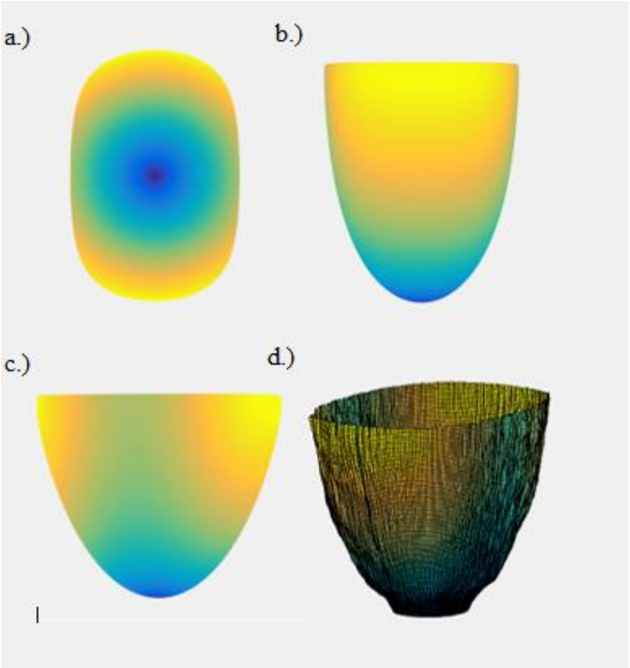

Fig 7 Views of shape 3 used for reconstruction a.)Top View b.) Side view c.)Front view d.) Reconstructed Shape

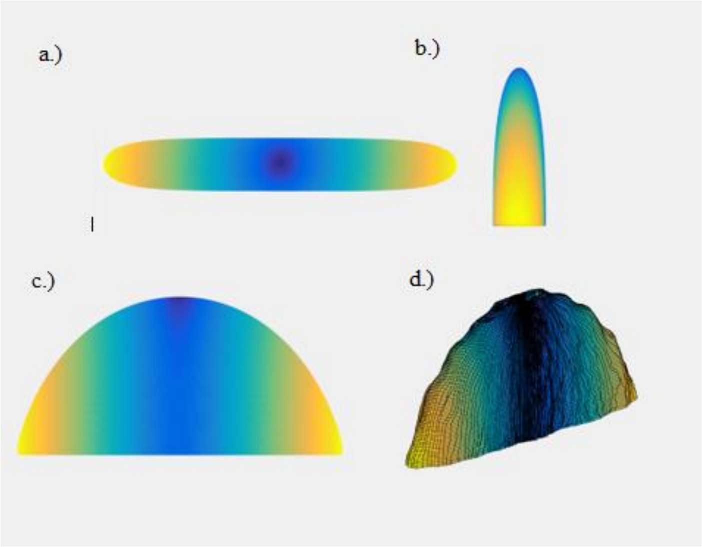

Fig 8 Views of shape 4 used for reconstruction a.)Top View b.) Side view c.)Front view d.) Reconstructed Shape

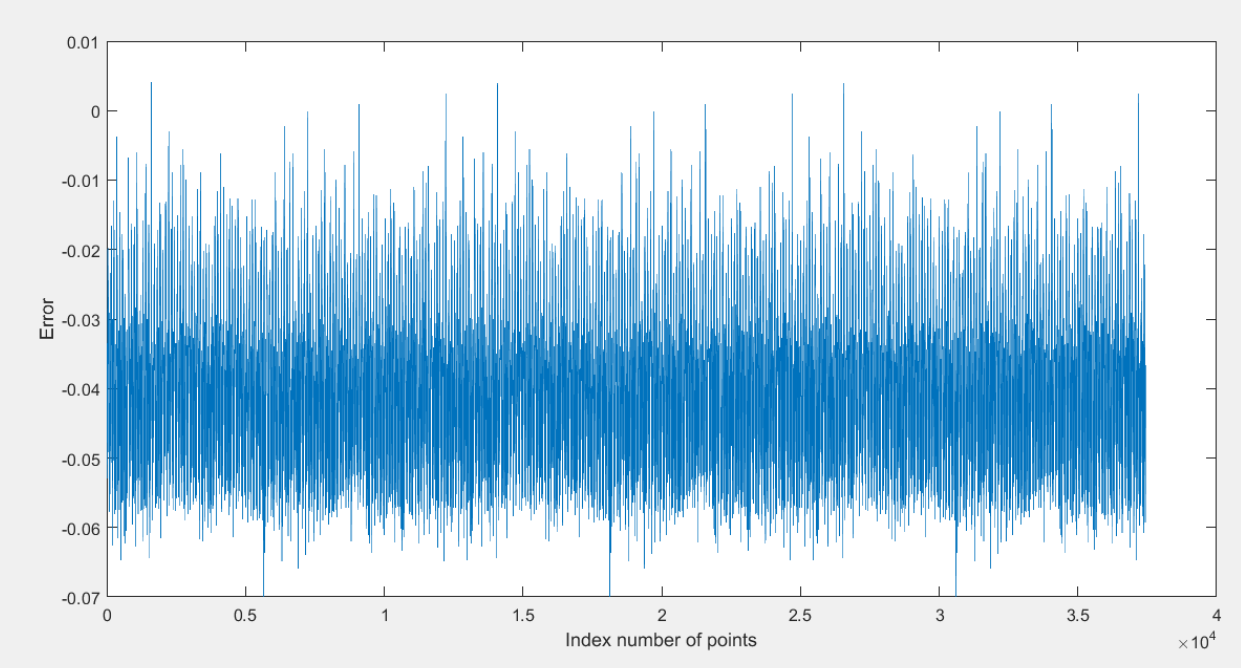

Fig 9 Error vs Index Number for each point on sphere

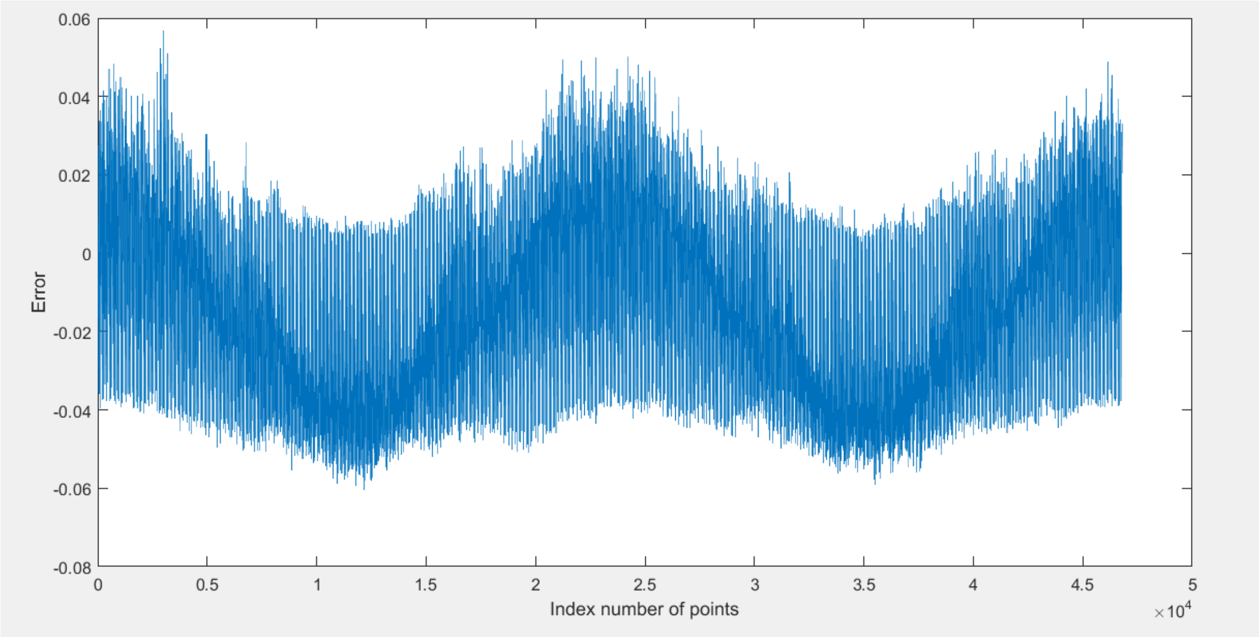

Fig 10 Error vs Index Number for each point on shape 1

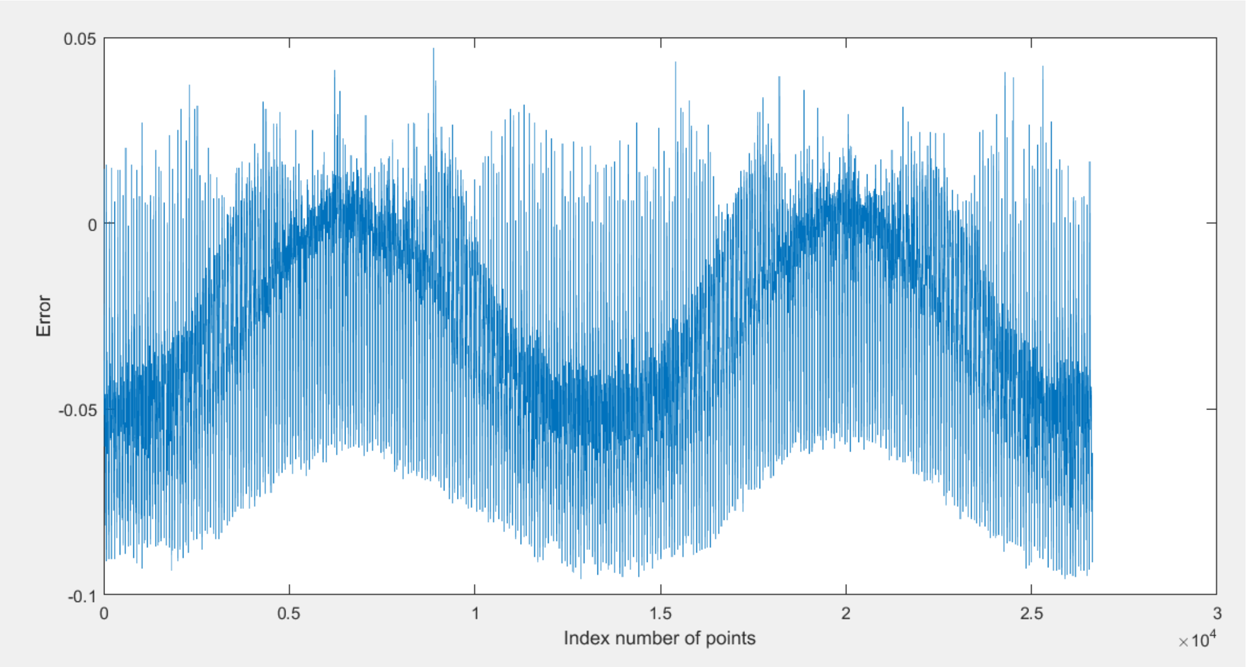

Fig 11 Error vs Index Number for each point on shape 2

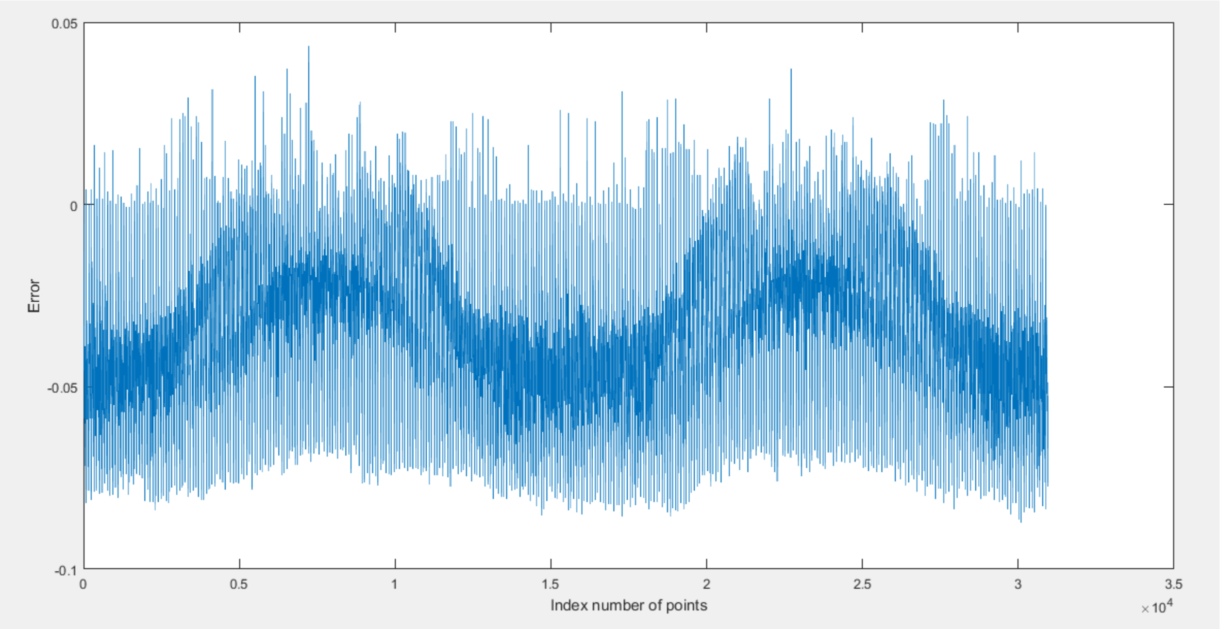

Fig 12 Error vs Index Number for each point on shape 3

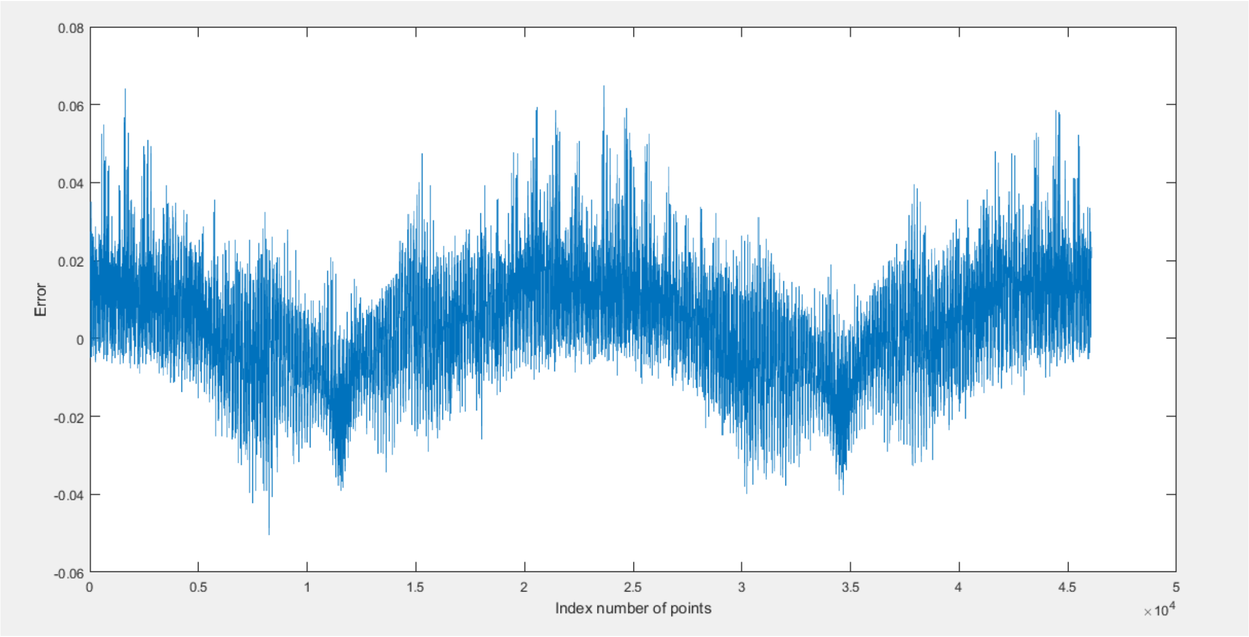

Fig 13 Error vs Index Number for each point on shape 4

Fig.14,15,16,17,18 below shows the effect of curvature on the error for the five shapes. The Gaussian Curvature was calculated and plotted for the corresponding point on the X axis [8] The errors were calculated based on the technique described above and plotted on the Y axis.

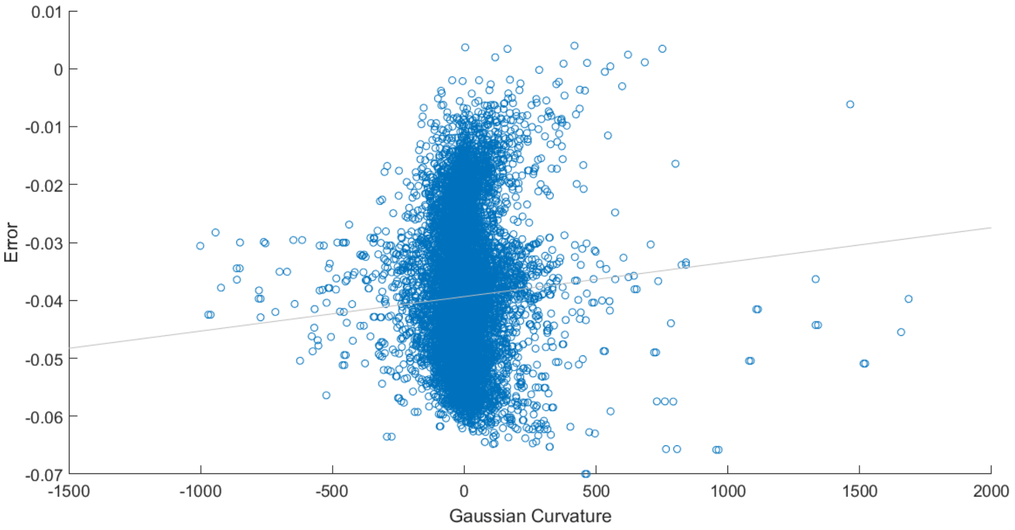

Fig 14 Gaussian Curvature Vs Error for sphere

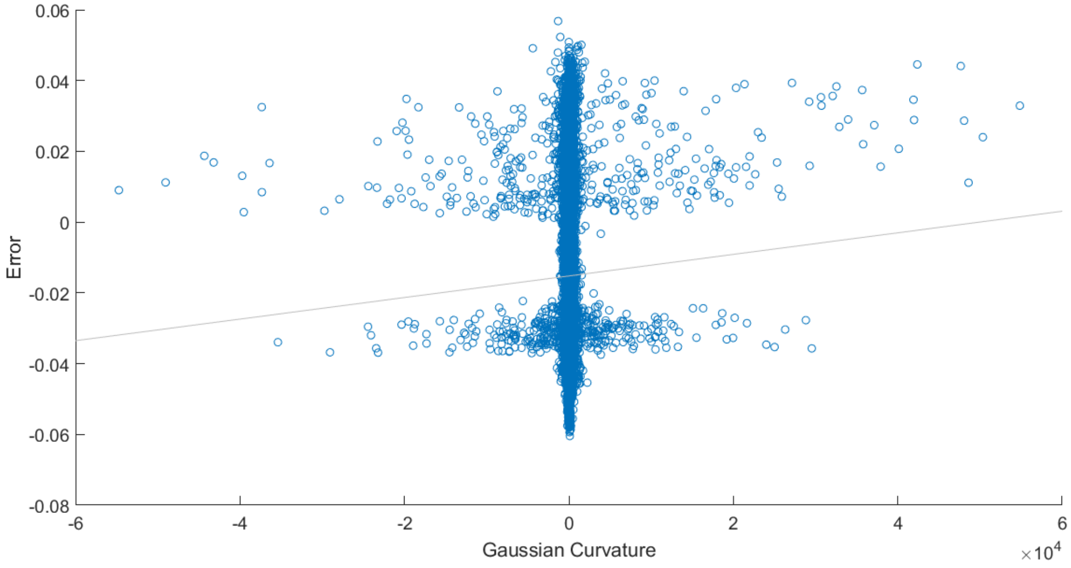

Fig 15 Gaussian Curvature Vs Error for shape1

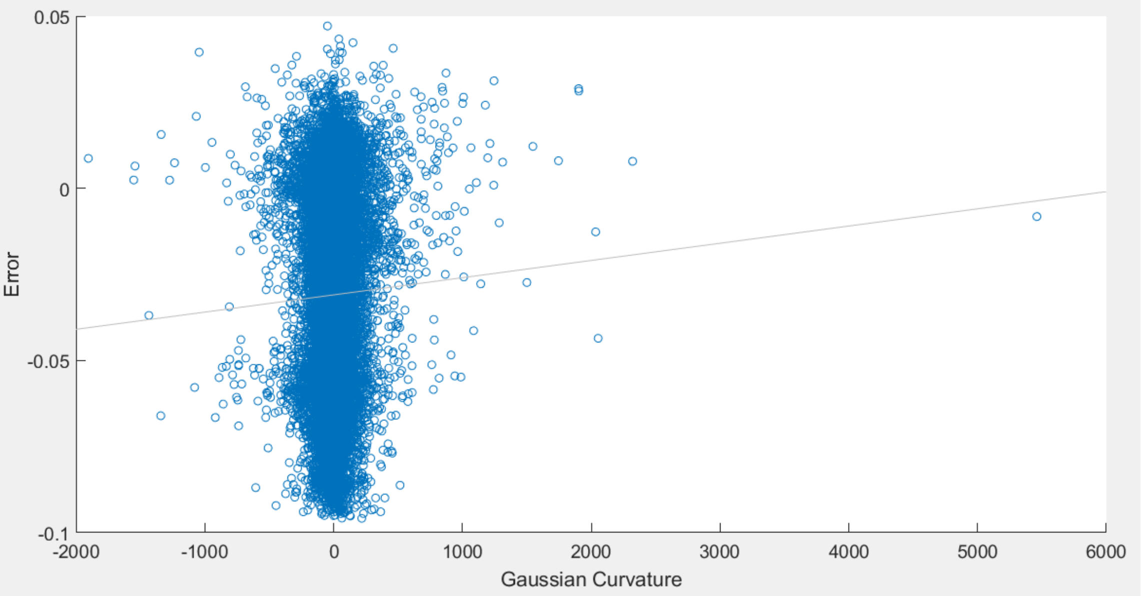

Fig 16 Gaussian Curvature Vs Error for shape 2

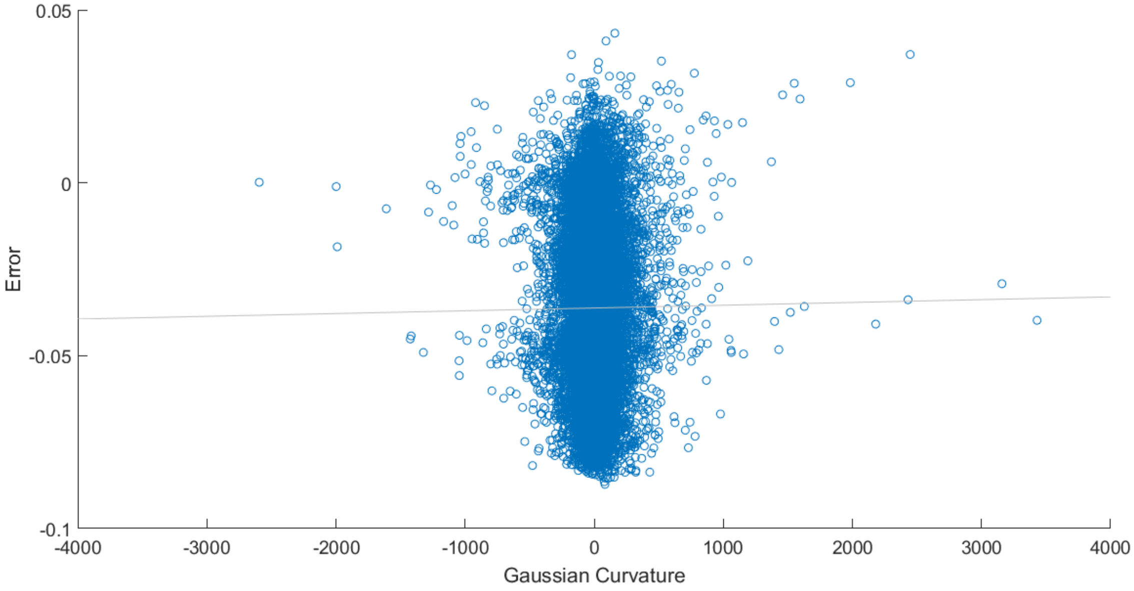

Fig 17 Gaussian Curvature Vs Error for shape 3

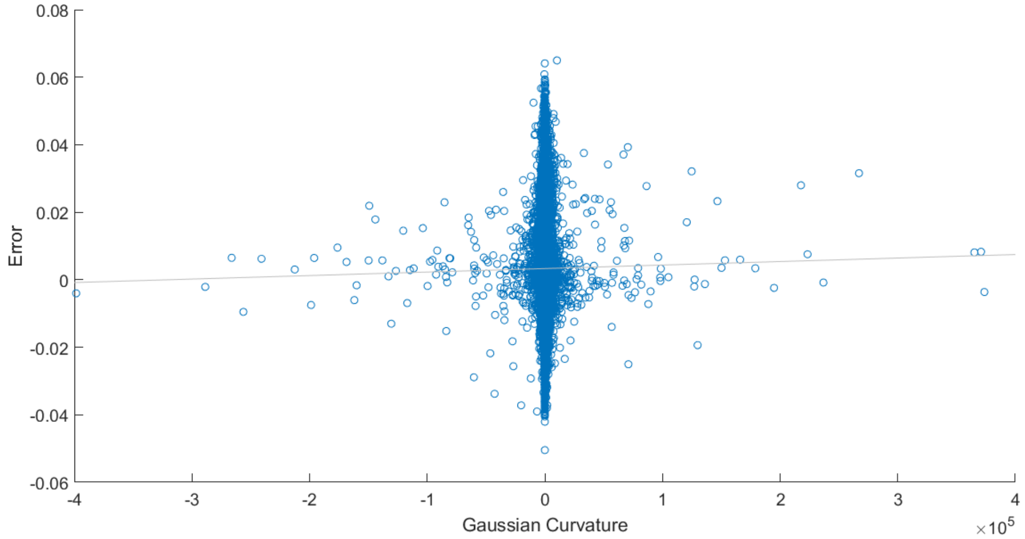

Fig 18 Gaussian Curvature Vs Error for shape 4

As can be seen from the least squares line fitted on the scatter plots in 14,15,16,17,18, there might possibly exist a positive significant relationship between the error and curvature.

3.6 Bi-Cubic-spline interpolation routine for reducing error

As mentioned in the preceding chapter that improving the interpolation routine will have an effect on the error. The reason for not increasing the degrees of the polynomial was to avoid Runge's Phenomenon [9]. Hence a new interpolation routine based on a cubic spline was implemented in place of 2nd order polynomial interpolation [10]. Bicubic splines perform better than cubic polynomial interpolation since they generate lower systematic errors as seen from the previous chapter [11]. The results are shown in the table below for all shapes considered.

We can see from Table 3.1 that the error in the reconstructed figures show a significant reduction by the improvement in interpolation. The improvements in the reconstructed figure can also be seen visually in the Fig. 19 below. While changing the interpolation routine, it was observed that introducing higher order coefficients in the interpolation kernel improved the accuracy. This shall be used in the experimental phase of the thesis to generate better scans of objects.

Table 3.1: Error Metrics

	Max error (2nd Order Polynomial/Cubic Spline)	Mean error (2nd Order Polynomial/Cubic Spline)	Min error (2nd Order Polynomial/Cubic Spline)
Sphere	0.0040/0.0038	-0.0394/-0.0384	-0.070/-0.0659
Shape 1	0.0568/0.0550	-0.0152/ -0.0106	-0.0604/-0.0532
Shape 2	0.0471/ 0.0442	-0.0310/-0.0232	-0.0958/ -0.0912
Shape 3	0.0433/ 0.0366	-0.0362/-0.0342	-0.0872/ -0.0854
Shape 4	0.0650/ 0.0615	0.0033/ 0.0030	-0.0505/-0.0469

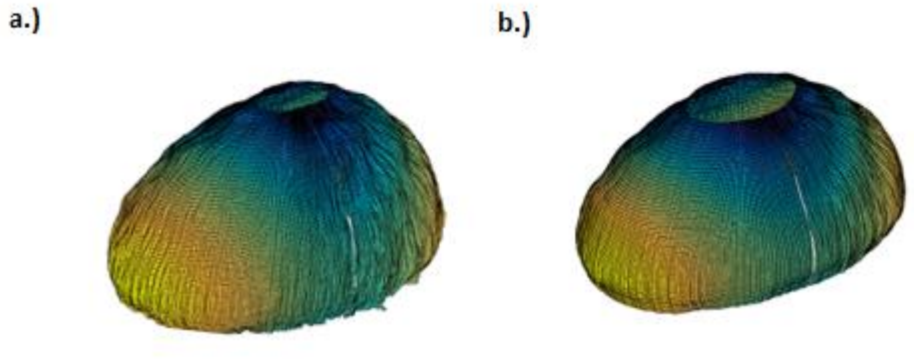

Fig 19: a.) Original reconstructed figure b.) Reconstructed figure using bi-cubic spline interpolation

3.7 Summary

The reconstruction technique proposed was evaluated by simulation in this chapter and was shown to be effective in reconstructing the figures. Also, a method for evaluating the accuracy of the technique was also demonstrated. An improvement in interpolation that improved the accuracy was shown by the implementation of the bicubic spline.

3.8 References:

[1] The interpretation of structure from motion, S Ullman, Proceedings of Royal Society of London, 1979

[2] The Kinetic Depth Effect, H Wallach, D O'Conell, Journal of Experimental Psychology, 1953

[3] view documentation, MATLAB

[4] savesas documentation, MATLAB

[5] DIC Code, R Thompson, D Gianola, C Eberl

https://www.mathworks.com/matlabcentral/fileexchange/12413-digital-image-correlation-and-tracking

[6] pdist2 documentation, MATLAB

[7] Curves and Surfaces, M Abate, F Tovena

[8] Gaussian Curvature: MATLAB code, Ahmed Elnaggar

[9] Runge's Phenomenon: A virtual artifact in image processing, X Shen, F Mohd-Zaid, R Francis, Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition, 2012

[10] interp2 documentation, MATLAB

[11] Digital image correlation technique, H Xie, Y Kang, Optics and Lasers in Engineering, 2015

Chapter 4: Experimental reconstruction and applications

Note: Sections of this chapter are derived from the working manuscript mentioned in the Preface

4.1 Introduction

The concept of reconstruction explained in the previous chapter is validated experimentally using a variety of objects. A variety of applications of the technique will be shown. A detailed explanation of the hardware and code written for this purpose shall also be explained. A few complications and issues arising out of the technique including calibration shall be explained and solved. Herein a new smartphone based three-dimensional reconstruction technique is developed and validated experimentally. A technique to measure the error in the generated point cloud is also presented, which is missing in most reconstruction approaches. The technique is unique as it requires only a smartphone and components that can be printed on a MakerBot™ machine in contrast to other techniques. A benchmark study with a MakerBot™ Digitizer, a commercial scanning system is also presented. The motivation as explained in the earlier chapters is to build an open source flexible reconstruction experimental setup.

4.2 Experimental equipment

The experimental setup used for this study was adapted from the work of jsteuben on the website thingiverse.com. This consisted of a rotating stage powered by a motor and an Arduino controller [1]. A few modifications were made to the original design. The stage consists of a hyperbolic gear and spur gear whose reduction ratio is 72:1. As is known, having more teeth in mesh with each other, which is the case with hyperbolic worm gears reduces the contact stresses and hence increases the life of the assembly. The entire assembly was fabricated on a Makerbot™ machine using a Makerbot+ extruder with Makerbot PLA material. To ensure adequate strength for the worm gear, the infill percentage and number of cores can be optimized based on the torque and power of motor being used. Also since the mating parts must be high precision, a low layer height of 0.20 mm was used. A simple stage was affixed to the assembly consisting of a circular disc and sleeve. An Arduino™ is an integrated device with a built-in microcontroller and several input/output connections. The Arduino can be programmed using an Integrated Development Environment and several libraries with defined functions. In the present application, the library 'Servo' was used. The setup must be calibrated for every object as the mass and mass distribution is different and impacts the rotational speed. The motor used to drive the assembly was a PowerHD® AR 3606HB Continuous Servo which can operate at a maximum speed of 71 RPM at 6V with a torque of 6.7 kg-cm. Proper care was taken to ensure there is no vibration in the assembly as the stage rotates. This is essential to prevent errors during image acquisition from video. The Arduino was programmed in such a way that the stage rotated at a constant angular velocity of 2.5rpm. The setup is shown in Fig.1

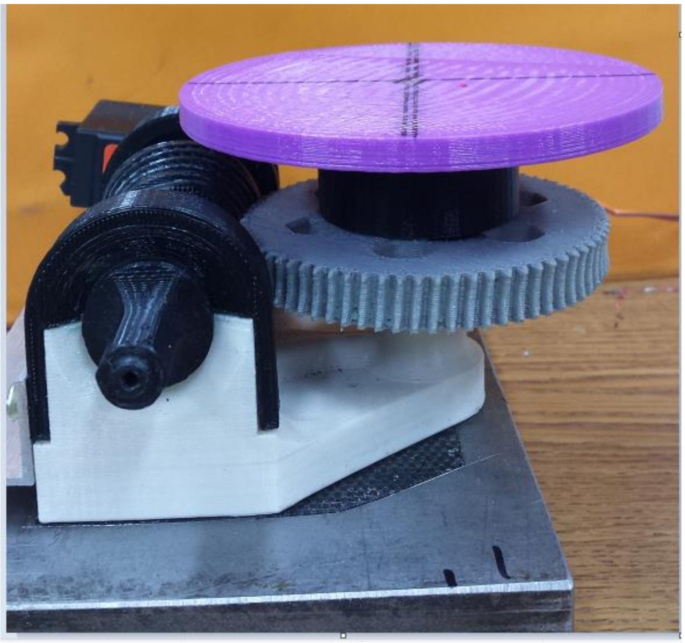

Fig 1 Experimental setup used

A smartphone camera was used for data acquisition. The smartphone used for this purpose was Samsung S4 equipped with thirteen-megapixel camera which could capture videos at 30 frames per second (fps). Appropriate lighting sources were placed close to the setup based on experience such that specular reflection on the surface did not interfere with the image correlation. The frame rate and the rotational speed of 2.5 rpm was used to extract images from the videos captured. The angular movement in each of these images of the object being scanned was 0.5 degrees. Thus 721 images were captured during one complete rotation of the object being scanned. A commercially available tripod was used to mount the camera during data acquisition. During the data acquisition by the camera, care was taken to ensure that the central axis of the camera is in line with the center of the stage and the object being scanned to minimize the effect of parallax. After raw images were extracted, a low pass filter was applied to the images to smooth out the images and ensure proper pixel tracking [2]. Also, powder particles were sprinkled on the objects being scanned to induce high contrast edges and thus aid in the image correlation process.

The idea behind using components made by additive manufacturing or which could be acquired commercially was to make the design and concept readily reproducible at other institutes and organizations. It is envisioned that this methodology will augment reverse engineering capabilities of AM platforms by providing accurate 3D-reconstruction using inexpensive open source components that could be fabricated by the user.

4.3 Initial experimental runs

The initial runs of the algorithm developed and described earlier resulted in a noisy point cloud. This was due to the inherent errors induced by the normalized cross correlation calculation in the frequency domain and can also be attributed to the rotational speed at which the images were captured. An example of the raw point cloud of a conical surface of a paper cup generated is shown in Fig. 2. As can be seen from the data that there exists a high frequency noise pattern which needs to be filtered out.

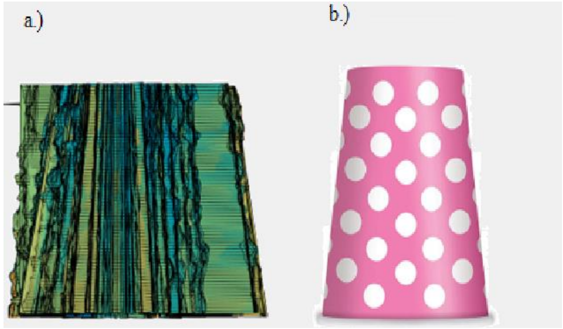

Fig 2 a.) “Noisy” point cloud b.) reference image showing a similar paper cup

4.3.1 Moving Average Filter

The moving average filter is an example of a filter that can attenuate high frequency noise. Essentially, it averages n data points into one data point. Here, n is called the window size of the filter and as it slides along the function of interest, a smooth signal is obtained. The value of ‘n’ determines the smoothness of the output. The input to the moving average filter will be the velocity magnitudes. Some reconstructions carried out with this filter are shown below in Fig. 3 and 4. However, the aspect ratios obtained were visually of poor quality.

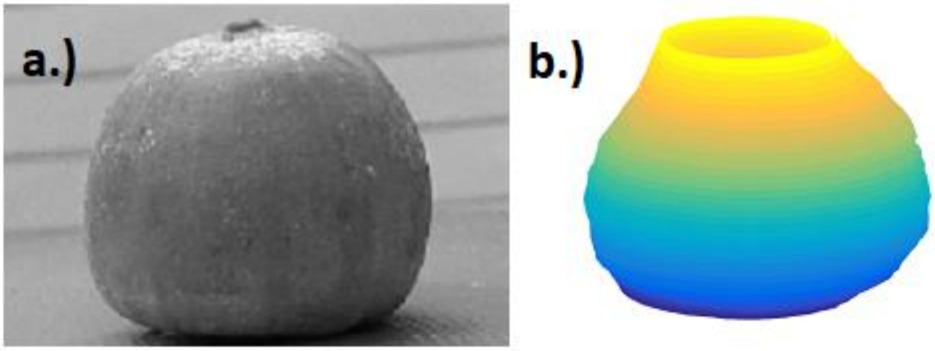

Fig 3: Reconstruction of plant seed pericarp using moving average filter a.) Real image of specimen b.) Reconstructed pericarp with poor aspect ratio

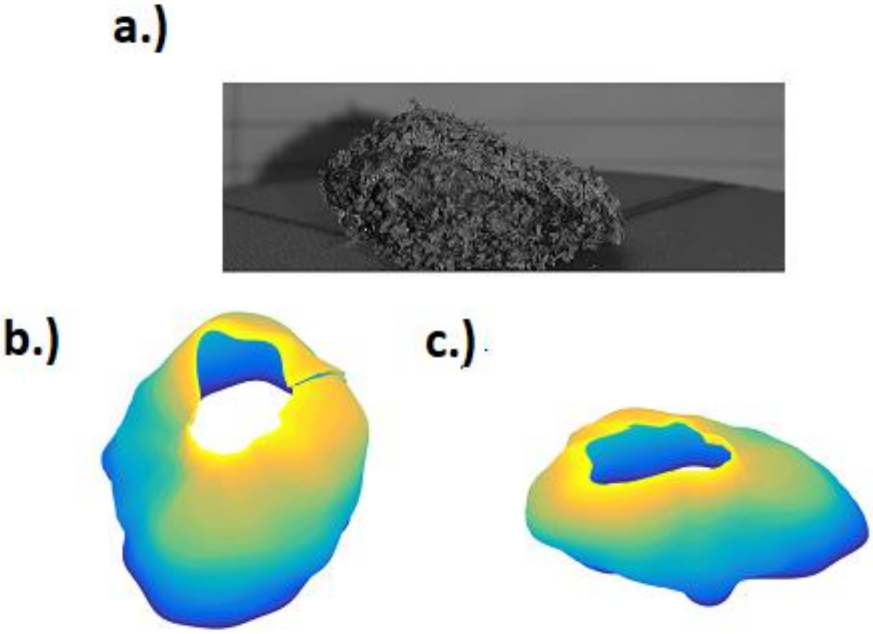

Fig 4: Reconstruction of a stone using moving average filter a.) Real image of stone b.) Reconstructed stone c.) Reconstructed stone with poor aspect ratio

4.3.2 Low pass filters

A moderate low pass filter was used to smooth out the data as it gave better results than the smoothing average filter previously used. This is based on attenuation of high frequency components, which contribute to the noise.

4.4 Pixel Correction, Calibration, and parallax rectification

The resolution N (mm/pixel) was plotted as a function of distance from camera to counter this phenomenon. Gauge blocks were used for accurately determining the resolution by precisely controlling their distance from the camera center. This plot is shown below in the Fig. 5.

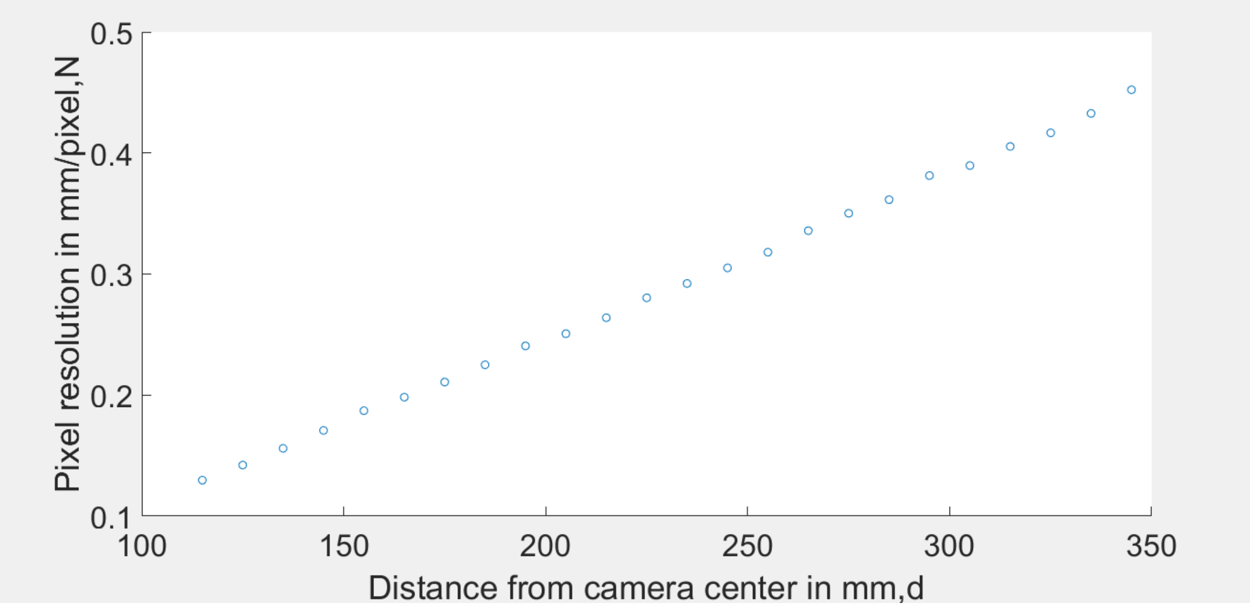

Fig 5 Resolution (mm/pixel) N vs Distance from Target(mm) d

Based on this linear trend, a regression model was fitted and was used to find a pixel correction factor for each point. However, this correction factor was found to be insignificant and not necessary for objects with modest aspect ratios. The relationship can be explained by the linear regression model, $\mathrm{N(d)} = 0.0014 * \mathrm{d(mm)} - 0.0307$ with an R^2 of 0.9994. This curve also serves as a calibration tool as it correlates pixel count to real world units. Subsequently, this curve was used to find the true resolution of the object from the images of the object. This is then used to scale the object. The parallax error associated with the camera was resolved by noting, $N(d) = |Vd/VdP|$, where Vd (mm/s) is the object surface speed in the real space at distance d (mm) from the camera and VdP (pixel/s) is the corresponding DIC-characterized surface speed in the pixel space. Practical implementation of the same necessitates a change in coordinate systems to the object space as, $|V(r)| = N(dA - r)|VrP|$, where dA is the fixed distance between the camera and the axis of rotation of the object and r is the radius of the point on the object under consideration, in real space. Note that the formulation recursively depends on the accurate characterization of r, since $r = N(r)|VrP|\Delta t/\omega\Delta t$ wherein $N(r)$ is a function of r to begin with. To mitigate this, we assumed a constant resolution for the entire object as governed by its distance from the camera as $|Vr| = N(dA - N(dA) * |VrP|\Delta t/\omega\Delta t)$. This assumption is valid while imaging objects with modest aspect ratios. Herein, it will be seen that the error incurred from this approximation is negligible.

4.5 Examples of reconstruction with applications

A few examples of reconstruction using the technique and correction factors described above are shown.

4.5.1 Reconstruction of semi crushed can

As can be seen in Fig. 6, the tool developed works well, in reconstructing the can from the images.

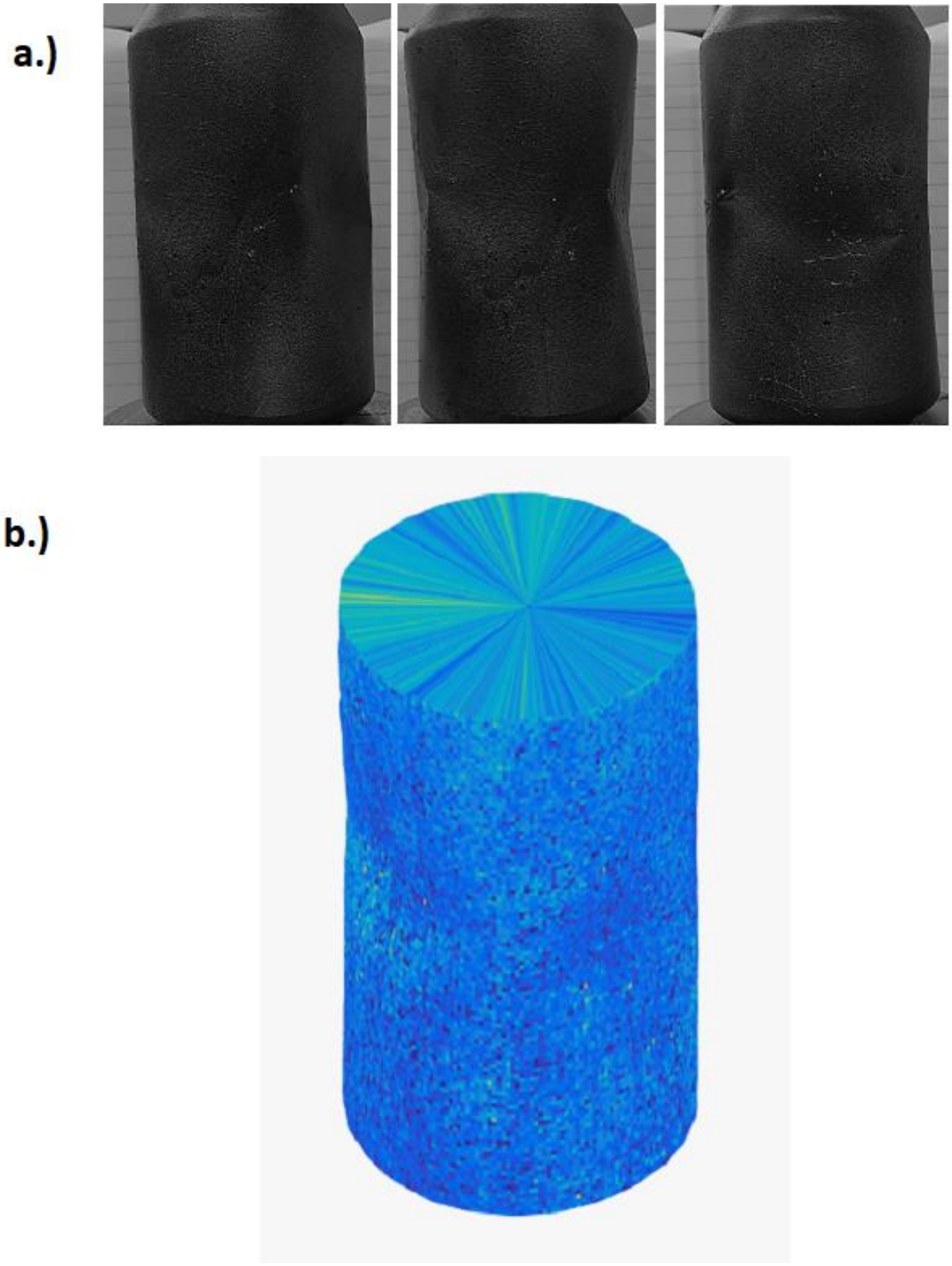

Fig 6 Reconstruction of semi crushed can with parallax scaling a.) Real images of can b.) Reconstructed can

4.5.2 Reconstruction of plant seed pericarp

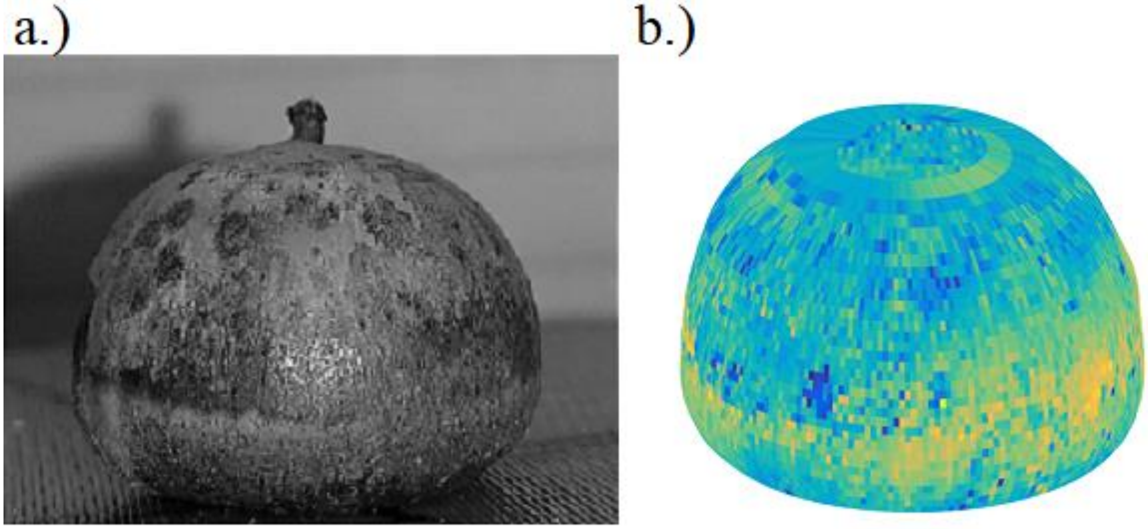

Fig 7 Reconstruction of plant seed pericarp a.) Real image of pericarp b.) Reconstructed shape

As can be seen, the reconstruction in Fig.7 is very close to the actual aspect ratio of the pericarp.

4.6 Benchmarking with MakerBot Digitizer

A test for validation with a commercial three-dimensional scanner, manufactured by MakerBot was carried out. This study was used as a benchmark for the reconstruction technique developed in this thesis. MakerBot digitizer uses two lasers mounted on the left and right of the object being scanned which create a laser line. The stage is rotated and a series of photographs of the laser lines by an inbuilt camera system is collected as shown in Fig.8. An STL file is generated which is then used to create a point cloud. This point cloud is then compared to the point cloud obtained from the same can by DIC reconstruction. The results of this exercise are shown in this section.

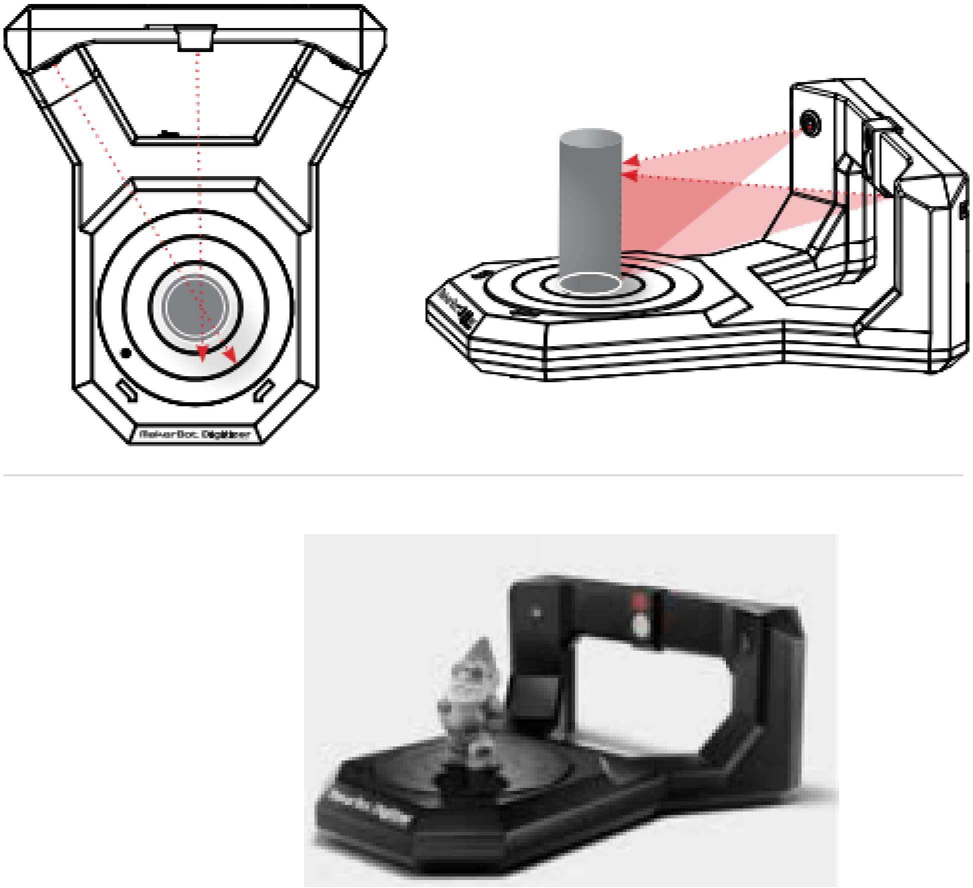

Fig 8 Maker bot digitizer [3]

For generation of error, proper scaling in real world units i.e. mm was done using the resolution curve described in Fig 5.

The interpolation routine used in the cross-correlation engine in this case was the bicubic spline which provably provides better accuracy. A semi crushed can, seen in previous scans was chosen for the benchmarking. A MakerBot image of the can is shown in Fig.9.

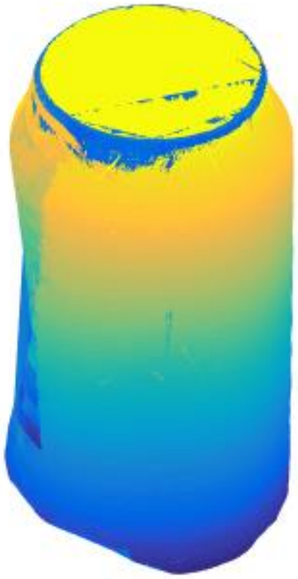

Fig 9 MakerBot Scan of the semi crushed scan

Cloud Compare, an open source project was used to generate surface error through a local modeling mechanism [4]. ‘Local’ modeling is significantly different from the nearest neighbor technique used in the previous chapter. A local model, in this case, a quadratic height function is fitted about the closest point and six neighboring points. This model approximates the surface with which the comparison is to be made. The results from the nearest neighbor technique used in the previous chapter will converge with the results from ‘local’ modeling when size of the point clouds compared is similar. The size of the point clouds compared was similar and controllable in the previous chapter and hence the nearest neighbor technique for error calculation was used. Thus, the error is calculated with an emphasis on accuracy and not approximation. All reported values are absolute magnitudes of error obtained from Cloud Compare. The error mapped on the surface of the DIC reconstructed scan is shown below in Fig 10.

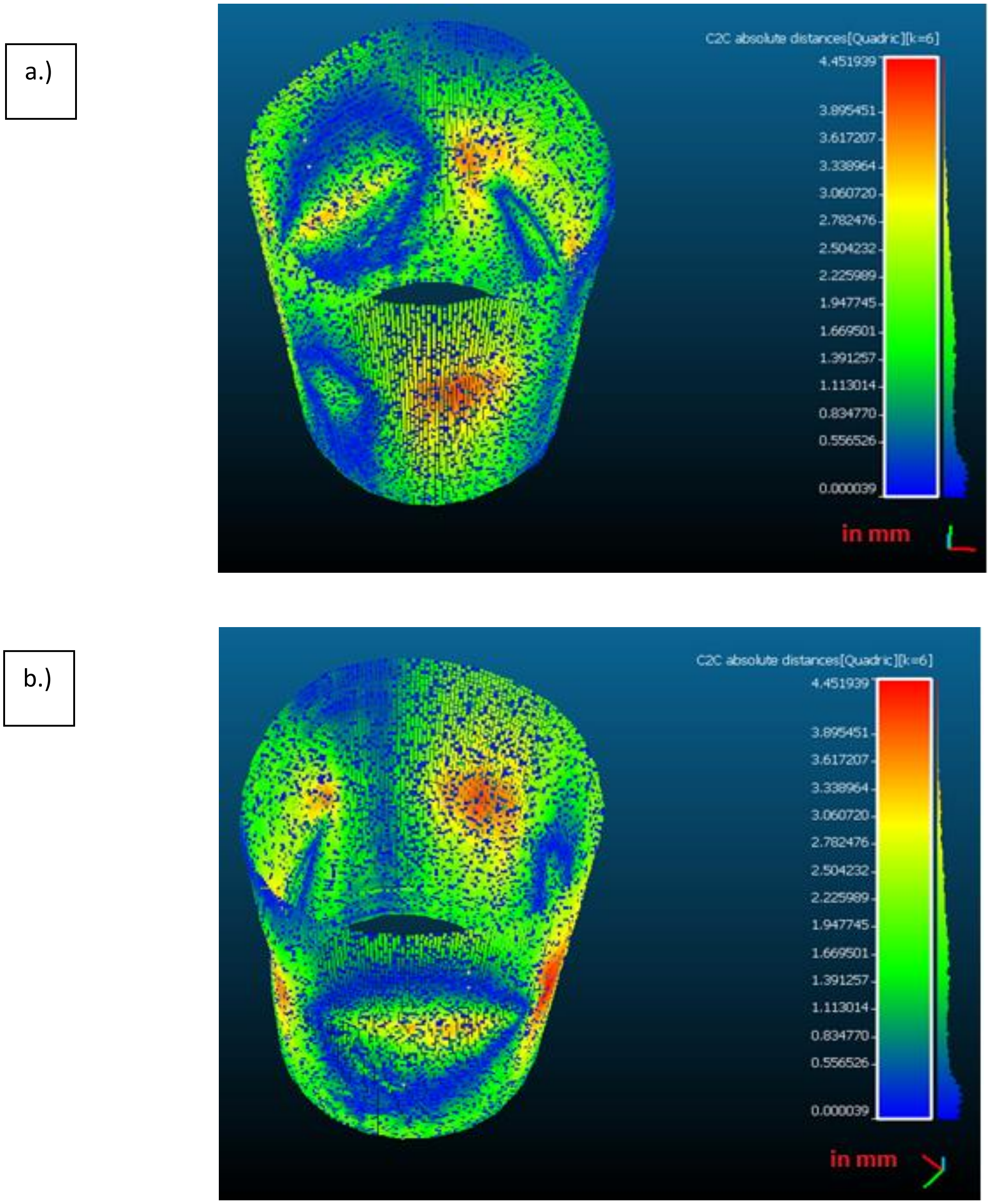

Fig 10a.) b.) Error produced by local modelling mapped on surface of scan generated by DIC

As can be seen, the maximum error is 4.45 mm and the minimum error is 0.000039 mm. With respect to the nominal dimensions of the can, the maximum and minimum errors in percentage are 13.3% and 0.0001172% respectively. The mean error was 1.2974 mm with a standard deviation of 0.9756mm. However, there are very few points with high maximum error resulting in a low overall error. The maximum radius of the can was 36.44 mm as can be seen in Fig 11. The consolidated results can be seen in Fig 12.

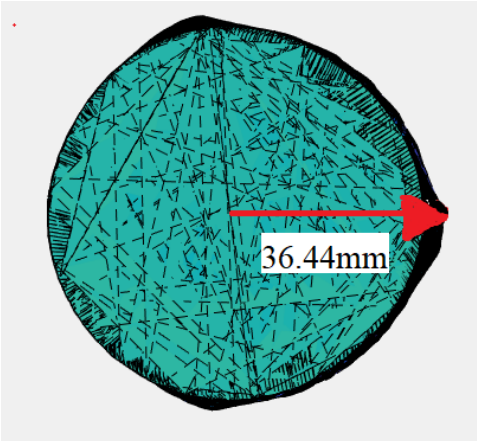

Fig 11 Maximum radius of can :36.44mm obtained from MakerBot model

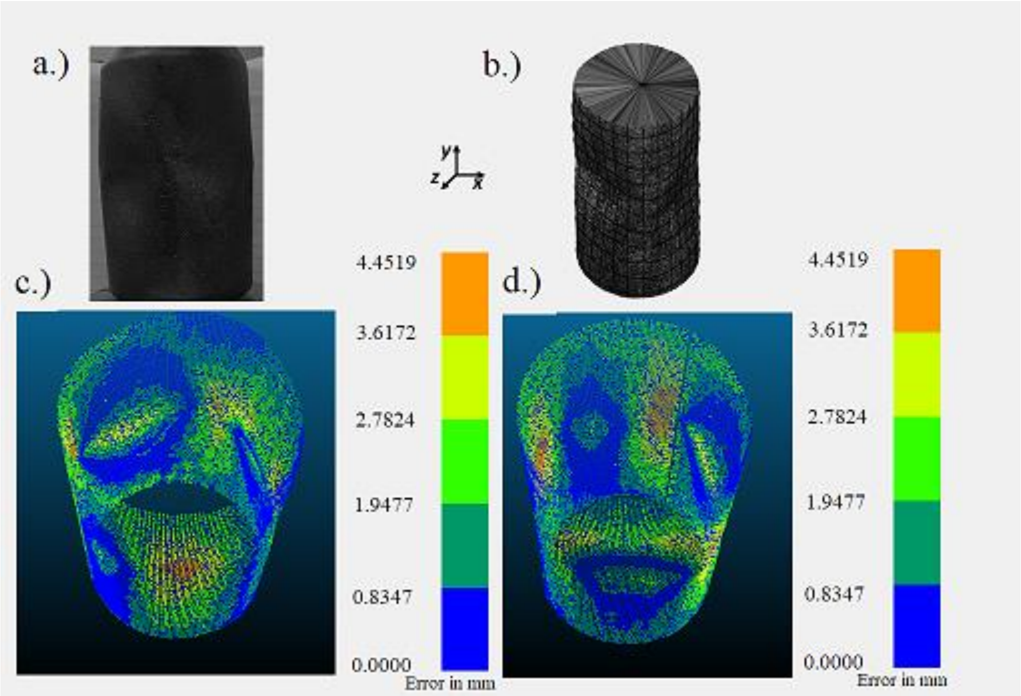

Fig 12 Consolidated figure for reconstructed can error surface

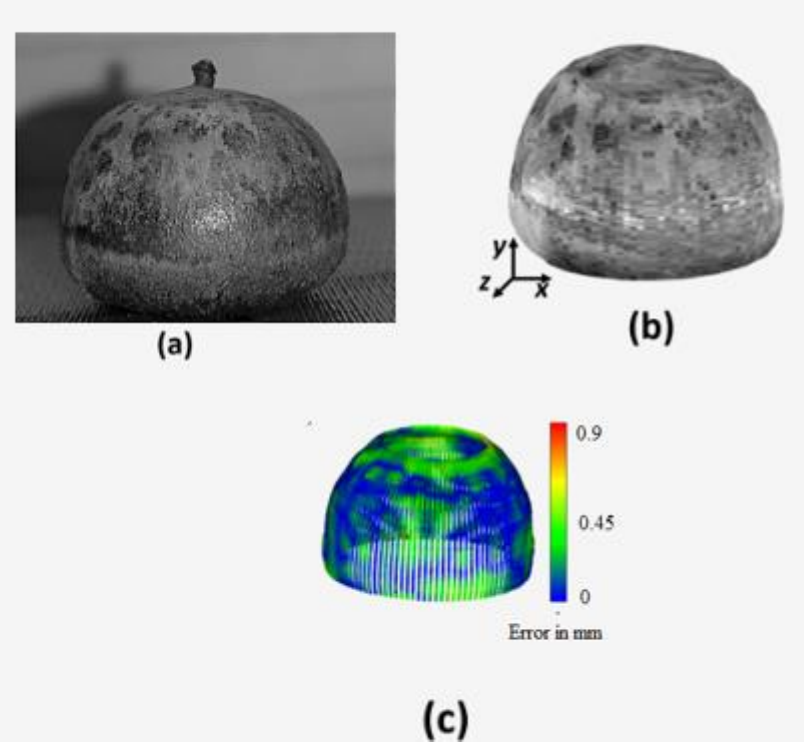

Fig 13 Plant seed pericarp error surface

A similar benchmark study carried out on the pericarp is shown above. As can be seen, the max error is 0.9 mm and the min error is 0mm.With respect to the maximum diameter of the pericarp, the maximum and minimum errors in percentage are 10.28% and 0% respectively. Hence, the reconstructed model from DIC, matches well with the MakerBot model of the pericarp. The maximum radius of the pericarp was 8.75 mm. The consolidated results can be seen in Fig.13.

4.7 Scan of a pristine can

A pristine can without deformation was scanned to verify the present approach to scaling as the nominal dimensions of a standard 12oz US can are known [5]. The results of the scan are shown in Fig.15.

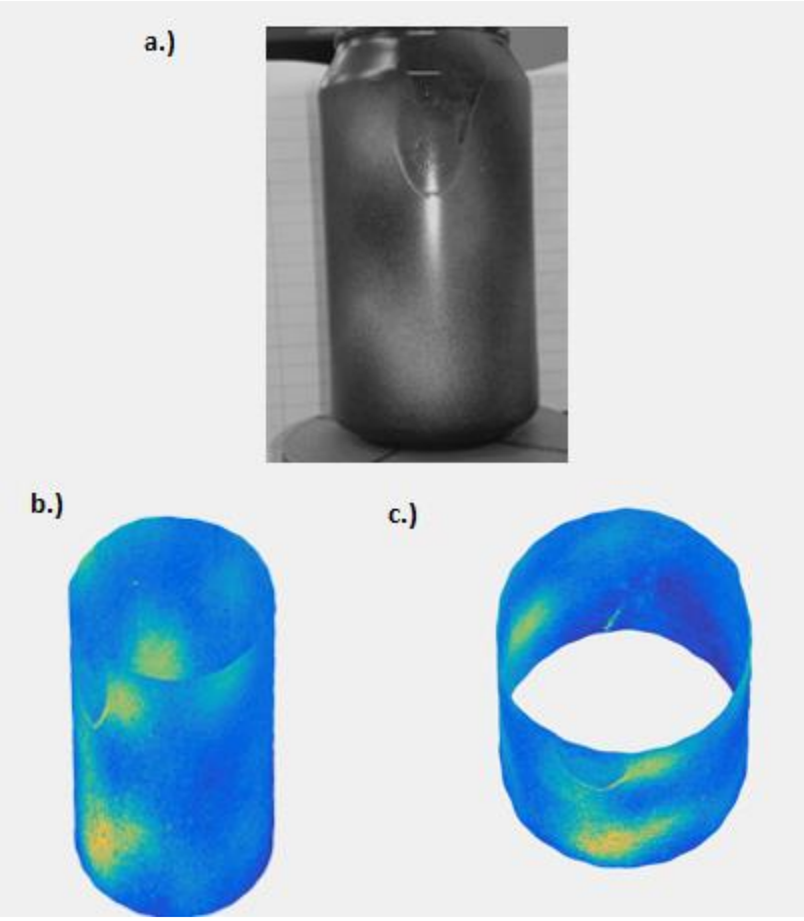

Fig 15 Reconstruction of pristine can a.) Real image of pristine can b.) Reconstructed can c.) Reconstructed can

The diameter of the can calculated from the scaling method adopted was 69.9276 mm which translates to an error of 5.07 %. Since, such an error compared to nominal dimensions is small. It validates the assumption of constant resolution on the surface of the object scanned.

4.8 Conclusion

A novel DIC based volume reconstruction tool fulfilling the requirements presented such as scalability and flexibility has been presented. A better interpolation routine to increase the accuracy was also developed and applied to the experimental setup.

As shown, the current technique can be used to characterize three dimensional objects with accuracy. A few engineering applications of such scans are also shown. A benchmark study that validates the approach is also presented. There is a close coincidence of the point clouds obtained from the MakerBot and DIC based reconstruction. Thus, this technique has been shown to be a viable alternative to commercial scanning systems. An estimate of the total cost of this setup including software can be made at $80, which is considerably lower than commercial three-dimensional scanners.

4.9 References:

[1] https://www.thingiverse.com/thing:343922, jsteuben

[2] Image Correlation for Shape, Motion and Deformation Measurements Basic Concepts, Theory and applications, M Sutton, J Orteu, H Schreier, Springer, 2009

[3] MakerBot Digitizer User Manual, MakerBot

[4] CloudCompare :3D point cloud and mesh processing software, Open Source Project User Manual

[5] Technical Specifications-12oz, Cask Brewing Systems

Chapter 5: Future Work

5.1 Focus areas in subsequent research

Some future work that can be carried out is the integration of volumes into a composite scan for scanning complete objects using Iterative Closest Point (ICP) algorithm and actuated camera movement. A more accurate Newton-Raphson method or Inverse Compositional Gauss Newton(ICGN) method may also be devised to improve accuracy in the spatial domain as opposed to performing correlation in the frequency domain which is known to improve accuracy [1]. Further validation with a Coordinate Measuring Machine (CMM) may be carried out to generate a point cloud with "real" physical data since the MakerBot Digitizer does not specify an error term associated with its measurements, apart from a 2mm envelope within which the object's true location will be found. Also, some errors may be attributed to the scaling method based on the resolution curve in Fig 5 in the previous chapter. A more accurate scaling method may be devised to bring down the errors. A perspective correction may be applied to images of tall objects, since it is suspected to introduce errors at the edges. Integration of this technique along with other reconstruction technique such as shape from silhouette and shape from shading may be used to devise a hybrid approach to reconstruction that can potentially recreate complete objects. The process of capturing and processing images may be automated with a trigger mechanism that can be programmed in the Arduino that can make scanning easier.

References:

[1] A fast digital image correlation method for deformation measurement, B Pan, K Li, Optics and Lasers in Engineering, 2011

Appendix: MATLAB Code

Chapter 3

Section 3.8: Deformation measurement

```
clc
clear
close all
image1=load('testimg1.dat');
image2=load('testimg2.dat');
size_=150;
Displx=zeros(size_);
Disply=zeros(size_);

moving=image1;
deform=zeros(2,2,100);
fixed=image2;
for i=1:1:size_
   for j=1:1:size_
movingPoints = [i j];
fixedPoints = [i j];
movingPointsAdjusted =
cpcorr(movingPoints,fixedPoints,moving(:,:,1),fixed(:,:,1));
deformx=movingPointsAdjusted(1)-[i];
deformy=movingPointsAdjusted(2)-[j];
deform=[deformx deformy];
Displx(i,j)=deform(1);
Disply(i,j)=deform(2);
   end
end
imagesc(Disply),colorbar;axis equal;title('Displacement y');
figure,imagesc(Displx),colorbar; axis equal;title('Displacement x');
```

Chapter 4

Section 4.4: Shape Creation Module

```
clc
clear
close all
figure
dimension=200;
speckle_dimension=300;

perturbation=0.3;
wave_number=3;
wave_number_theta=2;
speckle_pattern=[];
img=imread('Capture1.png');
img=double(squeeze(img(:,:,1)));
speckle_pattern_seed=img(1:speckle_dimension,size(img,2)-
speckle_dimension+1:size(img,2));
 wave_number_theta=wave_number-1;
% creating the speckle pattern to adorn the shape with a high resolution
field of asperities
```

```
for col_multiples=1: floor(dimension/speckle_dimension)
  speckle_pattern=[speckle_pattern speckle_pattern_seed];

end
speckle_pattern=[speckle_pattern
speckle_pattern_seed(:,1:mod(dimension,speckle_dimension))];
speckle_pattern_seed=speckle_pattern;
for row_multiples=1: floor(dimension/speckle_dimension)-1
  speckle_pattern=[speckle_pattern;speckle_pattern_seed];
end
speckle_pattern=[speckle_pattern;
speckle_pattern_seed(1:mod(dimension,speckle_dimension),:)];

% adding 1 extra layer. This layer is added to accommodate the way sphere
works
% The sphere command creates spherical coordinate spaces of size dimension+1
X dimension+1 X dimension+1
speckle_pattern=[speckle_pattern speckle_pattern(:,1)];
speckle_pattern=[speckle_pattern;speckle_pattern(1,:)];

[X, Y, Z]=sphere(dimension);



sin_phi=sin(wave_number*acos((X./(X.^2+Y.^2).^0.5)));
sin_phi(:,(size(sin_phi,2)+1)/2:size(sin_phi,2))=-
1*sin_phi(:,(size(sin_phi,2)+1)/2:size(sin_phi,2));
cos_phi=cos(wave_number*acos((X./(X.^2+Y.^2).^0.5)));
cos_theta=cos(wave_number_theta*acos((Z./(X.^2+Y.^2+Z.^2).^0.5)));
sin_theta=sin(wave_number_theta*asin(((X.^2+Y.^2).^0.5)./(X.^2+Y.^2+Z.^2).^0.
5));
X=X+perturbation*cos_phi.*sin_theta;
Y=Y+perturbation*sin_phi.*sin_theta;
Z=Z+perturbation*cos_theta;
radius=(X.^2+Y.^2).^0.5;
X=X/max(max(radius));
Y=Y/max(max(radius));
Z=Z/max(max(radius));
Z=Z-mean(reshape(Z,[],1));
radius=radius/max(max(radius));
surface(X,Y,Z,radius,'edgecolor','none');axis equal;axis off;
%surface(X,Y,Z,speckle_pattern,'edgecolor','none');axis equal;axis
off;colormap gray
axis vis3d
% keyboard
drawnow
projection_number=1;
for view_theta = 0:1:360
%view_theta=0.5;
view(view_theta,0)
drawnow
 %  wait(0.01)
 %   keyboard
file_name=['test3' num2str(projection_number,'%03d') '.tif'];
%file_name(6)=[];
saveas(gcf,file_name,'tiff');
```

```
display(sprintf('saving projection: ',projection_number));
projection_number=projection_number+1;
%if projection_number==100
 %      break;
  % end
end
```

Section 4.4: Reconstruction

```
load validx.dat
load validy.dat
load grid_x.dat
load grid_y.dat
omega=pi/180;
validx=validx-max(grid_x);
velocity=((validx.^2).^0.5)/omega;
Xc=zeros(size(velocity,1),size(velocity,2));
Yc=zeros(size(velocity,1),size(velocity,2));
Zc=repmat(grid_y,1,360);

for i=1:1:360
    Xc(:,i)=velocity(:,i).*cos(i*2*pi/360);
    Yc(:,i)=velocity(:,i).*sin(i*2*pi/360);
end
Xc=Xc/max(max(velocity));
Yc=Yc/max(max(velocity));
Zc=Z/max(max(velocity));
Zc=flipud(Zc);
theta=11;
R=[cos(theta) -sin(theta);sin(theta) cos(theta)];
rotatedxy=R*[Xc(:) Yc(:)]';

rotatedxy=rotatedxy';

Xc=reshape(rotatedxy(:,1),[size(Xc,1) size(Xc,2)]);
Yc=reshape(rotatedxy(:,2),[size(Yc,1),size(Yc,2)]);
surf(Xc,Yc,Zc,velocity);axis equal;axis vis3d;axis off
```

Section 4.5 Error Derivation

```
REXX=X(:); % These are original shape coordinates
REYY=Y(:);
REZZ=Z(:);
OriginalCo=[REXX REYY REZZ];
OriginalCo(any(isnan(OriginalCo),2),:) = [];
REXX=OriginalCo(:,1);
REYY=OriginalCo(:,2);
REZZ=OriginalCo(:,3);

[theta1,rho1,z1] = cart2pol(REXX,REYY,REZZ);
```

```
ReX=Xc(:);%These are reconstructed coordinates
ReY=Yc(:);
ReZ=Zc(:);
Newco=[ReX ReY ReZ];
[ttheta rrho zz]=cart2pol(ReX,ReY,ReZ);
REXXX=zeros(size(OriginalCo,1),1);
REYYY=zeros(size(OriginalCo,1),1);
REZZZ=zeros(size(OriginalCo,1),1);
for i=1:size(Newco,1)
    REXXX(i)=ReX(i);
    REYYY(i)=ReY(i);
    REZZZ(i)=ReZ(i);
end

NewCo1=[REXXX REYYY REZZZ];
[theta,rho,z]=cart2pol(REXXX,REYYY,REZZZ);
sim=[theta rho z];
sim1=[theta1 rho1 z1];
[D,I] = pdist2(sim1,sim,'euclidean','Smallest',4);
diff=zeros(size(Newco,1),1);
for i=1:size(Newco,1)


   diff(i)=rho(i)-(rho1(I(1,i))+rho1(I(2,i))+rho1(I(3,i))+rho1(I(4,i)))/4;




end

figure,set(gca,'fontsize',48),plot(1:size(Newco,1),diff),set(gca,'fontsize',1
2),xlabel('Index number of points');ylabel('Error');
```

Section 4.5 Gaussian Curvature

```
Gaussian_cu=gcurvature(X,Y,Z);

figure,scatter(Gaussian_cu(:),diff(:));set(gca,'Fontsize',14);xlabel('Gaussia
n Curvature');ylabel('Error')

d=fitlm(Gaussian_cu(:),diff(:));

lsline
```

Section 4.6 Bicubic spline Interpolation

Replace following script in place of findpeak to generate bicubic interpolation results

```
function [xpeak, ypeak, max_f] = findpeak222222(f,subpixel)
[max_f, imax] = max(abs(f(:)));
[ypeak, xpeak] = ind2sub(size(f),imax(1));

if ~subpixel || ...
```

```
    xpeak==1 || xpeak==size(f,2) || ypeak==1 || ypeak==size(f,1) % on edge
    return % return absolute peak

else

    x=-1:1;
    y=-1:1;
    [X,Y]=meshgrid(x,y);
        fXY=[f(ypeak-1,xpeak-1) f(ypeak-1,xpeak) f(ypeak-
1,xpeak+1);f(ypeak,xpeak-1) f(ypeak,xpeak) f(ypeak,xpeak+1);f(ypeak+1,xpeak-
1) f(ypeak+1,xpeak) f(ypeak+1,xpeak+1)];
    xq=-1:0.001:1;
    yq=-1:0.001:1;
    [Xq,Yq]=meshgrid(xq,yq);
    Vq=interp2(X,Y,fXY,Xq,Yq,'spline');
    surf(Xq,Yq,Vq);



    [max_fXY, imax] = max(abs(Vq(:)));
[ypeak1, xpeak1] = ind2sub(size(Vq),imax(1));
xpeak1 = -1+xpeak1*0.001;
    ypeak1 = -1+ypeak1*0.001;
    xpeak=xpeak+xpeak1;
    ypeak=ypeak+ypeak1;

end
```

Chapter 5
Section 5.3.1 : Moving average filter

```
clc
clear all
close all
load('validx.dat');
load('validy.dat');
validx_filtered=zeros(size(validx,1),size(validx,2));

for i=1:1:size(validx,1)
    nn=validx(i,:);
 xx=movmean(nn,57);
    validx_filtered(i,:)=xx;
end
load grid_x.dat
load grid_y.dat
omega=0.5*pi/180;
validx_filtered=validx_filtered-max(grid_x);
velocity=((validx_filtered.^2).^0.5)/omega;

Xc=zeros(size(velocity,1),size(velocity,2));
Yc=zeros(size(velocity,1),size(velocity,2));
Zc=repmat(grid_y,1,720);

for i=1:1:720
    Xc(:,i)=velocity(:,i).*cos(i*0.5*pi/180);
```

```
  Yc(:,i)=velocity(:,i).*sin(i*0.5*pi/180);
end
Xc=Xc/max(max(velocity));
Yc=Yc/max(max(velocity));
Zc=Zc/max(max(velocity));
Zc=flipud(Zc);
surf(Xc,Yc,Zc);axis equal;axis vis3d;axis off
```

Section 5.3.2 : Low pass filter

```
 clc
clear
close all
file_name='validx.dat';
data=load(file_name);

for n=1:size(data,1)

% n=2;
sgnl=data(n,:);
img_fft=fft(sgnl);
img_fft_shift=fftshift(img_fft);
fourier_space=log(1+abs(img_fft_shift));
% center_=round(size(img_fft,2)/2);
center_=find(fourier_space==max(fourier_space));
for row=1:size(img_fft,2)
    distance=abs(row-center_);
%    if distance==0
%      keyboard;
%    end
    img_fft_shift(row)=img_fft_shift(row)*exp(-1*(distance*0.14));
end
img_fft_reshift=fftshift(img_fft_shift);
filtered_sgnl=abs(ifft(img_fft_reshift));
data(n,:)=filtered_sgnl;
end
processed_file_name=[file_name(1:length(file_name)-4) '_filtered.dat'];
save(processed_file_name,'data','-ascii');

% plot(sgnl);hold on; plot(sgnl,'.');
% plot(filtered_sgnl);hold on; plot(filtered_sgnl,'.');
% keyboard
clc
close all
file_name='validy.dat';
data=load(file_name);

for n=1:size(data,1)
sgnl=data(n,:);
img_fft=fft(sgnl);
img_fft_shift=fftshift(img_fft);
fourier_space=log(1+abs(img_fft_shift));
```

```
% center_=round(size(img_fft,2)/2);
center_=find(fourier_space==max(fourier_space));
for row=1:size(img_fft,2)
        distance=abs(row-center_);
        img_fft_shift(row)=img_fft_shift(row)*exp(-1*(distance*0.14));
end
img_fft_reshift=fftshift(img_fft_shift);
filtered_sgnl=abs(ifft(img_fft_reshift));
data(n,:)=filtered_sgnl;
end
processed_file_name=[file_name(1:length(file_name)-4) '_filtered.dat'];
save(processed_file_name,'data','-ascii');
% plot(sgnl);hold on; plot(sgnl,'.');
% plot(filtered_sgnl);hold on; plot(filtered_sgnl,'.');
```

Section 5.4 Parallax Scaling

```
clear
% close all
figure
load validx_filtered.dat
validx=validx_filtered;
% load validy_filtered.dat
% validy=validy_filtered;
load grid_y.dat
load grid_x.dat
load filenamelist.mat
color_map=[];
col=1;
% figure
resolution=1/(size(filenamelist,1)/360);
d_theta=resolution*pi/180;
validx=validx-grid_x*ones(1,size(validx,2));
% validy=validy-grid_y*ones(1,size(validx,2));

velocity_mag=(validx.^2).^0.5;
velocity_mag=velocity_mag/d_theta;

distance=241;
cam_resolution=0.0014*distance-0.0307;
% distance=80;
scaling_=(0.0014*abs(distance-velocity_mag*cam_resolution)-0.0307);
velocity_mag=velocity_mag.*scaling_;
theta=0:resolution:360;
theta=theta*pi/180;
cos_theta=cos(theta);
sin_theta=sin(theta);
X=zeros(size(velocity_mag));
Y=zeros(size(velocity_mag));

for row=1:size(filenamelist,1)
    img=imread(filenamelist(row,:));
    vector=[];
    for col=1:size(grid_y,1)
```

```
        vector=[vector img(round(grid_y(col)),round(grid_x(col)))];
    end
    vector=vector';
    color_map=[color_map vector];
end
for row=1:size(X,1)-3
    for col=1:size(X,2)
        X(row,col)=velocity_mag(row,col)*cos_theta(col);
        Y(row,col)=velocity_mag(row,col)*sin_theta(col);
    end
end

Z=grid_y(:,1)*ones(1,size(X,2));
Z=Z-mean(reshape(Z,[],1));
Z=flipud(Z);
% Pix_SS=get(0,'screensize');
% screen_resolution_adjustment=Pix_SS(3)/Pix_SS(4);
Z=Z*cam_resolution;
% Z=Z*cam_resolution*screen_resolution_adjustment;
row=find(sum(X')==0);
X(row,:)=[];
Y(row,:)=[];
Z(row,:)=[];
color_map(row,:)=[];
X=[X X(:,1)];
Y=[Y Y(:,1)];
Z=[Z Z(:,1)];
% subplot(2,1,1)
imshow(imread('07062procantest3002.tif'));hold on
% plot(grid_x,grid_y,'.black');
% subplot(2,1,2)
figure
surface(X(3:end,:),Y(3:end,:),Z(3:end,:),color_map(3:end,:),'edgecolor','none
');axis equal;  axis off;view (0,0);axis vis3d;
alpha(1);hold on
% plot3(X(:,1),Y(:,1),Z(:,1),'.black');
cam_resolution/max(max(scaling_))
cam_resolution/min(min(scaling_))
```