

**OPTICS BASED LOW COST FULL-FIELD IN-PLANE
DISPLACEMENT MEASURING TECHNIQUE**

A. Diluxshan 168018G

Master of Science of Engineering

Department of Civil Engineering

University of Moratuwa

Moratuwa

Sri Lanka

December, 2017

OPTICS BASED LOW COST FULL-FIELD IN-PLANE DISPLACEMENT MEASURING TECHNIQUE

A. Diluxshan 168018G

The Research Thesis was submitted in partial fulfilment of the requirements for the
Degree of Master of Science of Engineering

Supervised by Dr. H.M.Y.C. Mallikarachchi



Department of Civil Engineering

University of Moratuwa

Moratuwa

Sri Lanka

December, 2017

DECLARATION

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

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

.....

Date: October 23, 2017

A. Diluxshan

The undersigned hereby certified that they have read and recommended the thesis for the acceptance in partial fulfilment of the requirements for the Degree of Bachelor of Science of Engineering.

.....

Date: October 23, 2017

Dr. H.M.Y.C. Mallikarachchi

ABSTRACT

Optics Based Low-cost Full-field In-plane Displacement Measuring Technique

Displacement/strain measurement plays a significant role in understanding structural and material behaviour. In the local context, commonly used contact based systems have a major limitation when it comes to measuring deformation of a surface. A measuring probe can record the displacement/strain only at a single location and hence require a larger number of pre-defined measuring points in order to calculate the deformation of the surface. Furthermore, the comparative stiffness between the measuring probe and the object under consideration has to be significantly larger. A non-contact optic based system is an ideal candidate for overcoming the problems encountered in such systems.

This research attempts to utilize image processing techniques in order to develop a low cost non-contact based deformation measuring technique to measure the in-plane deformation of an object. A set of pre-defined measuring points (targets) were marked on specimen surfaces and a series of images with a commonly available camera was taken while applying a load. The images were then processed using the image processing tools available in MATLAB software to track the motion of predefined targets. The calculated displacements were validated against the physical measurements taken during the bending test.

Key Words: *digital image correlation, DIC, image processing, non-contact measurements, pattern matching*

ACKNOWLEDGEMENT

I would like to make this opportunity paying gratitude for all the people who gave their helpful hand during my research study as it would never get succeeded without them.

I would like to express my profound gratitude and deep regards to the supervisor of this research, Dr. H.M.Y.C. Mallikarachchi, Senior Lecturer, Dept. of Civil Engineering, University of Moratuwa, for giving me the guidance and continuous supervision and sharing knowledge and time in work on the research project.

I appreciate the financial assistance given by Senate Research Council (SRC) of University of Moratuwa and National Research Council (NRC) of Sri Lanka.

I obligate to thank Prof. W.P.S. Dias, Dr. C.S. Lewangamage, Dr. Indika Perera and Dr. R.U. Halwatura for their valuable comments and advices given during research progress evaluations.

Last but not least, I am truly grateful to many people including the technical staffs in the laboratory and my colleagues who have helped me in various ways during my research. I would like to make this as an opportunity to express my sincere gratitude towards them all.

TABLE OF CONTENTS

DECLARATION	i
ABSTRACT	ii
ACKNOWLEDGEMENT	iii
TABLE OF CONTENTS	iv
LIST OF FIGURES	vi
LIST OF TABLES	vii
LIST OF ABBREVIATIONS	vii
1 INTRODUCTION	1
1.1 Background	1
1.1.1 Contact Based Systems	1
1.1.2 Non-contact Based System	2
1.2 Objectives of the Research Study	3
1.3 Outline	3
2 LITERATURE REVIEW	4
2.1 Digital Image Correlation	4
2.1.1 Selection of Experimental Setup	5
2.1.2 Digital Imaging	5
2.1.3 Pattern Matching	7
2.1.4 Pattern Recognition	8
2.2 Applications	9
2.3 Camera Calibration	12
2.3.1 Calibration Algorithm:	13
2.4 Correlation Methods:	14
3 DEVISING PHYSICAL SETUP AND CAMERA CALIBRATION	15
3.1 Identifying External Factors to be Considered	15
3.1.1 Contrasting Background	16
3.1.2 Lighting and Shades	16
3.1.3 Camera Position	17
3.1.4 Thickness Effect of the Targets	17
3.2 Camera Calibration	18
3.2.1 Experimental Setup	19
3.2.2 Processing Calibration Images	21
4 IDENTIFICATION OF PREDEFINED TARGETS	28
4.1 Measuring Rigid Body Motion with Predefined Targets	28
4.1.1 Preparation of Targets	28

4.1.2	Capturing Images-----	29
4.1.3	MATLAB Algorithm for Identifying Targets-----	30
4.1.4	Analysis and Results of Rigid Body Motion Test -----	32
4.1.5	Problems Encountered and the Solutions: -----	33
4.2	Measuring In-plane Deflection of Flexible Structures -----	34
4.2.1	Three-point Bending Setup -----	34
4.2.2	Image Processing – Flexible Beam -----	36
4.2.3	Comparison of Vertical Deflections -----	38
5	FULL-FIELD DEFORMATION MEASUREMENT-----	42
5.1	Template Matching: -----	42
5.2	Pattern Matching-----	43
5.3	Surface Deformation under Uniaxial Tensile Testing -----	47
5.3.1	Uniaxial Tensile Test Setup-----	47
5.3.2	Processing Images for Full-field Deformation -----	49
5.3.3	Comparison of Results from Uni-Axial Tensile Test -----	53
5.4	Cost Comparison of the Developed Technique-----	61
6	CONCLUSION AND FUTURE WORK -----	62
6.1	Future Work -----	62
	REFERENCES -----	64
	BIBLIOGRAPHY -----	65
	APPENDICES-----	I
	Appendix A: Equipment Used in the Experiment-----	I
	A.1 Camera: -----	I
	A.2 Apparatus: -----	I
	A.3 Universal Testing Machine-----	II
	A.4 Extensometer -----	III
	A.5 Camera EOS 700D -----	III
	Appendix B: MATLAB Algorithm -----	VI
	B.1 Calibration Algorithm-----	VI
	B.2 Output of the above (Appendix B.1) Algorithm:-----	VIII
	B.3 Finding Out the Centre of the Circles-----	X
	B.4 Timber Deflection Measurement -----	XV
	B.5:Template Matching Algorithm-----	XXI
	B.6 Pattern Recognition Algorithm -----	XXII
	Appendix C: Strain Contours-----	XXVIII
	Specimen 1:-----	XXVIII
	Specimen 2:-----	XXXI

LIST OF FIGURES

Figure 1.1: Contact-based measurement systems (a) dial gauge (b) strain gauges-----	1
Figure 1.2: Observing optical flow in an object using DIC -----	2
Figure 2.1: Optical analysisi of an object for deformation -----	4
Figure 2.2: Typical electronic textile of a material -----	5
Figure 2.3: (a) Important parameters in digital imaging (b) pixel arrangement -----	6
Figure 2.4: Histogram of an image (a) sample image (b) corresponding image histogram--	6
Figure 2.5: Pattern matching correlation parameters -----	7
Figure 2.6: Particle movement inside ROI and ROI movement-----	8
Figure 2.7: 3D ESPI Ssystem (Q-300)-----	9
Figure 2.8: Image analysisi (a) strain of bridge pier (b) vibration of pipes (red)-----	10
Figure 3.1: Experimental setup of Hounsfield Tensometer -----	15
Figure 3.2: Aluminium dog-bone specimen (Units: mm) -----	15
Figure 3.3: Zoomed image of targets -----	16
Figure 3.4: Clamped specimen-----	16
Figure 3.5: Effect of over illumination (a) image with camera flash (b) processed image with IPT	17
Figure 3.6: Checkerboard pattern for calibration -----	18
Figure 3.7: Experimental setup of camera calibration -----	19
Figure 3.8: Region of Interest (ROI)-----	19
Figure 3.9: Checkerboard pattern oriented in different planes -----	20
Figure 3.10: Flowchart of camera calibration -----	21
Figure 3.11: Possible radial distortion types-----	22
Figure 3.12: Tangential Distotion -----	23
Figure 3.13: Detected checkerboard square corner points (green circle) and reprojected points (red cross)-----	23
Figure 3.14: Zoomed view of a corner of a checkerboard square-----	24
Figure 3.15: Scatter plot of reprojection error of image 7 and image 30 -----	24
Figure 3.16: Mean reprojection error of calibration images -----	25
Figure 3.17: 3D view of the camera centric view-----	25
Figure 3.18: Extrinsic parameters visualization from pattern centric view-----	26
Figure 4.1: Four targets used to define coplaner rigid body -----	29
Figure 4.2: Different orientations of targets -----	29
Figure 4.3: Flow chart for the blob method-----	30
Figure 4.4: Process of target detection:(a) Undistorted image (b) Grayscale image (c) Binary image (d) Pseudo coloured image (e) Labelled image -----	31
Figure 4.5: Comparison of percentage error in measured distances -----	32
Figure 4.6: Region of Interest (ROI) of three-point bending experiment-----	34
Figure 4.7: Experimental setup of bending test -----	34
Figure 4.8: Calibration image samples -----	35
Figure 4.9: Images of initial configuration of three-point bbending setup (a) Original image(b) After undistorting with camera parameters -----	36
Figure 4.10: Projected centre of a target -----	37
Figure 4.11: Selection of appropriate greyscale colour datum range -----	37
Figure 4.12: Comparison of vertical displacements of targets from dial gauge readings and MATLAB readings-----	40
Figure 5.1: Template matching analysis-----	42
Figure 5.2: Template rotated (a) by 10 degrees (b) by 90 degrees -----	43
Figure 5.3: Image of a balloon (A) before stretching (b) after stretching-----	44
Figure 5.4: Detected SURF points -----	46
Figure 5.5: Matched features -----	46

Figure 5.6: Deformed region (Red quadrilateral) -----	46
Figure 5.7: Experimental setup of uniaxial tensile testing -----	47
Figure 5.8: (a) Clamped specimen (b) Extensometer -----	48
Figure 5.9: Load vs. Time curve for a specimen -----	49
Figure 5.10: Flow chart for image processing of uni-axial tensile test -----	50
Figure 5.11: Block diagram for pattern matching -----	55
Figure 5.12: ROI (a) before loading (b) during necking (c) strain contour during necking (d) overlapping of b & c (the strain contour during the necking sample) -----	58
Figure 0.1: ROI (a) before loading (b) during necking -----	XXX
Figure 0.2: ROI (a) before loading (b) during necking -----	XXXII

LIST OF TABLES

Table 4.1: Comparison of measured distance in between the target points -----	32
Table 4.2: Comparison of displacements obtained from five targets -----	38
Table 5.1: Types of detecting surface features -----	44
Table 5.2: Comparison of strain output from extensometer and MATLAB -----	53
Table 5.3: Strain contours of specimen 3 -----	56
Table 5.4: Comparison of strain contours with the same colormap range -----	59
Table 5.5: Mesh sensitivity analysis for the strain contours -----	60

LIST OF ABBREVIATIONS

DIC	Digital Image Correlation
ROI	Region of Interest
DOF	Depth of Field

1 INTRODUCTION

1.1 Background

Once a structure is subjected to any form of loading, it has to deform into a new equilibrium position. In Civil Engineering applications, deformation measurement plays a vital role in deciding the structural performance as well as understanding the material behaviour

The existing deformation measuring techniques can be mainly divided into two categories;

1. Contact based system
2. Non-contact based system

1.1.1 Contact Based Systems

Contact based systems where a measuring gauge is physically attached to the object under investigation is being widely used in Sri Lanka. Two such common examples are dial gauges and strain gauges, shown in Figure 1.1.

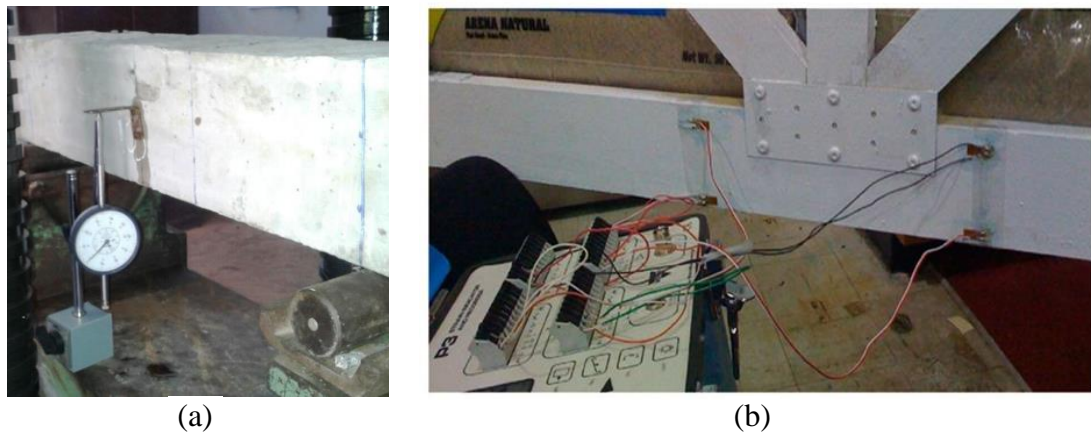


Figure 1.1: Contact-based measurement systems (a) dial gauge (b) strain gauges

In the case of a dial gauge or a displacement transducer a probe is placed to be in touch with the structure and the displacement at the point of contact is measured, Figure 1.1(a). In general dial gauges can measure up to fine displacement increment of 0.01mm. This type of sensors measures the overall deflection of the structure at selected locations which can be used to calculate the averaged strain over a pre-

selected region. In addition, the stiffness of the gauge and the stiffness of the structure under consideration has to be significantly different. Otherwise the deflection of the gauge itself affects the results.

On the other hand, a strain gauge is glued to a selected location (Figure 1.1(b)) and change in resistance due to material stretching is used to calculate the localized strain at a selected point. Strain gauge has to be discarded after a single usage. Though the accuracy is very high cost associated with strain gauges, inability to reuse and time-consuming process of preparation of the surface of the structure are the main drawbacks.

Applicability of contact based systems is limited by the comparative stiffness of the measuring probe and the structure under investigation. Furthermore, these systems are capable of measuring the localized deformation and hence a large number of probes has to be used in order to obtain deformation variation over a surface (Waterfall & McCormick, 2014).

1.1.2 Non-contact Based System

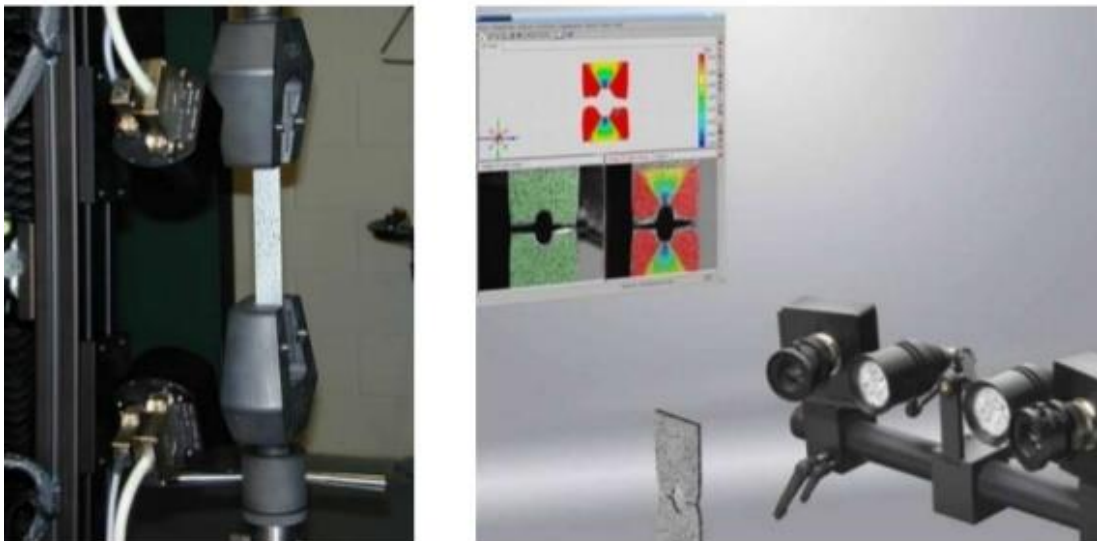


Figure 1.2: Observing optical flow in an object using DIC

Recent developments in optics and computer vision in past few decades have allowed non-contact based measuring techniques to overcome above mentioned drawbacks. In these systems, the measuring probe and the structure will not be having any physical contact. This system is mainly based on the video or images and electromagnetic wave techniques to measure deformations.

Optics based systems which involve taking photographs with camera and analysing the photographs with a image processing software. Digital Image Correlation (DIC) is one such solution where a series of digital images are analysed to track the movement of predefined targets in a structure. Here the difference between the images taken before and after certain deformation is compared with a known scale (Hild & Roux, 2006) which would result in a full-field deformation analysis as shown in Figure 1.2. These measuring systems are already available in the market but the initial cost associated with acquiring a such system is over \$ 30,000 (Lord & McCormick, 2012). The algorithm for the developed technique and the optimisation techniques in this research was unavailable in the online sources since the available systems are used only for commercial purpose. The developed technique was done with only MATLAB coding by the researcher. Since this technique costs only a digital camera and the license of the MATLAB software, the cost associated with the technique is comparatively very low.

1.2 Objectives of the Research Study

A developing country like Sri Lanka has a difficulty in acquiring such systems. Hence the main objective of this research is to develop a low cost non-contact based displacement measuring technique which can be easily adopted in local context by using a commonly available digital camera and an image processing algorithm based on MATLAB. The developed method was intended to measure the full-field deformation.

1.3 Outline

Following the introduction, Chapter 2 provides a detailed literature review on available techniques. Chapter 3 describes the important factors affecting this technique and devising the physical setup. Chapter 4 describes the identification of predefined targets. Chapter 5 compares the results of full-field deformation and Chapter 6 concludes the thesis.

2 LITERATURE REVIEW

This chapter focuses on the research related to DIC technique. DIC technique was combined with the computer vision to produce more innovative solutions for various types of optical measurements. Hence the main aspects of DIC technique is explained in detail. Further some of the applications that use this technique are explained in brief. Finally, the camera calibration and the correlation methods that is used to generate a common coordinate system to be used for measuring deformations are discussed

2.1 Digital Image Correlation

This technique allows to study deformations of a structure under applied loading conditions. Images captured with a camera at different loading steps are compared with a reference image with a known scale to measure the relative deformation of a structure including shape and magnitude. This method can be used for measuring full-field deformation with high accuracy. DIC investigates the optical flow in an object by analysing relative changes in two or more images as shown in Figure 2.1. (Lecompte et al., 2006)

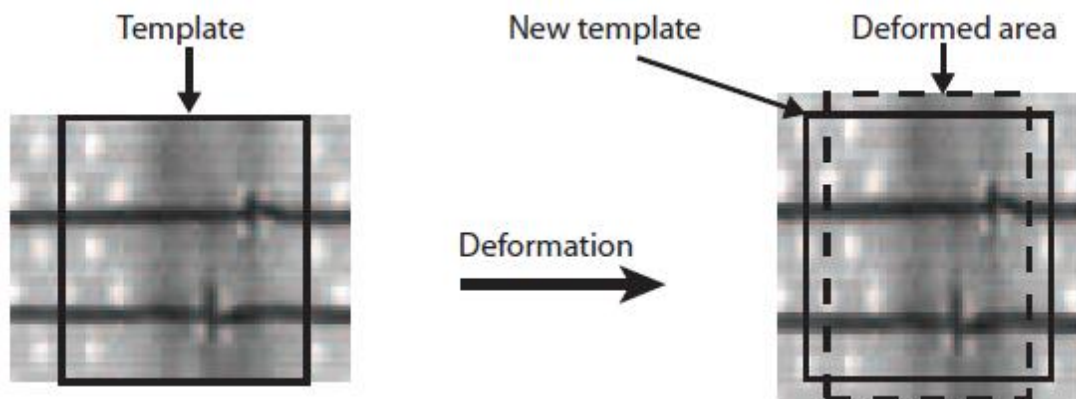


Figure 2.1: Optical analysisi of an object for deformation

During the process, local effects on the displacement field are captured by analysing the variation of optical flow. Hence a correlation algorithm has to be designed to achieve a higher accuracy at a low computational cost. Feron (2008) presented that deformation measurement with DIC technique into following steps,

- I. Selection of Experimental Setup
- II. Digital Imaging
- III. Pattern Matching
- IV. Pattern Recognition
- V. Analysis

2.1.1 Selection of Experimental Setup

Feron (2008) examined different experimental arrangements to find the surface strain of a structure. He showed that the result depends on various factors during the experiment such as the external lighting condition and preparing the specimen sample with appropriate targets and how the targets are going to be detected from the image. Depending on the application it has to be decided that how the experiment has to be set up and how the DIC algorithm has to be modified for each application. Therefore, the experiment and the following image processing steps should be well planned. It was proposed to identify the deformation of electronic textiles of the material during testing process in order to measure the surface deformation. A typical textile of a material is shown in Figure 2.2.

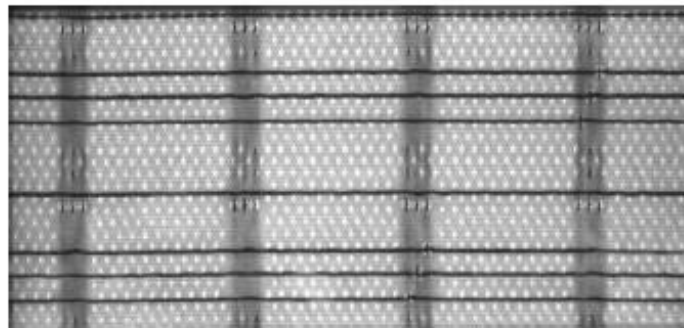


Figure 2.2: Typical electronic textile of a material

2.1.2 Digital Imaging

This stage is about capturing the regions of interest (ROI) with a camera. ROI is the region where our ultimate interest in finding out the surface deformation. Hence the camera should be located to capture the the whole ROI during the experiments. In this stage, the important parameters are (shown in Figure 2.3):

1. Resolution: This is composed of number of pixels in rows and columns of the captured image. If there are A number of rows and B number of columns it has a spatial resolution of $A \times B$.

2. Field of view: The area of the object captured by the camera. In order to utilize the pixels field of view should cover only the ROI.
3. Working distance: The distance in between the camera lens and the object.
4. Depth of field (DOF): Maximum depth that has been focused to the object
5. Camera sensor

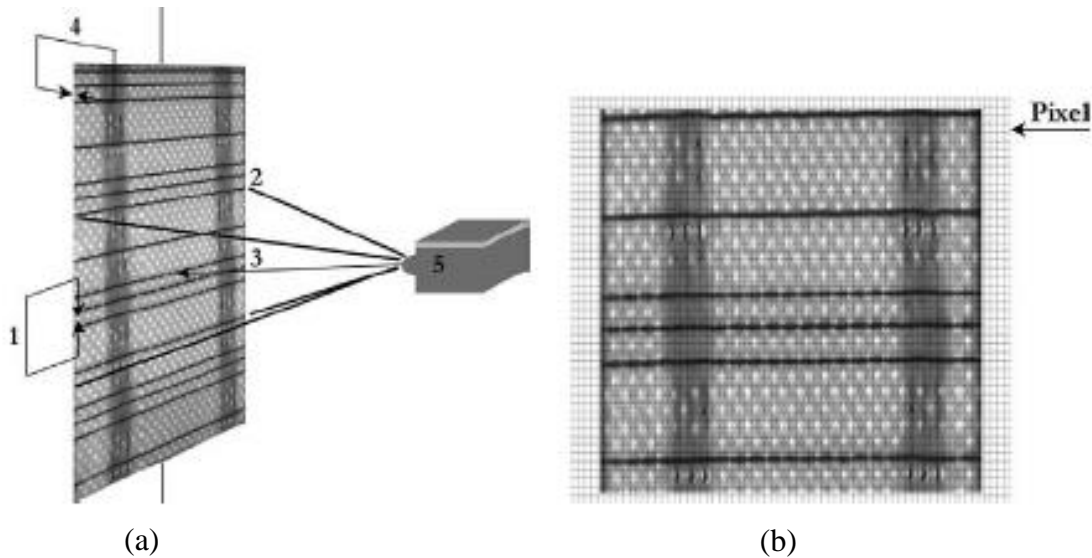


Figure 2.3: (a) Important parameters in digital imaging (b) pixel arrangement

After capturing image each pixel value will be represented by the corresponding image array. When the digital imaging is taking place, a histogram for the corresponding image can be produced (Figure 2.4). It will show how the total number of pixels at each each grayscale level. In order to identify the features from the background, eliminate the noises and produce a preferable output, two important parameters should be considered which are saturation (Light concentration experienced by the image sensor very high or very low) and contrast (value variation in greyscale for different features). But to optimize the results, there should be some adjustments made during the post-processing stage.

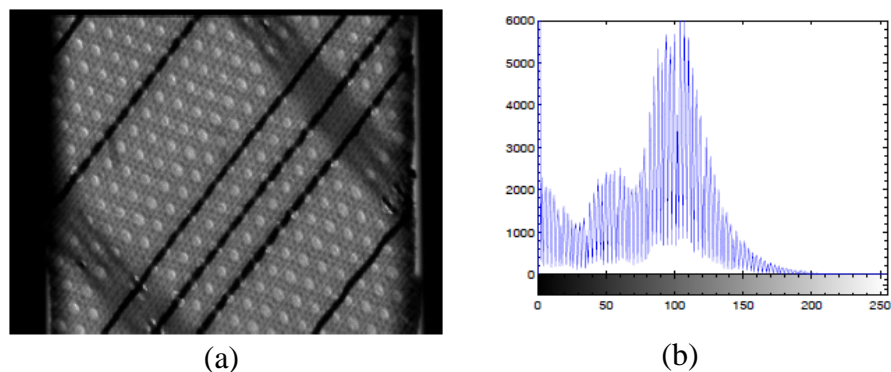


Figure 2.4: Histogram of an image (a) sample image (b) corresponding image histogram

2.1.3 Pattern Matching

This is a technique which helps to identify the region on a greyscale image with a specific pattern or template. The greyscale image will be helpful to analyse with less time consuming by making the image as a traceable mesh. But the result can be roused by considering certain other factors such as geometric transformation in the texture, lighting condition, position of the camera, etc.

The basic concept of the DIC can be expressed as follows during the transformation of arbitrary patterns, correlation between w & f at a point (i,j) is

$$C(i, j) = \sum_{x=0}^{L-1} \sum_{y=0}^{K-1} w(x, y) \cdot f(x + i, y + j)$$

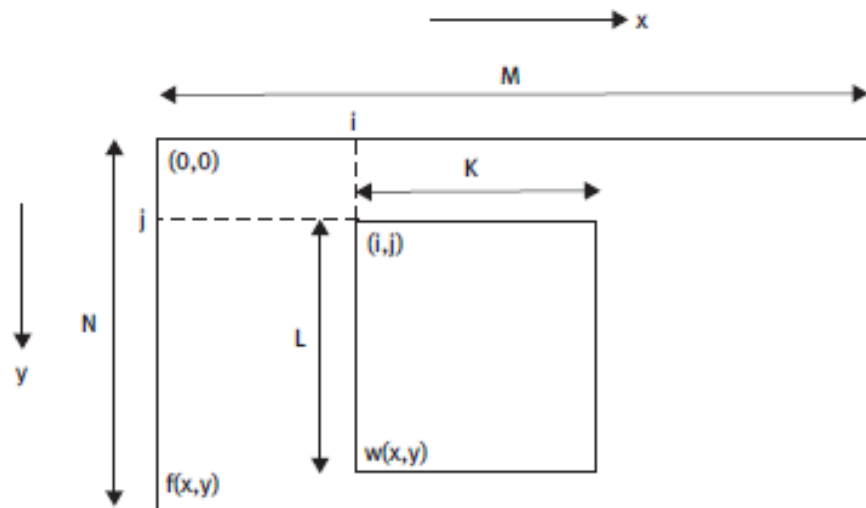


Figure 2.5: Pattern matching correlation parameters

where the C is the correlation coefficient by considering a sub image $w(x,y)$ with the spatial resolution of $K \times L$ in the main image $f(x,y)$ and the dimensions of the images are mentioned above in Figure 2.5. During the pattern matching process, some facts causes insensitivity in image properties which are,

- Texture of the image
- Position of the texture
- Orientation of the texture
- Repetitive textures

In order to track the sub image on the consecutive greyscale images, correlation coefficient value should be larger than a match score (defined by the user) which is known as the “Initial Minimum Score”. The researcher found that uniqueness of

features with irregular high contrast would produce best results, but since there is only one capturing device there is no choice for images but implementing the technique would be a better solution. Some of the disadvantages are low accuracy due to repetitive templates or non-uniqueness and high resolution leading to high data consumption and longer analysis time.

2.1.4 Pattern Recognition

- This technique tracks only certain characteristic features in a greyscale image array. It would track some distinguished features in a manually fed arbitrary region of interest (ROI). In order to improve the efficiency of the tracking, morphological functions (by considering the intensity of the nearby pixels transforming a pixel) can be used during analysis. But pattern recognition becomes ineffective when there is a dense displacement field.
- Pattern recognition provided a solution when there is a need to track particles in a structure that is subjected to large deformations where particles moving outside the ROI. This is done by recognizing special features with moving ROI and relating them by using a global co-ordinate system and local co-ordinate systems.

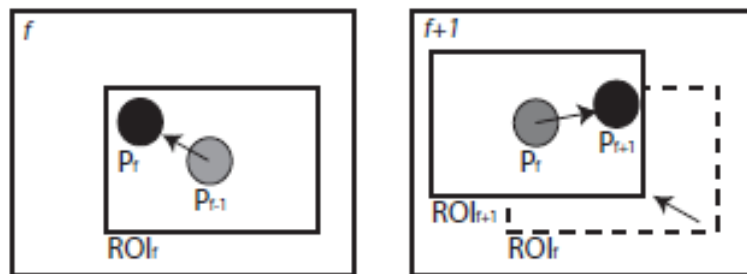


Figure 2.6: Particle movement inside ROI and ROI movement

This accuracy could be increased to sub-pixel level since it analyses regions separately. But again, when the special feature extended for a few pixels the accuracy of the method would decrease. Further than this, there will be ineffective results when there is out of plane deformation and it can be eliminated by using two cameras which would analyse out of plane deformation also. By using a camera with high resolution and improving the material texture in a more unique way would produce much more accurate results.

It was prescribed to use relearn algorithm in case of large deformations, since it would remove the original template and a new relearned template would be created by using morphological function pixels.

2.2 Applications

Full-field deformation becomes a main consideration to identify the crack propagation and settlement on the building in structural engineering applications which is fairly expensive. Since the technology available in the market is having high cost and provide the instructions how the equipment had to be installed to take continuous measurement with the help of the deformation sensors. As Feron (2008) addressed, here the large structure was divided into small parts and analysed. In order to analyse one plane deformation (2D) one camera is used and for both in and out plane deformations two cameras are used and the images were captured simultaneously.

There are some advanced techniques which can measure the 3D strain deformations very accurately and produce three dimensional holographic images which are very useful in the field such as DESPI (Digital Electronic Speckle Pattern Interferometry) and STI (Site Test Interferometer). One such example of 3D system (Q-300) using DESPI technique is made by “Dantec Dynamics GmbH” as shown in Figure 2. It measures the strain variation of a dog-bone metal specimen with a centre hole. But the affordability of such systems becomes questionable due to high capital and maintenance cost, since they are very high. (Løkberg, 1987)



Figure 2.7: 3D ESPI Ssystem (Q-300)

However, it should be noted that accuracy of results can deteriorate due to low image contrast, material type and finish, and low lighting condition. UK's NPL

(National Physical Laboratory) developed an advanced software to investigate this technique which uses initial integer calculations of approximate deformation followed by sub-pixel interpolation using Newton Raphson's method. This method can be used to identify crack openings which are not visible for naked eyes. However, this technique cannot identify internal crack formation since this is a surface deformation measuring system. (McCormick and Lord, 2010)

When the case study was carried forward to calculate deflection in bridges under different type of loadings, some of the strains were not seemed correct (Figure 2.8 (a)). As due to the fact that image sensor plane and the bridge were not parallel since the field of view was larger i.e. there would be a change in scale in wide angle images. This could be eliminated by including the algorithm which contains the angle of images taken. When another case study on vibration of pipework was conducted (Figure 2.8 (b))it was observed that shiny thermal insulation covering produced local contrast which was insufficient. In some other cases noises produced by the background caused pattern recognition a bit difficult (McCormick and Lord, 2010).

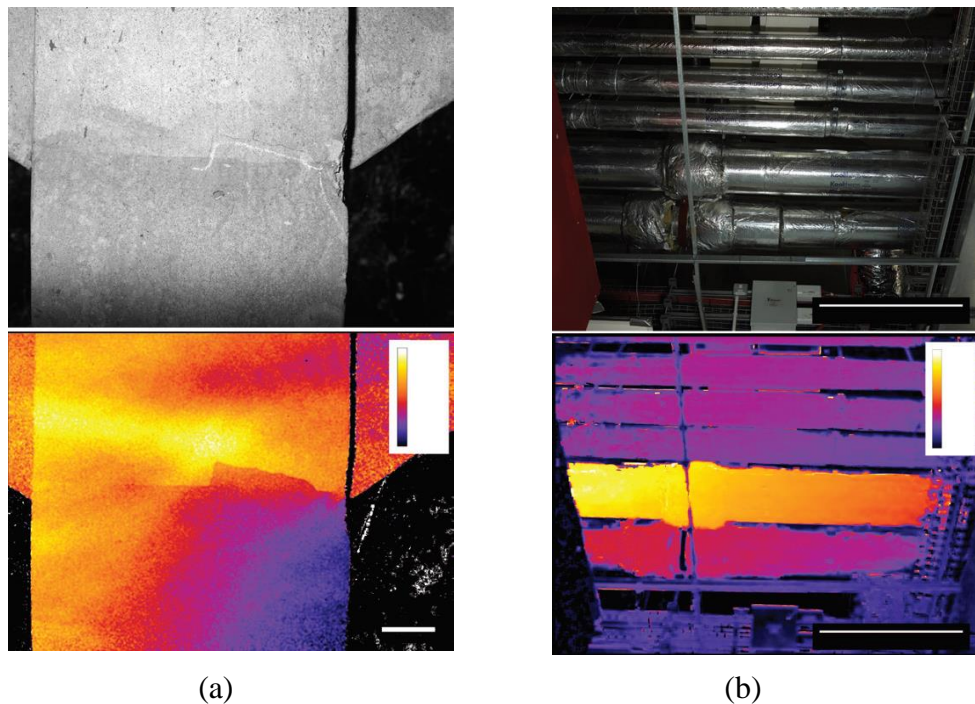


Figure 2.8: Image analysis (a) strain of bridge pier (b) vibration of pipes (red)

McCormick and Lord (2010) addressed a major challenge on measuring long term deformations. Here the issue is camera repositioning after a long period. Since the orientation and position of camera should be kept same for DIC technique.

Kinematic mount becomes a solution for this problem which would allow the users to position the camera with a proper alignment. In addition to that they introduced a system called “Imetrum” which is a video DIC technique which would produce real time strain measurements by capturing images in video format with high resolution. They predicted that the future improvements should be made to create a panoramic displacement field which was created by stitching small images together to analyse large structures.

Even though there are some optical based displacement measuring techniques available commercially, the cost became questionable. In order to investigate this problem and find a solution, a research was done by McCormick and Waterfall (2013). In short-term and long-term monitoring of civil engineering structure deformations has to be measured with a high precision, low cost and at a large scale. This DIC measurements can be taken even if there is no access or dangerous access to the object to be measured. Commercially Imetrum system is widely used which allows the user to select the ROI, taking minimum setting-up time, and producing real time and offline analysis measurements such as strain, rotation, and displacement.

Before analysing the displacement field, anomalous movement which are affecting the direct sight to the object such cob webs or insects in the intermediate region or out-of-plane deformation. It was prescribed to use a high frequency, image capturing devices when there is a highly deforming structure. In order to avoid refraction errors in measurements due to heat gases it was proposed to use laser beam projections. When there is lighting variation like moving shadows, the initial minimum matching score should be decided. Some of the approaches to be followed when the image capturing taking place cameras positioning in a vibration isolated zone, use of carbon fibre made cameras which are resilient to thermal expansion in daytime and reorientation of the cameras should be done with high precision inclinometers. (McCormick & Waterfall, 2013)

Another research was done on strain distribution analysis by Tung et al. (2005). Researcher criticized two traditional methods which were used to find strain. 1) Average strain was calculated by considering the relative displacement in between two points where the surface strain couldn't be obtained. 2) A grid system was introduced to calculate the strain by comparing the initial and deformed images. But the analysis was very time consuming. As expressed earlier DIC was then used to calculate the

strain by creating a structural fleck tracking. By comparing the greyscale images before and after the deformation displacement vector, and strain in the plane, shear strain and Von Mises strain could be calculated. Comparison of the theoretical value and the precision of the software was almost same. In actual case strain distribution would not be homogenous but heterogeneous as predicted and visualized. (Tung et al., 2005)

DIC was used in another research containing a novel integrated approach coupling image correlation to produce two modes of stress intensity factors. A silican carbide plate was subjected to 3-point bend load test and the crack formation (size and geometry) was tried to be detected. Since it's really hard to find the properties of brittle materials such as remaining lifetime or ultimate load to be applied for failure using this DIC technique helped to produce accurate results. In brittle materials, it is difficulty to identify the initial crack tip location, so that producing a stress intensity factors and toughness becomes a challenge. Since the modern techniques allow to measure at sub-pixel level crack was measured up to hundreds of nanometres which is not visible for bare eyes. The benefits compared to the other methods are no special sample preparation is required and no sophisticated equipment is needed. The advancement in the conventional microscope effect on camera led the experiment need not to be done by fixing the camera near to the required material. Even if there is no accessibility to the material or structure this high-resolution camera still provides the required accuracy to measure the strains. Therefore, this method was so helpful to identify the mechanical properties of the elastic materials and could be extended for complex scenarios also with some assumptions. (Hild & Roux, 2005)

2.3 Camera Calibration

Camera calibration (or camera resectioning) is the process which is used to determine the intrinsic (optical and internal camera geometric characteristics) and extrinsic parameters (orientation of the camera in a 3D space relative to the world coordinate system) (Ali, Logeswaran & Bister, 2004) and the lens distortion of the camera (Tsai, 1987). By finding out the values of these parameters, the distortions of an image can be eliminated or reduced and dimensions of objects can be measured, and the 3D structure of an object can be reconstructed. After evaluating the parameters, the calibration accuracy can be improved by selecting appropriate

calibration images, removing the images with high distortion and recalibrate the good images with less distortion.

2.3.1 Calibration Algorithm:

Calibration algorithm of a camera can be defined as (“Single Camera Calibration App - MATLAB & Simulink,” n.d.)

$$w \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = [X \quad Y \quad Z \quad 1] \begin{bmatrix} R \\ t \end{bmatrix} K$$

where (X,Y,Z): are world coordinates of a point, (x,y): coordinates of the corresponding point in the image, w: arbitrary homogeneous coordinates scale factor and K: camera intrinsic matrix.

Intrinsic parameter consists of 5 parameters and is represented by a 3x3 matrix. This can be defined as:

$$K = \begin{bmatrix} f_x & 0 & 0 \\ s & f_y & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

where (cx, cy): principal point of the image (i.e. centre of the image), x & y are perpendicular axis, c is the coefficient of skewness, and (fx, fy): focal length in pixel terms. From this the focal length of the camera, image centre coordinate, coefficients of radial and tangential distortion and aspect ratio can be estimated.

Position of the camera C can be expressed by the following equation.

$$C = -R^{-1}T = -R^T T$$

where R, t are the extrinsic parameters which denote the rotation matrix and translation vector respectively. T is the origin of the world coordinate system. Generally T is expressed in terms of camera centered coordinate system.

MATLAB software has image processing toolbox which consists of both single and stereo camera calibrator. By default, the calibration pattern is the checkerboard pattern and this can be changed by rewriting the calibration code.

2.4 Correlation Methods:

When evaluating the surface displacement field, an appropriate correlation method should be used. There are mainly 3 correlation methods. All of them comes under the category of sum of squared differences (SSD). (Riccio et al., 2017)

- Sum of squared differences criterion (SSD): This is not very robust, since this method is sensitive to variation in brightness. And this criterion is the fastest out of three methods.
- Normalized sum of squared difference criterion (NSSD): This is more robust than SSD since this is not sensitive to change in brightness scale.
- Zero-normalized sum of squared difference criterion (ZNSSD): This is the most robust method out of three since this is not sensitive to change in brightness and brightness scale. But it is time consuming.

3 DEVISING PHYSICAL SETUP AND CAMERA CALIBRATION

In devising an optics based measuring technique it is important to eliminate external disturbances like contrast, background noise as much as possible. This chapters discusses the external factors to be taken into consideration when using DIC technique. Second half of the chapter describes the distortions due to camera and its position, and use of image calibration for correction.

3.1 Identifying External Factors to be Considered

In order to identify the external factors a uniaxial tensile test with a Hounsfield Tensometer (Type 6645, Tensometer Limited, England). Tensile force was applied to an Aluminium dog bone shape specimen as indicated in Figure 3.2. The shape was specifically chosen to prevent failure occurring within the clamped region (Figure 3.1). This allows the elongation to be visible without any obstacles during the test.

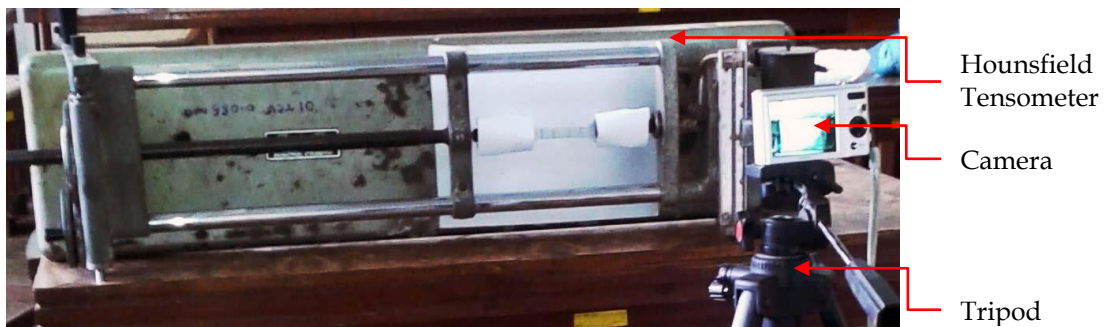


Figure 3.1: Experimental setup of Hounsfield Tensometer

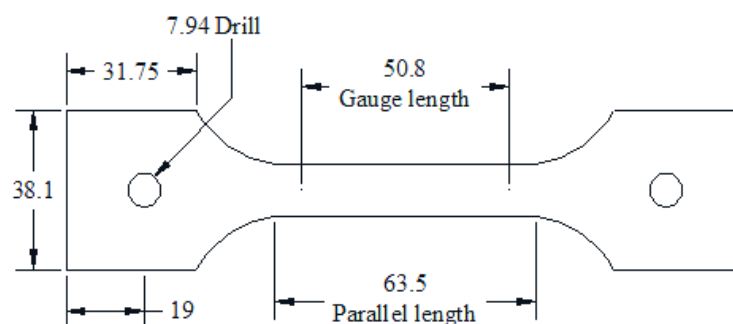


Figure 3.2: Aluminium dog-bone specimen (Units: mm)

The entire specimen was painted with white colour first and then divided into three regions with four sharp dark black lines in order to have a set of contrasting targets (points of interest) as shown in Figure 3.3

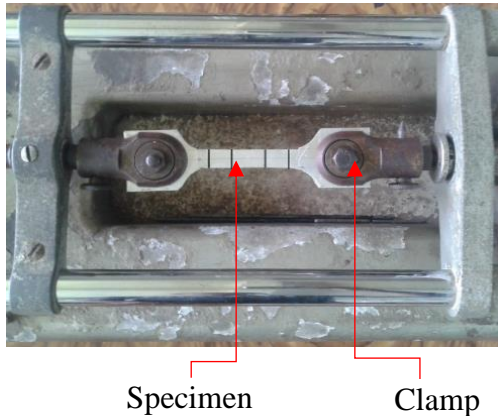


Figure 3.4: Clamped specimen

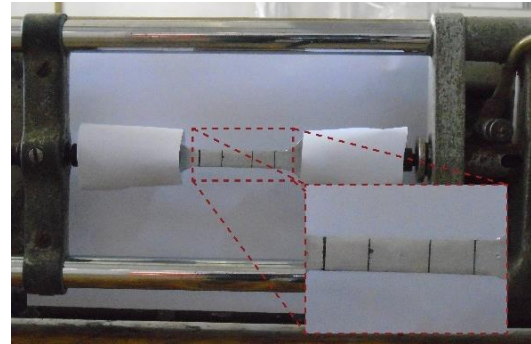


Figure 3.3: Zoomed image of targets

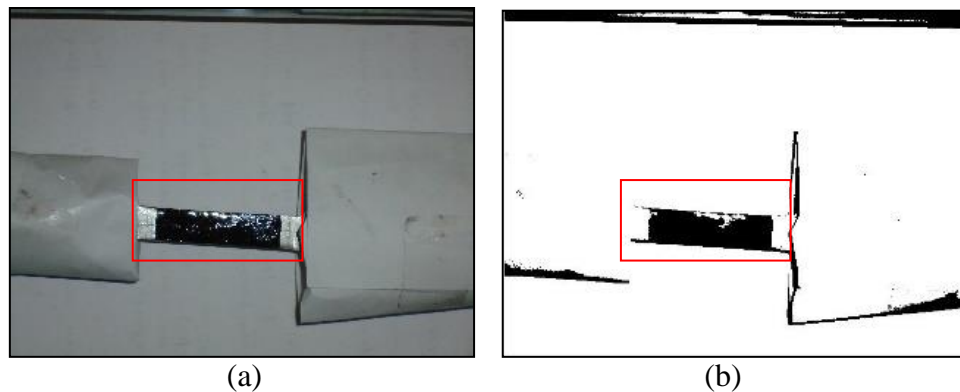
3.1.1 Contrasting Background

Disturbances in the background make it difficult to clearly identify the intended targets and hence reduce accuracy during image processing stage. Therefore, it is important to have a contrasting background to avoid as much as possible. In this case, a piece of white colour paper was placed behind the specimen and the clamps were covered with white paper as in Figure 3.4. If the disturbances cannot be avoided during the experimental stage they can be reduced by using the noise reduction algorithms during the image processing stage.

3.1.2 Lighting and Shades

Adequate lighting condition is important to illuminate the targets and get rid of unnecessary shadows to have better results in the image processing stage. LED lighting systems provide better lighting condition over ambient light to eliminate shadows (Haddadi & Belhabib, 2008). Over illumination of an object leads to distortions around boundaries. Common causes of over illumination can be due to higher light flux created by the lighting system, environmental light sources, camera flash or having specimen with reflective surfaces. Hence the lighting condition of the object factors

has to be checked before conducting the measurements. Figure 3.5 illustrates how the camera flash and the shadows affect the images during the image processing stage.



**Figure 3.5: Effect of over illumination (a) image with camera flash
(b) processed image with IPT**

3.1.3 Camera Position

The precision of this technique is governed by the number of pixels available within the region of interest (ROI). Hence it is important to focus on the ROI (Figure 3.4) and position the camera as close as possible to the experimental setup. While keeping the camera closer makes the precision higher, it makes the field of view (a solid angle which can be observed with the camera) shorter. In addition, the angle of incidence of further most positions within the ROI becomes larger and hence the scale across ROI varies at a higher rate (Murray, Take, Hoult & Hoag, 2015). Therefore, the camera was positioned at the furthestmost point from the apparatus while maintaining ROI to fit into the field of view of the camera. By doing this, full resolution of the camera can be utilised.

3.1.4 Thickness Effect of the Targets

Selection of targets and marking them plays an important part in this technique. It is better to mark the targets using a thin outline since the thickness of the marker itself lead to reduce the pointing accuracy. Further this makes the validation process with basic measuring techniques harder.

3.2 Camera Calibration

It is well known that any measuring system first needs to be calibrated to obtain accurate measurements. In the case of DIC, a predefined image is used to calibrate the system. This process is known as “Image calibration” where several calibration images are taken with different orientations and processed to estimate the camera parameters like camera position, focal length, scale and resolution. Therefore, it is important to select suitable images for calibration.

A checkerboard square pattern was selected as the calibration pattern. MATLAB algorithm (“Single Camera Calibration App - MATLAB & Simulink,” n.d.) provides the viability of using the checkerboard patterns to get the scale over different regions in the same plane. Calibration checker squares should have odd number of squares in one direction and even number of squares in the other direction for algorithm to differentiate length and width. Size of the calibration grid pattern should cover the whole area of the ROI. Hence the size of it will vary depending on the application. Figure 3.6 shows the checkerboard pattern used for this experiment where each square has the dimensions of 21.51 mm \times 21.51 mm. Printed checkerboard pattern was pasted on a 15 mm thick plywood plate (calibration frame) to make sure the checkerboard pattern remains in a single plane.

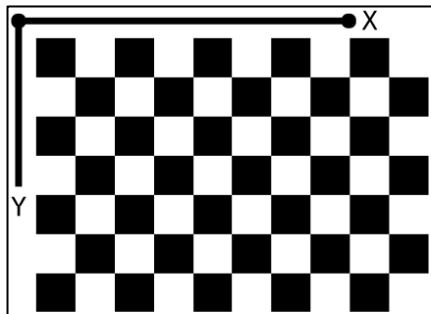


Figure 3.6: Checkerboard pattern for calibration

In general calibration process requires at least 10 images taken at different orientations. Capturing more than 20 images is recommended to have a better accuracy and some images may have to be discarded due to distortion or blur arising due to external factors. These distortions can result from following causes:

- Reflections of target surface due to excessive ambient light or external lighting systems and camera flash
- Camera shaking while pressing the trigger
- Improper focus of target

3.2.1 Experimental Setup



Figure 3.7: Experimental setup of camera calibration

Figure 3.7 shows the physical setup used for capturing the calibration images. A Sony DSC-W800 cybershot digital camera was mounted on a Velbon PH-368 fluid head tripod and placed approximately 540 mm away from the plane of interest.

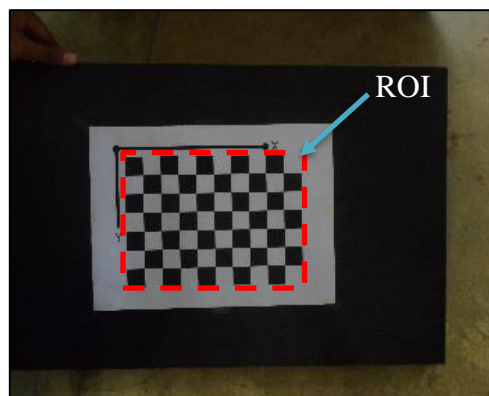


Figure 3.8: Region of Interest (ROI)

Precision of measurement depends on the resolution of the camera and the distance between the camera and the measuring plane. Thus, camera should be zoomed and focused on the ROI where the intended targets to be placed within. The size of the calibration frame should be selected to cover the ROI to correct distortions associated with angle of incidence. Figure 3.8 shows an image of calibration frame. Note that the checkerboard pattern covers the complete ROI.

Once the camera is setup to focus on the ROI the checkerboard pattern was first placed in the intended focal plane and a photograph was taken. Then a series of 35 photographs were carefully taken by orienting the checkerboard in different planes to cover all possible translations, rotations and inclinations within the depth of field of the camera. It should be noted that the plane of calibration pattern should not be kept more than 45 degrees to the camera plane. The camera flash was switched off at all times and external light was barricaded to minimize reflections. Photographs were taken with ambient light since it was sufficient to capture the details. However, in the case of low contrasting background, it is recommended to use non-reflective LED lighting system in order to capture a clarity image by eliminating the shadows. Figure 3.9 shows a selected set of images taken for calibration.



Figure 3.9: Checkerboard pattern oriented in different planes

3.2.2 Processing Calibration Images

After the target images have been captured, MATLAB image Processing Toolbox (IPT) was used to analyse the targets. Figure 3.10 shows a flow chart explains the important steps followed during the camera calibration process.

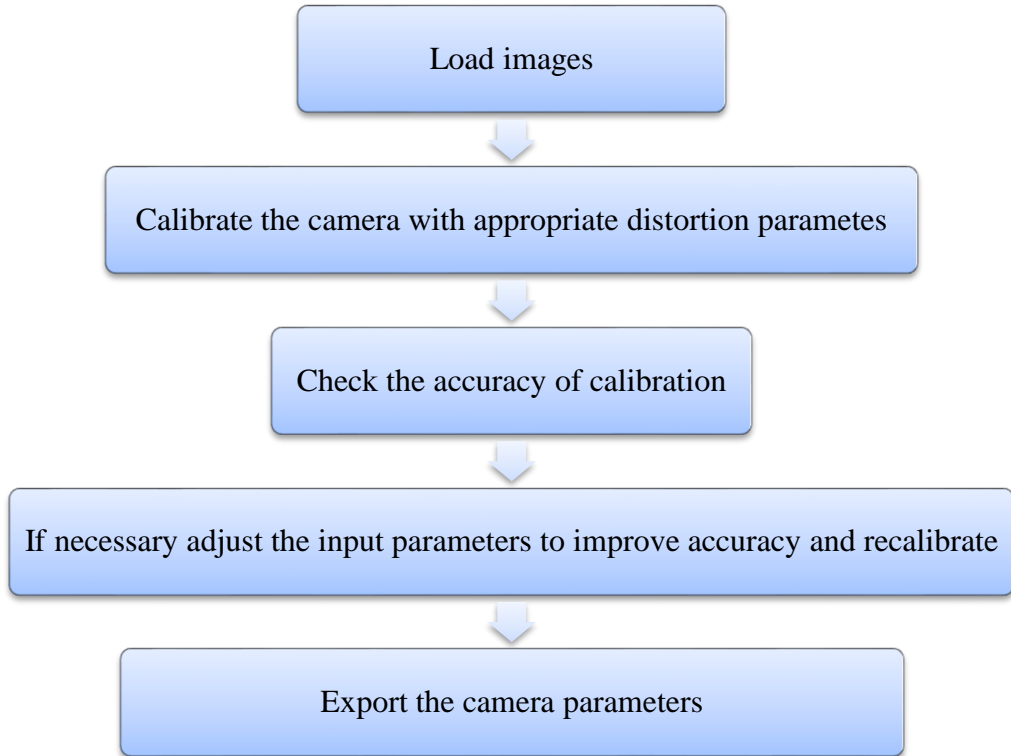


Figure 3.10: Flowchart of camera calibration

The Appendix B.1 provides a sample calibration algorithm. The calibration images captured during experiment were processed using the IPT available in MATLAB. When calibrating the images IPT helps to reject the distorted images due to not being focused well, being outside the depth of field, or reflection, etc. and select appropriate images for calibration. Once the defective images are taken out the remaining were used to estimate the camera parameters. There are three main parameters that contribute to the accuracy of calibration.

Radial distortion Coefficients:

This depends on the camera lens which can be specified by a distortion coefficient having values of 2 or 3. This distortion is caused by the deflection of the light rays at the lens of the camera. For normal calibration process, radial distortion coefficient of 2 would be enough. In case of high distortion which may be caused by wide angle lenses, coefficient of 3 would be preferable to calibrate the images.

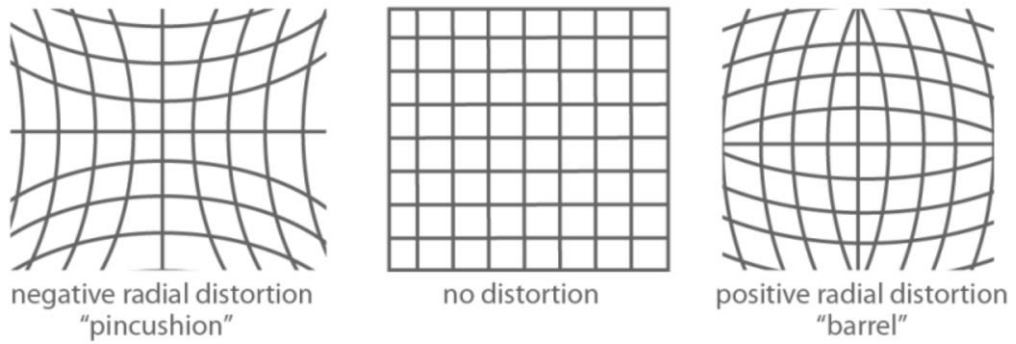


Figure 3.11: Possible radial distortion types

Figure 3.11 shows how a image will be distorted under different kinds of radial distortions. The distorted points are denoted as $(x_{\text{distorted}}, y_{\text{distorted}})$:

$$x_{\text{distorted}} = x(1 + k_1*r^2 + k_2*r^4 + k_3*r^6)$$

$$y_{\text{distorted}} = y(1 + k_1*r^2 + k_2*r^4 + k_3*r^6)$$

x, y : undistorted pixel locations

k_1, k_2 , and k_3 : radial distortion coefficients of the lens

$$r^2 = x^2 + y^2$$

The undistorted pixel locations are in normalized image coordinates, with the origin at the optical centre. The coordinates are expressed in world units (non-default spatial coordinate system).

Skew:

Imperfections in a camera sensor would result in the axis (x & y) of the image not to be considered as perpendicular. Modern cameras have a built-in correction to eliminate this fault.

Tangential Distortion:

This occurs when the image plane and camera lens are not in parallel to each other. This can be explained by Figure 3.12;

The distorted points are denoted as $(x_{\text{distorted}}, y_{\text{distorted}})$:

$$x_{\text{distorted}} = x + [2 * p_1 * y + p_2 * (r^2 + 2 * x^2)]$$

$$y_{\text{distorted}} = y + [p_1 * (r^2 + 2*y^2) + 2 * p_2 * x]$$

p_1 and p_2 : tangential distortion coefficients of the lens

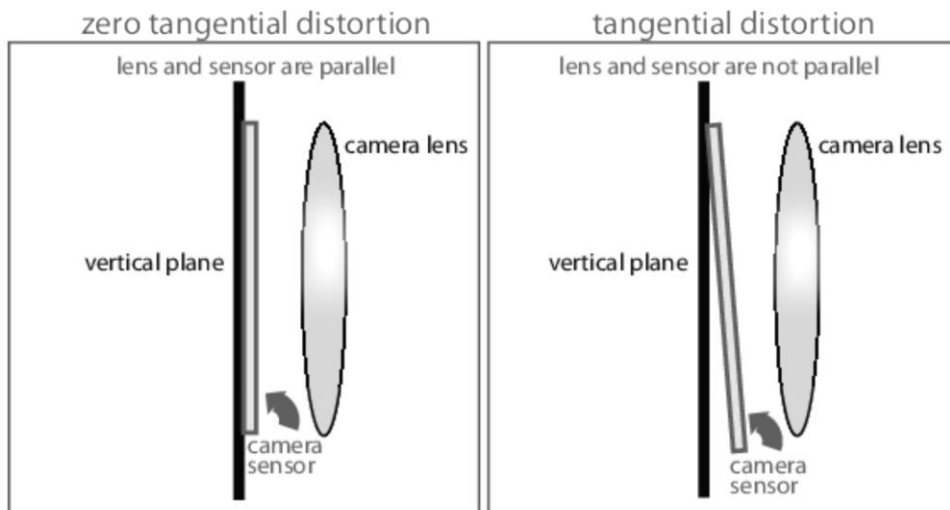


Figure 3.12: Tangential Distortion

When set the “Compute Tangential Distortion” parameter in the MATLAB algorithm true, the calibrator estimates the tangential distortion coefficients. Otherwise, the calibrator sets the tangential distortion coefficients to zero (i.e. ‘false’).

In this experiment only the radial distortion with coefficient of 2 was used. Skew and tangential distortion corrections were not considered as the camera has built-in facilities for such corrections.

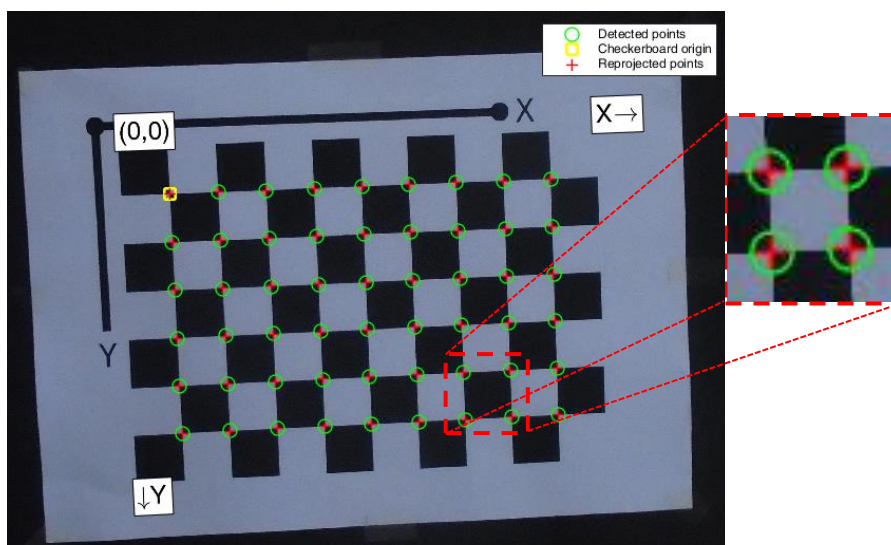


Figure 3.13: Detected checkerboard square corner points (green circle) and reprojected points (red cross)

After making corrections for camera distortion, the calibration images can be processed to estimate the camera parameters. During this process, the software detects the corners of the squares in the checkerboard by analysing the colour difference and hence the plane of the checkerboard. As described earlier imperfections in the captured

images can lead to errors in identifying the plane. This error is calculated by reprojecting the estimated points back into the original image as shown in Figure 3.13.

Difference between detected point and predicted point is known as “reprojection error”. Figure 3.14 shows a zoomed view of a corner of a checkerboard square. Note that the reprojection error is less than a pixel.

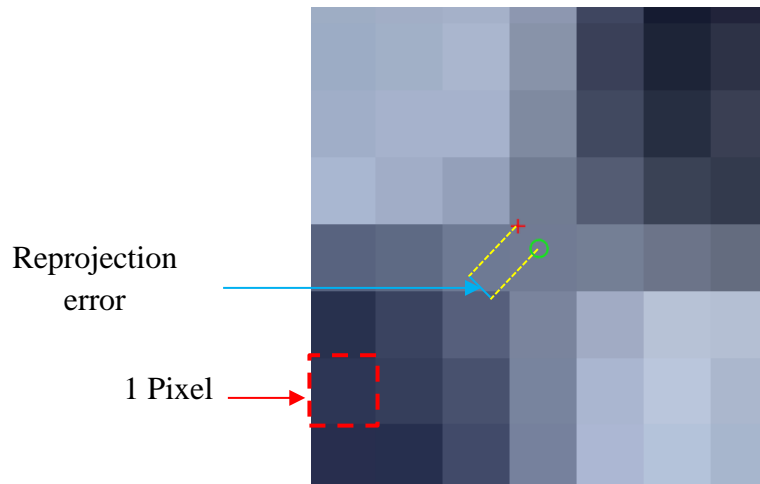


Figure 3.14: Zoomed view of a corner of a checkerboard square

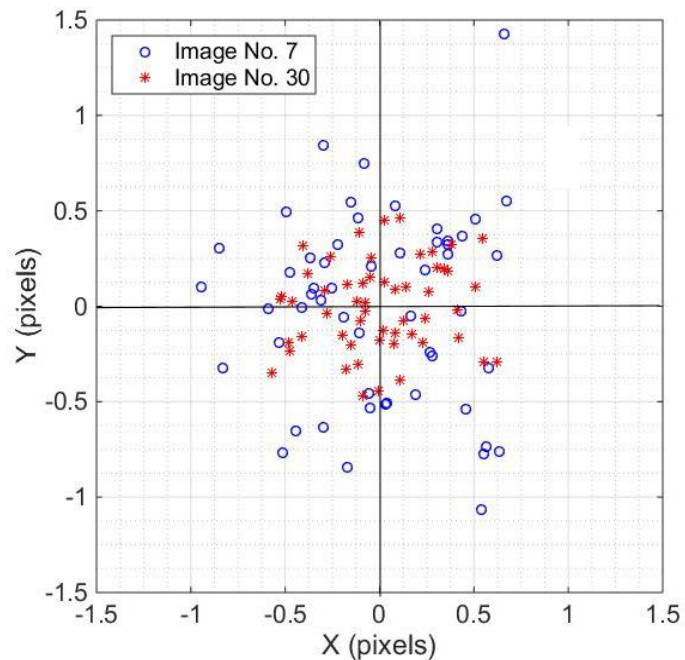


Figure 3.15: Scatter plot of reprojection error of image 7 and image 30

Reprojection error can be used to calculate the accuracy of reprojection and hence useful in identifying suitable calibration images. Figure 3.16 illustrates the mean reprojection error of the 35 calibration images taken in the form of a bar chart. Figure 3.15 shows the scatter associated with detected corner points of images no. 7 and no. 30 corresponding to the highest and lowest mean reprojection errors, respectively. Note that points related to image no. 30 has a low scatter and hence give rise to better results. The mean reprojection error of all 35 calibration images selected was maintained at 0.46 pixels.

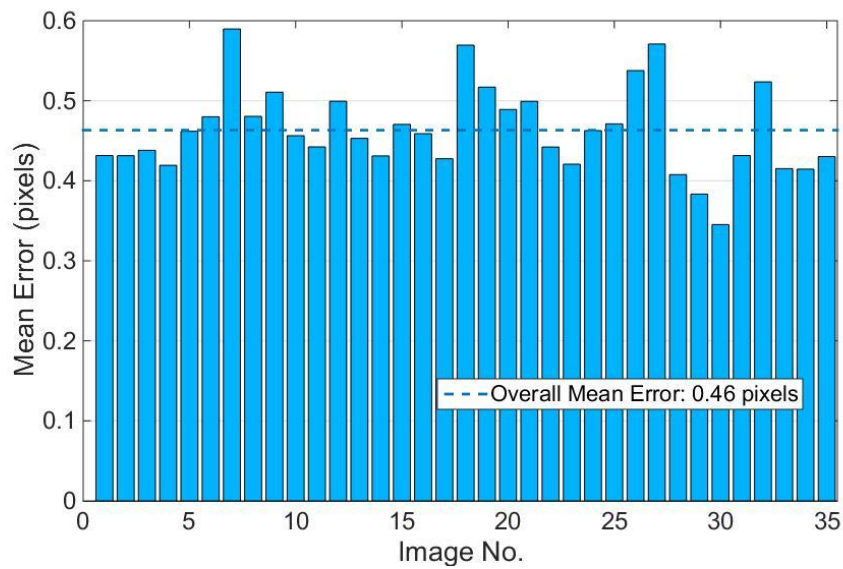


Figure 3.16: Mean reprojection error of calibration images

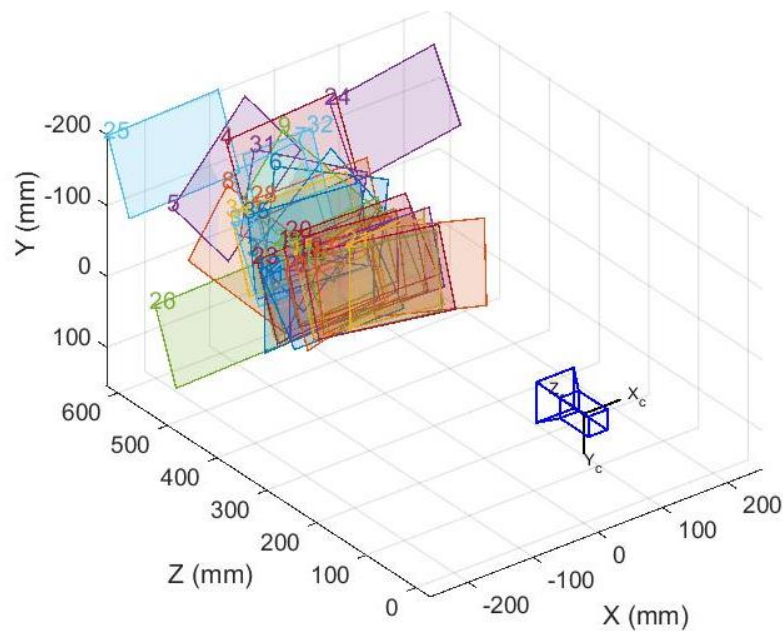


Figure 3.17: 3D view of the camera centric view

As described in Chapter 3.1.3 the camera was fixed to a tripod and the target plane was moved. This corresponds to camera centric view definition where the camera is static. Hence camera-centric view was selected while using MATLAB IPT in calculating camera parameters. Figure 3.17 shows the orientation of the calibration frames.

Camera parameters consists of two components: intrinsic and extrinsic parameters. Intrinsic parameter matrix consists of focal length, image sensor format which constitutes the skew coefficient, principal point of the image, radial distortion and tangential distortion. Extrinsic parameter matrix consists of rotation vector and the translation vector which is used to calculate the coordinates the targets with respect to predefined origin. Apart from the camera parameters the distance to the camera from the object can also be measured. The intrinsic and extrinsic parameters of the camera used in this experiment has been extracted and shown in Appendix B.2.

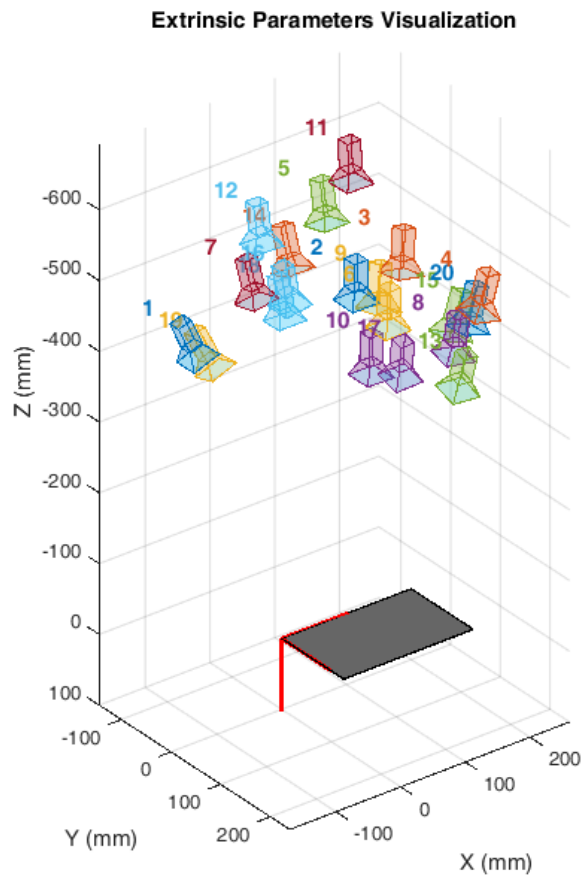


Figure 3.18: Extrinsic parameters visualization from pattern centric view

If the calibration pattern is static and the camera has been relocated to capture the images at different locations each time that method is called as “pattern centric view” which is another way to find out the camera parameters. A sample extrinsic view of the pattern centric view is shown in Figure 3.18.

4 IDENTIFICATION OF PREDEFINED TARGETS

This chapter explores the possibility of tracking movement of set of predefined targets to measure the deformation of a given structure. This procedure consists of two parts where first is to identify the predefined target through image processing and then to verify the deflection values by means of an independent measuring technique. The chapter describes two physical experimental arrangements followed by image processing procedures to track predefined targets. Section 4.1 presents a preliminary measurement on four predefined targets printed on a sheet of paper to verify rigid body motion of a four-sided polygon. Section 4.2 describes how this technique can be adopted to measure vertical deflection of a flexible beam.

4.1 Measuring Rigid Body Motion with Predefined Targets

Accuracy of measurement with predefined targets depends on the method of marking the targets, level of deformation of targets, shape due to flexibility of the structure under consideration, coplanarity of targets, etc. Hence four circles printed on an A4 sheet of paper which was then glued to a flat hardboard was selected for preliminary verification to minimize the effects of these unfavourable conditions. Rigid body motion was selected to eliminate effects due to varying these conditions.

4.1.1 Preparation of Targets

Targets that are used to track the motion of points of interest have to be contrasting from the background in order to locate the exact position. Disturbances in the background make it difficult to identify the exact location of a target leading to reduction in accuracy of the measurements. Hence four dark grey colour circles printed on a white colour paper was used to define the targets as shown in Figure 4.1. Centre of each circle was marked with a small cross in order to facilitate physical measurements for validation as well as check the accuracy in identifying the centre through computer program (Figure 4.1). The distance between the centroids of the targets were measured with a Vernier Calliper having an accuracy of 0.02 mm for validation.

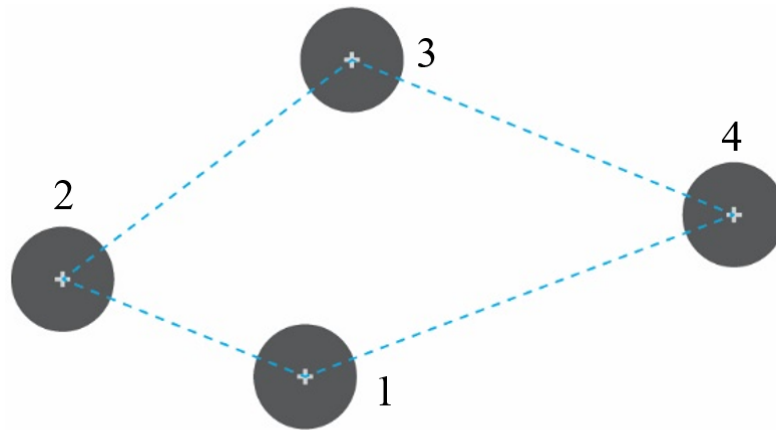


Figure 4.1: Four targets used to define coplaner rigid body

4.1.2 Capturing Images

A calibration grid slightly larger than the hardboard with predefined targets was prepared using the procedure described in Section 3.2 and 35 calibration images were taken covering the intended region of interest after the camera was fixed. Then the rigid plane was kept at the intended region of interest and a photograph was taken. Similarly, another four photographs were taken while changing the orientation of the targets by moving, rotating (can be done for 360 degrees) and tilting (up to 45 degrees) the rigid-plane in order to simulate rigid body motion.

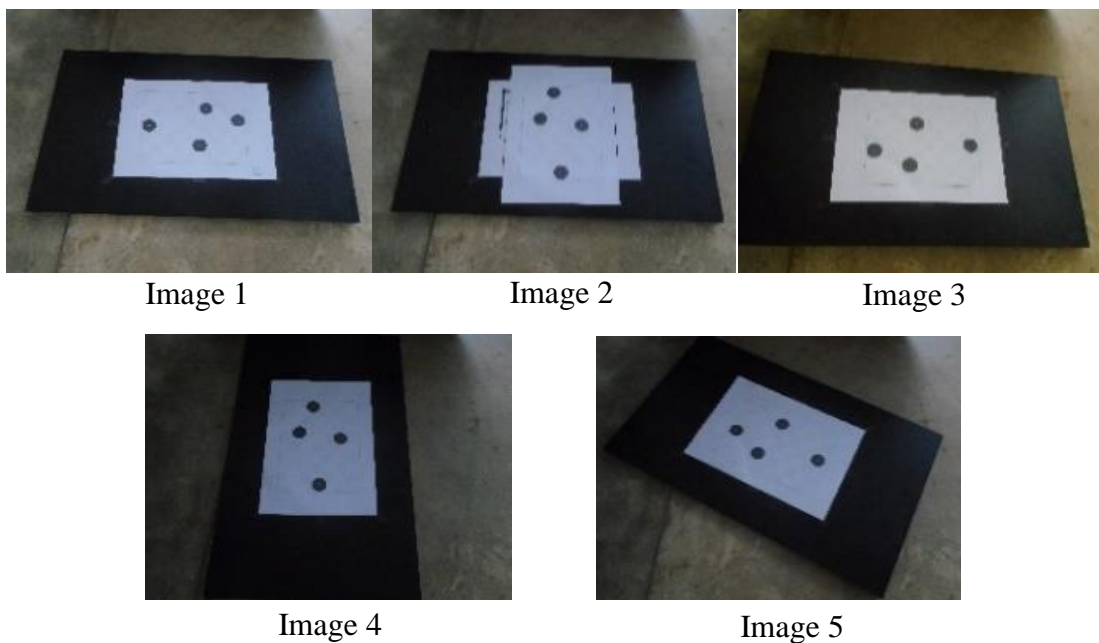


Figure 4.2: Different orientations of targets

4.1.3 MATLAB Algorithm for Identifying Targets

The targets first need to be segmented from the background in order to perform further calculation in estimating deformations. In order to identify and track the targets the “blob method” was used and it is explained in detail in the following passage referring to the flow chart shown in Figure 4.3.

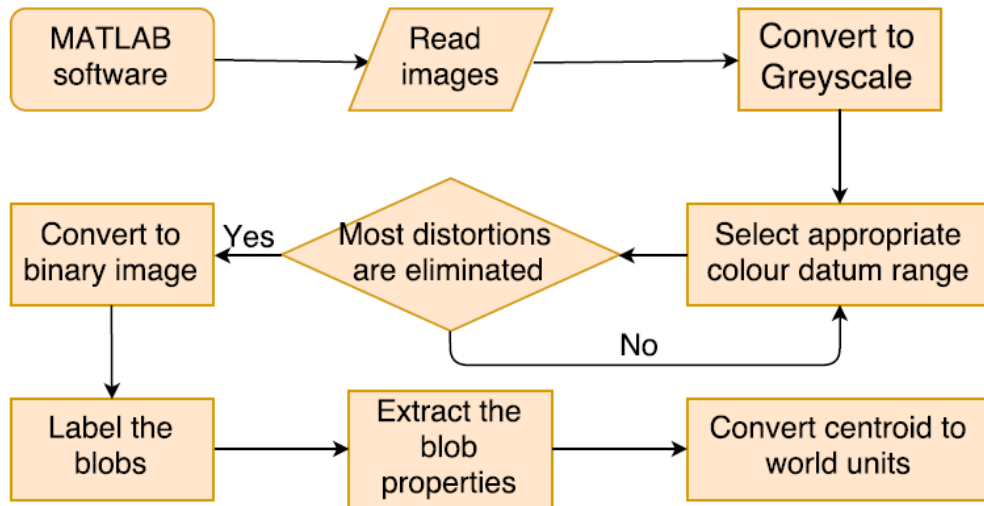


Figure 4.3: Flow chart for the blob method

A target image is first corrected for distortions using the camera parameters described in Chapter 3.2. Then the image is converted to grayscale and a colour datum was selected to remove the distortions. In a grayscale image colour map varies in integer form from 0 (black) to 255 (white). Once a suitable colour datum is selected the darker circles can be identified and converted to a binary image. Histogram of the greyscale image helps in selecting the grayscale colour datum range. This is a trial and error process. Then the surrounding pixels of the identified targets were filled to get a smooth target. Each interconnected pixel group is called a “blob”. Then these blobs were labelled by giving them different pseudo colour for each blob. Figure 4.3 shows images from different stages of image processing process. The algorithm in detecting the targets and measuring the distances is presented in Appendix B.3.

The physical properties of labelled blobs were then obtained using the algorithm developed. The properties include blob centroid, blob area, blob perimeter, blob boundaries and mean colour intensity. These parameters are obtained in terms of pixel. In order to convert this pixel term to standard measuring unit, extrinsic camera parameters were used. Finally, the centroid coordinates were converted to world units. After finding out the centroid of the targets in world units the distance in between the

target circles were calculated for all 5 images. The analysed results were compared with the manual measurements taken with the Vernier Calliper.

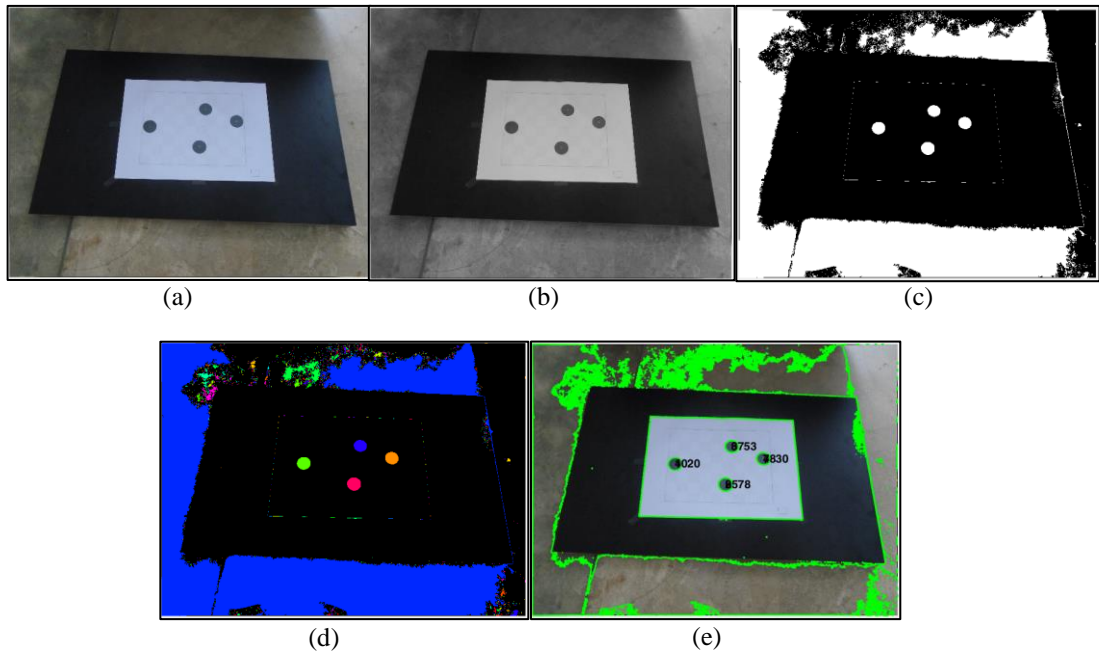


Figure 4.4: Process of target detection:(a) Undistorted image (b) Grayscale image (c) Binary image (d) Pseudo coloured image (e) Labelled image

4.1.4 Analysis and Results of Rigid Body Motion Test

Table 4.1 presents the measured distance between each 2-target point. Note that physical measurements and the calculated values of all 5 images through MATLAB algorithm are presented.

Table 4.1: Comparison of measured distance in between the target points

Target points	Vernier reading (mm)	MATLAB measurement (mm)				
		Image 1	Image 2	Image 3	Image 4	Image 5
1 - 2	66.98	67.26	67.24	67.11	67.02	67.13
2 - 3	92.78	93.02	93.17	93.11	93.12	93.06
3 - 4	106.06	106.19	106.27	105.84	106.01	106.36
1 - 3	82.16	82.01	82.09	82.29	82.07	82.53
1 - 4	117.98	118.21	118.09	117.82	118.31	118.15

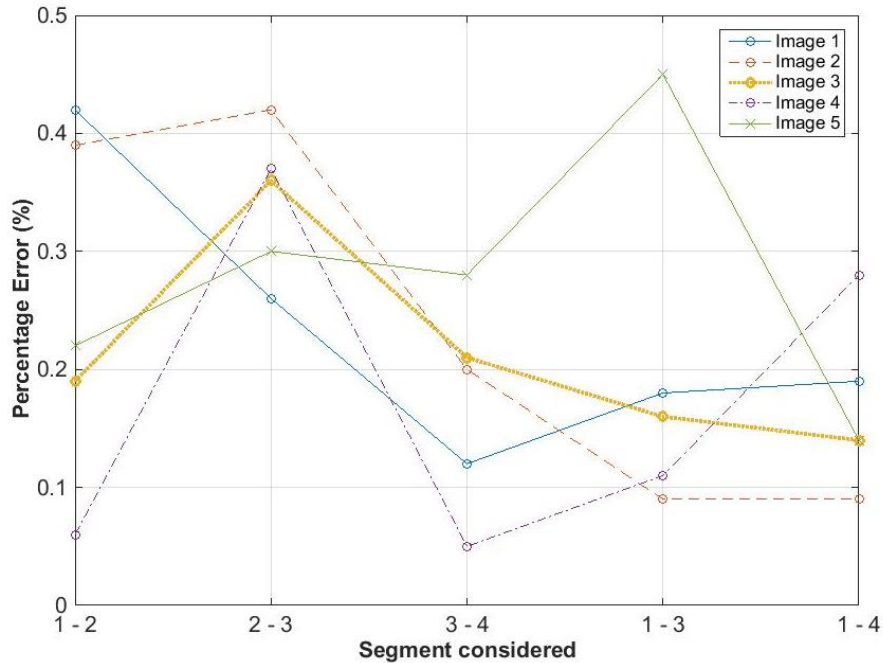


Figure 4.5: Comparison of percentage error in measured distances

It can be noticed that the maximum error of 0.39 mm is in Image 2 between target points 2 and 3. When looking into the pixel spacing, it is approximately 0.21 mm. Hence the error in terms of pixel size is 2. When considering the percentage, the maximum error occurred in Image 5, between points 3 and 1 which is 0.45%.

Therefore, the developed displacement measuring technique has a minimum count of 0.39 mm. Figure 4.5 plots the absolute error % with respect to physical measurements

Precision of this method can be further increased by keeping the targets closer to the camera while fully utilizing the ROI in the image frame or devising a high-resolution camera. It should be also noted that the Vernier readings were taken by manually placing the Vernier calliper to the targets with the naked eye. This could lead to human error in getting the precision to the second decimal place during the experiment.

In this research study, the targets and checkerboard squares were kept in the same region of the image. During this experiment, extrapolation of the coordinates was avoided to ensure the results to be more accurate.

4.1.5 Problems Encountered and the Solutions:

Some cameras are having different distortion coefficients. Hence the distortion parameters have to be changed and checked for the appropriate values for those parameters. Having folds on the target sheet or calibration pattern sheet will cause errors in readings. Thus, they have to be pasted on a flat plane. A sample photo of the targets has to be checked with the image processing before proceeding the experiment since some light distortions cannot be seen with naked eyes.

4.2 Measuring In-plane Deflection of Flexible Structures

Through the Section 4.1 focused on identification of pre-defined targets by considering only rigid body motion focus of this research is to measure the deformations of flexible elements. Hence this section investigates the possibility of using this technique to measure in-plane deflection of flexible structures. Vertical deflection of a beam subjected to three-point bending was selected for this purpose.

4.2.1 Three-point Bending Setup

Figure 4.6 shows the three-point bending configuration setup used. Here a 620 mm long simply supported timber beam with a cross section of 15.37 mm (vertical) x 25.93 mm (horizontal) was used. A loading hook was kept at the mid-span of the beam to hang weights. Five circles made out of white paper were pasted on the front vertical surface toward the upper end at preselected locations as shown in Figure 4.7. Then five dial gauges were placed on the top surface of the directly above the centre of the pasted circles. Saginamiya (TDM 1050), Mitutoyo (QDV 806) and Baker (K06) dial gauges were used for this purpose. The distance in between the dial gauges and the supports were measured using a Vernier Calliper having an accuracy of 0.02mm..



Figure 4.6: Region of Interest (ROI) of three-point bending experiment



Figure 4.7: Experimental setup of bending test

Then a Sony DSC-W800 Cyber Shot (Sony Corporation, Japan) (20MP) digital camera was fixed on to a Velbon PH-368 fluid head type tripod and set at a similar

height (1120 mm from ground) to the specimen while maintaining the face of the camera approximately parallel (not necessarily) to the face of the specimen (Figure 4.5). Then the camera was focused to capture the ROI (Figure 4.6) with a zoom of 1.6x at a distance of 710 mm from the specimen to capture the whole area of experiment. The pixel spacing at this range was 0.09 mm. Then the experiment was carried out under natural lighting.

After the experiment had been set up calibration process was carried out. In this experiment camera was kept at a static position. Here a checkerboard pattern with 7x 10 squares (each square having a size of 21.79 mm x 21.79 mm) was selected and printed on a white sheet of paper which as then pasted on a flat cardboard. This calibration grid pattern size was selected to cover the complete region of the experiment which includes configuration of the fully deflected beam. Then the checkerboard pattern was kept at different orientations (translation, rotation and slight inclination) and 21 images were captured with the camera as shown in Figure 4.8. One of the calibration image should be taken exactly on the plane of the targets (vertical surface of timber faced towards camera).

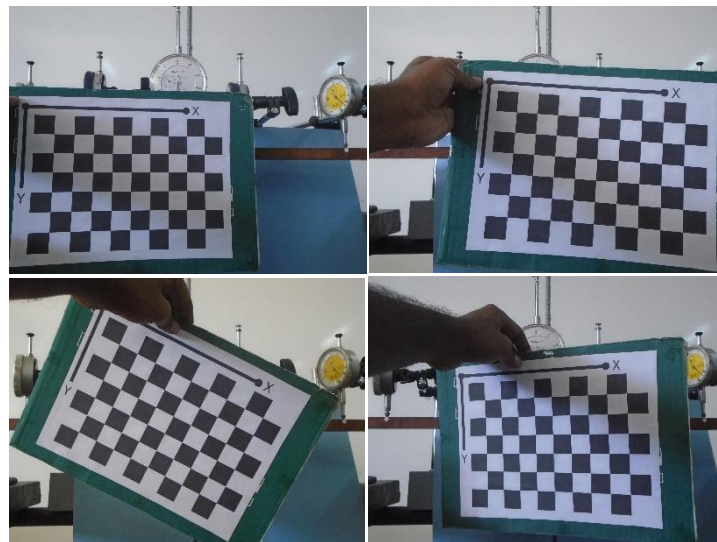


Figure 4.8: Calibration image samples

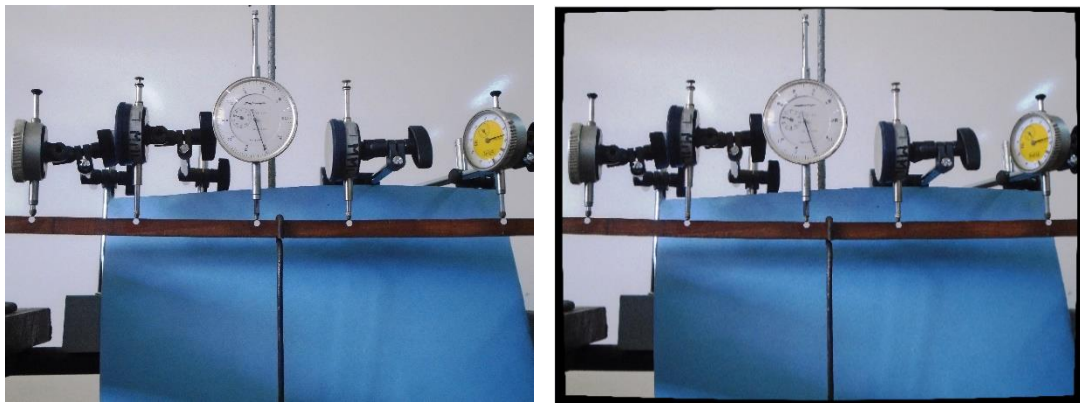
Once the camera and lighting systems were positioned a photograph of the unloaded specimen was taken. Then the specimen was loaded in steps by hanging weights on the hook. A photograph was taken at each loading step and the corresponding readings of each dial gauge were recorded.

4.2.2 Image Processing – Flexible Beam

First camera calibration process was performed to obtain the camera parameters as described in Section 3.2.

The same flow-chart (Figure 4.3) which has been used in the previous experiment can be used for this experiment to identify the target locations in the images since both experiments follow the same concept (“blob method”).

The selected target images were undistorted using the camera parameters as shown in Figure 4.9. After that they were converted to greyscale images to easily differentiate the targets from the background, since mean colour intensity of the targets can be easily defined in integers than having it in the coloured (RGB) image.



**Figure 4.9: Images of initial configuration of three-point bending setup
(a) Original image(b) After undistorting with camera parameters**

Then only the ROI was analysed to minimise the analysing time. In this process selection of a colour datum is an important to differentiate black and white pixels. A suitable colour datum range was selected through a trial and error process to minimize the distortions due to shadows in the photographs as shown in Figure 4.11. This colour datum selection is mainly dependent on the amount of light flux used to lighten the specimen. This is done mainly due to differentiate the targets from the background. Then the adjacent pixels were filled with 8-connected neighbourhood pixels to get a perfect binary image.

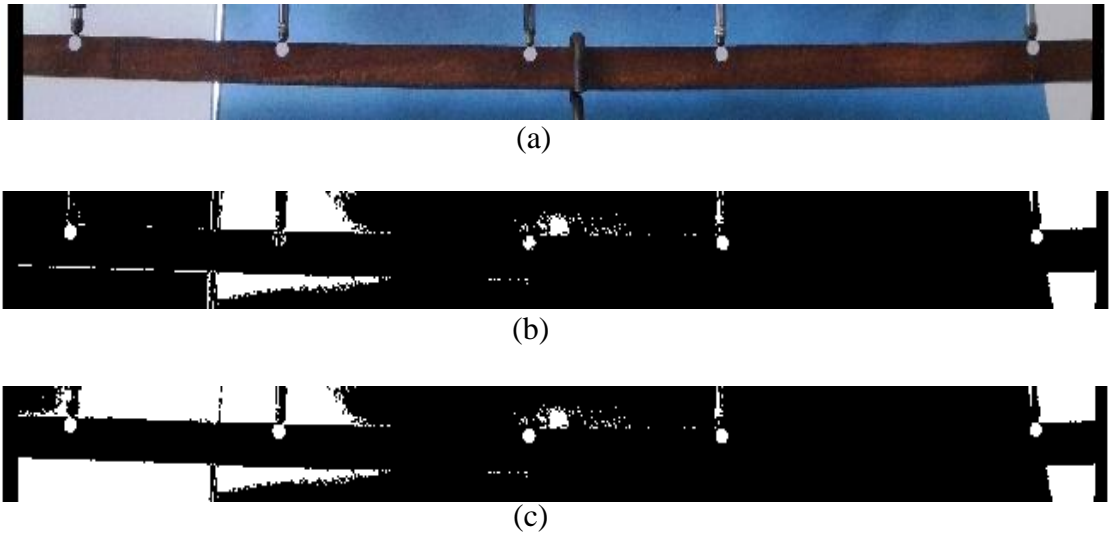


Figure 4.11: Selection of appropriate greyscale colour datum range
(a) ROI (b) inappropriate range (c) appropriate range

.In order to calculate the deformation in the predefined targets all the images under different loadings were processed through the developed MATLAB correlation algorithm. Here the centroids of the circles were identified in pixel terms. In order to convert this pixel co-ordinate system into world co-ordinate system (mm) the calibration image which was taken on the same plane of the vertical surface of the timber beam was used. This world co-ordinate system had been transferred to the target image and the vertical displacements were measured in millimetres. To cross check the world co-ordinate system the centroids of the circles were projected back on the same image as shown in Figure 4.10.

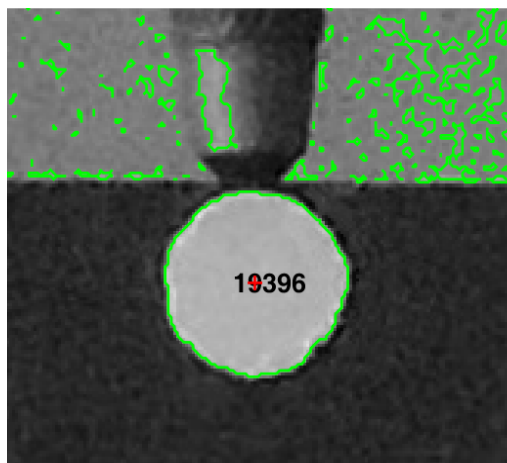


Figure 4.10: Projected centre of a target

While calculating the vertical displacements, there are two angles that must be taken into consideration.

- Camera inclination to the horizontal plane
- Corresponding calibration pattern co-ordinate system inclination to the horizontal plane

In Chapter 4.2 the distances measured with the dial gauges and the output from the MATLAB were compared for the analysis purpose.

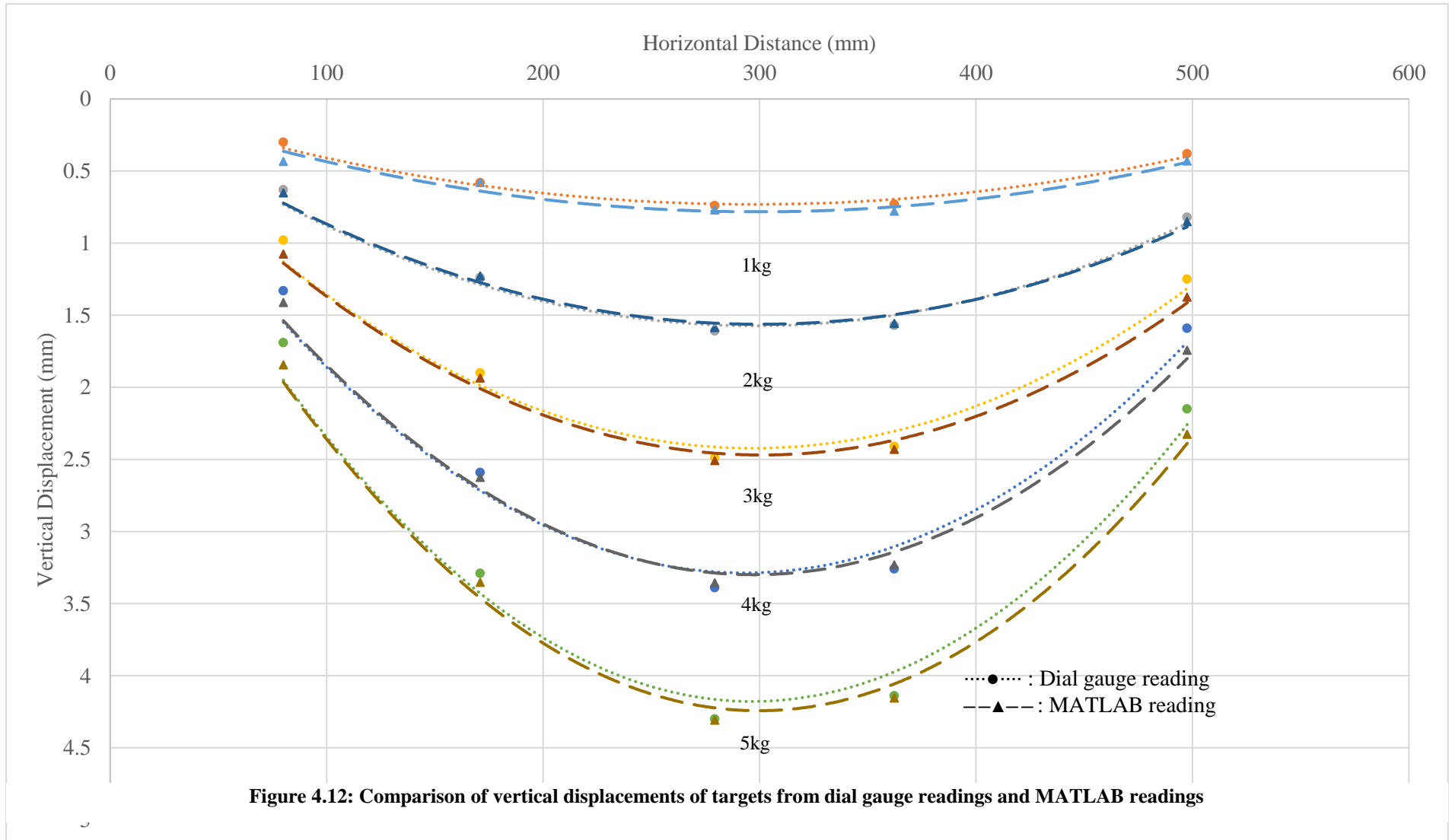
4.2.3 Comparison of Vertical Deflections

Table 4.2 presents the results obtained for vertical displacement in between initial and the current position of the targets calculated with MATLAB algorithm against the experimental results from dial gauge readings.

Table 4.2: Comparison of displacements obtained from five targets

Image	Loading (kg)	Vertical Displacement of Targets (mm)										
		Experimental readings					MATLAB results					
		1	2	3	4	5	1	2	3	4	5	
01	0.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
02	1.0	0.30	0.58	0.74	0.73	0.38	0.43	0.58	0.77	0.78	0.43	
03	1.5	0.45	0.90	1.16	1.14	0.59	0.53	0.95	1.21	1.21	0.67	
04	2.0	0.63	1.24	1.61	1.57	0.82	0.65	1.23	1.59	1.56	0.85	
05	2.5	0.80	1.59	2.08	2.00	1.05	0.89	1.60	2.08	2.01	1.13	
06	3.0	0.98	1.90	2.49	2.41	1.25	1.08	1.93	2.51	2.43	1.37	
07	3.5	1.16	2.26	2.95	2.84	1.48	1.32	2.35	2.98	2.87	1.56	
08	4.0	1.33	2.59	3.39	3.26	1.59	1.41	2.63	3.36	3.23	1.74	
09	4.5	1.51	2.93	3.84	3.69	1.92	1.66	3.01	3.86	3.72	2.08	
10	5.0	1.69	3.29	4.30	4.14	2.15	1.84	3.3i5	4.31	4.16	2.33	
Maximum Error		-	-	-	-	-	0.16	0.09	0.05	0.07	0.18	

From Table 4.2 it can be observed that there is a maximum error of 0.18 mm. Figure 4.12 shows the deformed beam profile which is drawn by fitting quadratic polynomial trendline through to the measured deflections. Both trendlines show similar deformation profile. Note that here a pixel in an image corresponds to 0.09 mm in world units. This shows that the readings have a good correlation and hence the technique can be used to measure in-plane deformation of a flexible structure to two sub-pixel accuracy.



However, it should be noted that there the calibration images had covered only three targets in the middle region and image calibration had to be extrapolated to cover the rightmost and leftmost targets. When only the middle three are considered the maximum error reduces to one pixel (0.09 mm). This emphasizes the importance of carefully selecting a calibration pattern which covers the entire region of interest.

Here the camera should not be repositioned by any means throughout the experiment. In case of repositioning, the entire experiment has to be repeated including recalibration of the camera.

The algorithm developed here is focusing on multi-directional variation of the targets and measures the distances in millimetres directly. In this scenario, the algorithm was modified to measure only the intended vertical displacements of the predefined targets. This method can be further developed to analyse the strain distribution on the elements in contour form which is done by creating multiple targets on the surface of an object and tracking the targets displacement over the surface. This method was followed in next Section which illustrates measuring full-field strain variation over a surface.

5 FULL-FIELD DEFORMATION MEASUREMENT

In order to measure the surface deformation there should be significant number of targets to cover the ROI. Manual marking or pasting of predefined targets becomes a tedious operation in this regard. Further as it was mentioned in Chapter 2.2 the multiple point targets should have unique feature or numbering in order to differentiate from each other. Hence it is preferred to have a random target patterns on the surface of an object there are two main methods that can be used to measure the surface deformation which are namely Template Matching and Pattern Matching.

5.1 Template Matching:

In this method, a template has been analysed for its texture/ pattern as a whole. The template with high correlation texture in the next image will be selected as the best matching template here. Appendix B.5 provides the algorithm which uses Normalized Cross Correlation and Sum of Squared Difference to analyse for template matching.

Figure 5.1 shows that how this technique works. Here, a picture of garlic has been selected as the template and the best matching template from a group of vegetables was found. By analysing the texture, the best correlation was found as the final output which was marked with a green rectangle in the picture in the bottom right corner.

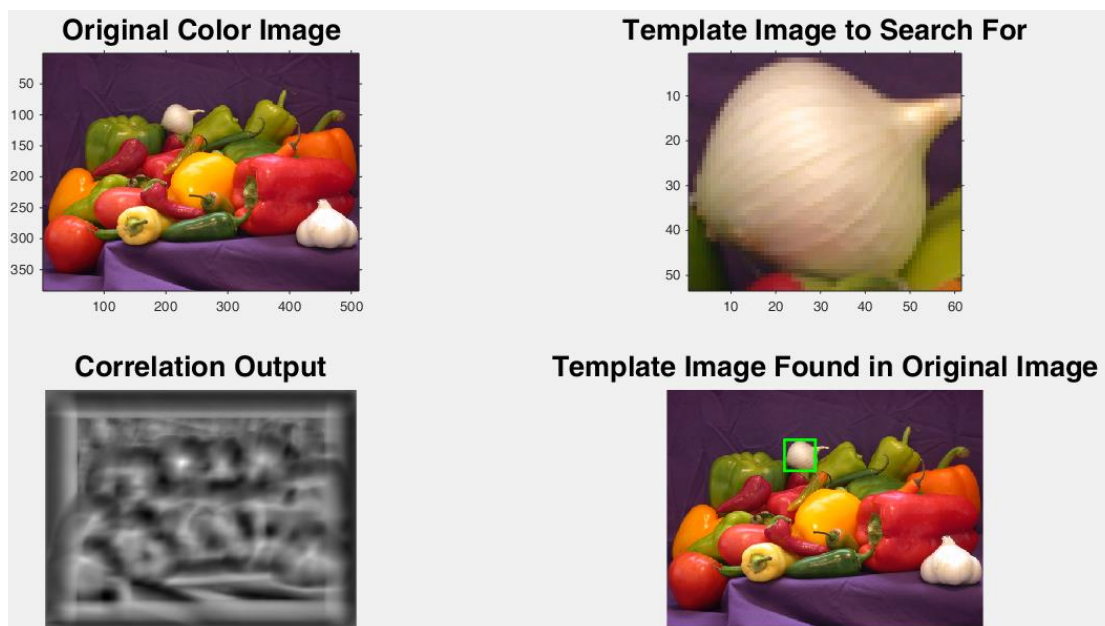


Figure 5.1: Template matching analysis

But it has been identified that there this method is having a difficulty in tracking the object when the template has been distorted excessively in scale or rotation. Figure 5.2 (a) shows a situation where the template was rotated by 10 degrees and in Figure 5.2 (b) it was rotated by 90 degrees. Hence in Figure 5.2(a), the identified locations are slightly varying and in Figure 5.2(b) the identified location with high correlation template is completely wrong. Hence this method is not reliable when there is an excessive deformation in the structure.

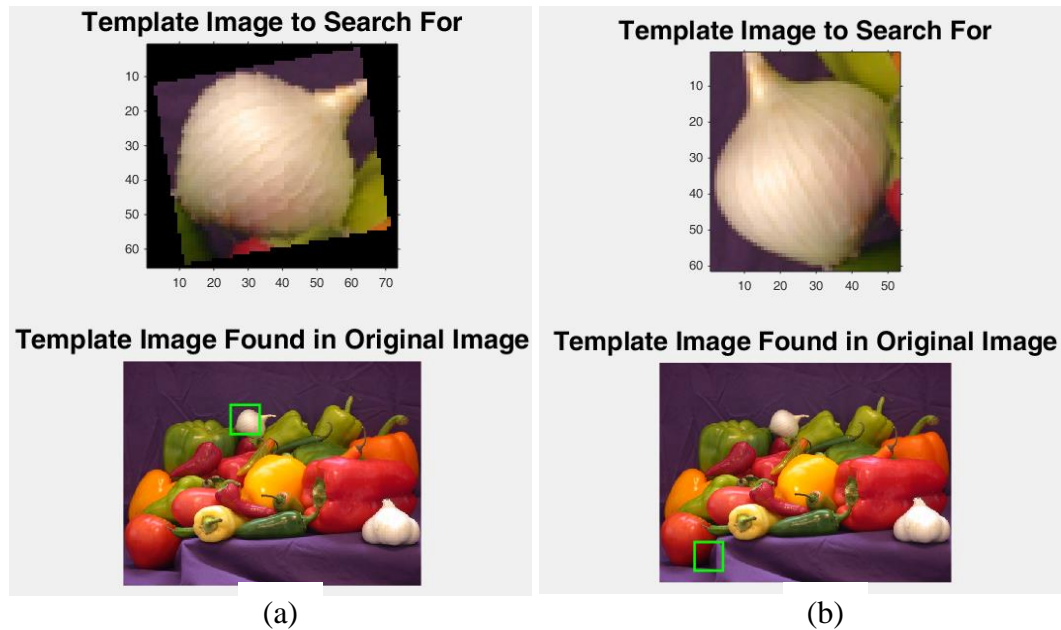


Figure 5.2: Template rotated (a) by 10 degrees (b) by 90 degrees

5.2 Pattern Matching

This technique tracks only some certain characteristic features (unique features) in the greyscale image array. Unlike a template matching, by manually inputting an arbitrary ROI it would track the distinguished features. This helps to identify excessive deformation until the targets get separated into parts. As preliminary test part of a balloon was taken as the specimen and random patterns were marked with a pen. Then two photographs were taken in its undistorted form and once it stretched as shown in Figure 5.3. The points highlighted with yellow rectangle are some points which illustrates that the second picture has been stretched non-uniformly.

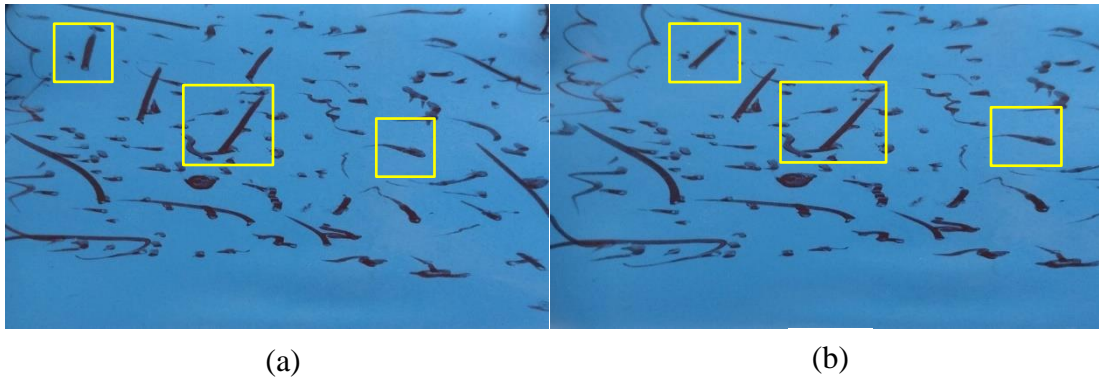


Figure 5.3: Image of a balloon (A) before stretching (b) after stretching

In detecting the surface features, there are different techniques available. Some of the main techniques and the corresponding detectable features are tabulated in Table 3.1.

Table 5.1: Types of detecting surface features

Type of Points	Algorithm	Detectable Feature
Corner Points	detectFASTFeatures (Features from accelerated segment test (FAST)) (Uses an approximate metric to determine corners)	Corners Single-scale detection Point tracking, image registration with little or no scale change, corner detection in scenes of human origin, such as streets and indoor scenes.
	detectMinEigenFeatures (Minimum eigen value algorithm) (Uses minimum eigen value metric to determine corners)	
	detectHARRISFeatures (Harris-Stephens algorithm) (More efficient than the minimum eigenvalue algorithm)	

BRISK Points	detectBRISKFeatures (Binary Robust Invariant Scalable Keypoints (BRISK) algorithm)	Corners Multiscale detection Point tracking, image registration, handles changes in scale and rotation, corner detection in scenes of human origin, such as streets and indoor scenes
SURF Points	detectSURFFeatures (Speeded-up robust features (SURF) algorithm)	Blobs Multiscale detection Object detection and image registration with scale and rotation changes
MSER Regions	detectMSERFeatures (Maximally stable extremal regions (MSER) algorithm)	Regions of uniform intensity Multi-scale detection Registration, wide baseline stereo calibration, text detection, object detection. Handles changes to scale and rotation. More robust to affine transforms in contrast to other detectors

Out of these techniques SURF feature detection method is the best since it will yield maximum matching points than the other detecting algorithms and the relevant detectable features when the subject experiences scale and rotation deformation. This tracks the shape of the target and surrounding targets in the consecutive images when the object deforms. Figure 5.4 shows the detected SURF points (green cross) in the greyscale image of the non-stretched balloon.

An arbitrary region has been selected in yellow square to analyse how that region was deformed in the latter image. Hence that part was cropped, and the corresponding unique features were searched in the latter image. Figure 5.5 shows that how the unique features have been matched. At this point it should be noted that there should be at least 3 points to be matched to identify the region from the latter image. Because if there are 2 points with similar properties the output would be wrong. Thus, using the SURF feature detection method, the following experiment was carried out to

detect random multiple targets under deformation. Figure 5.6 shows that how the square region was deformed into a quadrilateral polygon.

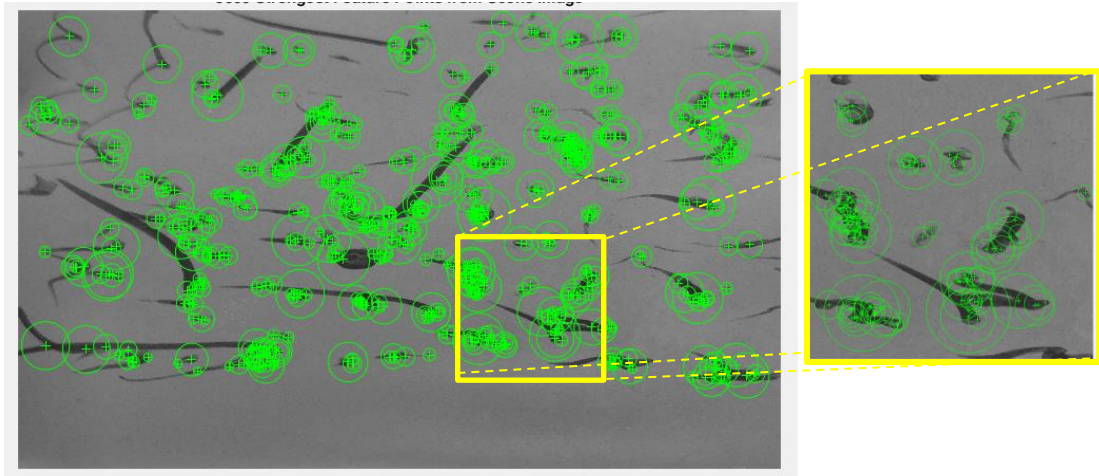


Figure 5.4: Detected SURF points

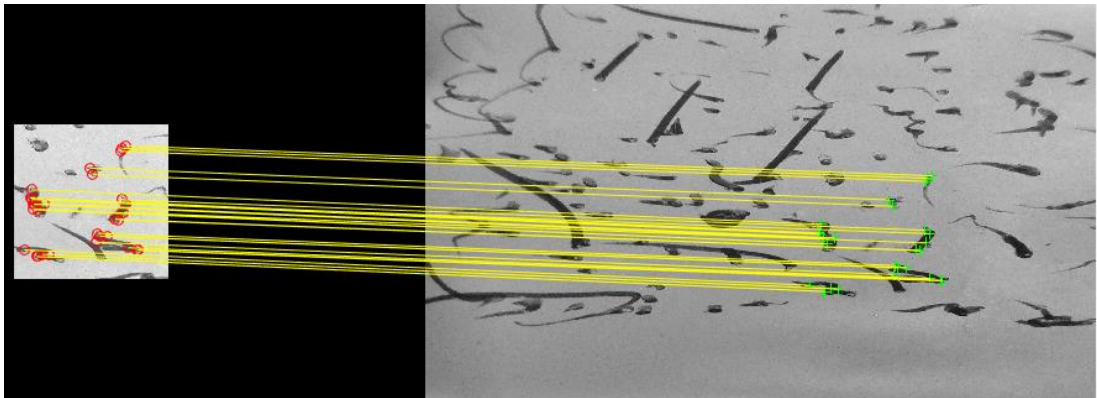


Figure 5.5: Matched features

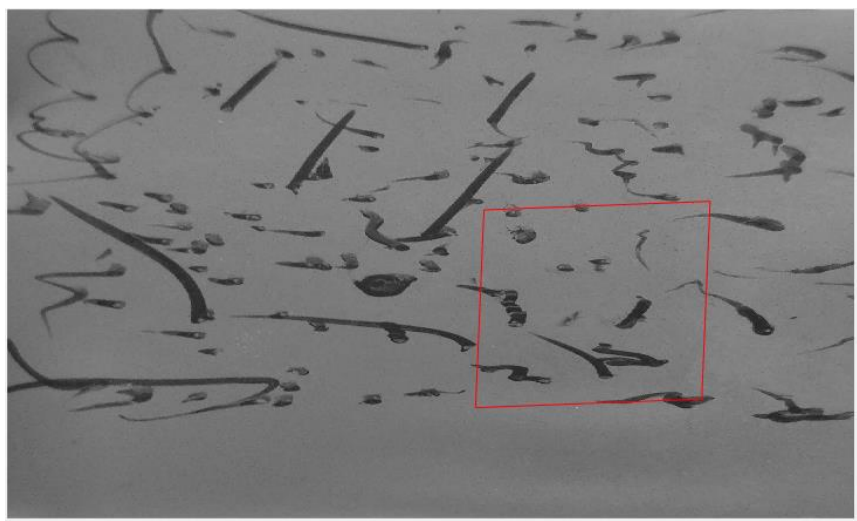


Figure 5.6: Deformed region (Red quadrilateral)

5.3 Surface Deformation under Uniaxial Tensile Testing

Strain contour is an appropriate way to visualize how the strain varies over a surface. Here an attempt was made to measure the surface deformation for a specimen subjected to uniaxial tension.

5.3.1 Uniaxial Tensile Test Setup

A similar Aluminium dog bone specimen which was used in the Chapter 3.1 was fixed to a A-1000E Universal Testing Machine (UTM) as shown in Figure 5.7. In order to prepare the sample, it was first painted with silver colour paint on its whole surface and then black paint was sprayed using the toothbrush to have random pattern. As mentioned in the literature survey having a uniform pattern leads to lower the accuracy. If the original surface has random pattern it does not require to be painted again. Three of their samples were selected for this purpose.

The prepared Aluminium dog-bone specimen was clamped to a specially designed clamp to have the analysing surface to be facing the camera. Then the extensometer was attached to the specimen with a gauge length 50 mm as shown in Figure 5.8. Then the specimen was clamped at to ends with the hydraulically controlled



Figure 5.7: Experimental setup of uniaxial tensile testing

clamps of the UTM (specifications of the extensometer and the UTM are mentioned in Appendix A.3 and Appendix A.4 respectively. Canon EOS 700D digital single reflex lens (DSLR) camera (specifications in Appendix A.5) was fixed to a tripod and placed at a similar level to the centre of the specimen as shown in figure 5.7.

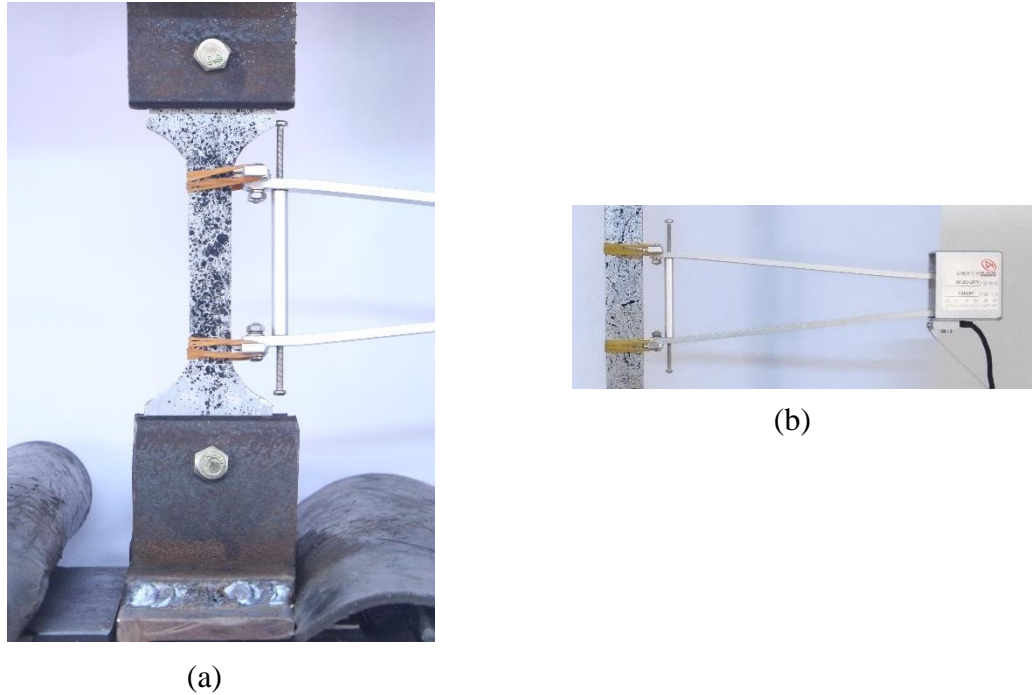


Figure 5.8: (a) Clamped specimen (b) Extensometer

Before starting the experiment, it is important to consider the external factors that are mentioned in Section 3.1. Here, only a room fluorescent light and natural light were used. Background of the specimen was covered with a white colour sheet of paper to avoid background distortions. Camera was kept 715 mm away from the specimen in order to physical restrictions with the UTM and to cover the area of full extension of the specimen at the the end of the test. However, the camera was then manually zoomed to capture only the area of interest to maximise the uuse of resolution of the camera. Here it is extremely important not to disturb the camera even by pressing the button to activate the camera shutter. Hence a remote trigger was attached to the camera for taking pictures when required. First a photograph of the sample was taken and analysed with IPT to check whether the external factors are appropriate to proceed.

Then a calibration checkerboard square pattern was prepared to cover the ROI of the experimental setup. Following the procedure described in Section 3.2, 35 number of photographs were taken for image calibration process.

The specimen was loaded by moving the head at a rate of 1 mm per minute. Both the camera and the machine was started together and a photograph was taken every 30 seconds, while the UTM was set to record displacement and load at every 0.2 s interval. The test was stopped when the specimen starts yielding since the extensometer configuration is not suitable beyond that point.

This procedure was repeated for three specimens.

5.3.2 Processing Images for Full-field Deformation

To begin image processing, it is first important to select the suitable images from the experiment. Figure 5.9 shows the load vs. time variation obtained for first specimen. Note that there are noises before 140 seconds and it starts fluctuating again after 660 seconds. Hence the analysis was restricted to the monotonic loading region between 140 seconds and 600 seconds to minimize effects due to sudden fluctuations in the experiment.

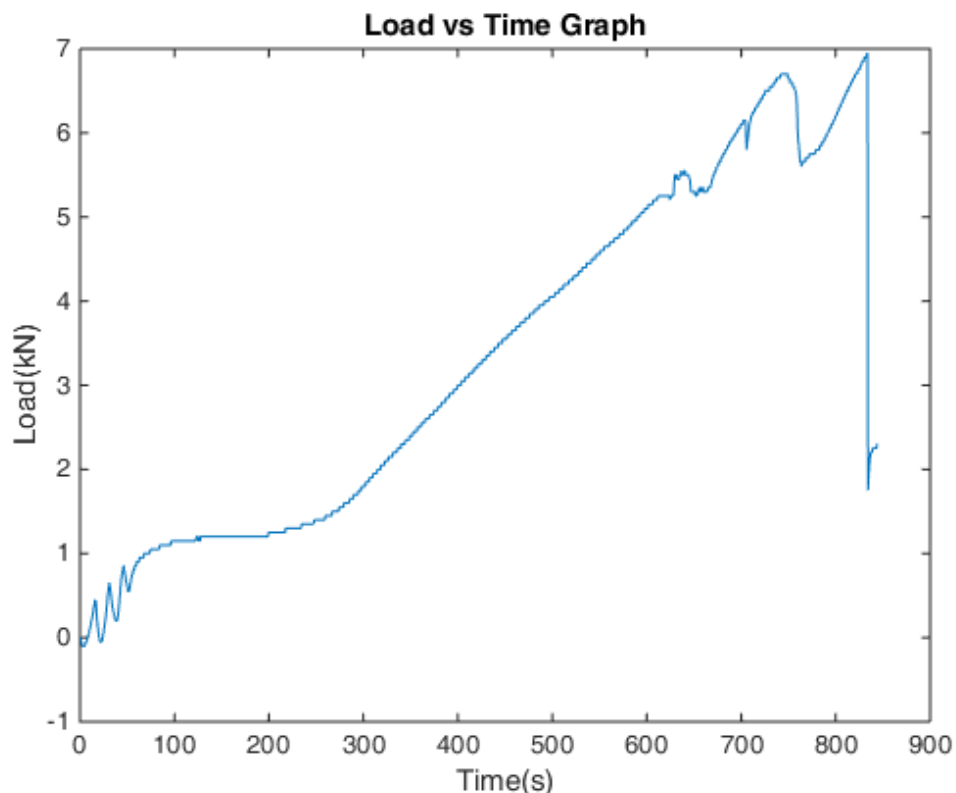


Figure 5.9: Load vs. Time curve for a specimen

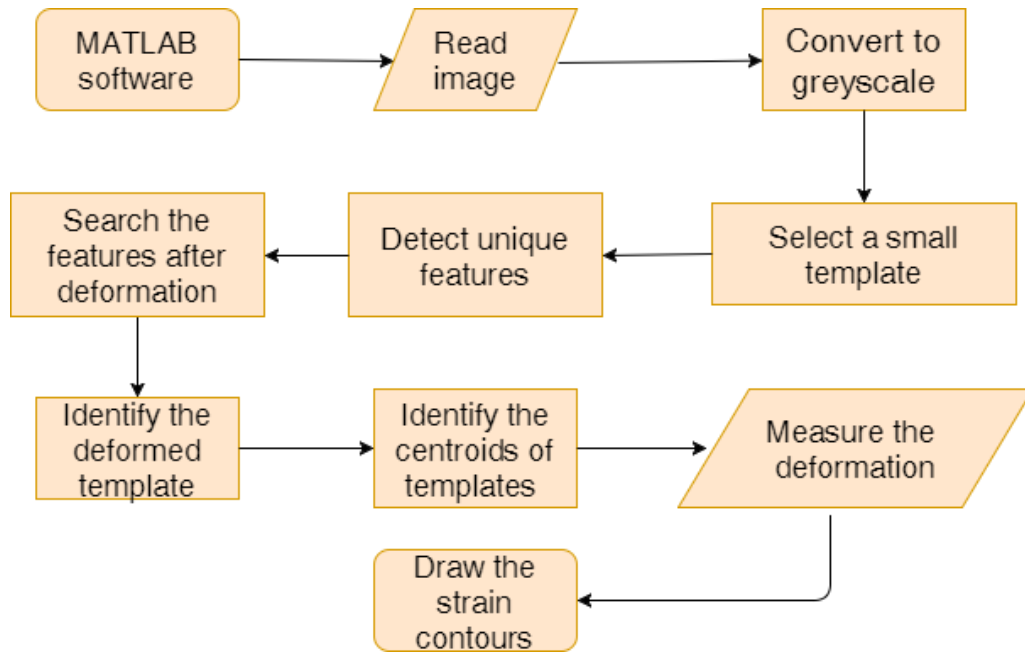


Figure 5.10: Flow chart for image processing of uni-axial tensile test

The selected images were then processed as shown in Figure 5.10 flow chart using the algorithm presented in Appendix B.6.

Here the “pattern recognition” is used to capture the full-field deformation over the previously used “blob method” which is used for identifying predefined targets. As explained in Section 4.2, initially images were converted to greyscale. Then an arbitrary square template (size of the square in pixel terms can be defined by the user) was cropped from the original image. Higher the size of arbitrary template will yield lesser accurate results. Hence the size of the arbitrary template should be selected in a way that most of the templates should have at least 3 matching points. Then the unique points on that ROI was identified with the detectSURFFeatures algorithm. Then from the consecutive images the matching points were compared and the best matching points will be identified. Then the corresponding identical template was analysed how it was deformed. Afterwards the corresponding centroid can be calculated for the deformed quadrilateral.

This procedure was continuously repeated by moving the arbitrary template by 10 pixels on both horizontal and vertical directions. Since the test is a uniaxial tensile test, the strain in the vertical strain has been calculated by considering the strain in

between two vertical arbitrary templates. However, this part of the algorithm can be easily modified to measure the strain direction depending on the application.

After the strain measurements are calculated, they were put into a matrix array which consists of X coordinate and Y coordinate of a point and the respective strain at that location. By producing this matrix array the strain contour of the specimen can be produced.

If there are not enough matching points the algorithm will move to the next arbitrary template. Hence some identical templates cannot be detected while it's deforming. This may happen due to less number of unique characteristic features within that template or the original target is separated in parts due to excessive deformation at that point. The deformation values in these missing regions were interpolated using natural interpolation algorithm.

5.3.3 Comparison of Results from Uni-Axial Tensile Test

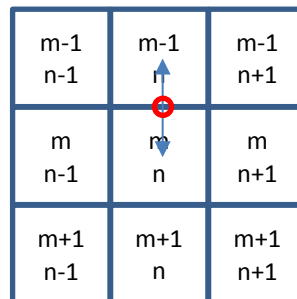
Here, the strain between points of extensometer connections were calculated using MATLAB algorithm and compared with the reading obtained from Extensometer. For this purpose, only the Extensometer connected region as the arbitrary template and compared with the other images. Table 5.2 presents the summary of the output for all three specimens.

Table 5.2: Comparison of strain output from extensometer and MATLAB

Specimen 1				Specimen 2				Specimen 3			
MATLAB	Measured	Time (min)	% error	MATLAB	Measured	Time (min)	% error	MATLAB	Measured	Time (min)	% error
-	0	0	-	-	0	0	-	-	0	0	-
0.1307	0.130	1	0.54	0.1304	0.130	1	0.31	0.0279	0.028	1.02	0.36
0.2204	0.220	2	0.18	0.2326	0.232	2	0.26	0.1582	0.158	2	0.13
0.2974	0.298	3	0.20	0.3174	0.318	3	0.19	0.2475	0.248	3	0.20
0.3961	0.396	4	0.03	0.4438	0.444	4	0.05	0.3583	0.358	4	0.08
0.5876	0.588	5	0.07	0.8276	0.828	5	0.05	1.3622	1.362	5	0.01
1.3366	1.336	6	0.04	1.9443	1.944	6	0.02	2.7638	2.764	6	0.01
2.4563	2.456	7	0.01	3.1787	3.178	7	0.02	4.4604	4.460	7	0.01
3.7658	3.766	8	0.01	4.7562	4.756	8	0.00	5.5177	5.518	7.30	0.01
5.7236	5.724	9	0.01	5.8117	5.812	8.30	0.01	6.0345	6.034	7.43	0.01

From this it can be observed that the strain measured with Extensometer and the MATLAB output are having closer values to each other up to third decimal places and maximum error observed was 0.54%. Further than this all the specimens have higher error margins when there is smaller strain. When the strain increases the error decreases.

In order to calculate the vertical surface strain variation, the whole image was divided into square templates which was having a size of 30 x 30 pixels.



After dividing the templates in squares, location of the corresponding deformed quadrilateral was identified from the deformed image. Since this test is only to measure the vertical deformation, the distance in between the centroids of the consecutive two vertical templates were considered in the algorithm calculation. For instance, in the second vertical grid, $[m-1, n]$, $[m, n]$ template distance was compared to measure the strain in the mid-point (red circled point) of both template centroids. The divided squares can be overlapped. Overlapping the templates will give high number of strain points. The algorithm can be modified to measure the strain in multi-directions by modifying the algorithm in a way in which direction the templates are going to be related.

The block diagram Figure 5.11 explains how the pattern matching algorithm has been developed to find out the strain variation over the surface.

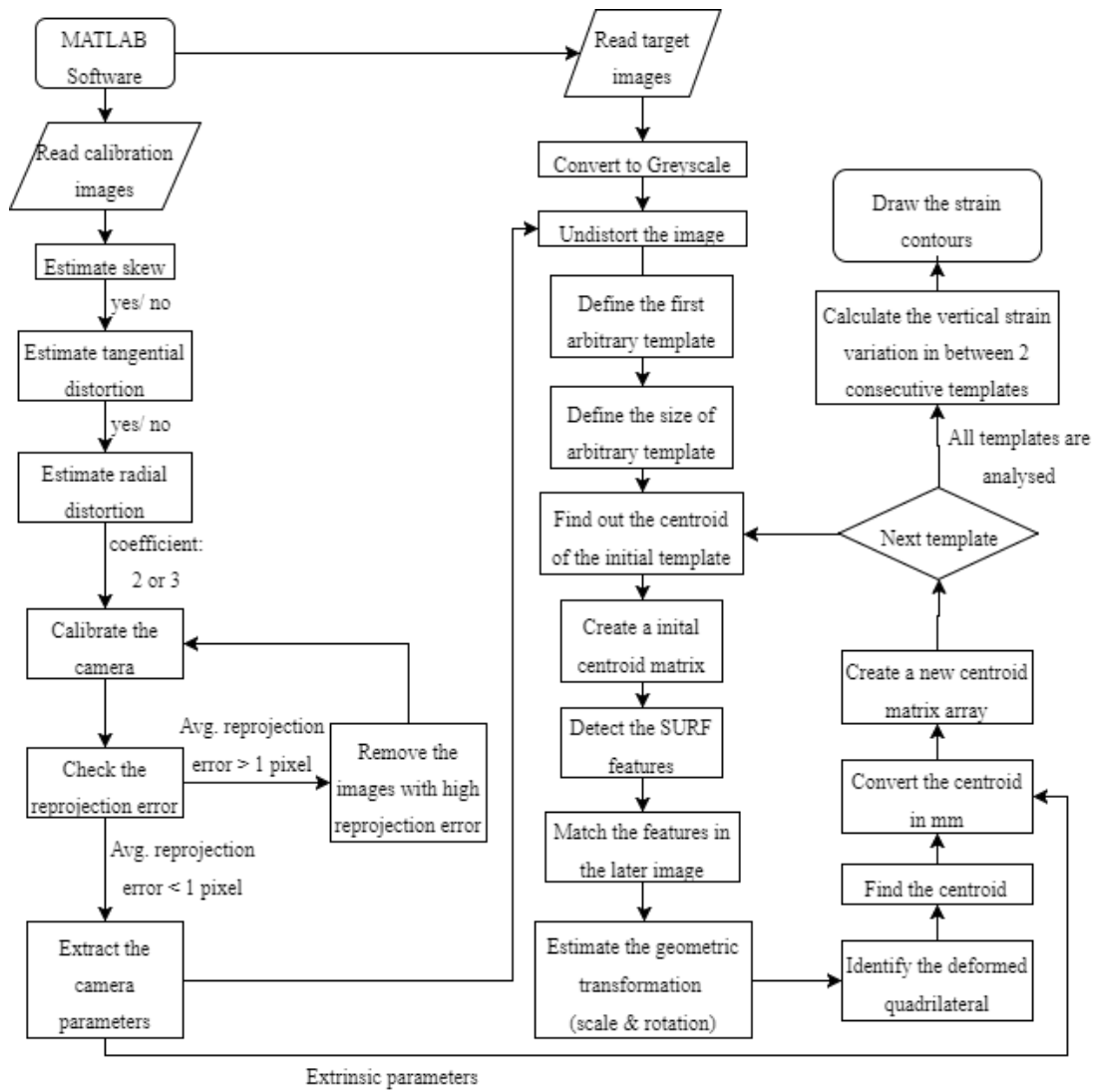
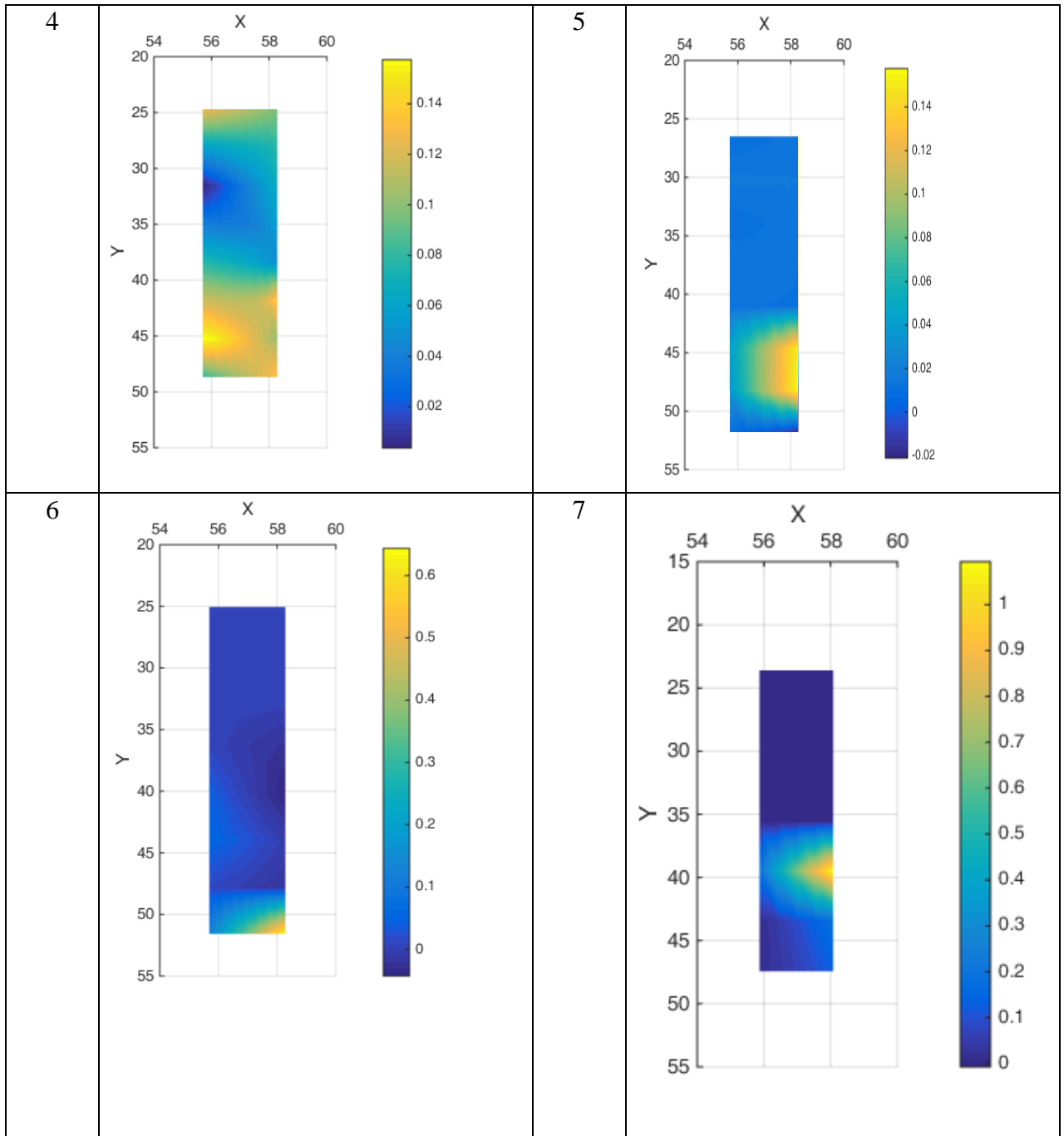


Figure 5.11: Block diagram for pattern matching

The following Table 5.3 illustrates that for specimen 3 (Results of the other two specimens are shown in Appendix) how the strain contour varies over time for each specimen: Figure 5.11 (b) provides the actual image before it got failed.

Table 5.3: Strain contours of specimen 3

Time (mins)	Strain Contour	Time (mins)	Strain Contour
0		1	
2		3	



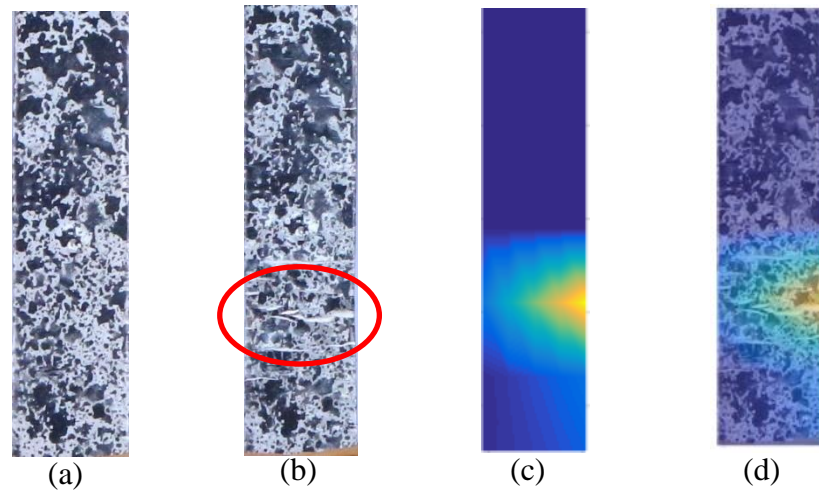


Figure 5.12: ROI (a) before loading (b) during necking (c) strain contour during necking (d) overlapping of b & c (the strain contour during the necking sample)

From this, it can be noticed that using this method the strain over any surface in-plane full-field deformation can be calculated and the failure location can also be easily identified. However, when the strain goes over 100%, the targets getting excessively distorted. Hence it is recommended to use this method having strain lesser than 100%.

When the strain contours were analysed the 0th minute image was compared for the strain with the same image to verify the accuracy of the identified quadrilateral during the pattern matching. Even though the expected contour is exactly zero, in the result it was showing an existing strain which has a value in the range of 10^{-6} . This error was caused due to the negligible deviation in the identified quadrilateral.

With the time strain varies over differently over the regions. Before failure at the 6th minute the strain has a high value at the bottom and but at the 7th minute it suddenly increases in the middle at the seventh minute. There is a possibility of the strain variation was getting affected due to the clamp slipping.

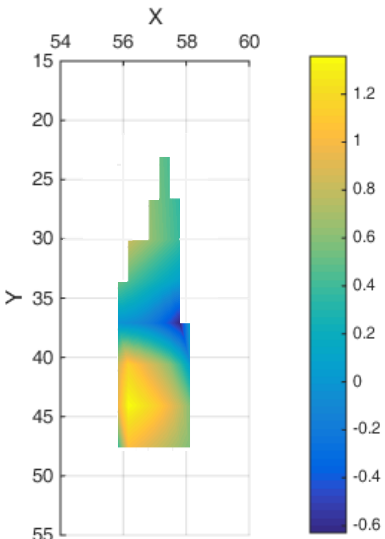
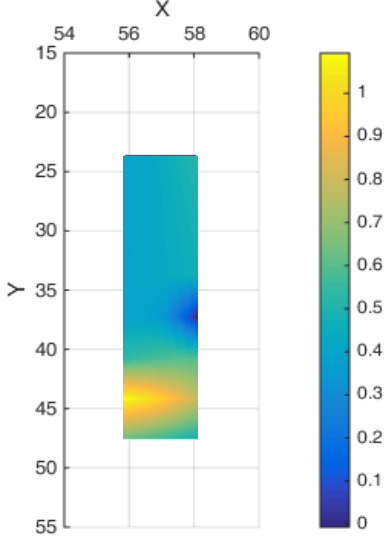
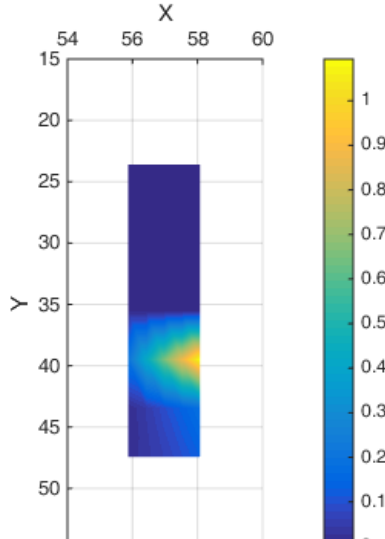
Table 5.4 illustrates the change in 7th minute strain contours for specimen 3 with the same colormap range (legend) which can be easily visualized.

Table 5.4: Comparison of strain contours with the same colormap range

Time (mins)	Strain Contours	Time (mins)	Strain Contours
0		4	
6		7	

After analysing the strain contours there was another image processing step was carried out to find out how the mesh sensitivity (size of the arbitrary template) affects the strain contour plots. The following Table 5.5 shows different sizes of arbitrary template analysed for strain contours.

Table 5.5: Mesh sensitivity analysis for the strain contours

Size of arbitrary template (Pixel x Pixel)	Output	Size of arbitrary template (Pixel x Pixel)	Output
20x20		40x40	
30x30			

It can be observed that when the size of arbitrary template reduces accuracy increases but due to the lesser matching points it is difficult to plot the whole area full-field deformation. When the size increases accuracy reduces. Hence the arbitrary template size has to be selected by analysing one of the deformed photograph and chosen correctly. From this, it was finalized that 30 x 30 size arbitrary template was able to produce more reasonable strain contour plots. Hence for the final strain contour producing purpose the same size was used.

However, for the strain measurements the accuracy has to be increased when there will be only a small amount of elongation by using high resolution camera.

In the application of this method in the industry, the scale instrument cannot be brought to the place when there is no accessibility. Hence to get the scale at the location where the displacement is going to be measured, parallel laser beams which have a constant width in between can be used. This method can be used for very long-distance measurements as well.

5.4 Cost Comparison of the Developed Technique

In this research, the camera which was used to capture the earlier experiments has a cost of 16,000 LKR (104 USD) and the DSLR camera which was used in the final experiment was 67500 LKR (440 USD). The MATLAB software for the academic institutions were having a value of 550 USD. Hence the minimum total cost for using this technique will be around 650 USD for an institution. For the commercially available systems the minimum cost will be 30,000 USD. Hence the developed technique is considerably having a low-cost and can be adopted easily in Sri Lanka.

6 CONCLUSION AND FUTURE WORK

An attempt is made to develop a low-cost optics based displacement measuring technique using DIC. An algorithm was developed to measure the displacements of a set of predefined targets. Camera was calibrated to identify the plane of the object and scale. Results implied that this method can measure displacement to an accuracy of 0.39 mm during the rigid body motion and the measurements can be done up to the subpixel level.

Then the algorithm was developed to measure the vertical displacement of a bending experiment timber specimen. Results showed that this method can measure the displacement to an accuracy of 0.1 mm. However, it should be noted that this test could measure only the vertical deformation at different points whereas the technique developed in this research can measure the elongation between much smaller predefined regions as well.

The algorithm then modified to multiple predefined targets to measure the full-field deformation using pattern recognition method. Strain contours also could be produced using this method where the full-field deformation will be easily visualized. MATLAB output is having closer values to each other up to third decimal places and maximum error observed was 0.54%. Further than this all the specimens have higher error margins when there is smaller strain.

Therefore, this technique can be used for surface strain measurements different applications and can be adopted in local context at a low cost.

6.1 Future Work

Furthermore, the present technique is limited to in-plane measurements and hence cannot be used for any out-of-plane deformations like bending or buckling which are common in civil engineering applications. Hence it is intended to expand this technique to use two cameras instead of a single camera which will allow to capture out-of-plane deformation as well as provide a cross-reference between cameras better accuracy. But still this solution cannot be used to the internal deformation in structures.

The arbitrary template size cannot be reduced after a certain limit though the accuracy increases. Hence, in order to achieve total surface deformation with a high accuracy a further modified method has to be developed.

In order to identify short term strain changes, input data has to be changed to video. But still the developed algorithm can be used for that purpose. If the algorithm can be further developed to measure the real-time strain measurements that will be really useful for different applications.

REFERENCES

- Hild, François, and Stéphane Roux. "Measuring Stress Intensity Factors with a Camera: Integrated Digital Image Correlation (I-DIC)." *Comptes Rendus Mécanique* 334, no. 1 (January 2006): 8–12. doi:10.1016/j.crme.2005.11.002.
- Waterfall, Paul, Nick McCormick, and Alan Owens. "Optical Imaging for Low-Cost Structural Measurements." *Proceedings of the ICE - Bridge Engineering* 167, no. 1 (March 1, 2014): 33–42. doi:10.1680/bren.11.00055.
- Tung, S.H., Kuo J.C. and Shih, M.S. "Strain distribution analysis using digital image correlation techniques." The Eighteenth KKCNN Symposium on Civil Engineering-NTU29 (December 19,2005)
- Hild, F., & Roux, S. (2006). Digital Image Correlation: from Displacement Measurement to Identification of Elastic Properties – a Review. *Strain*, 42(2), 69–80. <https://doi.org/10.1111/j.1475-1305.2006.00258.x>
- Løkberg, O. J. (1987). Electronic Speckle Pattern Interferometry. In O. D. D. Soares (Ed.), *Optical Metrology* (pp. 542–572). Springer Netherlands. Retrieved from http://link.springer.com/chapter/10.1007/978-94-009-3609-6_36
- Single Camera Calibration App - MATLAB & Simulink. (n.d.). Retrieved July 24, 2016, from <http://www.mathworks.com/help/vision/ug/single-camera-calibrator-app.html>
- J. Lord and N. McCormick, "Digital image correlation for structural measurements," *Proc. ICE - Civ. Eng.*, vol. 165, no. 4, pp. 185–190, Nov. 2012.
- H. Haddadi and S. Belhabib, "Use of rigid-body motion for the investigation and estimation of the measurement errors related to digital image correlation technique," *Opt. Lasers Eng.*, vol. 46, no. 2, pp. 185–196, Feb. 2008.
- C. Murray, W. A. Take, N. A. Hoult, and A. Hoag, "Field monitoring of a bridge using digital image correlation," *Proc. ICE - Bridge Eng.*, vol. 168, no. 1, pp. 3–12, Mar. 2015.
- M. J. . Feron, "Deformation analysis using image processing," *Mechanics of Electronic Textiles* (May, 2008)
- P. Bergeron, "Parallel Lasers for Remote Measurements of Morphological Traits," *J. Wildl. Manag.*, vol. 71, no. 1, pp. 289–292, Feb. 2007.
- B. Pan and K. Li, "A fast digital image correlation method for deformation measurement," *Opt. Lasers Eng.*, vol. 49, no. 7, pp. 841–847, Jul. 2011.
- "Imetrum: non-contact precision measurement." [Online]. Available: <http://www.imetrum.com/>. [Accessed: 23-Mar-2017].
- Ali, Aziah, Rajasvaran Logeswaran, and Michel RM Bister. "Optimal camera calibration for 3 D object tracking." *WSEAS Transactions on Computers* 3, no. 1 (2004): 262-266
- A. Diluxshan and H.M.Y.C. Mallikarachchi, "A Low Cost Optics Based Displacement Measuring Technique," presented at the *Society of Structural Engineers Annual Sessions 2016* (SSEAS 2016), Sep. 2016.
- A. Diluxshan and H.M.Y.C. Mallikarachchi, "Image Based Measuring Technique for In-plane Loading", presented at the *Internation Conference on Sustainable Built Environment 2016* (ICSBE 2016), Sep. 2016.

BIBLIOGRAPHY

- Hild, François, and Stéphane Roux. "Measuring Stress Intensity Factors with a Camera: Integrated Digital Image Correlation (I-DIC)." *Comptes Rendus Mécanique* 334, no. 1 (January 2006): 8–12. doi:10.1016/j.crme.2005.11.002.
- Lord, Jerry, and Nick McCormick. "Digital Image Correlation for Structural Measurements." *Proceedings of the ICE - Civil Engineering* 165, no. 4 (November 1, 2012): 185–90. doi:10.1680/cien.11.00040.
- Waterfall, Paul, Nick McCormick, and Alan Owens. "Optical Imaging for Low-Cost Structural Measurements." *Proceedings of the ICE - Bridge Engineering* 167, no. 1 (March 1, 2014): 33–42. doi:10.1680/bren.11.00055.
- Feron, M.J.M. "Deformation analysis using image processing." *Mechanics of Electronic Textiles* (May, 2008)
- Tung, S.H., Kuo J.C. and Shih, M.S. "Strain distribution analysis using digital image correlation techniques." The Eighteenth KKCNN Symposium on Civil Engineering-NTU29 (December 19,2005)
- Hild, F., & Roux, S. (2006). Digital Image Correlation: from Displacement Measurement to Identification of Elastic Properties – a Review. *Strain*, 42(2), 69–80. <https://doi.org/10.1111/j.1475-1305.2006.00258.x>
- Løkberg, O. J. (1987). Electronic Speckle Pattern Interferometry. In O. D. D. Soares (Ed.), *Optical Metrology* (pp. 542–572). Springer Netherlands. Retrieved from http://link.springer.com/chapter/10.1007/978-94-009-3609-6_36
- Single Camera Calibration App - MATLAB & Simulink. (n.d.). Retrieved July 24, 2016, from <http://www.mathworks.com/help/vision/ug/single-camera-calibrator-app.html>
- Helfrick, Mark N., Christopher Niezrecki, Peter Avitabile, and Timothy Schmidt. "3D Digital Image Correlation Methods for Full-Field Vibration Measurement." *Mechanical Systems and Signal Processing* 25, no. 3 (April 2011): 917–27. doi:10.1016/j.ymsp.2010.08.013.
- Pan, Bing, and Kai Li. "A Fast Digital Image Correlation Method for Deformation Measurement." *Optics and Lasers in Engineering* 49, no. 7 (July 2011): 841–47. doi:10.1016/j.optlaseng.2011.02.023.
- Tong, W. "An Evaluation of Digital Image Correlation Criteria for Strain Mapping Applications." *Strain* 41, no. 4 (November 1, 2005): 167–75. doi:10.1111/j.1475-1305.2005.00227.x.
- Zhang, Zhi-Feng, Yi-Lan Kang, Huai-Wen Wang, Qing-Hua Qin, Yu Qiu, and Xiao-Qi Li. "A Novel Coarse-Fine Search Scheme for Digital Image Correlation Method." *Measurement* 39, no. 8 (October 2006): 710–18. doi:10.1016/j.measurement.2006.03.008.
- Kamaya, Masayuki, and Masahiro Kawakubo. "A Procedure for Determining the True Stress–strain Curve over a Large Range of Strains Using Digital Image Correlation and Finite Element Analysis." *Mechanics of Materials* 43, no. 5 (May 2011): 243–53. doi:10.1016/j.mechmat.2011.02.007.
- Périé, Jean Noël, Hugo Leclerc, Stéphane Roux, and François Hild. "Digital Image Correlation and Biaxial Test on Composite Material for Anisotropic Damage Law Identification." *International Journal of Solids and Structures* 46, no. 11–12 (June 1, 2009): 2388–96. doi:10.1016/j.ijsolstr.2009.01.025.
- Moerman, Kevin M., Cathy A. Holt, Sam L. Evans, and Ciaran K. Simms. "Digital Image Correlation and Finite Element Modelling as a Method to Determine Mechanical Properties of Human Soft Tissue in Vivo." *Journal of Biomechanics* 42, no. 8 (May 29, 2009): 1150–53. doi:10.1016/j.jbiomech.2009.02.016.

- Pan, Bing, Anand Asundi, Huimin Xie, and Jianxin Gao. "Digital Image Correlation Using Iterative Least Squares and Pointwise Least Squares for Displacement Field and Strain Field Measurements." *Optics and Lasers in Engineering* 47, no. 7–8 (July 2009): 865–74. doi:10.1016/j.optlaseng.2008.10.014.
- Murray, Chris, W. Andy Take, Neil A. Hoult, and Adam Hoag. "Field Monitoring of a Bridge Using Digital Image Correlation." *Proceedings of the ICE - Bridge Engineering* 168, no. 1 (March 1, 2015): 3–12. doi:10.1680/bren.13.00024.
- Pan, Bing, Dafang Wu, and Yong Xia. "High-Temperature Deformation Field Measurement by Combining Transient Aerodynamic Heating Simulation System and Reliability-Guided Digital Image Correlation." *Optics and Lasers in Engineering* 48, no. 9 (September 2010): 841–48. doi:10.1016/j.optlaseng.2010.04.007.
- Abanto-Bueno, Jorge, and John Lambros. "Investigation of Crack Growth in Functionally Graded Materials Using Digital Image Correlation." *Engineering Fracture Mechanics* 69, no. 14–16 (September 2002): 1695–1711. doi:10.1016/S0013-7944(02)00058-9.
- V. Richter-Trummer, P.M.G.P. Moreira, S.D. Pastrama, M.A.P. Vaz, and P.M.S.T. de Castro. "Methodology for in Situ Stress Intensity Factor Determination on Cracked Structures by Digital Image Correlation." *International Journal of Structural Integrity* 1, no. 4 (April 1, 2010): 344–57. doi:10.1108/17579861011099178.
- J. Zhang, M. Li, C.Y. Xiong, J. Fang, and S. Yi. "Thermal Deformation Analysis of BGA Package by Digital Image Correlation Technique." *Microelectronics International* 22, no. 1 (April 1, 2005): 34–42. doi:10.1108/13565360510575530.
- Berryman, James G., and Stephen C. Blair. "Use of Digital Image Analysis to Estimate Fluid Permeability of Porous Materials: Application of Two-point Correlation Functions." *Journal of Applied Physics* 60, no. 6 (September 15, 1986): 1930–38. doi:10.1063/1.337245.
- A. Diluxshan and H.M.Y.C. Mallikarachchi, "A Low Cost Optics Based Displacement Measuring Technique," presented at the *Society of Structural Engineers Annual Sessions 2016* (SSEAS 2016), Sep. 2016.
- A. Diluxshan and H.M.Y.C. Mallikarachchi, "Image Based Measuring Technique for In-plane Loading," presented at the *International Conference on Sustainable Built Environment 2016 (ICSBE 2016)*, Sep. 2016.
- H. Haddadi and S. Belhabib, "Use of rigid-body motion for the investigation and estimation of the measurement errors related to digital image correlation technique," *Opt. Lasers Eng.*, vol. 46, no. 2, pp. 185–196, Feb. 2008.
- C. Murray, W. A. Take, N. A. Hoult, and A. Hoag, "Field monitoring of a bridge using digital image correlation," *Proc. ICE - Bridge Eng.*, vol. 168, no. 1, pp. 3–12, Mar. 2015.
- P. Bergeron, "Parallel Lasers for Remote Measurements of Morphological Traits," *J. Wildl. Manag.*, vol. 71, no. 1, pp. 289–292, Feb. 2007.
- Ali, Aziah, Rajasvaran Logeswaran, and Michel RM Bister. "Optimal camera calibration for 3 D object tracking." *WSEAS Transactions on Computers* 3, no. 1 (2004): 262-266

APPENDICES

Appendix A: Equipment Used in the Experiment

A.1 Camera:

Type:	Sony W800 compact camera
Sensor:	
Sensor type	0.31 in type Super HAD CCD
No. of pixels	20.1MP
Lens:	
F-No.	F3.2 (W) – 6.4 (T)
Focal length	f=4.6-23mm
Focus range	1.97 in – infinity (W), 1.97 ft – infinity (T)
Camera:	
Exposure compensation	+/- 2.0 EV (1/3 EV steps)
ISO sensitivity	ISO 100-3200
Zoom	5x (2.6x used)
External flash mode	Auto (Off during experiment)
Shutter speed	Auto (1 – 1/1500)

A.2 Apparatus:

Hounsfield Tensometer:	
Elongation per revolution	0.0085mm

Vernier Caliper:	
Accuracy	0.02mm

Specimen:	
Dog bone specimen with following dimensions,	
Thickness	3mm
Width	38mm
Gauge length	50.8mm

Parallel length 63.5mm

A.3 Universal Testing Machine

Model	WAW-1000E
Max. Load(kN)	1000
Load measuring range	2~100%F.S. (0.2/0.4~100%F.S optional)
Load accuracy (%)	±1
Deformation measuring range	2~100%F.S. (0.2/0.4~100%F.S optional)
Deformation accuracy (%)	±1
Displacement position(mm)	0.001
Test loading speed(mm/min)	0.5-50 (0.01-50 if configured with EDC220 controller & Moog servo valve)
Max. Crosshead moving speed (mm/min)	200
Stress control range	1~60(N/mm ²)S-1
Strain control range	0.00025/s~0.0025/s
Tensile space1(mm)	750
Compression space(mm)	620
Piston stroke(mm)	250
Column Distance(mm)	570
Column Diameter(mm)	90
Working table size(mm)	650x800
Flat jaw (mm)	0-40
Round jaw(mm)	Φ20-Φ60
Jaw length(mm)	110
Jaw width(mm)	110
Platen size(mm)	Φ148x40

Bending span(mm)	50-500
Roller diameter (mm)	Φ50
Roller length (mm)	160
Bending depth (mm)	180
Net weight (kg)	3500
Max. height(mm)	2750
Dimension of load frame(mm)	900X650X2500
Size of power pack(mm)	550x550x1410
Oil tank volume(L)	110
Oil pressure (MPa)	26
Footprint (L x W)	1600x1600
Gross weight (kg)	3700
Shipping dimension (mm)	2700x1160x1100 1540x980x1725
Power supply	3PH, 380VAC, 50H, 5Kw

A.4 Extensometer

The strain gauge resistance:	350 ohms
Bridge voltage value: acuties	6V (dc, ac all can)
Output sensitivity:	about 2mV/V
Extensometer gauge length:	YYU series20~200 mm, YYJ series 5~25 mm
Maximum deformation:	YYU series25 mm, YYJ Series 4mm
Output terminal connector:	four core or five core plugs, etc

A.5 Camera EOS 700D

IMAGE SENSOR

Type	22.3 x 14.9mm CMOS
------	--------------------

Effective Pixels	Approx. 18.0 megapixels
Total Pixels	Approx. 18.5 megapixels
Aspect Ratio	3:2
Low-Pass Filter	Built-in/Fixed
Sensor Cleaning	EOS integrated cleaning system
Colour Filter Type	Primary Colour

LENS

Lens Mount	EF/EF-S
Focal Length	Equivalent to 1.6x the focal length of the lens

FOCUSING

Type	TTL-CT-SIR with a CMOS sensor
AF System/ Points	9 cross-type AF points (f/2.8 at centre)
AF Working Range	EV -0.5 -18 (at 23°C & ISO100)
AF Modes	AI Focus One Shot AI Servo
AF Point Selection	Automatic selection, Manual selection
Selected AF Point Display	Superimposed in viewfinder and indicated on LCD monitor
Predictive AF	Yes, up to 10m ¹
AF Lock	Locked when shutter button is pressed half way in One Shot AF mode.
AF Assist Beam	Intermittent firing of built-in flash or emitted by optional dedicated Speedlite
Manual Focus	Selected on lens

SHUTTER

Type	Electronically-controlled focal-plane shutter
Speed	30-1/4000 sec (1/2 or 1/3 stop increments), Bulb (Total shutter speed range. Available range varies by shooting mode)

VIEWFINDER

Type	Pentamirror
Coverage (Vertical/Horizontal)	Approx. 95%
Magnification	Approx. 0.85x (4)
Eye point	Approx. 19mm (from eyepiece lens centre)
Dioptre Correction	-3 to +1 m-1 (diopetre)
Focusing Screen	Fixed
Mirror	Quick-return half mirror (Transmission: reflection ratio of 40:60, no mirror cut-off with EF600mm f/4 or shorter)
Depth of Field Preview	Yes, with Depth of Field preview button.
Eyepiece Shutter	On strap

OTHER FEATURES

Custom Functions

Appendix B: MATLAB Algorithm

B.1 Calibration Algorithm

```
[1] %Camera Calibration
[2] %-----

[3] % Define calibration images to process
[4] imageFileNames = {'F:\20160702_205334.jpg',...
[5] 'F:\20160702_205356.jpg',...
[6] 'F:\20160702_205404.jpg',...
[7] 'F:\20160702_205410.jpg',...
[8] 'F:\20160702_205417.jpg',...
[9] 'F:\20160702_205423.jpg',...
[10] 'F:\20160702_205427.jpg',...
[11] 'F:\20160702_205433.jpg',...
[12] 'F:\20160702_205437.jpg',...
[13] 'F:\20160702_205442.jpg',...
[14] 'F:\20160702_205446.jpg',...
[15] 'F:\20160702_205449.jpg',...
[16] 'F:\20160702_205504.jpg',...
[17] 'F:\20160702_205510.jpg',...
[18] 'F:\20160702_205520.jpg',...
[19] 'F:\20160702_205547.jpg',...
[20] 'F:\20160702_205623.jpg',...
[21] 'F:\20160702_205654.jpg',...
[22] 'F:\20160702_205738.jpg',...
[23] 'F:\20160702_205743.jpg',...
[24] };

[25] % Detect checkerboards in images
[26] [imagePoints, boardSize, imagesUsed] =
[27] detectCheckerboardPoints(imageFileNames);
[28] imageFileNames = imageFileNames(imagesUsed);

[29] % Generate world coordinates of the corners of the squares
[30] squareSize = 25; %calibration 1 square size in units of 'mm'
[31] worldPoints = generateCheckerboardPoints(boardSize, squareSize);

[32] % Calibrate the camera
[33] [cameraParams, imagesUsed, estimationErrors] =
[34] estimateCameraParameters(imagePoints, worldPoints, ...
[35] 'EstimateSkew', true, 'EstimateTangentialDistortion', false, ...
[36] 'NumRadialDistortionCoefficients', 2, 'WorldUnits', 'mm');

[37] % View reprojection errors
[38] h1=figure; showReprojectionErrors(cameraParams, 'BarGraph');

[39] % Visualize pattern locations
[40] h2=figure; showExtrinsics(cameraParams, 'PatternCentric');

[41] % Display parameter estimation errors
[42] displayErrors(estimationErrors, cameraParams);

[43] % For example, we can use the calibration data to remove effects of
[44] lens distortion.
[45] originalImage = imread(imageFileNames{1});
[46] %undistortedImage = undistortImage(originalImage, cameraParams);
```



```
[44] [undistortedImage,newOrigin]=undistortImage(originalImage,  
cameraParams);  
[45] imshow(undistortedImage);  
  
[46] % See additional examples of how to use the calibration data. At  
the prompt type:  
[47] % showdemo('MeasuringPlanarObjectsExample')  
[48] % showdemo('SparseReconstructionExample')
```

B.2 Output of the above (Appendix B.1) Algorithm:

Standard Errors of Estimated Camera Parameters

Intrinsics

Focal length (pixels): [2414.6298 +/- 8.5917 2435.9077 +/- 8.5264]
Principal point (pixels): [1294.2716 +/- 2.0554 909.5780 +/- 2.6184]
Skew: [-6.1185 +/- 0.5285]
Radial distortion: [0.1609 +/- 0.0079 -0.7222 +/- 0.0607]

Extrinsics

Rotation vectors:

[0.5757 +/- 0.0022 0.1639 +/- 0.0026 3.0168 +/- 0.0006]
[0.0254 +/- 0.0041 -0.0163 +/- 0.0057 3.1235 +/- 0.0003]
[0.2953 +/- 0.0026 -0.2413 +/- 0.0030 -3.1148 +/- 0.0004]
[-0.3264 +/- 0.0030 -0.1310 +/- 0.0035 3.0844 +/- 0.0004]
[0.0189 +/- 0.0035 -0.3056 +/- 0.0040 -3.0788 +/- 0.0005]
[-0.0051 +/- 0.0036 0.0349 +/- 0.0050 -3.1282 +/- 0.0003]
[-0.1588 +/- 0.0027 -0.3494 +/- 0.0030 -3.0924 +/- 0.0005]
[-0.1368 +/- 0.0025 -0.1282 +/- 0.0030 2.9796 +/- 0.0003]
[0.3424 +/- 0.0023 -0.2966 +/- 0.0025 -3.0627 +/- 0.0004]
[0.0354 +/- 0.0026 -0.0192 +/- 0.0037 -3.1404 +/- 0.0003]
[0.0018 +/- 0.0067 0.0966 +/- 0.0093 3.1110 +/- 0.0005]
[0.0953 +/- 0.0035 0.1436 +/- 0.0039 3.0006 +/- 0.0004]
[-0.0876 +/- 0.0023 -0.3696 +/- 0.0026 3.0722 +/- 0.0004]
[0.0146 +/- 0.0024 -0.5755 +/- 0.0027 -3.0821 +/- 0.0006]
[-0.5878 +/- 0.0023 0.0969 +/- 0.0026 3.0785 +/- 0.0006]
[0.2899 +/- 0.0035 0.0142 +/- 0.0041 3.0834 +/- 0.0005]
[0.1053 +/- 0.0027 -0.2824 +/- 0.0032 3.0814 +/- 0.0005]
[0.0910 +/- 0.0023 0.2455 +/- 0.0026 3.0053 +/- 0.0004]
[0.5013 +/- 0.0019 0.4959 +/- 0.0022 3.0515 +/- 0.0006]
[-0.4972 +/- 0.0028 0.1027 +/- 0.0031 3.0984 +/- 0.0005]

Translation vectors (mm):

[38.4922 +/- 0.4421 66.3069 +/- 0.5466 506.1958 +/- 1.8718]
[75.3411 +/- 0.4455 56.9876 +/- 0.5623 522.0011 +/- 1.8702]
[119.9770 +/- 0.4152 44.0626 +/- 0.5397 500.8516 +/- 1.7495]
[93.8129 +/- 0.4505 101.1754 +/- 0.5727 539.3369 +/- 1.8309]
[114.6758 +/- 0.4777 41.6797 +/- 0.6051 556.8783 +/- 2.0105]
[88.7685 +/- 0.4210 90.9182 +/- 0.5316 494.2845 +/- 1.7489]

[53.9191 +/- 0.4205	40.9917 +/- 0.5278	484.8456 +/- 1.7683]
[145.0586 +/- 0.3816	66.6888 +/- 0.4878	460.7869 +/- 1.6038]
[93.4762 +/- 0.3761	39.9689 +/- 0.4875	451.8492 +/- 1.5621]
[80.4392 +/- 0.3482	68.5073 +/- 0.4423	410.0927 +/- 1.4529]
[84.1385 +/- 0.5727	69.7984 +/- 0.7252	667.0181 +/- 2.4494]
[36.4667 +/- 0.4912	0.3751 +/- 0.6220	573.1633 +/- 2.0598]
[89.2566 +/- 0.4146	106.1776 +/- 0.5230	495.4692 +/- 1.6913]
[100.0227 +/- 0.4139	50.6485 +/- 0.5254	483.7036 +/- 1.7705]
[79.8624 +/- 0.3872	64.2443 +/- 0.4888	460.6280 +/- 1.4878]
[64.8680 +/- 0.4756	65.0443 +/- 0.5904	547.8949 +/- 1.9632]
[93.5285 +/- 0.4042	80.1241 +/- 0.5084	479.1034 +/- 1.6615]
[76.9006 +/- 0.3785	-11.5440 +/- 0.4778	440.2027 +/- 1.5679]
[74.0567 +/- 0.3537	42.4776 +/- 0.4334	401.2955 +/- 1.5124]
[77.2469 +/- 0.4089	123.1866 +/- 0.5167	484.6958 +/- 1.6176]

B.3 Finding Out the Centre of the Circles

```
[1] %Finding out the centre of circles with calibration
[2] %-----

[3] clc;
[4] clearvars; % Get rid of variables from prior run of this m-file.
[5] imtool close all; % Close all imtool figures.
[6] fprintf('Running the programme...\n'); % Message sent to command
    window.
[7] workspace; % Make sure the workspace panel with all the variables is
    showing.
[8] imtool close all; % Close all imtool figures.
[9] format long g;
[10] format compact;
[11] captionFontSize = 14;

[12] % Define images to process
[13] imageFileNames = {'F:\20160702_205334.jpg',...
[14] 'F:\20160702_205356.jpg',...
[15] 'F:\20160702_205404.jpg',...
[16] 'F:\20160702_205410.jpg',...
[17] 'F:\20160702_205417.jpg',...
[18] 'F:\20160702_205423.jpg',...
[19] 'F:\20160702_205427.jpg',...
[20] 'F:\20160702_205433.jpg',...
[21] 'F:\20160702_205437.jpg',...
[22] 'F:\20160702_205442.jpg',...
[23] 'F:\20160702_205446.jpg',...
[24] 'F:\20160702_205449.jpg',...
[25] 'F:\20160702_205504.jpg',...
[26] 'F:\20160702_205510.jpg',...
[27] 'F:\20160702_205520.jpg',...
[28] 'F:\20160702_205547.jpg',...
[29] 'F:\20160702_205623.jpg',...
[30] 'F:\20160702_205654.jpg',...
[31] 'F:\20160702_205738.jpg',...
[32] 'F:\20160702_205743.jpg',...
[33] };

[34] % Detect checkerboards in images
[35] [imagePoints, boardSize, imagesUsed] =
    detectCheckerboardPoints(imageFileNames);
[36] imageFileNames = imageFileNames(imagesUsed);

[37] % Generate world coordinates of the corners of the squares
[38] squareSize = 25; % in units of 'mm'
[39] worldPoints = generateCheckerboardPoints(boardSize, squareSize);

[40] % Calibrate the camera
[41] [cameraParams, imagesUsed, estimationErrors] =
    estimateCameraParameters(imagePoints, worldPoints, ...
[42] 'EstimateSkew', true, 'EstimateTangentialDistortion', false, ...
[43] 'NumRadialDistortionCoefficients', 2, 'WorldUnits', 'mm');

[44] % View reprojection errors
[45] %h1=figure; showReprojectionErrors(cameraParams, 'BarGraph');
```

```

[46] % Visualize pattern locations
[47] h2=figure; showExtrinsics(cameraParams, 'PatternCentric');

[48] % Display parameter estimation errors
[49] displayErrors(estimationErrors, cameraParams);

[50] % For example, we can use the calibration data to remove effects of
    lens distortion.
[51] %originalImage = imread(imageFileNames{17});
[52] %undistortedImage = undistortImage(originalImage, cameraParams);
[53] imOrig=imread('F:\20160702_205701.jpg');
[54] %imshow(undistortedImage);

[55] % Display one of the calibration images
[56] magnification = 25;

[57] subplot(3,3,1);
[58] imshow(imOrig, 'InitialMagnification', magnification);
[59] title('Input Image');

[60] [im,newOrigin]=undistortImage(imOrig, cameraParams);
[61] subplot(3,3,2);
[62] imshow(im);
[63] title('Undistorted Image');

[64] im=rgb2gray(im);
[65] subplot(3,3,3);
[66] imshow(im, 'InitialMagnification', magnification);
[67] title('GreyScale Image');

[68] %%binaryImage = (im>50 & im<170);
[69] %%imCoin = imfill(binaryImage, 'holes');
[70] %%labeledImage = bwlabel(binaryImage, 8);

[71] %%subplot(3,3,4);
[72] %%imshow(imCoin, 'InitialMagnification', magnification);
[73] %%title('Segmented Coins');

[74] %Histogram
[75] %[pixelCount, grayLevels] = imhist(im);
[76] %subplot(3, 3, 4);
[77] %bar(pixelCount);
[78] %title('Histogram of original image');
[79] %xlim([0 grayLevels(end)]);
[80] %grid on;

[81] %thresholdValue = 100;
[82] binaryImage = im<50; % Bright objects will be chosen if you use >.

[83] binaryImage = imfill(binaryImage, 'holes');
[84] %hold on;
[85] %maxYValue = ylim;
[86] %line([thresholdValue, thresholdValue], maxYValue, 'Color', 'r');
[87] % Place a text label on the bar chart showing the threshold.
[88] %annotationText = sprintf('Thresholded at %d gray levels',
    thresholdValue);
[89] % For text(), the x and y need to be of the data class "double" so
    let's cast both to double.
[90] %text(double(thresholdValue + 5), double(0.5 * maxYValue(2)),
    annotationText, 'FontSize', 10, 'Color', [0 .5 0]);

```

```

[91] %text(double(thresholdValue - 70), double(0.94 * maxYValue(2)),
      'Background', 'FontSize', 10, 'Color', [0 0 .5]);
[92] %text(double(thresholdValue + 50), double(0.94 * maxYValue(2)),
      'Foreground', 'FontSize', 10, 'Color', [0 0 .5]);

[93] % Display the binary image.
[94] subplot(3, 3, 4);
[95] imshow(binaryImage);
[96] title('Binary Image, obtained by thresholding');

[97] %-----
-----

[98] labeledImage = bwlabel(binaryImage, 8);      % Label each blob so we
      can make measurements of it
[99] %labeledImage is an integer-valued image where all pixels in the
      blobs have values of 1, or 2, or 3, or ... etc.
[100] %subplot(3, 3, 5);
[101] %imshow(labeledImage, []); % Show the gray scale image.
[102] %title('Labeled Image, from bwlabel()');

[103] % Let's assign each blob a different color to visually show the user
      the distinct blobs.
[104] coloredLabels = label2rgb (labeledImage, 'hsv', 'k', 'shuffle'); %
      pseudo random color labels
[105] % coloredLabels is an RGB image. We could have applied a colormap
      instead (but only with R2014b and later)
[106] subplot(3, 3, 5);
[107] imshow(coloredLabels);
[108] axis image; % Make sure image is not artificially stretched because
      of screen's aspect ratio.
[109] caption = sprintf('Pseudo colored labels');
[110] title(caption);

[111] % Get all the blob properties. Can only pass in originalImage in
      version R2008a and later.
[112] blobMeasurements = regionprops(labeledImage, im, 'all');
[113] numberOfBlobs = size(blobMeasurements, 1);

[114] %-----
-----

[115] % bwboundaries() returns a cell array, where each cell contains the
      row/column coordinates for an object in the image.
[116] % Plot the borders of all the coins on the original grayscale image
      using the coordinates returned by bwboundaries.
[117] subplot(3, 3, 6);
[118] imshow(im);
[119] title('Outlines, from bwboundaries()', 'FontSize', captionFontSize);
[120] axis image; % Make sure image is not artificially stretched because
      of screen's aspect ratio.
[121] hold on;
[122] boundaries = bwboundaries(binaryImage);
[123] numberOfBoundaries = size(boundaries, 1);
[124] for k = 1 : numberOfBoundaries
[125] thisBoundary = boundaries{k};
[126] plot(thisBoundary(:,2), thisBoundary(:,1), 'g', 'LineWidth', 2);
[127] end
[128] hold off;

```

```

[129] textFontSize = 14; % Used to control size of "blob number" labels
      put atop the image.
[130] labelShiftX = -7; % Used to align the labels in the centers of the
      coins.
[131] blobECD = zeros(1, numberOfBlobs);
[132] % Print header line in the command window.
[133] fprintf(1, 'Blob #      Mean Intensity Area Perimeter Centroid
      Diameter\n');
[134] % Loop over all blobs printing their measurements to the command
      window.
[135] i=0;
[136] for k = 1 : numberOfBlobs % Loop through all blobs.
[137] % Find the mean of each blob. (R2008a has a better way where you
      can pass the original image
[138] % directly into regionprops. The way below works for all versions
      including earlier versions.)
[139] thisBlobsPixels = blobMeasurements(k).PixelIdxList; % Get list of
      pixels in current blob.
[140] meanGL = mean(im(thisBlobsPixels)); % Find mean intensity (in
      original image!)
[141] meanGL2008a = blobMeasurements(k).MeanIntensity; % Mean again, but
      only for version >= R2008a

[142] blobArea = blobMeasurements(k).Area; % Get area.
[143] blobPerimeter = blobMeasurements(k).Perimeter; % Get perimeter.
[144] blobCentroid = blobMeasurements(k).Centroid; % Get centroid
      one at a time
[145] blobECD(k) = sqrt(4 * blobArea / pi); % Compute
      ECD - Equivalent Circular Diameter.
[146] if(blobArea<30000 & blobArea>18000 & blobPerimeter<30000)
[147] i=i+1;
[148] fprintf(1, '%#2d %17.1f %11.1f %8.1f %8.1f %8.1f % 8.1f\n', k,
      meanGL, blobArea, blobPerimeter, blobCentroid, blobECD(k));
[149] % Put the "blob number" labels on the "boundaries" grayscale image.
[150] text(blobCentroid(1) + labelShiftX, blobCentroid(2), num2str(k),
      'FontSize', textFontSize, 'FontWeight', 'Bold');
[151] blobMeasurements(i)=blobMeasurements(k);
[152] end
[153] end

[154] allBlobCentroids = [blobMeasurements.Centroid];
[155] centroidsX = allBlobCentroids(1:2:end-1);
[156] centroidsY = allBlobCentroids(2:2:end);

[157] x=[centroidsX;centroidsY]; %combining centroid matrix as Mx2 matrix
[158] combinedCentroids=transpose(x);

[159] %Distance in between blobs-----
[160] for j=1:i-1
[161] fprintf('Distance in between blobs ');
[162] deltax=(centroidsX(j)-centroidsX(j+1));
[163] deltay=(centroidsY(j)-centroidsY(j+1));
[164] distance=hypot(deltax,deltay);
[165] fprintf('%d & %d is %f pixels\n',j,j+1,distance);
[166] end

[167] %-----

[168] % Detect the checkerboard.
[169] %corimage=imread(imageFileNames{17});

```

```
[170] [imagePoints, boardSize] =  
      detectCheckerboardPoints(imageFileNames{18});  
  
[171] % Compute rotation and translation of the camera.  
[172] [R, t] = extrinsics(imagePoints, worldPoints, cameraParams);  
  
[173] % Get the world coordinates of the corners  
[174] worldPoints2 = pointsToWorld(cameraParams, R, t, combinedCentroids);
```


B.4 Timber Deflection Measurement

```
[1] %Analyse the digital camera (fixed on tripod) images experiment 2

[2] clc;
[3] clearvars; % Get rid of variables from prior run of this m-file.
[4] imtool close all; % Close all imtool figures.
[5] fprintf('Running the programme...\n'); % Message sent to command
window.
[6] workspace; % Make sure the workspace panel with all the variables is
showing.
[7] imtool close all; % Close all imtool figures.
[8] format long g;
[9] format compact;
[10] captionFontSize = 14;
[11] %-----

[12] % Define images to processa
[13] %Total Images = 35
[14] % Define images to process
[15] imageFileNames = {'F:\DSC03297.JPG',...
[16] 'F:\DSC03299.JPG',...
[17] 'F:\DSC03300.JPG',...
[18] 'F:\DSC03302.JPG',...
[19] 'F:\DSC03305.JPG',...
[20] 'F:\DSC03306.JPG',...
[21] 'F:\DSC03307.JPG',...
[22] 'F:\DSC03308.JPG',...
[23] 'F:\DSC03311.JPG',...
[24] 'F:\DSC03312.JPG',...
[25] 'F:\DSC03313.JPG',...
[26] 'F:\DSC03316.JPG',...
[27] 'F:\DSC03317.JPG',...
[28] 'F:\DSC03318.JPG',...
[29] 'F:\DSC03320.JPG',...
[30] 'F:\DSC03322.JPG',...
[31] 'F:\DSC03325.JPG',...
[32] 'F:\DSC03326.JPG',...
[33] 'F:\DSC03327.JPG',...
[34] 'F:\DSC03328.JPG',...
[35] 'F:\DSC03335.JPG',...
[36] };

[37] %Image 26 is compared for scaling

[38] % Detect checkerboards in images
[39] [imagePoints, boardSize, imagesUsed] =
detectCheckerboardPoints(imageFileNames);
[40] imageFileNames = imageFileNames(imagesUsed);

[41] % Generate world coordinates of the corners of the squares
[42] squareSize = 2.17943e+01; % in units of 'mm'
[43] worldPoints = generateCheckerboardPoints(boardSize, squareSize);

[44] % Calibrate the camera
[45] [cameraParams, imagesUsed, estimationErrors] =
estimateCameraParameters(imagePoints, worldPoints, ...
```

```

[46] 'EstimateSkew', true, 'EstimateTangentialDistortion', true, ...
[47] 'NumRadialDistortionCoefficients', 2, 'WorldUnits', 'mm');

[48] % View reprojection errors
[49] h1=figure; showReprojectionErrors(cameraParams, 'BarGraph');

[50] % Visualize pattern locations
[51] %h2=figure; showExtrinsics(cameraParams, 'CameraCentric');

[52] % Display parameter estimation errors
[53] %displayErrors(estimationErrors, cameraParams);

[54] % For example, you can use the calibration data to remove effects of
    lens distortion.
[55] %originalImage = imread(imageFileNames{17});
[56] %undistortedImage = undistortImage(originalImage, cameraParams);
[57] imOrig=imread('F:\DSC03371.JPG');
[58] %imshow(undistortedImage);

[59] % Display one of the calibration images
[60] magnification = 25;

[61] subplot(3,3,1);
[62] imshow(imOrig, 'InitialMagnification', magnification);
[63] title('Input Image');

[64] [im,newOrigin]=undistortImage(imOrig, cameraParams);
[65] subplot(3,3,2);
[66] imshow(im);
[67] title('Undistorted Image');

[68] im=rgb2gray(im); %earlier it was not 'imOrig' but 'im'
[69] subplot(3,3,3);
[70] imshow(im, 'InitialMagnification', magnification);
[71] title('GreyScale Image');

[72] %%binaryImage = (im>50 & im<170);
[73] %%imCoin = imfill(binaryImage, 'holes');
[74] %%labeledImage = bwlabel(binaryImage, 8);

[75] %%subplot(3,3,4);
[76] %%imshow(imCoin, 'InitialMagnification', magnification);
[77] %%title('Segmented Coins');

[78] %Histogram
[79] %[pixelCount, grayLevels] = imhist(im);
[80] %subplot(3, 3, 4);
[81] %bar(pixelCount);
[82] %title('Histogram of original image');
[83] %xlim([0 grayLevels(end)]);
[84] %grid on;

[85] %thresholdValue = 100;
[86] binaryImage = im>130 & im<200; % Bright objects will be chosen if
    you use >.

[87] binaryImage = imfill(binaryImage, 'holes');
[88] %hold on;
[89] %maxYValue = ylim;
[90] %line([thresholdValue, thresholdValue], maxYValue, 'Color', 'r');
[91] % Place a text label on the bar chart showing the threshold.

```

```

[92] %annotationText = sprintf('Thresholded at %d gray levels',
thresholdValue);
[93] % For text(), the x and y need to be of the data class "double" so
let's cast both to double.
[94] %text(double(thresholdValue + 5), double(0.5 * maxYValue(2)),
annotationText, 'FontSize', 10, 'Color', [0 .5 0]);
[95] %text(double(thresholdValue - 70), double(0.94 * maxYValue(2)),
'Background', 'FontSize', 10, 'Color', [0 0 .5]);
[96] %text(double(thresholdValue + 50), double(0.94 * maxYValue(2)),
'Foreground', 'FontSize', 10, 'Color', [0 0 .5]);

[97] % Display the binary image.
[98] subplot(3, 3, 4);
[99] imshow(binaryImage);
[100] title('Binary Image, obtained by thresholding');

[101] %-----
-----

[102] labeledImage = bwlabel(binaryImage, 8); % Label each blob so we
can make measurements of it
[103] %labeledImage is an integer-valued image where all pixels in the
blobs have values of 1, or 2, or 3, or ... etc.
[104] %subplot(3, 3, 5);
[105] %imshow(labeledImage, []); % Show the gray scale image.
[106] %title('Labeled Image, from bwlabel()');

[107] % Let's assign each blob a different color to visually show the user
the distinct blobs.
[108] coloredLabels = label2rgb (labeledImage, 'hsv', 'k', 'shuffle'); %
pseudo random color labels
[109] % coloredLabels is an RGB image. We could have applied a colormap
instead (but only with R2014b and later)
[110] subplot(3, 3, 5);
[111] imshow(coloredLabels);
[112] axis image; % Make sure image is not artificially stretched because
of screen's aspect ratio.
[113] caption = sprintf('Pseudo colored labels');
[114] title(caption);

[115] % Get all the blob properties. Can only pass in originalImage in
version R2008a and later.
[116] blobMeasurements = regionprops(labeledImage, im, 'all');
[117] numberOfBlobs = size(blobMeasurements, 1);

[118] %-----
-----

[119] % bwboundaries() returns a cell array, where each cell contains the
row/column coordinates for an object in the image.
[120] % Plot the borders of all the coins on the original grayscale image
using the coordinates returned by bwboundaries.
[121] subplot(3, 3, 6);
[122] figure;imshow(im);
[123] title('Outlines, from bwboundaries()', 'FontSize', captionFontSize);
[124] axis image; % Make sure image is not artificially stretched because
of screen's aspect ratio.
[125] hold on;
[126] boundaries = bwboundaries(binaryImage);
[127] numberOfBoundaries = size(boundaries, 1);
[128] for k = 1 : numberOfBoundaries

```

```

[129] thisBoundary = boundaries{k};
[130] plot(thisBoundary(:,2), thisBoundary(:,1), 'g', 'LineWidth', 2);
[131] end
[132] hold off;

[133] textFontSize = 20; % Used to control size of "blob number" labels
put atop the image.
[134] labelShiftX = -7; % Used to align the labels in the centers of the
coins.
[135] blobECD = zeros(1, numberOfBlobs);
[136] % Print header line in the command window.
[137] fprintf(1, 'Blob #      Mean Intensity Area   Perimeter   Centroid
Diameter\n');
[138] % Loop over all blobs printing their measurements to the command
window.
[139] i=0;
[140] for k = 1 : numberOfBlobs % Loop through all blobs.
[141] % Find the mean of each blob. (R2008a has a better way where you
can pass the original image
[142] % directly into regionprops. The way below works for all versions
including earlier versions.)
[143] thisBlobsPixels = blobMeasurements(k).PixelIdxList; % Get list of
pixels in current blob.
[144] meanGL = mean(im(thisBlobsPixels)); % Find mean intensity (in
original image!)
[145] meanGL2008a = blobMeasurements(k).MeanIntensity; % Mean again, but
only for version >= R2008a

[146] blobArea = blobMeasurements(k).Area; % Get area.
[147] blobPerimeter = blobMeasurements(k).Perimeter; % Get perimeter.
[148] blobCentroid = blobMeasurements(k).Centroid;% Get centroid one at a
time
[149] blobECD(k) = sqrt(4 * blobArea / pi); % Compute
ECD - Equivalent Circular Diameter.
[150] if(blobArea<3500 & blobArea>2000 & blobPerimeter>100 &
blobPerimeter<240 & meanGL>100 &meanGL<210)
[151] i=i+1;
[152] fprintf(1, '#%2d %17.1f %11.1f %8.1f %8.1f %8.1f % 8.1f\n', k,
meanGL, blobArea, blobPerimeter, blobCentroid, blobECD(k));
[153] % Put the "blob number" labels on the "boundaries" grayscale image.
[154] text(blobCentroid(1) + labelShiftX, blobCentroid(2), num2str(k),
'FontSize', textFontSize, 'FontWeight', 'Bold');
[155] %allBlobCentroids = [blobMeasurements.Centroid];
[156] %centroidsX = allBlobCentroids(1:2:end-1);
[157] %centroidsY = allBlobCentroids(2:2:end);
[158] %plot(centroidsX(k), centroidsY(k), 'r+', 'MarkerSize', 10,
'LineWidth', 2);
[159] blobMeasurements(i)=blobMeasurements(k);
[160] end
[161] end

[162] allBlobCentroids = [blobMeasurements.Centroid];
[163] centroidsX = allBlobCentroids(1:2:end-1);
[164] centroidsY = allBlobCentroids(2:2:end);

[165] x=[centroidsX;centroidsY]; %combining centroid matrix as Mx2 matrix
[166] combinedCentroids=transpose(x);

[167] % Detect the checkerboard.
[168] %corimage=imread(imageFileNames{17});

```

```

[169] [imagePoints, boardSize] =
        detectCheckerboardPoints(imageFileNames{21});

[170] % Compute rotation and translation of the camera.
[171] [R, t] = extrinsics(imagePoints, worldPoints, cameraParams);

[172] % Get the world coordinates of the corners
[173] worldPoints1 = pointsToWorld(cameraParams, R, t, combinedCentroids);

[174] worldPoints1(1:i,:)

[175] % blobAnalysis = vision.BlobAnalysis('AreaOutputPort', true,...
[176] %     'CentroidOutputPort', false,...
[177] %     'BoundingBoxOutputPort', true,...
[178] %     'MinimumBlobArea', 2000, 'MaximumBlobArea', 3000,
        'ExcludeBorderBlobs', true);
[179] % [areas, boxes] = step(blobAnalysis, binaryImage);
[180] %
[181] % % Sort connected components in descending order by area
[182] % [~, idx] = sort(areas, 'Descend');
[183] %
[184] % % Get the two largest components.
[185] % boxes = double(boxes(idx(1:2), :));
[186] %
[187] % % Adjust for coordinate system shift caused by undistortImage
[188] % boxes(:, 1:2) = bsxfun(@plus, boxes(:, 1:2), newOrigin);
[189] %
[190] % % Reduce the size of the image for display.
[191] % magnification=25,
[192] % scale = magnification / 100;
[193] % imDetectedCoins = imresize(im, scale);
[194] %
[195] % % Insert labels for the coins.
[196] % imDetectedCoins = insertObjectAnnotation(imDetectedCoins,
        'rectangle', ...
[197] %     scale * boxes, 'blob');
[198] % figure; imshow(imDetectedCoins);
[199] % title('Detected blobss');
[200] %
[201] % % Detect the checkerboard.
[202] % [imagePoints, boardSize] =
        detectCheckerboardPoints(imageFileNames{21});
[203] %
[204] % % Compute rotation and translation of the camera.
[205] % [R, t] = extrinsics(imagePoints, worldPoints, cameraParams);
[206] %
[207] % % Get the top-left and the top-right corners.
[208] % box1 = double(boxes(1, :));
[209] % imagePoints1 = [box1(1:2); ...
[210] %     box1(1) + box1(3), box1(2)];
[211] %
[212] % % Get the world coordinates of the corners
[213] % worldPoints1 = pointsToWorld(cameraParams, R, t, imagePoints1);
[214] %
[215] % % Compute the diameter of the coin in millimeters.
[216] % d = worldPoints1(2, :) - worldPoints1(1, :);
[217] % diameterInMillimeters = hypot(d(1), d(2));
[218] % fprintf('Measured diameter of one penny = %0.2f mm\n',
        diameterInMillimeters);
[219] %
[220] % % Get the top-left and the top-right corners.

```

```

[221] % box2 = double(boxes(2, :));
[222] % imagePoints2 = [box2(1:2); ...
[223] %             box2(1) + box2(3), box2(2)];
[224] %
[225] % % Apply the inverse transformation from image to world
[226] % worldPoints2 = pointsToWorld(cameraParams, R, t, imagePoints2);
[227] %
[228] % % Compute the diameter of the coin in millimeters.
[229] % d = worldPoints2(2, :) - worldPoints2(1, :);
[230] % diameterInMillimeters = hypot(d(1), d(2));
[231] % fprintf('Measured diameter of the other penny = %0.2f mm\n',
diameterInMillimeters);
[232] %
[233] % % Compute the center of the first coin in the image.
[234] % center1_image = box1(1:2) + box1(3:4)/2;
[235] %
[236] % % Convert to world coordinates.
[237] % center1_world = pointsToWorld(cameraParams, R, t, center1_image);
[238] %
[239] % % Remember to add the 0 z-coordinate.
[240] % center1_world = [center1_world 0];
[241] %
[242] % % Compute the distance to the camera.
[243] % distanceToCamera = norm(center1_world + t);
[244] % fprintf('Distance from the camera to the first penny = %0.2f
mm\n', ...
distanceToCamera);
[245] %

```

B.5: Template Matching Algorithm

```
[1] % % Find maximum response
[2] I = im2double(imread('lena.jpg'));
[3] I=imrotate(I,30); %angle of rotation is 30 degrees

[4] % % Template of Eye Lena
[5] T=I(124:200,124:200,:);

[6] % % Calculate SSD and NCC between Template and Image
[7] [I_SSD,I_NCC]=template_matching(T,I);

[8] % % Find maximum correspondence in I_SDD image
[9] [x,y]=find(I_SSD==max(I_SSD(:)));

[10] % % Show result
[11] figure,
[12] subplot(2,2,1), imshow(I); hold on; plot(y,x,'r*'); title('Result');
[13] subplot(2,2,2), imshow(T); title('The eye template');
[14] subplot(2,2,3), imshow(I_SSD); title('SSD Matching');
[15] subplot(2,2,4), imshow(I_NCC); title('Normalized-CC');
```

B.6 Pattern Recognition Algorithm

```
[1]  clc;clear;
[2]  clearvars; % Get rid of variables from prior run of this m-file.
[3]  imtool close all; % Close all imtool figures.
[4]  fprintf('Running the programme...\n'); % Message sent to command
      window.
[5]  workspace; % Make sure the workspace panel with all the variables is
      showing.
[6]  format long g;
[7]  format compact;
[8]  captionFontSize = 14;
[9]  %-----

[10] %Camera Calibration

[11] % Define images to process % overall mean error 2.24 pixel
[12] imageFileNames = {'F:\IMG_3510.JPG',...
[13] 'F:\IMG_3511.JPG',...
[14] 'F:\IMG_3512.JPG',...
[15] 'F:\IMG_3530.JPG',...
[16] 'F:\IMG_3531.JPG',...
[17] 'F:\IMG_3532.JPG',...
[18] 'F:\IMG_3533.JPG',...
[19] 'F:\IMG_3536.JPG',...
[20] 'F:\IMG_3539.JPG',...
[21] 'F:\IMG_3540.JPG',...
[22] 'F:\IMG_3541.JPG',...
[23] 'F:\IMG_3542.JPG',...
[24] 'F:\IMG_3547.JPG',...
[25] 'F:\IMG_3552.JPG',...
[26] 'F:\IMG_3556.JPG',...
[27] 'F:\IMG_3559.JPG',...
[28] 'F:\IMG_3560.JPG',...
[29] 'F:\IMG_3561.JPG',...
[30] 'F:\IMG_3564.JPG',...
[31] 'F:\IMG_3565.JPG',...
[32] 'F:\IMG_3566.JPG',...
[33] 'F:\IMG_3567.JPG',...
[34] 'F:\IMG_3569.JPG',...
[35] 'F:\IMG_3570.JPG',...
[36] 'F:\IMG_3571.JPG',...
[37] 'F:\IMG_3572.JPG',...
[38] 'F:\IMG_3574.JPG',...
[39] 'F:\IMG_3575.JPG',...
[40] 'F:\IMG_3578.JPG',...
[41] 'F:\IMG_3579.JPG',...
[42] 'F:\IMG_3580.JPG',...
[43] 'F:\IMG_3581.JPG',...
[44] 'F:\IMG_3582.JPG',...
[45] 'F:\IMG_3583.JPG',...
[46] 'F:\IMG_3588.JPG',...
[47] };

[48] % Detect checkerboards in images
[49] [imagePoints, boardSize, imagesUsed] =
      detectCheckerboardPoints(imageFileNames);
[50] imageFileNames = imageFileNames(imagesUsed);
```



```

[51] % Generate world coordinates of the corners of the squares
[52] squareSize = 13.64; % in units of 'mm'
[53] worldPoints = generateCheckerboardPoints(boardSize, squareSize);

[54] % Calibrate the camera
[55] [cameraParams, imagesUsed, estimationErrors] =
estimateCameraParameters(imagePoints, worldPoints, ...
[56] 'EstimateSkew', false, 'EstimateTangentialDistortion', false, ...
[57] 'NumRadialDistortionCoefficients', 2, 'WorldUnits', 'mm');
[58] % Display parameter estimation errors
[59] %displayErrors(estimationErrors, cameraParams);

[60] %Read the image and process them (only 2 for the moment)
[61] clc;
[62] Im1=rgb2gray(imread('IMG_3513.jpg'));
[63] Im2=rgb2gray(imread('IMG_3525.jpg'));

[64] [Im1,newOrigin]=undistortImage(Im1, cameraParams);
[65] [Im2,newOrigin]=undistortImage(Im2, cameraParams);

[66] % Detect the checkerboard.
[67] %corresponding plane image : (imageFileNames{8});
[68] [imagePoints, boardSize] =
detectCheckerboardPoints(imageFileNames{5});

[69] % Compute rotation and translation of the camera.
[70] [R, t] = extrinsics(imagePoints, worldPoints, cameraParams);

[71] %Im2=rgb2gray(Im2);
[72] %Im1=rgb2gray(Im1);
[73] Im1=Im1(1049:1792,2805:2996,:);
[74] x1=2805;
[75] y1=1049;
[76] x2=2996;
[77] y2=1792;
[78] P=[x1 y1;x1 y2]; %Vertical line in the image
[79] P= pointsToWorld(cameraParams, R, t,P); % World coordinate of P
[80] theta=atan((P(2,2)-P(1,2))/(P(2,1)-P(1,1)));
[81] % Angle theta (in radians) to the horizontal of the vertical line in
world coordinate

[82] %Im2=imrotate(Im2,1); %Angle is anticlockwise
[83] a=size(Im1);
[84] i=1;x=1;y=1;j=1;
[85] s=60; %Size of the template sxs
[86] %--k=1;

[87] for (x=1:s:(a(2)-s-1))   %%s/8
[88] j=1;
[89] for (y=1:s:(a(1)-s-1))   %%s/8
[90] boxImage=Im1(y:y+s-1,x:x+s-1,:); %x=y coordiante, y=x coordinate
[91] sceneImage=Im2;
[92] boxPoints = detectSURFFeatures(boxImage);
[93] dimbP=size(boxPoints);

[94] if (dimbP(1,1)>2) % at least 3 unique points
[95] % for (m=1:2s:(b(1)-4s))
[96] %     n=1
[97] %     for (n=1:2s:(b(2)-4s))
[98] %         sceneImage=Im2(m:m+4s,n:n+4s,:);
[99] scenePoints = detectSURFFeatures(sceneImage);

```

```

[100] %figure;
[101] %imshow(boxImage);
[102] %title('200 Strongest Feature Points from Box Image');
[103] %hold on;
[104] %plot(selectStrongest(boxPoints, 100));

[105] %figure;
[106] %imshow(sceneImage);
[107] %title('5000 Strongest Feature Points from Scene Image');
[108] %hold on;
[109] %plot(selectStrongest(scenePoints, 5000));

[110] %Extract Feature Descriptors
[111] [boxFeatures, boxPoints] = extractFeatures(boxImage, boxPoints);
[112] [sceneFeatures, scenePoints] = extractFeatures(sceneImage,
    scenePoints);

[113] %Find Putative Point Matches
[114] boxPairs = matchFeatures(boxFeatures, sceneFeatures);
[115] dimbPs=size(boxPairs);

[116] if(dimbPs(1,1)>2)

[117] matchedBoxPoints = boxPoints(boxPairs(:, 1), :);
[118] matchedScenePoints = scenePoints(boxPairs(:, 2), :);

[119] %figure;
[120] %showMatchedFeatures(boxImage, sceneImage, matchedBoxPoints, ...
[121] %matchedScenePoints, 'montage');
[122] %title('Putatively Matched Points (Including Outliers)');

[123] [tform, inlierBoxPoints, inlierScenePoints] = ...
[124] estimateGeometricTransform(matchedBoxPoints, matchedScenePoints,
    'affine'); %affine is the original

[125] %figure;
[126] %showMatchedFeatures(boxImage, sceneImage, inlierBoxPoints, ...
[127] %inlierScenePoints, 'montage');
[128] %title('Matched Points (Inliers Only)');

[129] boxPolygon = [1, 1;...           % top-left
[130] size(boxImage, 2), 1;...         % top-right
[131] size(boxImage, 2), size(boxImage, 1);... % bottom-right
[132] 1, size(boxImage, 1);...        % bottom-left
[133] 1, 1]; % top-left again to close the polygon

[134] newBoxPolygon = transformPointsForward(tform, boxPolygon);

[135] % Get the world coordinates of the corners
[136] newBoxPolygon = pointsToWorld(cameraParams, R, t, newBoxPolygon);

[137] %figure;
[138] %imshow(sceneImage);
[139] %hold on;
[140] %line(newBoxPolygon(:, 1), newBoxPolygon(:, 2), 'Color', 'r');
[141] %title('Detected Box');

[142] %Finding centroid of the detected area
[143] centroidsX(i,j)=(newBoxPolygon(1,1)+newBoxPolygon(3,1)+newBoxPolygon
    (2,1)+newBoxPolygon(4,1))/4;

```

```

[144] centroidsY(i,j)=(newBoxPolygon(1,2)+newBoxPolygon(3,2)+newBoxPolygon
      (2,2)+newBoxPolygon(4,2))/4;

[145] initialCentroid=[x1+(x-1)+s/2 y1+(y-1)+s/2];
[146] initialCentroid = pointsToWorld(cameraParams, R, t,initialCentroid);
[147] centroidsX1(i,j)=initialCentroid(1,1);
[148] centroidsY1(i,j)=initialCentroid(1,2);
[149] % Calculate SSD and NCC between Template and Image
[150] %--
      [sceneImage_SSD,sceneImage_NCC]=template_matching(boxImage,sceneImage);
[151] % Find maximum correspondence in I_SDD image
[152] %--[x1,y1]=find(sceneImage_SSD==max(sceneImage_SSD(:)));
[153] %--else
[154] %--end
[155] %--centroidsX(i,j)=y1;
[156] %--centroidsY(i,j)=x1;
[157] %disp(x1);
[158] %disp(y1);%figure;imshow(sceneImage);title('image');plot(x1,y1,'r*')
      ;
[159] %--k=k+1;
[160] j=j+1;
[161] %y=y+s/2;
[162] end
[163] end
[164] end

[165] i=i+1;
[166] %x=x+s/2;
[167] end

[168] fprintf('First loop completed...\n');
[169] dimTemplate=size(centroidsX);
[170] T1=dimTemplate(1,1);
[171] T2=dimTemplate(1,2);
[172] i=1;j=1;
[173] bmw=centroidsX;
[174] while(j<(T2+1)) %Rearrange the matix
[175] fprintf('loop1 \n');
[176] while(i<T1)
[177] %fprintf('loop2 \n');
[178] if(centroidsX(i,j)==0)
[179] if(centroidsX(i+1,j)~=0)
[180] fprintf('analysing %d %d \n',i,j);
[181] centroidsX(i,j)=centroidsX(i+1,j);
[182] centroidsX1(i,j)=centroidsX1(i+1,j);
[183] centroidsY(i,j)=centroidsY(i+1,j);
[184] centroidsY1(i,j)=centroidsY1(i+1,j);
[185] centroidsX(i+1,j)=0;
[186] centroidsX1(i+1,j)=0;
[187] centroidsY(i+1,j)=0;
[188] centroidsY1(i+1,j)=0;
[189] i=0;
[190] end
[191] end
[192] i=i+1;
[193] end
[194] j=j+1;
[195] i=1;
[196] fprintf('i,j = %d %d',i,j)
[197] end

```

```

[198] m=2;n=1;
[199] z=0;X=0;Y=0;Z=0;
[200] a=2;
[201] %Finding out the deformation and centroid of connecting lines

[202] for(m=2:1:T1)
[203] b=1;
[204] for(n=1:1:T2)
[205] if(centroidsX(m,n)~=0)
[206] strain(a,b) = (-1)*(sqrt(((centroidsY1(m,n)-centroidsY1(m-1,n))^2)+((centroidsX1(m,n)-centroidsX1(m-1,n))^2))-sqrt(((centroidsY(m,n)-centroidsY(m-1,n))^2)+((centroidsX(m,n)-centroidsX(m-1,n))^2)))/sqrt(((centroidsY1(m,n)-centroidsY1(m-1,n))^2)+((centroidsX1(m,n)-centroidsX1(m-1,n))^2));
[207] centroidLineY(a,b)=(centroidsY1(m,n)+centroidsY1(m-1,n))/2;
[208] centroidLineX(a,b)=(centroidsX1(m,n)+centroidsX1(m-1,n))/2;
[209] %if(abs(centroidsLineY(a,b)-centroidsLineY(a-1,b))>2s) % maximum strain 300%

[210] if(strain(a,b)~=0)
[211] Z(a-1,b)=strain(a,b);
[212] fprintf('\n strain(%d,%d) = %3f, centroidsY(%d,%d)-centroidsY(%d-1,%d) = %3f, centroidsY1(m,n)-centroidsY1(m-1,n) = %3f', a,b, strain(a,b), m,n,m,n, (centroidsY(m,n)-centroidsY(m-1,n)), (centroidsY1(m,n)-centroidsY1(m-1,n)));
[213] %if(m==2 & n==2)
[214] %Y=[centroidLineY(m,n-1) centroidLineY(m+1,n-1) centroidLineY(m+1,n) centroidLineY(m+1,n+1) centroidLineY(m,n+1) centroidLineY(m-1,n+1) centroidLineY(m-1,n) centroidLineY(m-1,n-1)];
[215] %X=[centroidLineX(m,n-1) centroidLineX(m+1,n-1) centroidLineX(m+1,n) centroidLineX(m+1,n+1) centroidLineX(m,n+1) centroidLineX(m-1,n+1) centroidLineX(m-1,n) centroidLineX(m-1,n-1)];
[216] %else

[217] Y=[Y centroidLineY(a,b)];
[218] X=[X centroidLineX(a,b)];
[219] b=b+1;
[220] end
[221] end
[222] end
[223] a=a+1;
[224] end
[225] m=1;
[226] B=(reshape(Z,[],1)); %convert Z matrix into row matrix
[227] p=size(B);
[228] for(k=1:1:p(1,1))
[229] if(B(k,1)~=0)
[230] A(m,1)=B(k,1);
[231] m=m+1;
[232] end
[233] end

[234] X=(X(2:end))';
[235] Y=(Y(2:end))';

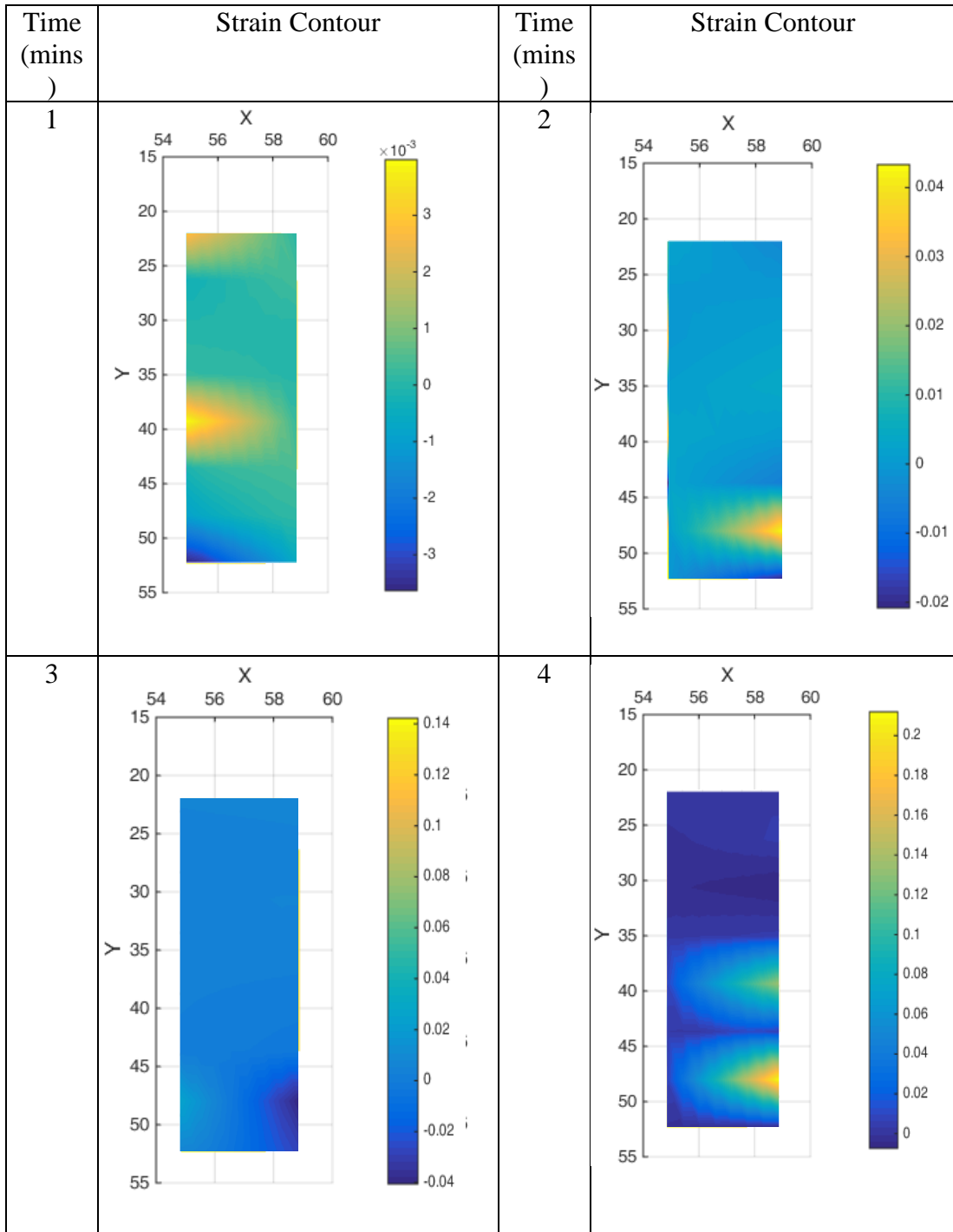
[236] n=10; %number of subdivisions for plotting purposes
[237] [xi,yi]=meshgrid(linspace(min(X),max(X),n),linspace(min(Y),max(Y),n)); %set limits for x and y
[238] %[xi,yi]=meshgrid(min(X):0.1:max(X),min(Y):0.1:max(Y));%

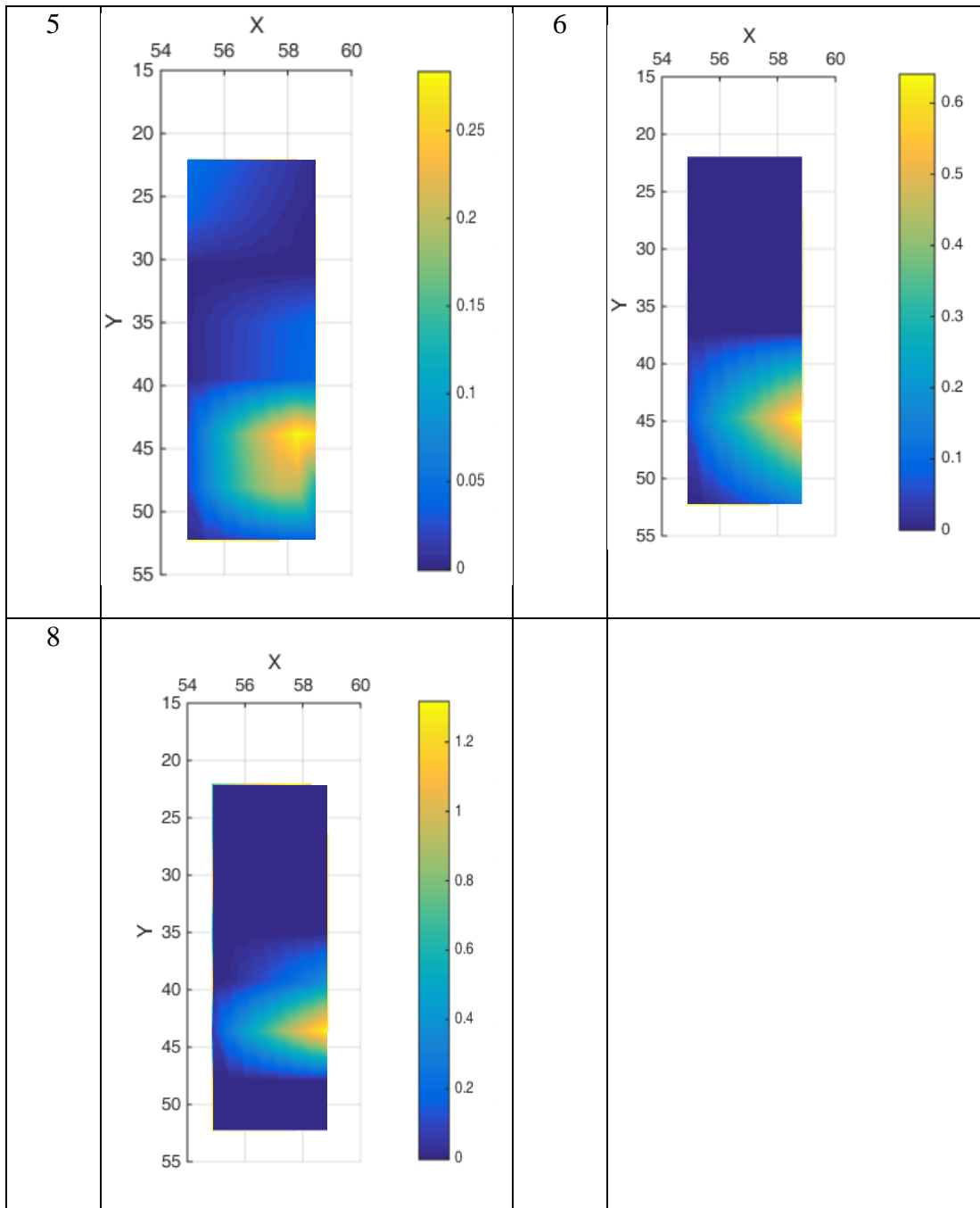
```

```
[239] zi=griddata(X,Y,A,xi,yi,'natural'); %interpolate Z to generate zi
      for complete grid
[240] %contour(xi,yi,zi);
[241] %figure;
[242] surf(xi,yi,zi); shading('interp');
[243] %scatter(xi,yi,zi);
[244] xlabel('X'); ylabel('Y'); zlabel('Z');
```

Appendix C: Strain Contours

Specimen 1:





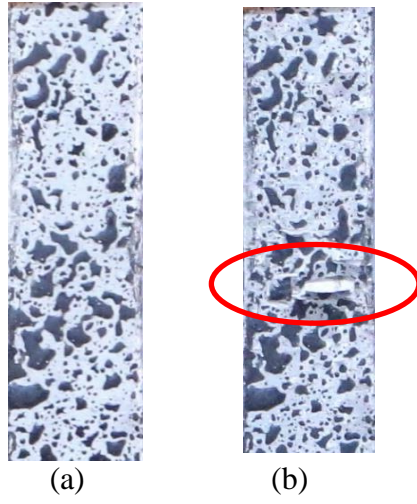
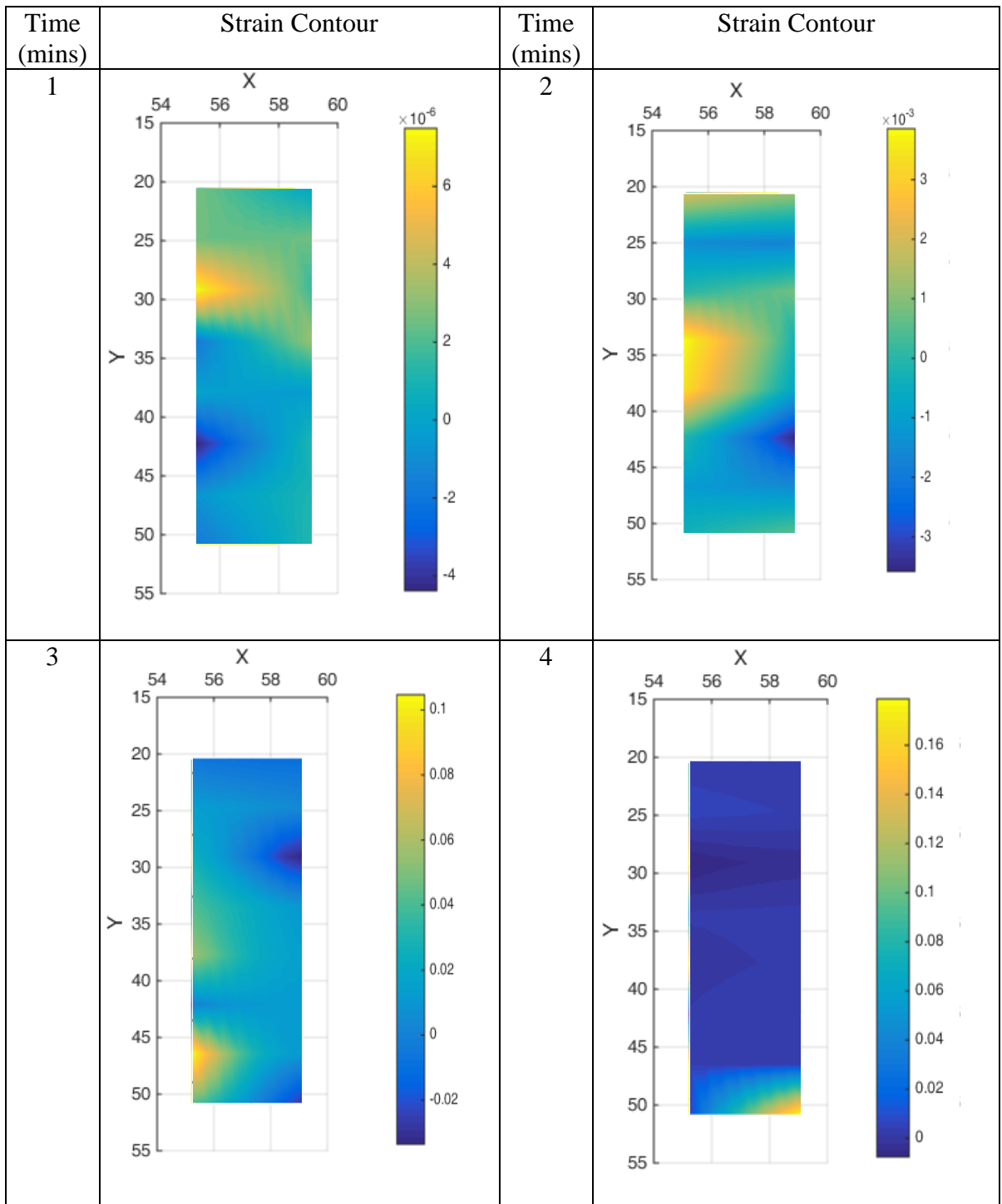
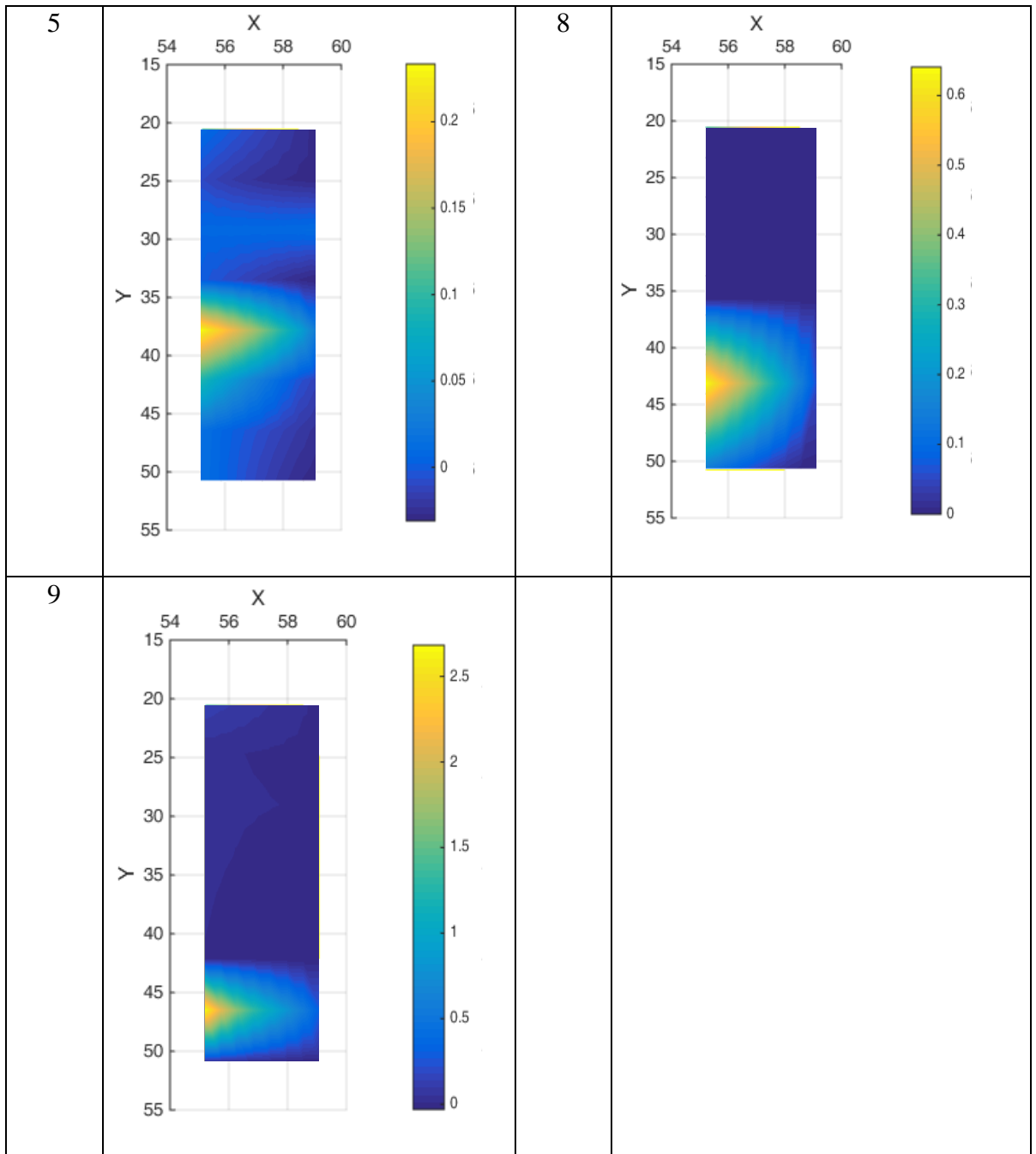


Figure 0.1: ROI (a) before loading (b) during necking

Specimen 2:





(a) (b)

Figure 0.2: ROI (a) before loading (b) during necking