The Pennsylvania State University

The Graduate School

College of Engineering

QUANTITATIVE RADIOGRAPHIC INVESTIGATION OF MECHANICAL FASTENERS IN WOOD CONNECTIONS

A Thesis in

Civil Engineering

by

Andrew R. Adams

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2009

The thesis of Andrew R. Adams was reviewed and approved* by the following:

Bohumil Kasal
Hankin Chair of Residential Building Construction
Professor of Civil and Environmental Engineering
Professor of Architectural Engineering
Thesis Advisor

Maria Lopez de Murphy
Will Assistant Professor of Civil Engineering

Charles Bakis
Distinguished Professor of Engineering Science & Mechanics

Peggy Johnson
Department Head, Civil and Environmental Engineering
Professor of Civil Engineering

*Signatures are on file in the Graduate School

ABSTRACT

This project deals with the use of digital x-ray imaging and image processing to quantitatively determine information about the strain behavior of dowels in wood connections. The developed method uses image processing techniques to extract the edge/boundary of the deflected dowel from an x-ray image, and then calculates the bending strain from the edge data using beam theory principles. This newly developed method has applications both in laboratory settings and also when investigating structures in situ after severe loading events. Feasibility and effectiveness of the method were evaluated using simplified tests consisting of dowel specimens instrumented with strain gages, so that the accuracy of the calculated strain values from digital x-ray images could be checked. The experiments provided valuable data for the development of suitable strain extraction procedures, which have been found to provide inexact but nonetheless useful information about the behavior of dowels in wood connections. Overall, the method presented here forms a useful tool which can be used by researchers and engineers, and also establishes a solid, verified starting point for the development of more sophisticated and accurate methodologies to calculate strains from deflected edge data found by digital image processing.

Table of Contents

List of Tables

List of Figures

ACKNOWLEDGEMENT

The author would like to thank his project's advisor, Dr. Bohumil Kasal, as well as the members of his thesis committee — Dr. Maria Lopez de Murphy and Dr. Charles Bakis. Furthermore, the help of the following people who have assisted in this work is acknowledged: Paul Kremer for assisting with the laboratory experiments, Dr. Richard Tutwiler for his helpful suggestions dealing with image processing techniques, and also the other members of Dr. Kasal's research group (Maral Amini, Tanya Myers, Brian Wolfgang, and Robert Blass) and the Pennsylvania Housing Research Center secretaries at Penn State (Tracy DeSoto and Rhonda Decker) for their constant help and support.

Chapter 1. Introduction

For many years wood has been primarily used in the residential housing industry, where loads do not require the use of connections with dowel type fasteners which are loaded laterally. However, recent increases in the use of wood for light-commercial or industrial structures have caused significant research to be carried out to investigate the complex behavior of bolted timber connection. The focus on wood connections is due to the fact that in wood structures the connections are frequently the limiting factors in design.

The increase in the use of wood structures has been due to many factors, including economic aspects, the desire to use regional and renewable resources, and the excellent behavior of wood under seismic loading. This behavior is due to both the low mass density of wood and the potentially high ductility of wood connections. While this ductility is desirable from the standpoint of dissipating more energy, it is undesirable due to larger drifts and deflections which can cause serviceability issues.

In order to develop models which can be used to accurately and appropriately design wood connections, research has typically involved comparing mathematical models with experimental data. Unfortunately, it is difficult to measure or quantify many behaviors of the connection which are important, and therefore should be present in validated mathematical models. Load-displacement (slip) values and observed failure states can easily be determined during experimental testing and typically are the only parameters used to verify or validate models. Surface strains found through methods such as digital image correlation can also be compared, however this only yields relevant strain information for the exterior bold holes of the side member.

As noted by Humphrey and Ostman [1],[2], observing the behavior of joints during loading may help understand the complex interactions which take place at the dowel-wood interface. Humphrey and Ostman observed the loaded behavior of doweled connections by taking x-ray images during loading. The bolt deflection was determined at various stages of loading, and when combined with results of dowel bearing tested "wafers" of wood, the reactions against the bolt were estimated.

The work outlined by Humphrey and Ostman [1],[2] in the late 1980s appears to have not garnered much attention, due to several possible reasons: the difficulties in replicating their findings, the lack of availability of x-ray sources for use in structural engineering research purposes, or doubts in usefulness of their models/methods.

The method used to quantify the bolt deflection from the x-ray images was crude and involved projecting the x-ray image and scaling off the deflection values. Humphrey and Ostman also noted the dangers and difficulties of working with the high energy x-ray machines which were available at the time. Now, after twenty years, both of these issues are easily circumvented as safer, battery powered, low energy x-ray sources with digital detectors are available and have found widespread use in non-destructive testing applications. The process of obtaining deflected bolt images inside of connections can now be carried out via x-ray imaging without the extreme safety hazards. Also, the task of determining deflection data from an image can be carried out more exactly by using digital image processing edge extraction techniques, of which there are numerous methods that are well known and frequently used in various digital image processing applications.

While Humphrey and Ostman [1],[2] focused their attention on estimating the reaction of the wood against the dowel, allowing for potential advancements in the modeling of wood-fastener interaction during loading, the digital form of the deflection data allows the opportunity for the information to be processed and used in other ways as well. Perhaps most importantly, the edge data could potentially be used to determine the strain distribution along the length dowel. This could be done by invoking geometry and simple beam theory principles, and only requiring the calculation of curvature from the data.

The eventual goal of research in wood connections should be to understand the complex interactions taking place between the anisotropic wood and isotropic dowels. Greater understanding would lead to better mathematical models, which can emulate this behavior, and to more accurate simplified models or empirical formulas, which can be used for safe and efficient design. The empirical models available for dowel behavior are often very conservative, while complex mathematical models have no real validation of results other than load-displacement [1],[2],[3].

It is evident that there is a need for new innovative test methods to capture important but previously unquantifiable behaviors of connections, so that a better understanding of the complex interactions can lead to the development of better mathematical models. Additionally, the quantification of behaviors from non-destructive test methods (like x-ray imaging) also has potential uses when investigating structures in situ. Most methods which obtain quantitative information involve destructive or semi-destructive techniques. Any method which quantifies behavior from a non-destructive

test would better enable the assessment of a structural component, without the shortcomings of destructive testing methods.

Chapter 2. Literature Review

Timber connections have been examined by numerous researchers from all over the world for more than 75 years. This has produced an extensive amount of information which has led to the theoretical and empirical models and failure criteria which are used in analysis and design today.

Most connection design methods used today are based on the work of Johansen [4] which was carried out in the 1940s. Often called Johansen's Yield Theory or the European Yield Model, this model considers failure occurring by one of several modes. Each mode involves the plastic deformation of wood, the plastic yielding of the connector, or a combination of both. This model has been verified experimentally various times, most recently in the United States by Aune and Patton-Mallory [5] who examined nailed connections with domestic species and construction methods.

While doweled connections can be designed for large timber structures, often their use is limited by serviceability issues. Reinforcement of timber connections has been a widely studied research subject, aiming to both increase the strength and the ductility, especially when members are loaded perpendicular to the grain. In a dissertation by Heiduschke [6], reinforced moment connections were compared with un-reinforced moment connections to evaluate the influence of the reinforcement on ductility and stiffness between columns and beams. It was found that the improved ductility due to the reinforcement allowed both larger dowels and wood densification to be used, both of which increase the overall connection capacity. The improved ductility is also advantageous energy dissipation, as long as the dowels yield before the wood crushes, as seen in Figure 1.

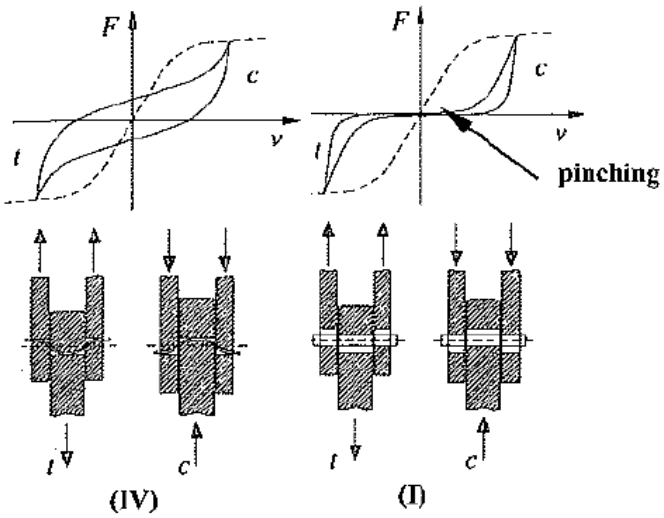


Figure 1: Shear connection failure modes and energy dissipation [6]

For a connection to dissipate the energy efficiently it is essential that plastic deformation of the steel dowels is reached prior to significant crushing of the wood. The Mode IV force-displacement curve shown in Figure 1 encloses a much greater energy than the Mode I failure, where dowel yielding does not occur.

For determining information other than load-displacement data from a connection test, non-contact measurement techniques seem to be a viable option. Sjödin et al. [3] used digital image correlation (DIC) techniques to examine the strains present in the side members of connections containing many dowels. A random pattern was applied to the outside of the edge member and two CCD cameras took images of the connection from different angles during loading. The commercially available DIC program that was used identified facets of the random pattern and then tracked their movement during loading in three dimensions to calculate strain values. The overall global results of an elastic numerical model compared favorably with the DIC method, and the influence of knots and defects in the wood specimens was clearly present in the results from DIC. It was also noted that this method is ideal for studying crack initiation and development. The

method allows strains to be calculated over relatively large areas, so even if the crack location is not known beforehand, the information captured by the method still would contain the pre- and post-crack strain fields in the vicinity of the crack.

Pollock, et. al. [7] investigated bolts in timber connections via ultrasonic techniques. By probing the bolt head and recording the pulse echoes which travel through the bolt and back to the head, the variations in the time history signal amplitudes were used to determine the formation of a plastic hinge and to quantify the total damage. It was found that the amplitude of the echoes shifted in relation to the damage taken (i.e. increasing plastic hinge angle). Correlations were developed to determine such things as plastic hinge angle, bolt deflection and connection load from the echo time history. The methods developed performed with higher confidence levels when comparisons to the unloaded bolt could be made, however even without the information from the unloaded bolt, R^2 values for the empirically derived formulas were found to be close to 0.9. It should be noted that the work focused only on connections which failed as a Mode III double shear failure. The impact of additional plastic hinges (corresponding to a Mode IV failure) was not investigated.

As mentioned previously, Humphrey and Ostman [1],[2] have been the only researchers to use x-ray imaging to investigate doweled wood connections and attempt to use the information from the images in a quantitative fashion. Recently, more and more researchers and engineers have been using low energy x-ray sources to investigate structures in situ. Similar to the use of x-rays technologies in almost any field, these projects have focused on simply using x-ray imaging to display objects hidden from the naked eye, and then qualitatively assessing the condition of structural components,

construction methods, sizes of hidden members and fasteners, etc. This has been done successfully by a variety of researchers.

Hart [8] used x-ray analysis to examine the Narbonne House in Salem, Massachusetts. He successfully identified hidden structural bracing, fastener types, and found little decay of members. Wrenn [9] did similar work at various locations throughout New England, focusing on historic buildings built prior to 1800. Construction methods and materials were recognized easily from the images, however the difficulties in obtaining clear radiographic images of some components due to restricted placement of the source and detector, and also the difficulty in analyzing the images due to lack of depth perception, were noted.

Anthony [10] used digital radiography to investigate the connections in wood roof trusses of large military warehouses. Some of the construction methods were unknown and were required for an evaluation of the current load carrying capacity. By using x-ray imaging to look into the interior of the joints, split rings were found to be present and their size, number and condition were noted. With this knowledge, the load carrying capacities were then able to be estimated much more accurately. An example of the detection of split ring connectors in a double shear connection using x-ray imaging can be seen in Figure 2.

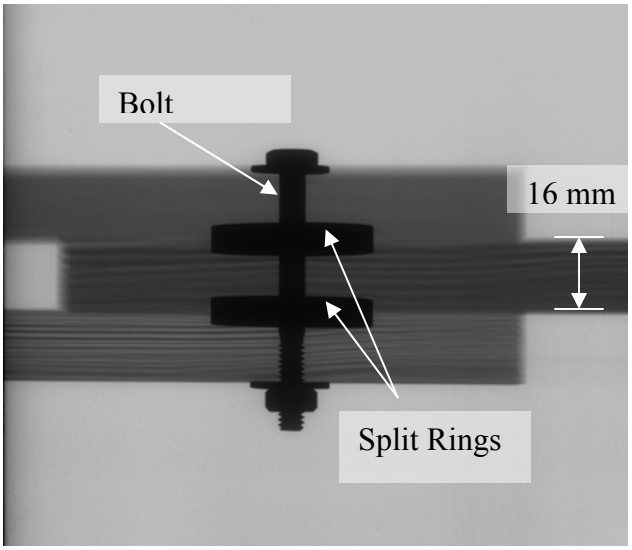


Figure 2: Detection of split ring connectors using x-ray imaging of a wood connection

In a thesis by Lear at NC State [11] various non-destructive and semi-destructive techniques for the in situ evaluation of wood structures were examined. The evaluation of radiography showed the capability of finding cracks, voids and deteriorations, however it also showed the difficulty in being able to quantify this information. This is mostly due to the collapsing of three dimensions into two dimensions when obtaining an x-ray image. Void volumes and crack depths cannot usually be accurately predicted from a two-dimensional image. Lear noted that other semi-nondestructive techniques are available (such as resistance drilling) to further probe problematic locations found via x-ray imaging. In the cases of a small crack with its plane perpendicular to the x-ray beam, the minute change in thickness does not provide sufficient contrast in the image to identify the crack. This shortcoming can be resolved by taking images from various angles, but doing this for all components to be investigated in a structure quickly becomes extremely time consuming and expensive, both from the standpoint of capturing the images and also having to interpret the results of multiple 2-dimensional images as a 3-dimensional space.

Chapter 3. Goal of Study

The goal of this study was to develop and verify a method which could be used to provide non-contact strain measurement of dowels in timber connections via x-ray imaging and digital image processing. The desired methods and procedures would be useful in the examination and understanding the dowel bending strain behavior in wood connections. By being able to obtain a strain distribution along each dowel in a loaded connection, the influence of many parameters (such as dowel size, number of dowels, dowel location, member size, member type, etc) on the distribution of load to each dowel could be determined. Also, the determination of a previously immeasurable but important quantity in the behavior of wood connections allows for this quantity to be used as an additional verification in mathematical models which are created to simulate the complex interactions associated with dowel bearing and bending.

The lack of a priori knowledge necessary to compute strain values in the dowels makes this method of investigation ideal for *in situ* evaluation, as long as the connection in question is easily accessible. Ideally, structures could be examined after severe loading events (earthquakes, hurricanes, etc) and the quantitative interpretation of behavior could give more insight into the complex interactions that not only take place within the structural connection, but also with the influence on its behavior from non-structural components. In these types of situations, typically the main consideration would be to determine if yielding of the dowel has occurred, and if so, to try to quantify the extent of yielding.

Chapter 4. Materials and Methods

4.1 Radiography

Radiography has been used in a variety of fields, such as medical, crystallography, security, art, and non-destructive testing. Mostly all methods, with the exception of crystallography which looks at diffraction patterns of x-rays, simply involve using x-ray radiation to create an image to visually inspect areas that cannot be seen with the naked eye.

Typically, radiography involves generating photons which attenuate when passing through a medium. A detector is used to capture the resulting energy which has passed through the target object. Since more energy is attenuated when passing through higher density or thicker materials, an image is created on the detector of the variation in density and thickness of the target.

The physics describing the radiographic process are well known. As an overview, to create x-rays the cathode supplies electrons via a current, which are then accelerated towards the anode due to the voltage potential. When impacting the anode, energy is released in the form of x-ray radiation and heat.

The attenuation of x-rays as they pass through a medium is described by Equation 1.

$$I_t = I_o \times e^{-\mu t}$$ **Equation 1**

Where: I_t = emergent intensity
I_o = original intensity
t = thickness of material
μ = linear attenuation coefficient per unit thickness

The linear attenuation coefficient is a material characteristic affected by the density of the material and the energy of the x-ray radiation. Wood is typically about 10-20 times less dense than steel. This difference is sufficiently large and allows the boundaries of each material in an assemblage (such as a connection) to be easily seen, regardless of the fact that the wood in a connection is significantly thicker than the steel dowel. The linear attenuation coefficients for wood, iron and concrete over a range of x-ray energy values are shown in Figure 3.

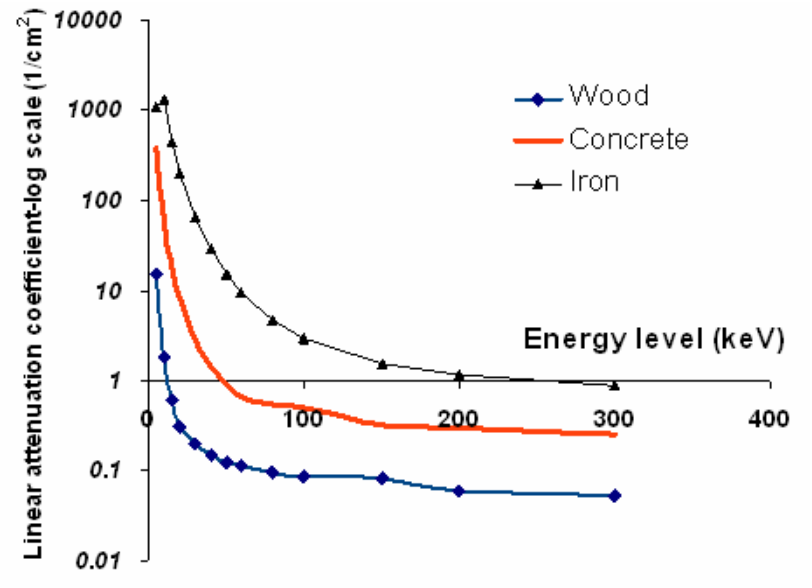


Figure 3: Linear attenuation coefficients as a function of x-ray energy [11]

Contrast and definition are the two parameters which influence the quality of x-ray generated images. The contrast is predominantly a function of the absorption differences in the different portions of the target, the wavelength (or energy) of the generated x-rays, and the film used. The wavelength, and therefore energy, can be controlled in the x-ray tube by changing the voltage of the anode. Lower voltages result in higher contrasts, but also result in smaller latitude (range of thicknesses or density values which can be shown). Although the properties of different types of film vary,

typically longer exposure times provide better contrast, as long as the film does not become oversaturated with energy. This is because a typical film's sensitivity increases with exposure. With imaging plates, which are read by scanners for digitizing, the sensitivity rate is typically constant over the entire range of exposure.

The definition in an x-ray image is mostly determined by the geometric unsharpness, a property based on the geometry of the x-ray imaging set-up. The geometric unsharpness can be thought of as a measure of the distortion caused by projecting a 3-dimensional target onto the 2-dimensional detector. Figure 4 shows how the x-ray set-up geometry results in distortion of the object edges, and Equation 2 defines the geometric sharpness value, U_g, based on the geometry.

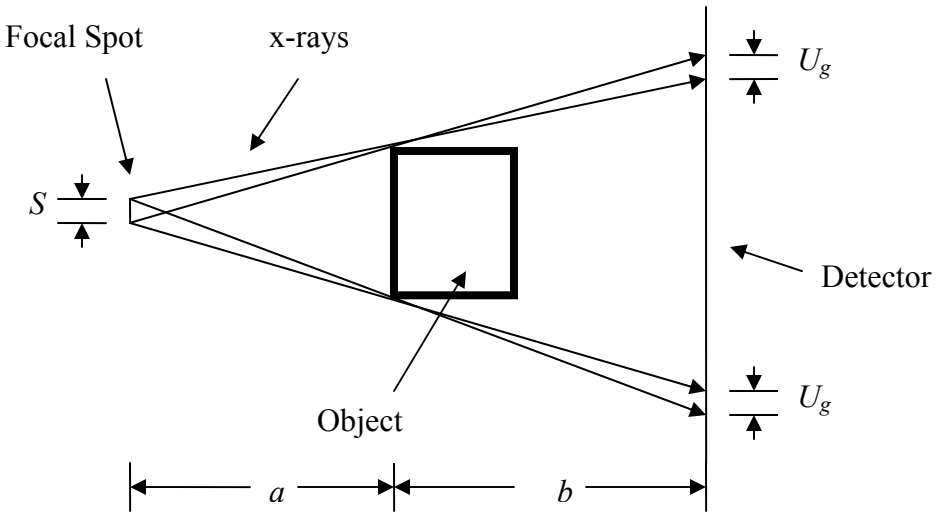


Figure 4: Definition of Geometric Unsharpness

$$U_g = S \times \frac{b}{a}$$ **Equation 2**

Where: U_g = the geometric unsharpness
S = size of the focal spot in the x-ray tube
a = distance from the x-ray source to the front of target
b = distance from the front of target to the detector

As can be seen from Figure 4, the edges of objects can become increasingly distorted as U_g increases. The focal spot size, S, is the size of the area which the x-ray radiation emanates from the source. Since the focal spot size cannot typically be changed on an x-ray source, the "sharpest" possible image is obtained by maximizing the distance from the source to the target and minimizing the distance from the target to the detector, however increasing the distance between the source and target decreases the intensity. Higher voltages may then be required to sufficiently penetrate the target which could result in the produced image having less contrast.

Because the geometry of the set-up influences the quality of images obtained, ASTM E 1742-08 [12] gives an equation for a minimum source to film distance for radiographic examinations. Rearranged to comply with the notation in Equation 2, the ASTM equation is shown below.

$$a + b = (S * b / U_g) + b$$ **Equation 3**

The maximum values of U_g are given in Table 1.

Table 1: ASTM maximum allowable geometric unsharpness values

Material Thickness (mm)	Maximum U_g (mm)
Less than 51	0.51
51-102	0.76
Greater than 102	1.02

4.2 X-ray Equipment

The usability of radiography for laboratory and in situ testing has been limited due to the environmental hazard produced by the equipment required. The electromagnetic radiation produced by x-ray sources can be very damaging to living tissue, and safety precautions must be taken when using such devices. In the past, the x-ray sources were very large and used much higher energies than the devices available today. Nowadays, while safety is still a concern, radiography can be carried out more safely and easily.

The x-ray source used was an ICM CP120B [13], a battery powered x-ray generator, shown in Figure 5.

Figure 5: ICM CP120B battery powered x-ray source

The voltage potential has a range of 40-120 kV, while the current has a range of 0.1-1.5 mA, although above 80 kV, 1.0 mA is the maximum usable current. The focal spot size was given from the manufacturer as 0.5x0.25 mm. The machine can be fired using a trigger connected by a long cable to the source, to ensure that the user was out of the range of harmful amounts of radiation. Figure 6 shows the x-ray spectrum for the CP-120B, which was provided by the manufacturer [13]. The *x*-axis is the energy of the

photons in KeV, while the *y*-axis shows the relative proportions of photon intensity, which is a function of mA and exposure time for various kV settings.

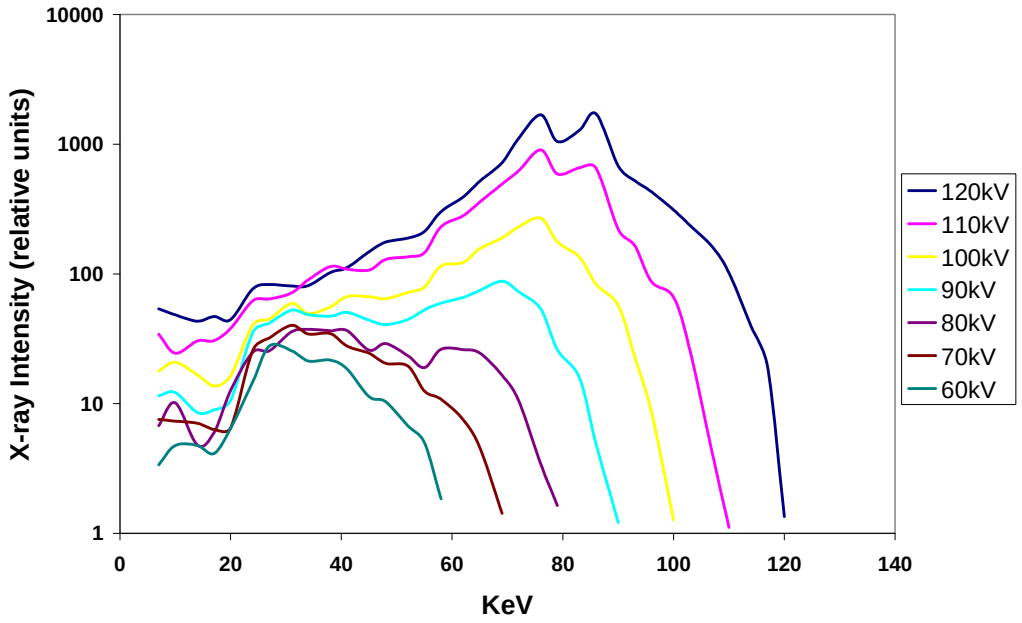


Figure 6: CP120B Energy Spectrum [13]

As can be seen, the source actually provides a continuous span of x-ray energy levels. This is due to the combination of the energy produced by the accelerated electrons impacting and removing elections from target atoms and the energy produced by the rapid deceleration of elections upon impact, which is referred to as Bremsstrahlung or "braking radiation". This continuous spectrum limits the use of Equation 1 to calculate appropriate source settings (kV, mA) and exposure times, since Equation 1 describes the phenomena of monoenergetic photons. To help determine correct settings and exposure time, empirically derived exposure charts can be used. These charts are particular to each model of x-ray source and target material. An example an exposure chart for the CP-120B for iron is shown in Figure 7. The *y*-axis represents a combination of mA-s which must be used for a specific kV to penetrate a thickness of material, *e*.

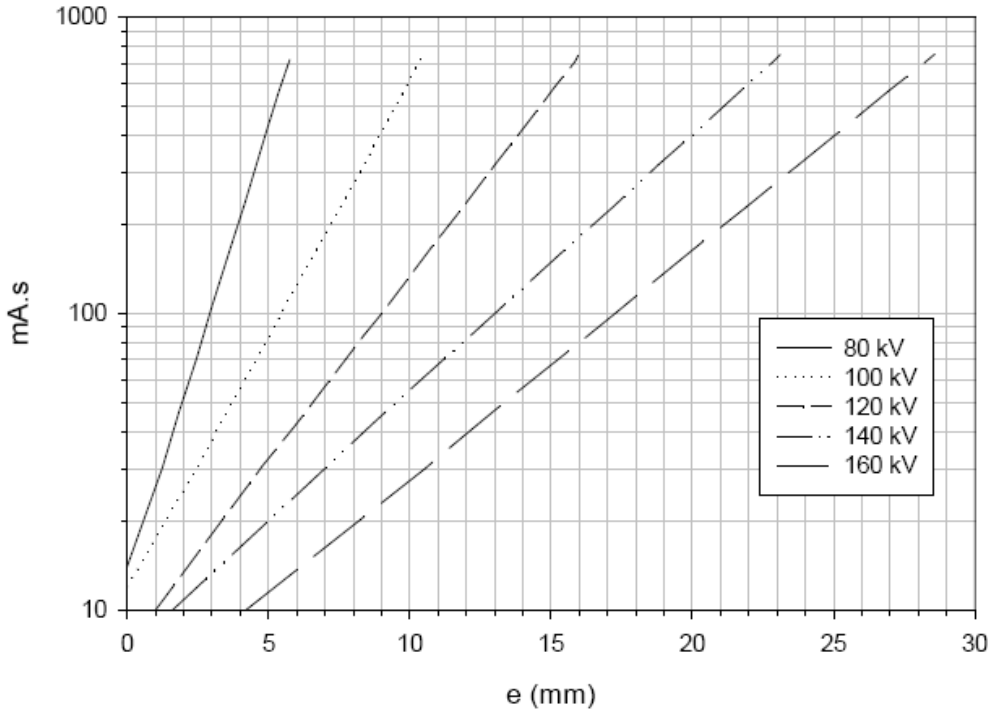


Figure 7: Exposure Chart for Iron (Fe) for CP120 and CP160 portable x-ray sources [13]

Another technological development which advanced the usability of radiography in non-destructive testing was the advent of the digital detector. Sheets of film no longer need to be developed after each x-ray is taken, and therefore the chemicals and time required to develop the image from the film are no longer present.

A Fuji medical imaging plate [14] and an A/T Scan X imaging plate made by Air Techniques [15] were used. Both plates were 250 mm x 305 mm (10" x 12") and use photo-stimulable phosphor as their main component. The phosphor electrons become excited but entrapped in the crystal lattice when stimulated by radiation, and then the energy is emitted upon the stimulation by electromagnetic radiation having a longer wavelength than that of the x-ray radiation. The amount of radiation emitted from the plate is proportional to the amount of radiation absorbed, and by reading this emitted radiation a digital image can be formed. The plates are placed in sheaths when put into

position to capture images. This protects the imaging plates from physical contact and also from light. Care must be taken to scan the plates soon after exposure because fading effects (loss of stored data) can occur as a result of the chemical processes involved in the plates.

The scanner used to convert data stored on the imaging plates to digital form was an ALLPRO Imaging's ScanX 12 Portable NDT Digital Imaging System [15]. The scanner operates by stimulating the imaging plate with energy and then collecting the energy emitted by the plate. The system has a maximum given resolution of 18 lp/mm (line pairs per millimeter). A ScanX eraser [16] was used to clear the image off of the plate for reuse. The eraser exposes the plate to a minimum of 25,000 lux light for roughly 35 seconds, which releases all of the energy stored in the plate. This light intensity is roughly equivalent to outdoor daylight conditions when not in direct sunlight. From this observation it is apparent that in order to not lose any captured information the plates should be exposed to as little light as possible when transferring them from their sheath to the detector.

4.3 Digital Image Processing and Strain Calculation

The fact that x-rays create intensity images makes digital image processing techniques very straightforward. Each pixel's intensity value is represented by a single number (as opposed to color images where several numbers, such as red-green-blue values, are typically used to represent a specific color) and each pixel's spatial location is represented by its location in the matrix. Techniques for handling the enhancement, filtering, restoration, and data extraction from images have existed since the 1960s. These techniques are commonplace now in software programs such as Adobe Photoshop,

which performs all mathematical processes (such as blurring, sharpening, thresholding, etc) "under-the-hood".

For this work, the program Matlab [17] was used. In Matlab, all variables are matrices (with scalar values having size 1x1 and vectors having size *n*x1 or 1x*n*), so manipulation and indexing of matrix values is relatively easy. Since digital images are matrices, this makes Matlab very popular for image processing, especially due to the fact that Matlab has an image processing toolbox with functions specifically designed for image operations.

Four steps are required to accomplish the established goal.

Step 1. Image preprocessing
Step 2. Extraction of edge data
Step 3. Data preprocessing
Step 4. Calculation of strain

Step 1 involves cropping the image to a region of interest in order to reduce the computational demand on the computer. Also, filtering or image enhancement techniques can be done in order to prepare the image for Step 2. This involves procedures such as low-pass filtering, high pass-filtering and contrast enhancement.

The extraction of edge data can be carried out by two methods. Either the edge pixels can be found by defining edge characteristics or the object and background pixels can be segmented and the object boundary can be traced. While the human vision system can identify edges instantaneously when looking at an image, programming a computer to recognize these features is a much more involved procedure.

The most straightforward method involves thresholding the image based on intensity levels. This can be done either locally or globally. Otsu's method [18] is typically used in this application when the histogram has a bimodal distribution, i.e. there

are two distinct regions of the image which are distinguishable by their difference in intensity values. The method is optimal because automatically selects a threshold value which maximizes the between-class variance of the created object and background classes. Once the image is segmented into a binary image consisting of object and background, a Matlab procedure can be used to extract the *x*-*y* values of the edge. Now instead of a 2-d signal depicting the dowel as an image, the data being used has been reduced to a 1-d signal of the edge data.

Instead of using a thresholding procedure, different types of edges in images can be defined as having certain properties. Figure 8 shows two types of ideal edges.

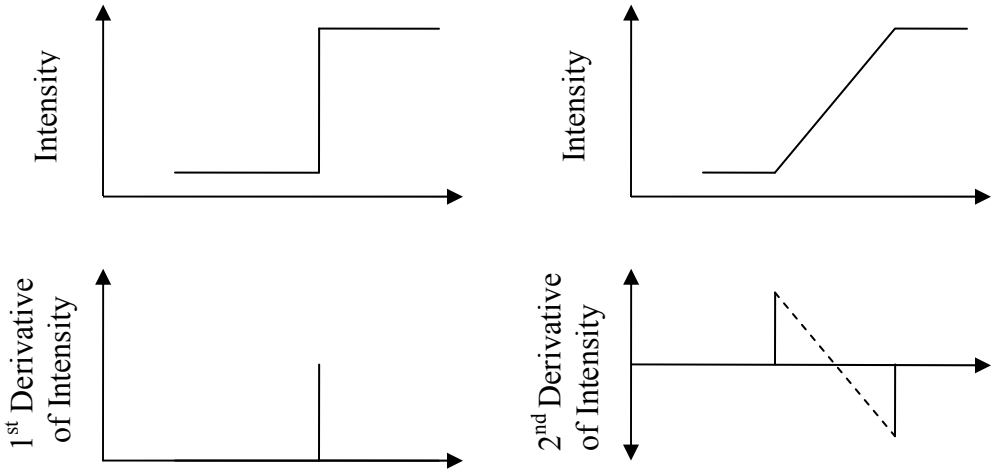


Figure 8: Ideal edges in intensity images

In (a) of Figure 8 the edge location is found by noting that the first derivative of the pixel intensity values at the edge of a step edge is a maximum. Therefore, first derivative difference operators can be used to emphasize and select these locations as the edge. In (b), the edge is not as abrupt or prominent as in (a), so a different approach is required to systematically define the edge location. Simply taking the first derivative as

in (a) would not always accurately result in the edge being selected, since in real images the ramp will never be perfectly smooth. Instead, taking the second derivative of the intensity results in maximum and minimum values at the onset and ending of the ramp. The edge location can then be consistently defined at the location of the zero-crossing between the maximum and minimum values of the second derivative. Since it is extremely rare that the second derivative will ever be exactly equal to zero, this method can result in sub-pixel resolution of edge locations. More information on this type of edge extraction, especially suited for images containing noise, can be found in [19].

In Step 3, the extracted edge data is preprocessed to facilitate the calculation of strain values. Points which are not actually of the dowel but were found by the edge extraction procedure can be removed, and data points can be added if small regions of the edge were left out but are required. Also, the top and bottom edges of the dowel can be averaged together to get the deflection profile of the dowel centerline, as demonstrated in Figure 9.

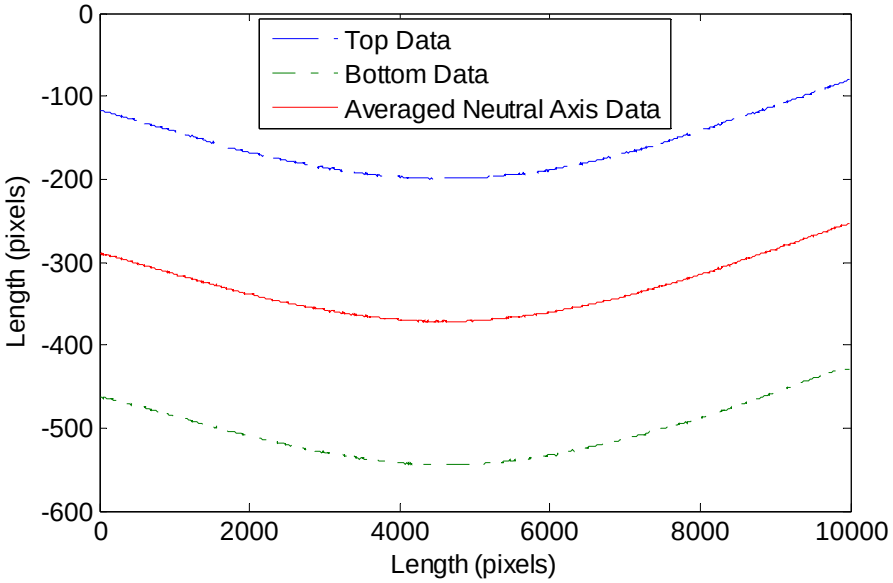


Figure 9: Averaging of top and bottom edge data to obtain neutral axis data

Aside from the fact that beam theory uses the dowel centerline for its relation of deflection to strain, averaging the top and bottom edges also suppresses some of the noise present in the digital signal.

As mentioned previously, the calculation of strain from the dowel edge deflection data is carried out by using the geometry of basic (Euler-Bernoulli) beam theory, shown below Figure 10 and described in Equation 4.

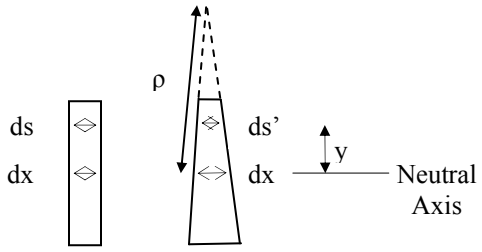


Figure 10: Geometry associated with deformations caused by internal moments in beams

$$\frac{1}{\rho} = -\frac{\varepsilon}{y}$$ **Equation 4**

Where: ρ = radius of curvature
ε = strain
y = distance from neutral axis to location of strain calculation

The inverse of the radius of curvature is curvature, often denoted as *K*. For a function expressed as *v(x),*

$$\mathrm{K} = \frac{1}{\rho} = \frac{v''}{(1+v'^2)^{3/2}}$$ **Equation 5**

where *v*'and *v*''are the first and second derivatives with respect to *x*, respectively.

Due to the small angle theory used in basic beam theory, the slope of the deflection, *v*', is much less than 1. Since this value is squared in Equation 5, its contribution can be neglected, resulting in the curvature being simply the second derivative of the displacement.

$$K = \frac{1}{\rho} = v'' \qquad \textbf{Equation 6}$$

Therefore, one only needs to calculate the second derivative of the deflection data and multiply by y (neglecting the negative sign in Equation 3) in order to find the strain value at any location x.

$$\varepsilon(x) = v'' \times y \qquad \textbf{Equation 7}$$

Alternatively, the sign of the calculated strain can be kept to distinguish between regions of opposite concavity.

No other information is required, such as load magnitude, boundary conditions or material properties. The largest strains are of the most interest and occur at the extreme fiber, where y is equal to one half of the depth for beams with doubly symmetric cross-sections.

Because the curvature is calculated based on the size of the pixels, its dimensions are inverse pixel length (l^{-1}). A link between the dimension of the pixels and the actual spatial dimensions of the beam is therefore needed to calculate the strain. This can be done by either defining y in Equation 7 as the number of pixels that make up one half of the beam depth, or by using a reference object placed at the same distance from the source and detector as the beam to determine a conversion factor to relate the dimensions of the pixels with the dimensions of beam in the image.

The calculation of strains involves multiplying extremely small second derivative values by a relatively large value y. This results in smaller y values producing more accurate calculations. This implies that the most accurate strain calculations are made when the dowel is very slender, and when the geometric unsharpness, U_g, is minimized.

It should be noted that there are several assumptions implied in the equations and the methodology stated for generating strain information from deflected dowels. Firstly,

it is assumed that any curvature in the dowel edge is due to bending, not due to shear deformations or factors such as corrosion or material defects. Dowels in connections can be considered to be a beam on an elastic foundation. Depending on the stiffness of the foundation, the shear in the beam can become significantly large, especially at the interfaces of the main and side members. This causes the "plane sections remain plane" assumption of basic beam theory to be violated. The true bending strain is then no longer proportional to the slope of the deflected shape, but is related to the angle of rotation of the cross-section. The use of higher order beam theories (such as Timoshenko Beam Theory) would be required to compensate for this discrepancy.

Secondly, it is assumed that the longitudinal axis of the dowel and the x-ray film are perpendicular to the incoming x-rays. It is also assumed that the bending of the dowel is about an axis parallel to the x-rays being fired. These assumptions are depicted by the schematic in Figure 11.

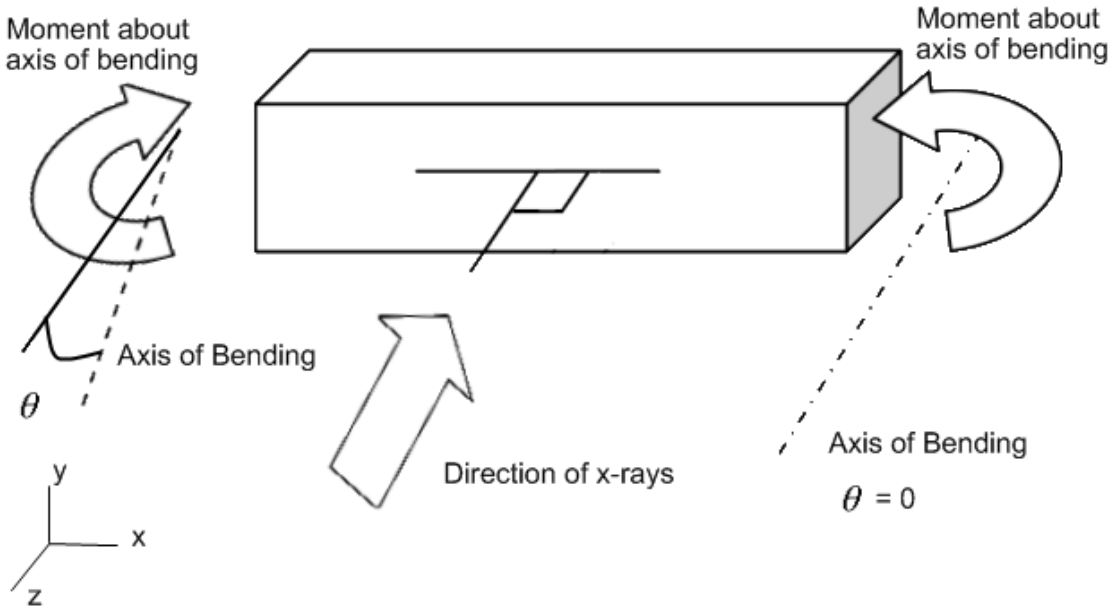


Figure 11: Geometry of the orientation of the dowel in relation to x-ray direction

The longitudinal axis of the dowel is along the *x*-axis, while the direction of the *x*-ray radiation is along the *z*-axis. When the angle θ (the angle formed by the difference between the direction of the x-ray radiation and the axis of bending) is equal to zero, the radiographic image captures the maximum bending deflection and curvature. As θ increases, the calculated curvature, and therefore strain, decreases. This is shown in a simplified sense in Figure 12, where with increasing values of θ the function $y = x^2$ is rotated about the *x*-axis and then projected onto the *x*-*y* plane.

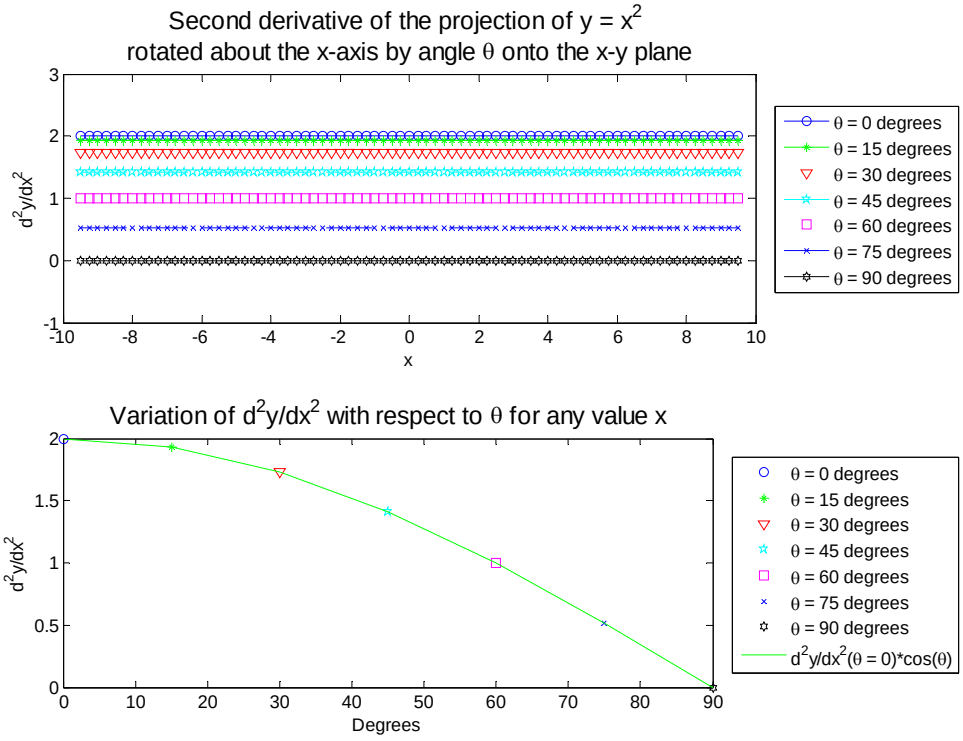


Figure 12: Changes in calculated curvature with changing angle θ

The relationship between the change in d^2y/dx^2 with respect to θ is shown in the bottom portion of Figure 12. Here it can be seen that the relationship follows the curve corresponding to the cosine of the angle θ. In situations where the second derivative is not constant (i.e. polynomials of order greater than 2 or experimental data sets which can not be accurately described by polynomials) the cosine function will be different for each

x-value but will always be of the form shown in the legend of the bottom portion of Figure 12.

This fact has two convenient results. Firstly, since the cosine function changes little in the region around $\theta = 0$, it can be stated that small deviations from the direction of x-rays being parallel to the axis of bending do not significantly influence the results. For instance, θ = 15 degrees results in an error of only about 3.4%. Therefore, the direction of the axis of bending does not need to be precisely known and the equipment does not need to be exactly placed (in relation to θ) to accurately calculate strain values. Secondly, in situations where it is known that θ has a nonzero value but can be estimated, the true strain in the dowel for any *x*-value along the dowel can be calculated from the relationship shown in Equation 8.

$$true\_strain = strain(\theta = 0) = \frac{strain(\theta_1)}{\cos(\theta_1)} \qquad \textbf{Equation 8}$$

In Equation 8, θ_1 is the known angle between the incident x-rays and the axis of bending, and $strain\ (\theta_1)$ is the calculated strain from the image taken at angle θ_1. It is apparent from the statements made previously and the cosine relationship in Equation 8 that as θ_1 increases, the potential for calculation errors due to incorrectly estimating θ_1 increases drastically. If necessary, other similar geometric corrections can be made for situations when the longitudinal axis of the beam and/or the x-ray imaging plate is not perpendicular to the direction of the x-ray radiation.

In circumstances where θ cannot be estimated with confidence, multiple x-ray images would be required to calculate the correct maximum bending strain. If two x-ray images are taken from different angles, the situation can be considered to be like the schematic shown in Figure 13.

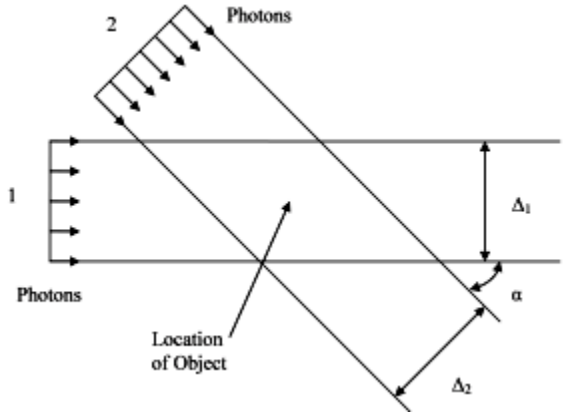


Figure 13: Geometry of object at an angle to photons

In this schematic, the object is considered to have an infinitesimally small thickness and a true length equal to Δ (which in this analogy to a bending dowel can be related to the true deflection of the dowel). From the two angles at which the x-ray images were taken the object appears to be of length Δ_1 and Δ_2. The angle between the direction of the two x-ray images, α, is known, but the angle of the object with respect to direction 1, θ, is not known prior to capturing the images. If θ were known then only one x-ray would be required since it would be known how to position the x-ray source so that the direction of the radiation is perpendicular to the orientation of the object.

For a given set of Δ_1, Δ_2, and α values there is not a single unique solution for the length of the object (Δ), as depicted in Figure 14.

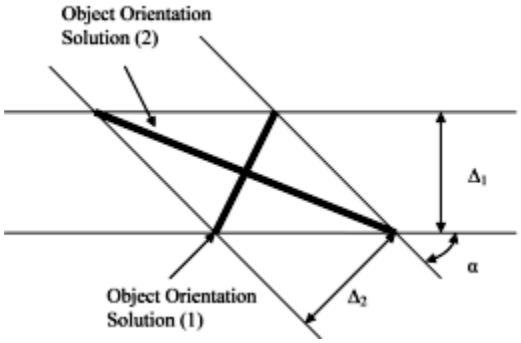


Figure 14: Two solutions to finding true object height, Δ, in Figure 13

However, as α approaches 90º the two solutions converge to a single value. The true length Δ is then simply the square root of the sum of the squares of Δ_1 and Δ_2. If something were to limit the ability to take images from two orthogonal directions, then three images would be necessary to establish the correct object orientation and length as displayed in Figure 15.

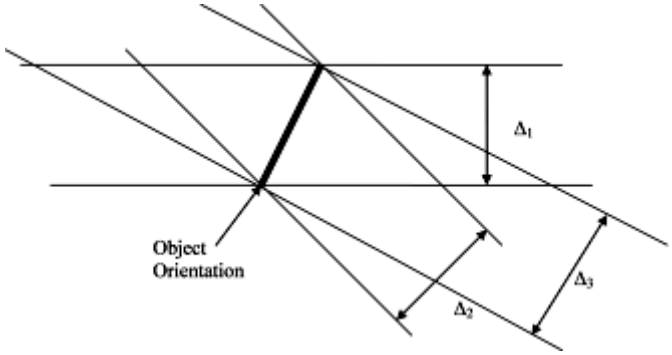


Figure 15: Three images yielding object orientation without the use of orthogonal directions

Another limitation of the method deals with the positioning of fasteners in connections with multiple fasteners. The collapsing of the 3-dimensional connection onto 2-dimentions obviously creates conflicts when there are rows with more than one bolt, as shown in Figure 16.

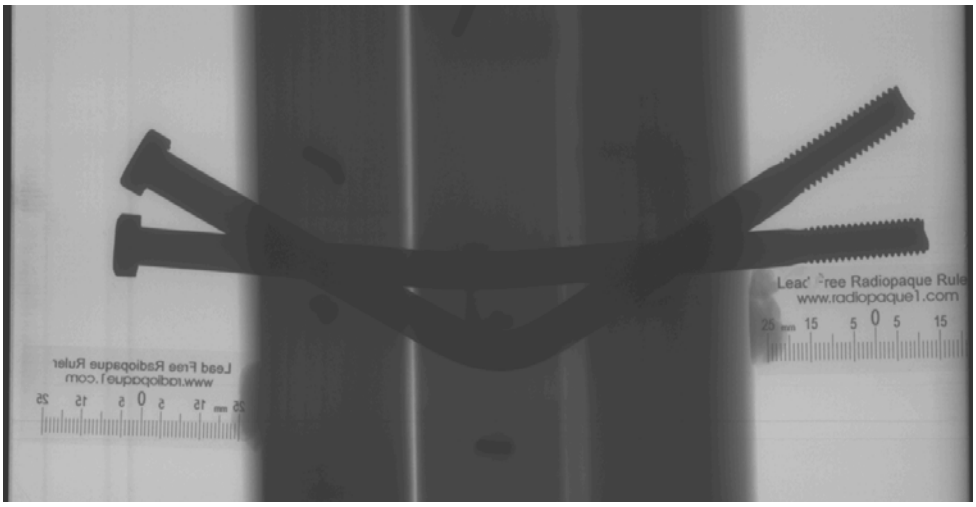


Figure 16: Radiographic image of a connection containing a row of two bolts

The edge data cannot be extracted along the full length of each of the bolts due to the overlapping. In the case of the image shown in Figure 16, the two dowels did not equally share the load, which resulted in drastically different deflections and strains. In cases like this some of the extractable edge data may still be useful to help quantify the unbalanced load sharing in the connection.

4.4 Differentiation of Inexact Discrete Data

Discrete data can be differentiated many different ways. The most straightforward is carried out by using finite difference formulas. These can be derived for uniformly spaced data using a second order Taylor Series expansion of a function,

$$f(x+h) = f(x) + hf'(x) + \tfrac{1}{2}h^2 f''(\xi) \qquad \textbf{Equation 9}$$

where h is some small value, and $\tfrac{1}{2}h^2 f''(\xi)$ is a higher order error term that can be neglected. Rearranging Equation 10 and disregarding the error term, one finds the well-known relation

$$f'(x) = \frac{f(x+h) - f(x)}{h} \qquad \textbf{Equation 10}$$

This is called a first order forward difference formula. Higher order formulas result in more accurate derivative approximations, since the error term is a function of h^n and n increases with the order. For example, subtracting the Taylor Series for $f(x-h)$ from the Taylor Series for $f(x+h)$ yields,

$$f(x+h) - f(x-h) = 2hf'(x) + \tfrac{2}{3!}h^3 f'''(x) + \ldots \qquad \textbf{Equation 11}$$

Solving for $f'(x)$ and neglecting higher order terms results in a second order central difference formula, given by Equation 12.

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h}$$ **Equation 12**

The finite difference approximation for the second derivative can be found by adding the Taylor Series for $f(x+h)$ and $f(x-h)$.

$$f(x+h) + f(x-h) = 2f(x) + h^2 f''(x) + 2[\tfrac{1}{4!} h^4 f^{iv}(x) + ...]$$ **Equation 13**

Solving for $f''(x)$ and ignoring the forth order and higher terms yields

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$ **Equation 14**

Often these equations are implemented using *for* loops, however, since Matlab is a built-up computer program which favors matrices and built in functions, it is more convenient and efficient to use the equation coefficients to form vectors and perform a convolution with the data. For instance, Equation 14 with $h = 1$ would become the vector $[1 \quad -2 \quad 1]$. This is commonly referred to as a filter. Subsequently, the discrete convolution can be carried out by Equation 15.

$$w(n) = \sum_{m=-\infty}^{\infty} f(m)\, g(n-m)$$ **Equation 15**

It should be noted that convolution and correlation have very similar definitions, and for symmetric 1-d filters, the two operations are equivalent. The definition of convolution in Equation 15 requires the "padding" of an infinite amount of zeros to both ends of the data and filter. Simplifying this to obtain only the usable output data, an example of convolution of a second derivative filter with a data set generated from a polynomial function is shown below.

Data: $y = x^3$, x = [0,5]:	[0	1	8	27	64	125]
2nd Derivative Filter:	[1	-2	1]	movement of filter through data→		
2nd Derivative:		[6	12	18	24]	

At the filter location shown above the calculation of the second derivative would be "$0\times1+1\times(-2)+8\times1=6$". The filter would then be shifted one value to the right and the operation would be repeated. For the endpoints, the derivative cannot be calculated because insufficient information is present. Non-centered finite difference approximations can be used here, however unless these values are specifically needed for the problem, they can simply be cropped out of the data set. Note that the number of data points cropped from each end is equal to rounding down the value of one half of the filter length.

As noted by many researchers, attempting to differentiate collected data is an ill-posed problem. It is called such because small errors in the collected data lead to large errors in the calculated derivatives. This is due to the fact that small errors (caused by noise, discretization, etc) introduce large variations in the slope of the data between points. Plotted below is a cubic function from x = [-5,5], by intervals of 0.1 with an additive normally distributed error (mean = 0, standard deviation = 1). As can be seen in Figure 17, this small error does not significantly corrupt the original signal but does produce large errors in the first derivative, and even larger errors in the second derivative (both of which were calculated by the 3 point finite difference formulas of Equation 12 and Equation 14, respectively).

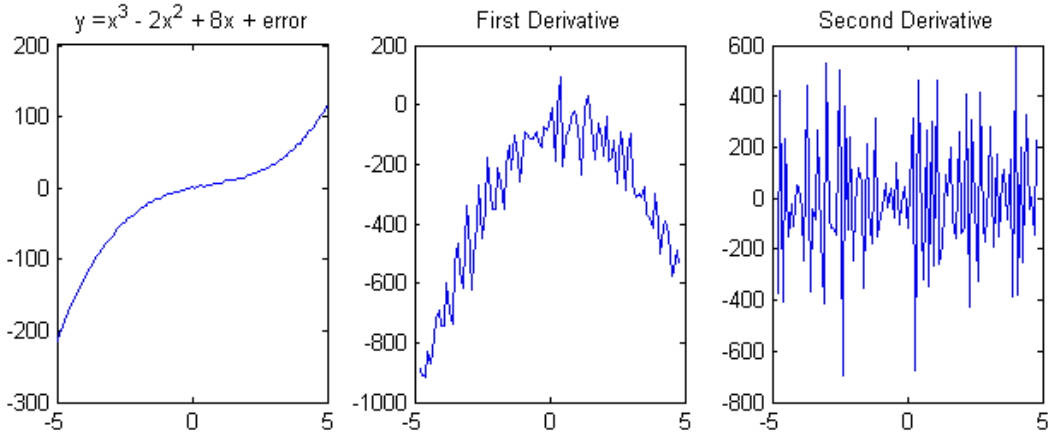


Figure 17: Polynomial function corrupted by noise and derivative calculated by finite difference methods

By convolving the data with an averaging filter (filters of the form $\left[\frac{1}{n} \quad \frac{1}{n} \quad \frac{1}{n} \ldots \frac{1}{n}\right]$, where n is an odd positive integer and the number of terms in the filter is equal to n), some of the noise in the data can be suppressed, as shown in Figure 18.

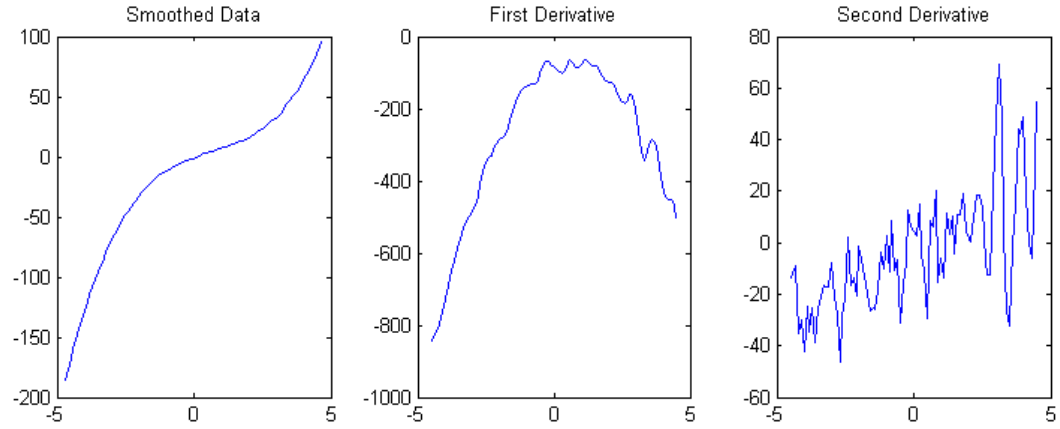


Figure 18: Data from Figure 17 smoothed by averaging filter

The data was smoothed by convolving the data with a three point averaging filter (coefficients $[\frac{1}{3} \quad \frac{1}{3} \quad \frac{1}{3}]$) three times. Running an averaging filter through the data multiple times typically is more effective than using a larger averaging filter only once because as the filter is passed through the data multiple times the overall effect becomes similar to the use of a Gaussian filter. This can be shown by convolving the averaging filter with itself, since convolution is associative, i.e. $\frac{d}{dx}(f * g) = \frac{df}{dx} * g = \frac{dg}{dx} * f$, where

$(f * g)$ denotes the convolution of vectors f and g. Figure 19 demonstrates the approach of using multiple passes of a seven point averaging filter to approximate a Gaussian filter.

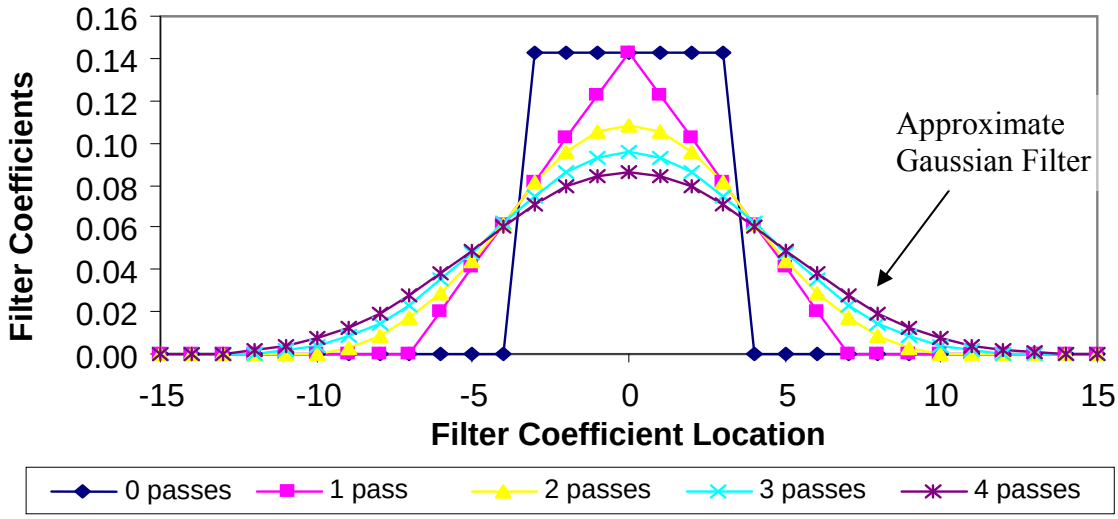


Figure 19: Approximation of a Gaussian filter by convolving an averaging filter with itself

The resulting filter has a standard deviation of $\sqrt{\frac{n^2-m}{12}}$, where n is the size of the filter and m is the number of passes plus 1.

Looking back at Figure 18, a majority of the noise from the first derivative has been removed, however there are still large variations present which oscillate about the actual value. The linear trend of the second derivative is now present, yet the variations are still too large to determine actual values. Using larger averaging filters or convolving the filter with the data more times will result in smoother data and derivatives, however it is extremely difficult to establish beforehand what size filter or number of passes should be used to obtain smooth and accurate derivatives.

If information about the true behavior of the noisy data is known, then a polynomial curve of the form $y = a_0 + a_1x + a_2x^2 + \ldots a_nx^n$ can be fit to the data using linear regression. From the fitted curve the smooth polynomial derivatives can be

obtained analytically. It is essential that the correct order polynomial is selected for the data, especially when the goal is to calculate derivatives. Selecting too low of an order results in the derivatives not being accurately represented, while selecting too high of an order can result in oscillations in the derivatives which are not actually present, especially if the data is non-exact or contains errors.

Since polynomial fitting is a linear procedure, filter coefficients can be derived that when convolved with the data will replace each data point with the smoothed value from a local polynomial fitting. These filters are known as Savitzky-Golay Filters [17] and are typically derived for orders 2 or 4, and for any odd span length. Their main role is for smoothing data while preserving local maximums/minimums, which can be significantly flattened if the data is smoothed using averaging filters. Also, derivatives at each data point can be found by simply differentiating the local polynomial approximation. This can be done by using filter coefficients as well, so the procedure of determining derivatives is merely convolution of the data with a filter which also provides smoothing. A main advantage of this method is the fact that, like the finite difference method, no a priori knowledge of the derivative is needed, but unlike the finite difference method noise is not amplified; instead it can be suppressed by the polynomial approximations if the selected window size is large enough.

Another method which can be implemented involves using the Fast Fourier Transform to determine coefficients for a function comprised of a sum of sines and cosines which fit the data. The transform gives coefficients so that the data can be completely reconstructed, but by using only lower frequency components a function representing an approximation of the signal can be found. The function can then be

differentiated analytically to obtain smooth derivatives. Obviously the number of terms included in the sum of sine and cosine function highly influences the behavior of the derivative.

Due to the ill-posedness of differentiation of non-exact data, researchers and mathematicians have developed various types of schemes in order to ensure that the calculated derivative from a noisy dataset is smooth, while not influencing the overall global behavior of the derivative or suppressing information (see [21], [22]). These methods are often very complex and require some input parameters which influence the smoothness of the calculated derivatives. Another method which can also be applied is to use a scheme to automatically select the appropriate size of a mollifier (a smooth function) to convolve with the original unsmooth data [23].

Table 2 shows all of the examined differentiation methods and summarizes their pros and cons for use in this research.

Table 2: Pros and cons of various differentiation methods

Method	Pros	Cons
Discrete Derivatives + Averaging Filters	• No influence of differentiation method on strain distribution	• Filter size and number of passes must be selected
Polynomial Fitting + Analytical Derivatives	• Suppresses noise without cons of smoothing filters	• Selection of polynomial order greatly influences results
Savitzky-Golay Filters	• Typically better smoothing of data than averaging filters • No influence of differentiation method on strain distribution	• Filter size, order and number of passes must be selected • Computationally intensive for large filter sizes
FFT + Analytical Derivatives	• No influence of method on strain distribution	• Number of terms to include greatly influences results
Regularization or Mollification Schemes	• No influence of differentiation method on strain distribution • Generates smooth discrete derivatives without selecting size for smoothing filters	• Very complex • Iterations, tolerance, and other parameters must be selected • Computationally intensive for large data sets

While various parameters need to be selected for the regularization or mollification schemes, it should be noted that the procedures are iterative and typically converge without significant differences, regardless of inputs. This was tested and verified for the mollification method outlined by Murio in [24].

As an overview of the mollification procedure, a filter created by a Gaussian function is convolved with the data to form a smoothed data set. The variable which must be selected for the filter is the standard deviation, σ. Since Gaussian functions extend infinitely to $\pm\infty$, a cut-off also needs to be selected so that the filter will have a finite length and can be implemented. 3σ is a typically selected cut-off for these types of procedures, since 99.7% of the area under a Gaussian curve is within 3σ of the mean. Therefore, selecting a cut-off at larger distances from the mean has little effect on the results.

The σ value is selected automatically by Generalized Cross Validation (GCV), which involves using a search algorithm to determine the σ value which minimizes a functional. The functional used is essentially the ratio of the sum of squared residuals to the filter size. This effectively creates a balance between smoothness and accuracy, resulting in output data which is smooth but also is still a good approximation of the original data. The only changes made to the algorithm from *The Inverse Engineering Handbook* [24] was to have the program warn the user and not perform any smoothing if the maximum or minimum values of the search procedure are selected by the GCV. This prevents the user input minimum and maximum values from influencing the results.

Chapter 5. Feasibility Study I: Simple Beam Test

5.1 Test Set-up

Since this type of study has not been done before, it is not known whether meaningful quantitative information about the strain distribution along the dowel can be ascertained from a radiographic image captured with current technologies. Therefore, simplified initial feasibility studies were carried out. For the first study, the complex interactions of a wood connection were simplified to a simply supported beam under mid-span point loading. With a strain gage attached at the location of the load, the strain readings from the gage were compared with the values calculated from the digital image processing. A schematic of the test set up is shown in Figure 20. Figure 21 shows a close up image of the LVDT, strain gage, and cable which provided the mid-span loading.

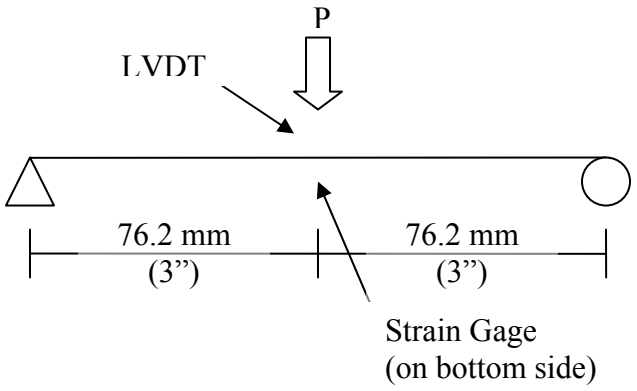


Figure 20: Schematic of 3 point bending simple beam test set-up

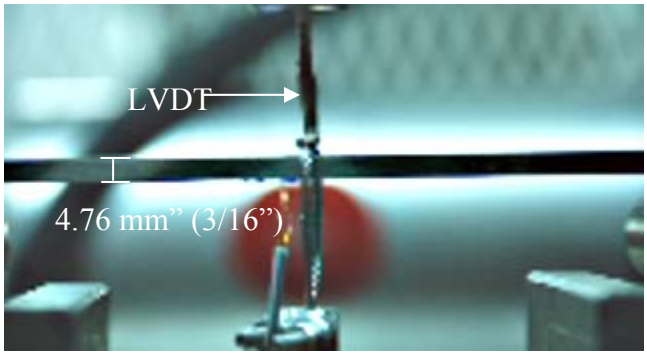


Figure 21: Close-up of cable applying load for simple beam test

The square steel dowel was instrumented with a strain gage on the bottom side at mid-span, and the load was applied via a cable which was looped about the dowel being pulled downwards. The LVDT rested on top of the cable looped around the beam, with a small amount of putty used to ensure that the LVDT does not slip off of the cable. The values from the LVDT were used with beam theory to ensure that there were no large errors and the gage was working properly.

In a subsequent test, a different loading condition was used. This consisted of a four point bending condition so that the moment (and therefore strain) across the gage would be uniform. A schematic of this test set-up is shown in Figure 22.

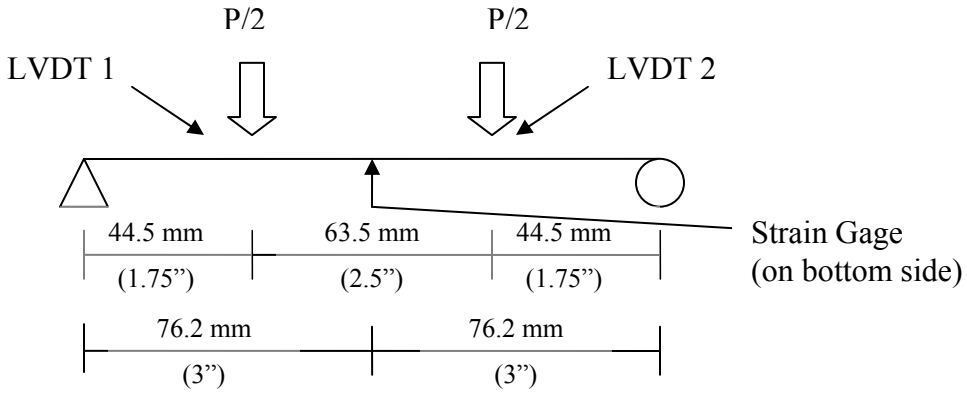


Figure 22: Schematic for 4 point bending test

A square steel dowel was used as the beam for this first test for three reasons: 1) a strain gage can be more easily attached to a square dowel than on a cylindrical dowel, 2)

rotation would not have to be monitored during testing to ensure that the gage remained perpendicular to the axis of bending, and 3) edges would be easier to identify with the square dowel. When radiographing a cylindrical object, the intensity varies as a function of the distance from the longitudinal centerline of the cylinder. This causes the boundary of the cylinder to be somewhat misrepresented and the edge of the cylinder more difficult to detect, since there is a gradual change in thickness from the diameter of the cylinder down to zero. Unlike a cylinder, a square dowel has constant thickness, so the edge is more easily identifiable from the background.

The equations of beam theory are developed from calculus and therefore are based on infinitesimally small distances, but the strain which was used to verify the method was obtained by a strain gage which has a finite length. Therefore, the values of strain over the length of the gage calculated from the digital image were averaged to obtain a value to compare with the strain gage readings.

5.2. Simple Beam Results

A typical x-ray image from Feasibility Study I is shown in Figure 23. The x-ray source was set to 50 kV and 1 mA, and an exposure time of roughly 3 seconds was used. Due to the spatial limitations of the laboratory set-up, the source was placed roughly 1 meter from the object, and the detector was placed as close to the object as possible, at a distance of about 100 mm. This yields a geometric unsharpness value of 2.5-5 μm depending on the direction, since the focal spot is rectangular. Images were scanned using the highest available resolution, given by the computer program which runs the scanner as 18 lp/mm. If it is assumed that there are 36 pixels in a millimeter then the size of each pixel is roughly 28 μm, which means that all of the unsharpness caused by the geometry of the set-up is contained well within a single pixel.

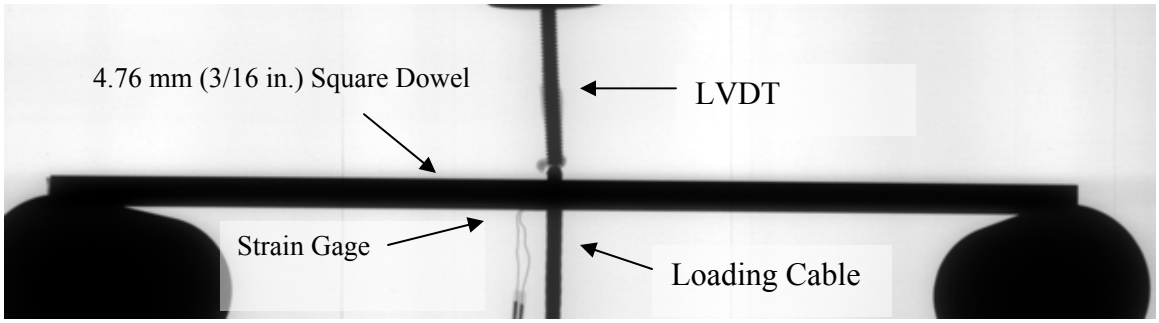


Figure 23: Phase 1 typical x-ray image

The contrast in the image shown in Figure 23 has been enhanced for better viewing and the image has been resized to only include 5% of the pixels. The true size of a 200 mm x 250 mm image scanned at 18 lp/mm resolution is extremely large, containing more than 100 million pixels and requiring greater than 300 megabytes to store in ".tif" format. This makes the pre-processing procedure for extraction of edge data extremely computationally intensive.

Because these test images contained easily distinguishable background (light pixels) and objects (dark pixels), Otsu's method was used to separate the image into two classes. The final result is a binary matrix consisting of object pixels and background pixels, shown in Figure 24.

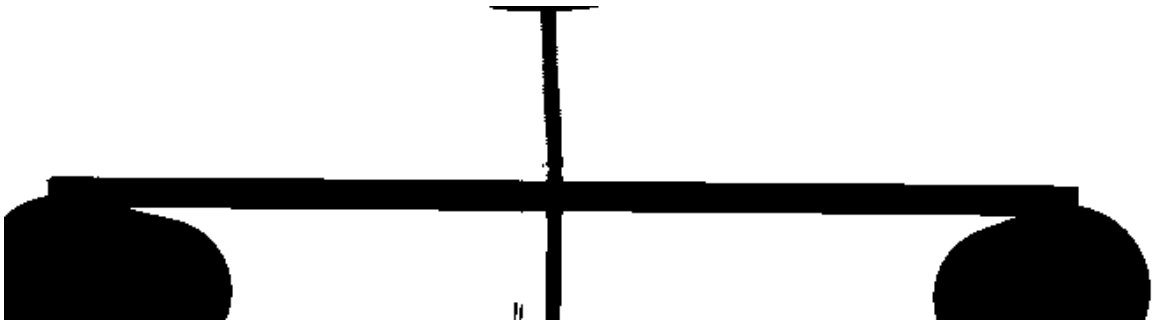


Figure 24: Figure 23 segmented by Otsu's method

The edge of the dowel was extracted by using a Matlab procedure which traces and records the *x-y* location values of points which are object pixels but contain background pixel neighbors. The Matlab code used to extract the edge can be found in Appendix B.

The load read by the load cell, the strains from the gage, and the amount of pixels per millimeter for the images are shown in Table 3 for Data Set B1 and Data Set B2 ("B" for beam). Data Set B1 was from a 3 point bending test, shown in Figure 20, while Data Set B2 was a 4 point bending test, shown in Figure 22. The variance in the pixels per millimeter between Data Set B1 and Data Set B2 is due to slightly different set-ups of the location of the x-ray source and detector. In all cases the strain gages were placed at mid-span.

Table 3: Load and Microstrain data for Data Set B1 and B2, images scanned at 18 lp/mm

Image	Load (N)	Microstrain from Gage
Data Set B1: 72.7 pixels/mm		
04	0	0
05	84.5	805
06	120	1236
07	27	205
08	160	1640
09	258	>2358
Data Set B2: 53.8 pixels/mm		
01	0	0
02	0	0
07	133	830
09	222	1370

During the testing which produced Data Set B2, other images were taken at a lower lp/mm to investigate the result of using lower quality images which would be computationally easier to work with. Load and strain values are shown in Table 4.

Table 4: Load and Microstrain data for Data Set B3 images scanned at 6 lp/mm

Image	Load (N)	Microstrain from Gage
Data Set B3: 14.7 pixels/mm		
00	0	0
03	67	400
04	67	400
06	133	820
08	222	1370
10	302	1760

Figure 25 shows a typical deflection curve for a beam loaded at mid-span. The close-up shows the lack of smoothness in the signal due to noise and digitization. For these tests, the amount of noise was contained within roughly 3 pixels, which corresponds to less than 0.07 mm for all images captured at 18 lp/mm. Note the lack of data present in the gage location due to the presence of the cable providing the load and also the strain gage.

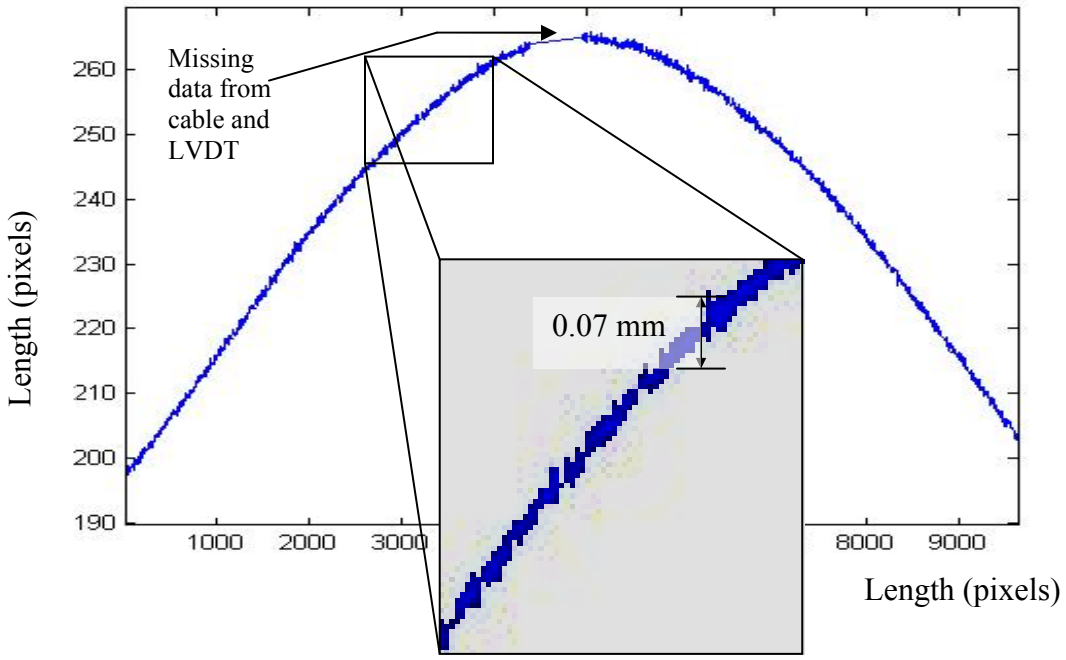


Figure 25: Typical *x-y* edge data for a 3 point bending test, with close-up of noise in data

For the use of finite difference formulas with averaging filters, the data must be continuous. Any method of generating data points would have an influence on the strain calculation dependent on the size of the averaging filter. Larger averaging filters would have less influence from the artificial data points since their influence would be averaged with many real data points. Linear interpolation between the two ends of missing data was used to generate the needed points, since attempting to fit a curve to visually match the trends in the data would require the selection of a window size and would also

influence the results. In real applications, this would not be a problem granted the data along the full edge of the dowel is usable.

The large size of the averaging filters necessary to obtain meaningful data makes the choice in size of the finite difference formulas used inconsequential. For example, using a 3 point second derivative finite difference formula yields results indistinguishable from those found from using a 9 point difference formula. Since 3 point finite difference formulas result in the least amount of data lost, they were used to calculate derivatives.

Results of the finite difference method are shown in Table 5 and Table 6 for Data Set B1 and B2. The filters were passed through the data three times. Filter sizes were selected to demonstrate the influence of their size on the calculated results, and to attempt to find a size which would consistently produce accurate results. Along with the filter size in pixels, the actual length of the filter in millimeters (along the beam length) is also noted. The calculated values closest to the measured strain are highlighted and marked with bullets, and those values were used to calculate the best percent difference displayed in the table.

Table 5: Data Set B1 results of finite difference method

Image	Microstrain from Gage	Calculated Microstrain			Best % Difference
		Averaging Filter Size (pixels, mm)			
		(501, 6.9)	(1251,17.2)	(2001,27.5)	
04	0	65	73	29	N/A
05	805	1317	1087	·981·	21.8
06	1236	1292	1376	·1232·	-0.3
07	205	316	385	·312·	52.2
08	1640	1836	1936	·1780·	8.5
09	>2358	2964	2640	2406	N/A

Table 6: Data Set B2 results of finite difference method

Image	Microstrain from Gage	Calculated Microstrain			Best % Difference
		Averaging Filter Size (pixels, mm)			
		(501, 9.3)	(1251, 23.3)	(2001,37.2)	
01	0	421	319	315	N/A
02	0	488	305	286	N/A
07	830	1257	1119	·1084·	30.6
09	1370	1860	1696	·1651·	20.5

The difference between the calculated strains in Image 01 and Image 02 demonstrates the variation of calculated strains due to the images having slightly different exposure times.

Significant strains calculated in images with no applied load indicate that either initial curvature is present in the beam prior to loading or that the edge-boundary extraction process produces artifacts which contribute to the curvature of the beam. If the curvature is actually present in the beam, then the strain from the first image should be subtracted from subsequent images, since the strain gage was set to zero just prior to capturing the image. In Data Set B1 the strain calculated in Image 04 is insignificant, however for Data Set B2 the strain calculated in the zero strain images (Image 01 and Image 02) is greater than 300 microstrain. Subtracting the averaged strain calculated for Image 01 and Image 02 of Data Set B2 from the other strain calculations significantly improves the accuracy as shown in Table 7.

Table 7: Results of finite difference method for Data Set B2, zeroed by average strain of Image 01 and Image 02

Image	Microstrain from Gage	Calculated Microstrain			Best % Difference
		Averaging Filter Size (pixels)			
		(501, 9.3)	(1251, 23.3)	(2001,37.2)	
07	830	803	·807·	784	-2.7
09	1370	1405	·1384·	1351	1.0

Figure 26 shows a typical plot of deflection and strain calculation along the beam for the 3 point loading test. Figure 27 shows a similar plot but for the 4 point bending test.

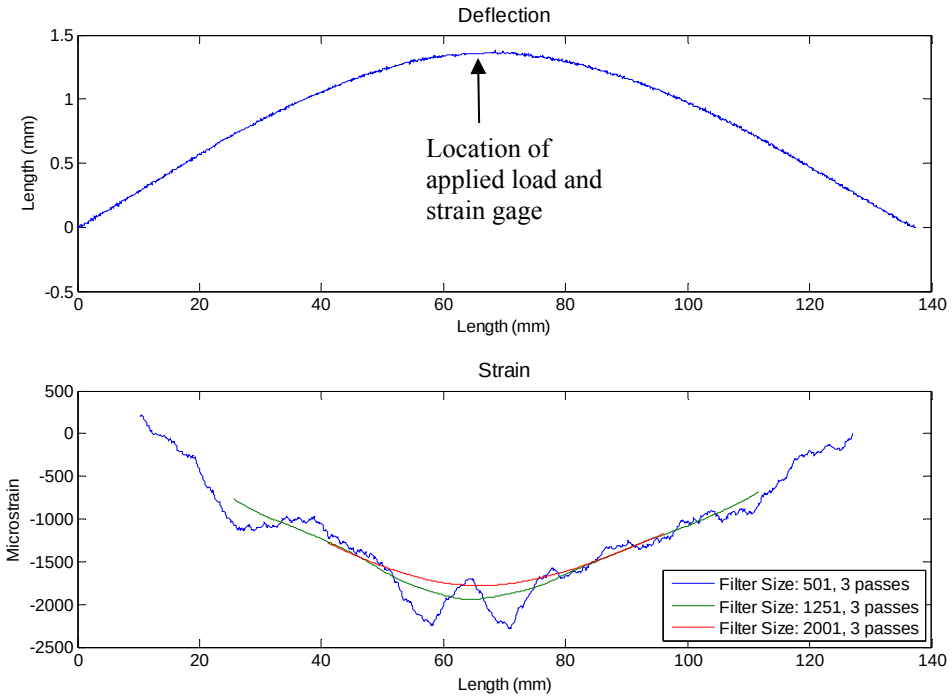


Figure 26: Deflection and strain for Data Set 1, Image 08, Gage Microstrain: 1640

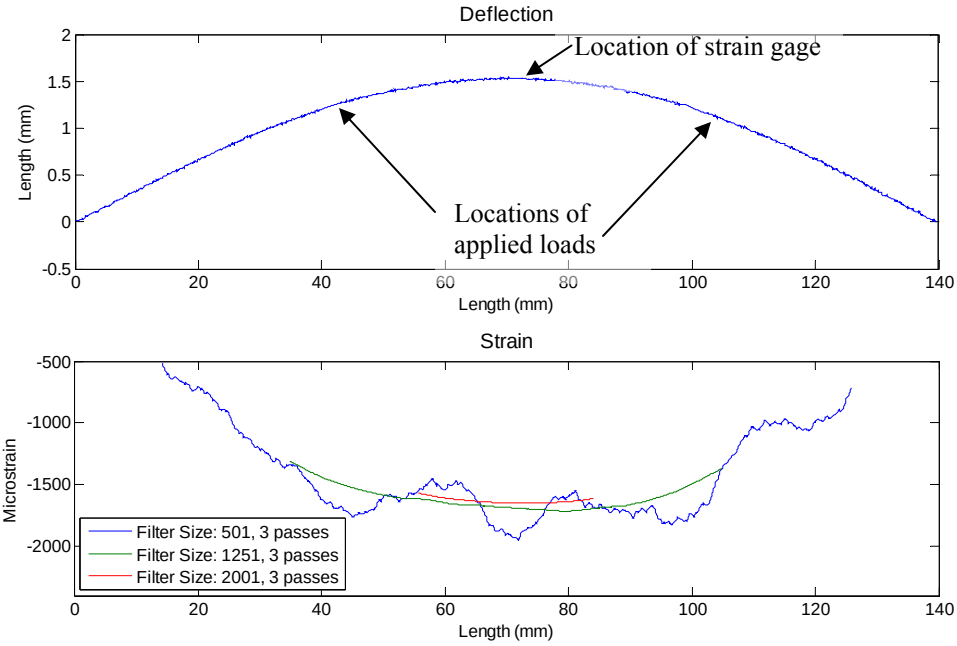


Figure 27: Deflection and strain for Data Set 2, Image 09, Gage Microstrain: 1370

The influence of the linear generated data points can be seen in the data of the smallest filter size in the region around 60 mm for Figure 26. At the beginning and end

of the linear portion the second derivative (and therefore strain) increases drastically due to an abrupt change in slope. Over the length of the linear region the strain greatly decreases since the second derivative of a linear function is zero.

In Figure 27, the constant moment/strain region caused by the four point bending is noticeable between 40-100 mm. The complete set of plots like Figure 26 and Figure 27 for all data can be seen in Appendix A. Load points and strain gage locations for all images of the same data set are the same.

Since the size of the pixels in relation to the size of the beam varies between data sets, the filter size can be selected to relate to the spatial dimensions of the beam instead of the filter size being based on a selected number of pixels. Table 8 and Table 9 show the results of the procedure of using averaging window sizes of 20 mm, 25 mm, and 30 mm on the images captured at 18 lp/mm for Data Set B1 and B2. The window sizes were selected based on the results in Table 5 and Table 6.

Table 8: Results of finite difference method for Data Set B1 using 20mm, 25mm and 30mm averaging filters

Image	Microstrain from Gage	Calculated Microstrain			Best % Difference
		Averaging Filter Size (pixels, mm)			
		(1455,20)	(1817,25)	(2181,30)	
04	0	62	39	21	N/A
05	805	1060	1009	·959·	19.1
06	1236	1343	·1268·	·1204·	± 2.6
07	205	362	330	·298·	45.4
08	1640	1896	1820	·1747·	6.5
09	>2358	2580	2465	2358	N/A

Table 9: Results of finite difference method for Data Set B2 using 20mm, 25mm and 30mm averaging filters

Image	Microstrain from Gage	Calculated Microstrain			Best % Difference
		Averaging Filter Size (pixels, mm)			
		(1075, 20)	(1345, 25)	(1615,30)	
01	0	333	325	326	N/A
02	0	330	311	304	N/A
07*	830	·797·	·797·	790	-3.9
09*	1370	·1373·	·1373·	1366	0.02

*** indicates data has been zeroed by the average of calculated strain of Image 01 and Image 02**

The results of using averaging filter sizes based on spatial length with Data Set B3 images (which were captured at 6 lp/mm) are shown in Table 10.

Table 10: Results of finite difference method for Data Set 3 using 20mm, 25mm and 30mm averaging filters

Image	Microstrain from Gage	Calculated Microstrain			Best % Difference
		Averaging Filter Size (pixels, mm)			
		(1455,20)	(1817,25)	(2181,30)	
04	0	418	376	328	N/A
05*	400	288	309	·365·	-8.8
06*	400	321	371	·415·	3.8
07*	820	741	805	·832·	1.5
08*	1370	1236	1278	·1319·	-3.7
09*	1760	1552	1654	·1711·	-2.8

*** indicates data has been zeroed by the calculated strain of Image 04**

Figure 28 shows the deflection and calculated strain from Image 08 of Data Set B3. The graphs of the other images from Data Set B3 can be found in Appendix A.

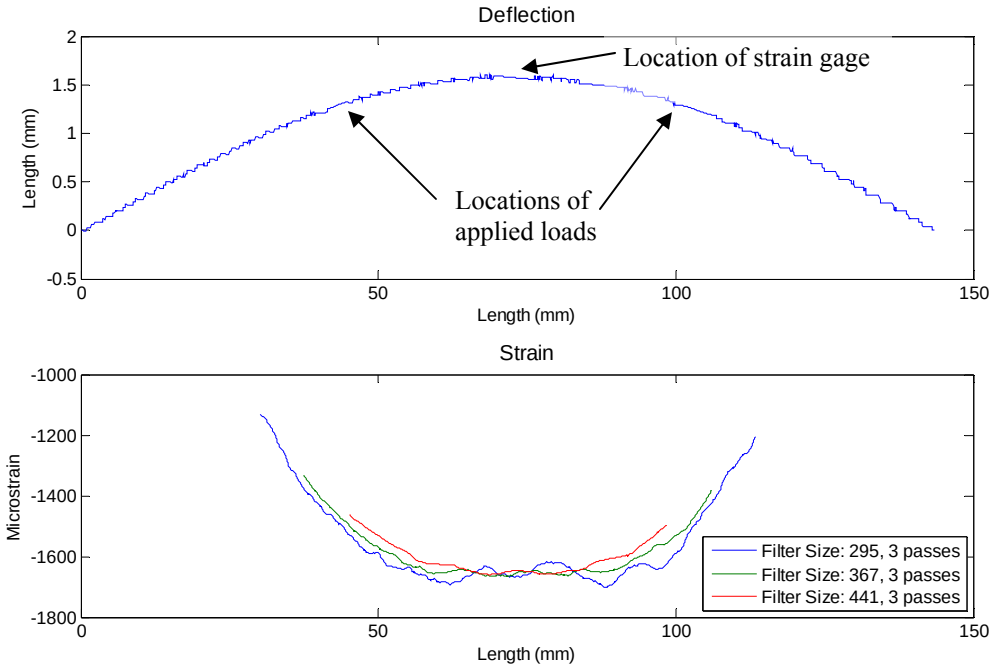


Figure 28: Data Set 3 Image 08 deflection and strain calculation

Overall, the results from all data sets using a 30 mm filter size are the best, even for Data Set B3 which was captured at 6 lp/mm and contains only about 1/4th as many data points as the other two. The region of constant moment/strain is again present, especially in the images which contain large deflections.

While requiring user inputs which could not be selected beforehand, the results have shown that the extracted edge data contains the necessary information to calculate accurate strain values. Aside from the selection of an appropriate filter size, another problem with the use of averaging filters is the extensive amount of data is consumed by the process. The data began with a length of about 140 mm, however after the strain calculation using the 30 mm averaging filter three times, the data was reduced to roughly 50 mm. This would be unacceptable for a connection test, since it would be desirable to obtain an accurate strain distribution over a larger area. Also, the large amount of

averaging done to produce accurate strain values would be unfavorable in situations where the strain actually does have large variations over short distances.

It should be noted that the larger averaging filters simply suppress the oscillations in strain, and therefore the actual strain can generally be considered to be somewhere within the band produced by the oscillations calculated by the smaller averaging filters. Figure 29 shows banded oscillations of strain data which has been filtered by a small averaging filter three times. As can be seen, this allows the attainment of quantitative information across a wider length of the beam than if a larger averaging filter was used to suppress the oscillations.

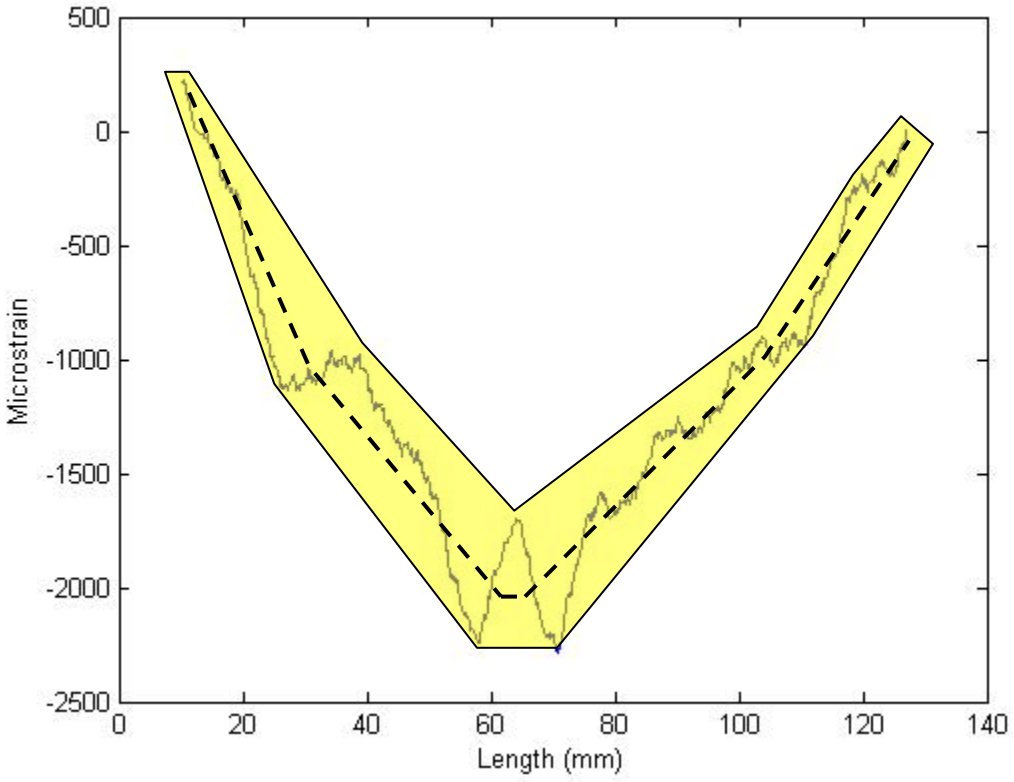


Figure 29: Bounding large oscillations of strain calculated using a small averaging filter. True gage strain at mid-span = 1640 microstrain.

Without any information about the loading, the true strain could be considered to be represented by any appropriate polynomial which fits within the bounds provided by the calculation of strain from the digital image; however, the problem still exists in selecting an appropriate filter size to create the bounded strain profile.

With it shown that meaningful information could be found from the obtainable image data, other differentiation methods were investigated in an attempt to find the methodology which would yield the best results and be the least influenced by user input. The data used to generate Figure 30 through Figure 33 is from Data Set B1 Image 06, which had a gage microstrain reading of 1236 at mid-span. Results and trends using the following methods for other images were typical of this selected image.

Polynomial fitting to all *x*-*y* data points from an image was not investigated since the results would be greatly influenced by the order selected. While the polynomial order to use for the simple beam test is known, it will not be known for the connection tests. Instead, a point at which a strain calculation is desired could be selected and a low order polynomial (order 2 through 4) could be fit to a neighborhood window to approximate the derivative at that location. The use of Savitzky-Golay filters [20] produces the same result at every data point because the filter provides a polynomial fitting to a window at each data point to calculate the second derivative.

In general, the results of using Savitzky-Golay filters were very similar to those using finite difference formulas and averaging filters, however more parameters were required. The results not only depended on filter size, but also on the polynomial order selected.

The influence of window size on calculated strain values is the same as for the method involving finite difference formulas and averaging filters; increasing the window size decreases the calculated strains. Increasing the order of the filter increases the oscillatory nature of the calculated strain, as seen in Figure 30.

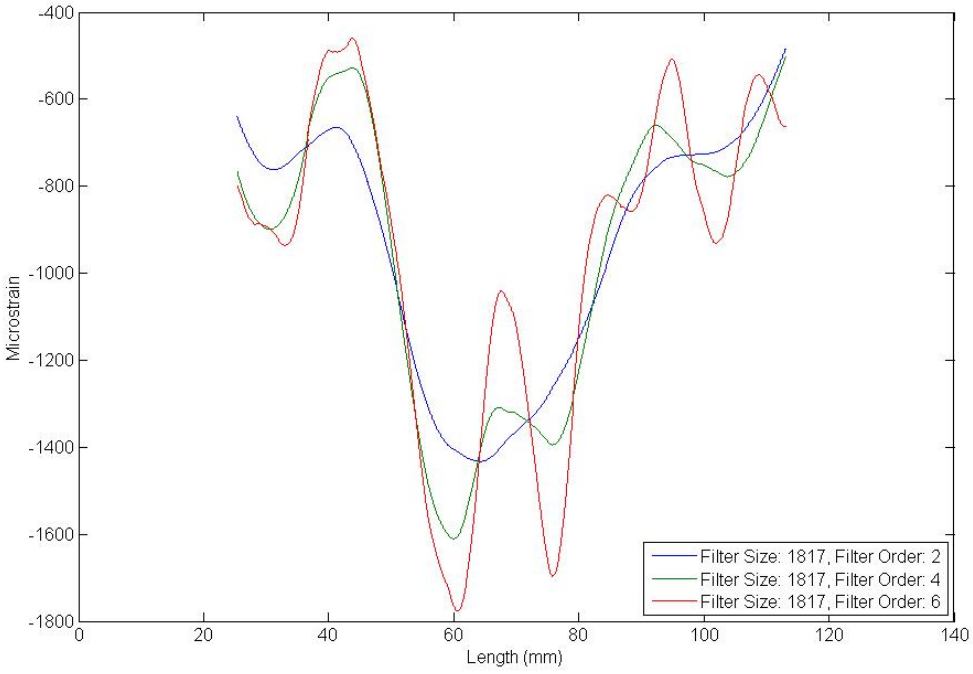


Figure 30: Influence of Savitzky-Golay filter order on strain calculation

The parameter which must be chosen for the fitting of sine and cosine terms to the edge data is the number of terms to include. The influence of this parameter is shown in Figure 31. Including more terms increases the amount and magnitude of the oscillations in strain along the beam. The strains calculated by using only 3 or 6 terms provides the best approximation of the strain distribution from the simple bending test. More terms would certainly be needed for connection tests where the strain distribution would be more complex, however determining the correct value a priori would not be possible without knowing the true behavior of the strain.

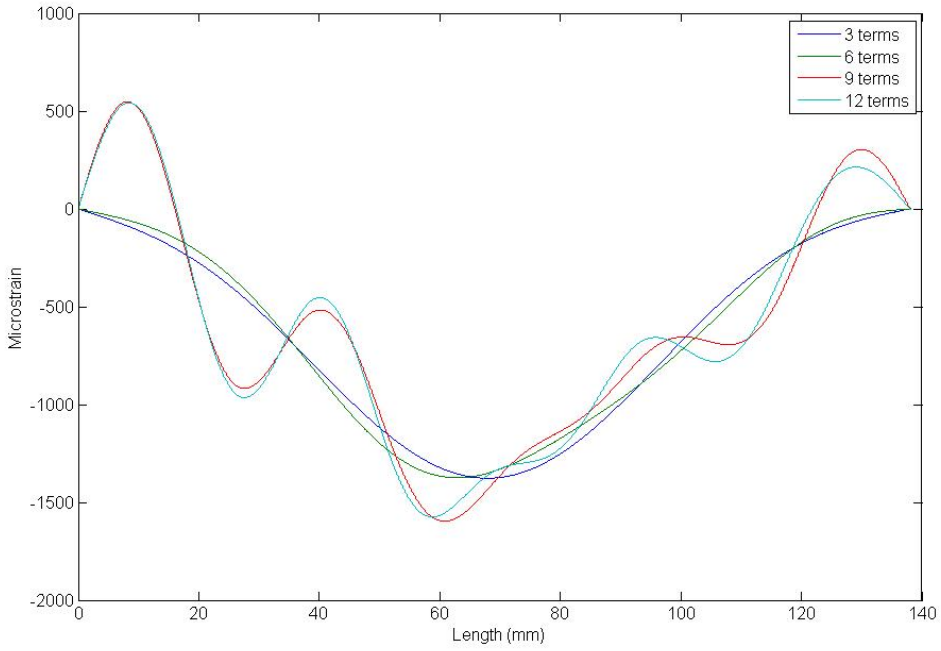


Figure 31: Influence of number of terms in sine and cosine fit of edge data on strain calculation

The Matlab code used to carry out the investigated mollification method comes directly from [23] and [24]. The method involves mollifying the data with a Gaussian function which has a standard deviation that is selected automatically and is influenced by the amount of noise in the data. The derivatives of the mollified data can then be found by finite difference formulas, and their result can again be mollified if desired. The results are similar to those from the finite difference method with averaging filters of the size which yielded accurate results. No data is lost in the procedure, although the calculated strains near the beginning and end of data are clearly an artifact of the procedure. This is shown in Figure 32.

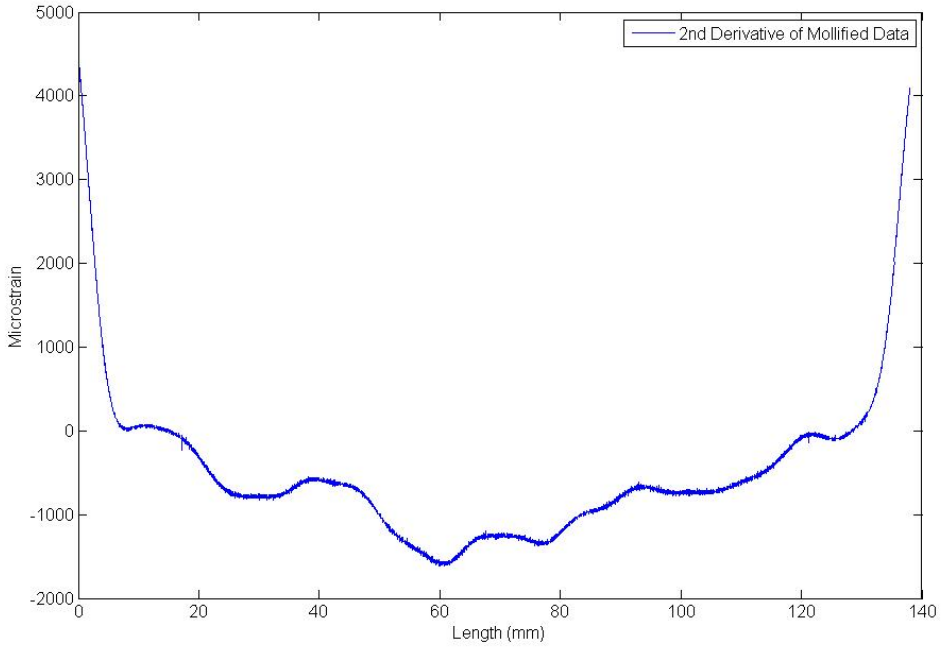


Figure 32: Strain calculated by using finite difference formulas on the mollified *x-y* edge data

The amount data points which are contaminated by the algorithm and need to be removed from each end is equal to the length of the automatically selected filter window size, which is an output from the Matlab mollification procedure. Like with the finite difference method with small averaging filters, the calculated strain contains oscillations about the theoretically smooth value, and the true strain in the beam can be thought of as being bounded by the oscillations. A close-up of Figure 32 which disregards the strains calculated at the ends of the beam is shown in Figure 33.

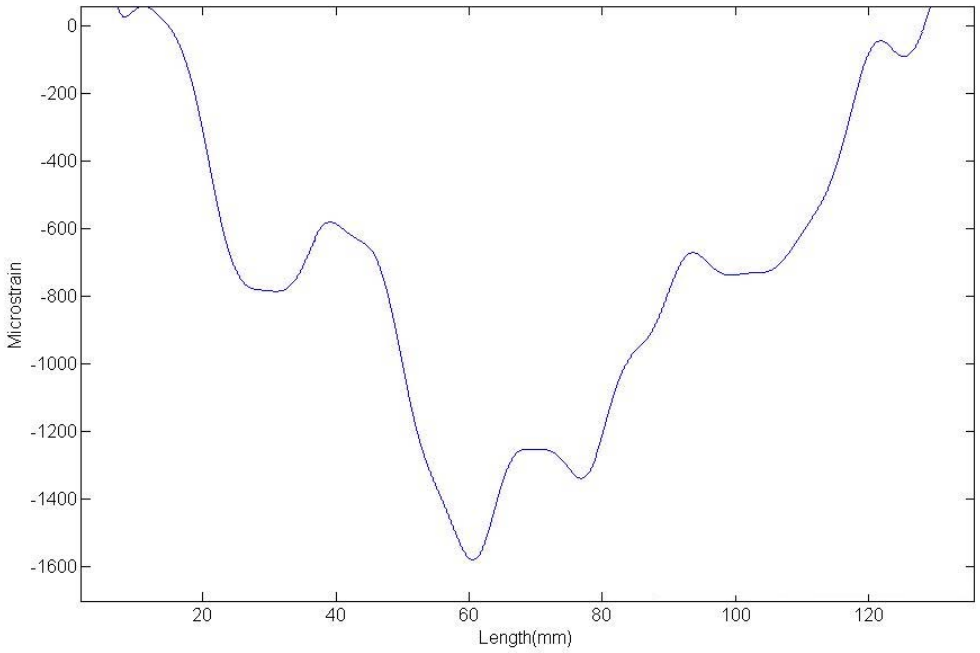


Figure 33: Close-up of Figure 32, disregarding the strain calculated at ends

The large oscillations at the mid-span of the beam (around 50-80 mm) are due to the linearly interpolated points which were used to fill in the missing data due to the loading cable. Again, this is because the slope (first derivative) changes abruptly at the beginning and end of the linearly interpolated data points, resulting in a large change in the second derivative. Graphs of strain calculations from all the images using the mollification method can be seen in Appendix A.

It is difficult to compare the values from the gage directly with the values from the algorithm because of the influence of the linearly interpolated points have on the strain in the vicinity of the location of the strain gage. However, despite this the oscillations in calculated strain at mid-span typically bound the actual strain gage value as long as the strain data is zeroed, as shown in Figure 34 for one of the 4 point bending test images. The oscillations in this case have amplitudes of roughly 400 microstrain, but

amplitudes in other images were found to be up to 1000 microstrain. These large oscillations can mostly be attributed to the influence of the linearly interpolated points at mid-span on the calculated second derivative.

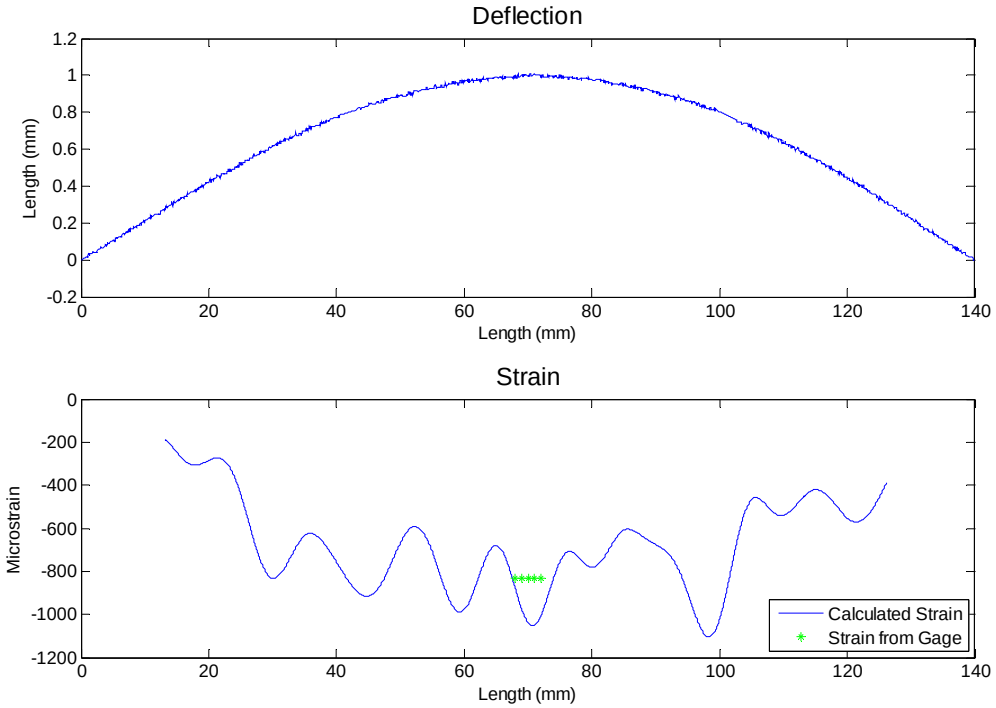


Figure 34: True strain bounded by oscillations in calculated strain (Data Set B2- Image 07)

From examining all of the data plots from the mollification method, it appears that the frequency of the oscillations is related to the rate of change of the strain. In regions where the strain is relatively constant there are frequent oscillations, as seen in all of the plots for Data Set B2 and B3, where there was a constant moment from roughly 40mm-110mm. In these cases, the correct strain is typically the average value of the oscillation, or slightly lower. There appear to be less oscillations, and of lower amplitudes, in regions where the strain is not constant but is changing, as seen in the data plots for Data Set B1.

The mollification method combined with finite difference formulas to calculate derivatives provides the most robust and least user influenced method for determining the strain along the dowel. Therefore, it was decided that this was the most suitable method to be used in the subsequent mock connection feasibility test.

The created Matlab functions and other ".m" files used to produce the figures can be found in Appendix C.

Chapter 6. Feasibility Study II: Mock Connection

6.1. Test Set-up

This test was run to ensure that the methodology developed for the simply supported beam test could be applied to connections, which have a much more complex strain distribution than that of a simply supported beam under point loading. To ensure that strain values from digital image processing match the actual values which would be found in a connection, a mock connection instrumented with strain gages was assembled and tested.

Care had to be taken to ensure that the gages were not damaged during assembly of the connection, and would not become damaged during the first application of load. The solution decided upon involved using acrylic glass (Polymethyl methacrylate, or PMMA) instead of wood as the material for the side and main members. As opposed to wood, PMMA can be easily machined, is isotropic, will not be influenced by temperature and moisture conditions, and the pieces would potentially be reusable. The PMMA pieces were machined to include grooves for the strain gages to safely exit the connection from their position on the dowel. The dowel used in the PMMA connection was the same type used for the simple beam test; a piece of 4.76 mm square steel. A simplified schematic of the mock connection center and edge pieces can be seen in Figure 35.

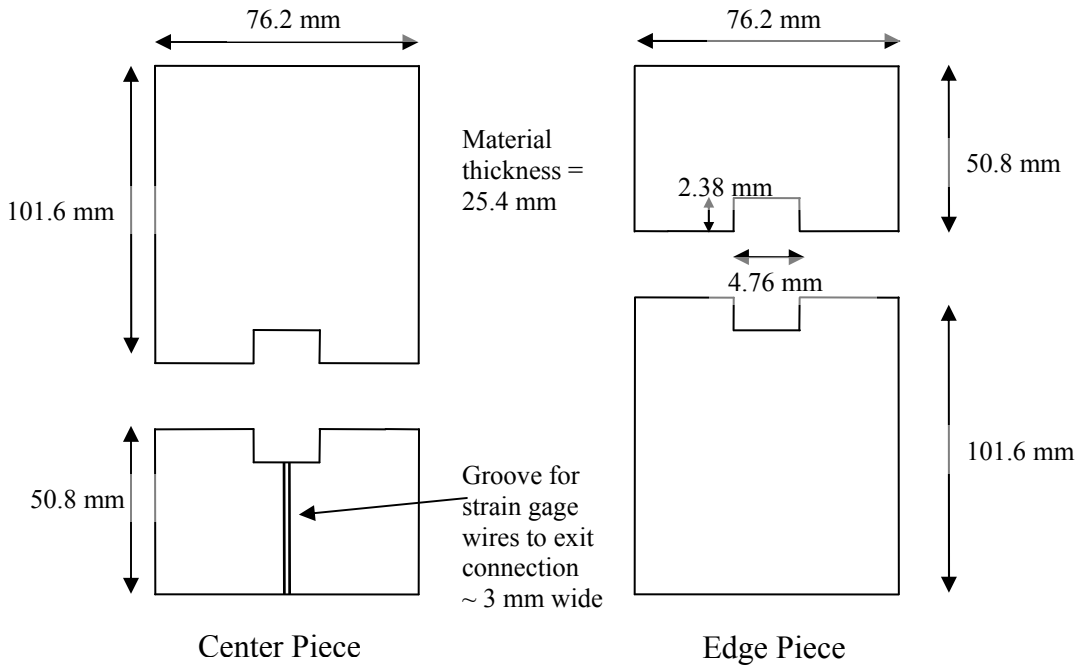


Figure 35: Schematic of PMMA pieces used to assemble mock connection (not to scale)

When assembling the connection, interfaces around the dowel were epoxied together. The fully assembled connection consisted of two center pieces epoxied together around the dowel, with one edge piece on either side. For the first test (Data Set MC1, "MC" for mock connection), strain gages were placed at the interface of the edge and center pieces, and also at the interface of the two center pieces, as shown in Figure 36.

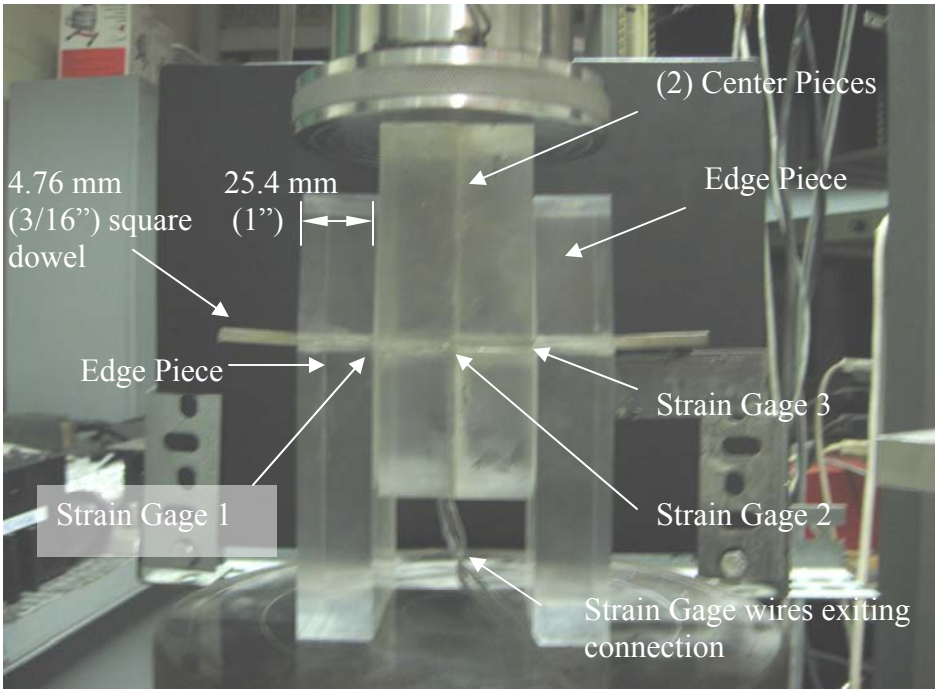
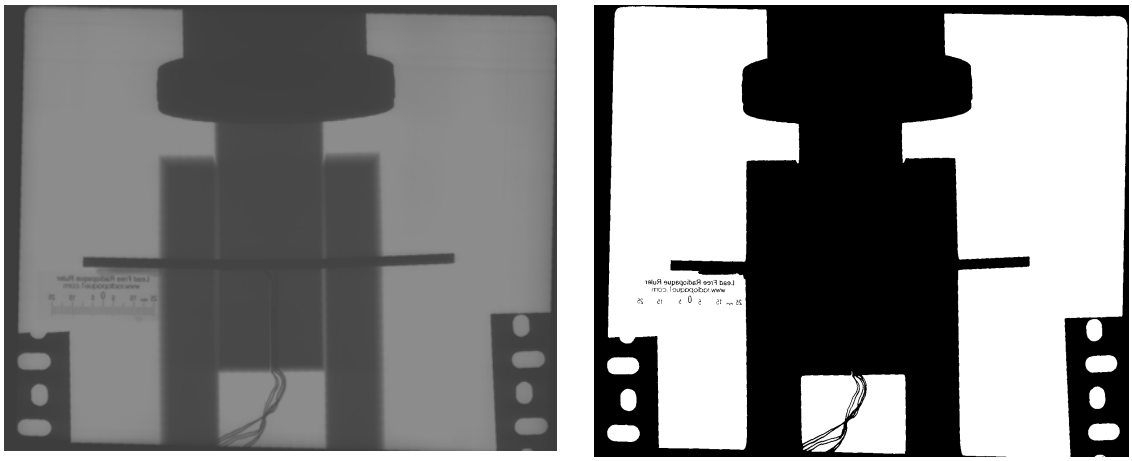


Figure 36: Strain gage locations in PMMA connection

In the subsequent test (Data Set MC2), only two gages were used (leaving out the center gage), due to the difficulty encountered when attempting to assemble the connection without damaging the gages.

6.2. Mock Connection Image Processing

Simply segmenting the images from the mock connection using Otsu's method does not result in the image being separated into the object and background required.



(a) (b)

Figure 37: (a) PMMA connection test image. (b) image in (a) segmented by Otsu's method

This is mostly due to the fact that unlike the simple beam test images, where there were only two distinct intensity regions (bimodal histogram), for a connection there are three regions (tri-modal histogram). This can be seen by examining the histograms of typical images from each type of test, shown in Figure 38.

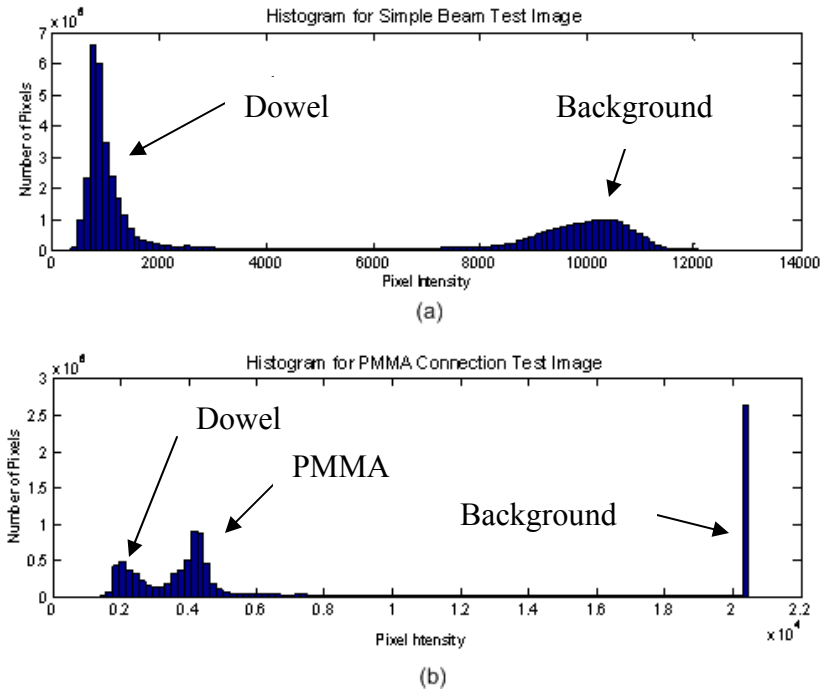


Figure 38: Comparison of histograms for simple beam test and connection test images

The difference in intensity ranges is due to the different exposure times. In (a) of Figure 38, the exposure time was short, causing background pixels to have a large variance in intensity, while in (b) the exposure time was longer, which caused all background pixels to be roughly the same intensity (about 20,000).

Despite this dilemma, the visibility of edge data to the human eye clearly indicates that the edges can be extracted image by some type of algorithm. From Figure 38 (b) it appears that by eliminating the background pixels from the histogram, Otsu's method could be used to separate the dowel from the PMMA. However, this is not the case because some of the pixel intensities in the dowel outside of the connection are the same as PMMA pixel intensities inside the connection. Therefore, any global thresholding would either result in pixels being left out of the dowel, or PMMA pixels being included as part of the dowel. Adaptive thresholding operations could have potentially been used, but it was decided to use known information about the system to help facilitate the development of an algorithm to extract the dowel edge.

The algorithm developed relies on the fact that the dowel edge can be approximated as an ideal step edge (i.e. the edge is the location where difference between adjacent pixels is a local maximum), and that the distance between the top and bottom edge of the dowel is roughly known. The developed algorithm was comprised of five steps.

Step 1. Crop the image to a small vicinity around the dowel and perform a smoothing operation to remove small local maximums

Step 2. Run a difference operation down each column of the cropped image, i.e. $f'(x) = f(x+1) - f(x)$, and find all local maximum values

Step 3. Input dowel depth (d), pixels per inch (ppi), and a plus/minus percent (per) value for the search. Therefore, the number of pixels between the top and bottom of the dowel is $p = d \times ppi$.

Step 4. For each column of the image, find points which are a local maximum difference value and are within $p \pm p \times per$ of each other

Step 5. Extract the maximum value of the found points and its partner at distance $p \pm p \times per$ away

Step 1 is not entirely necessary, but greatly decreases the computational time and has no adverse effects on the extracted edge values. Step 2 treats each column as a 1-d signal and determines the difference between adjacent pixels and also runs an operation to find the local maximums. Values less than a threshold value (chosen as 10) were set to 0. This eliminates the small changes in pixel intensity which do not correspond to the dowel edges and significantly decreases the computational time. For images which contain low contrast it can be necessary to select a lower value, since in low contrast images the edges are not as prominent due to there being less of a "step" at the edge. In Step 3 the dimensional inputs required for finding the edge are entered. The percent value typically used could be varied to account for the slope of the dowel in the image. The value could also be adjusted if in some areas the algorithm did not find edge pixels. Step 4 picks out the pixels which are on the edge of the dowel and sets them equal to 1, while setting the rest of the pixels in the image equal to 0. Finally, Step 5 finds and extracts the indices of pixels whose value is 1 from the binary image.

Figure 39 shows the process of obtaining the edges from one column of a connection image.

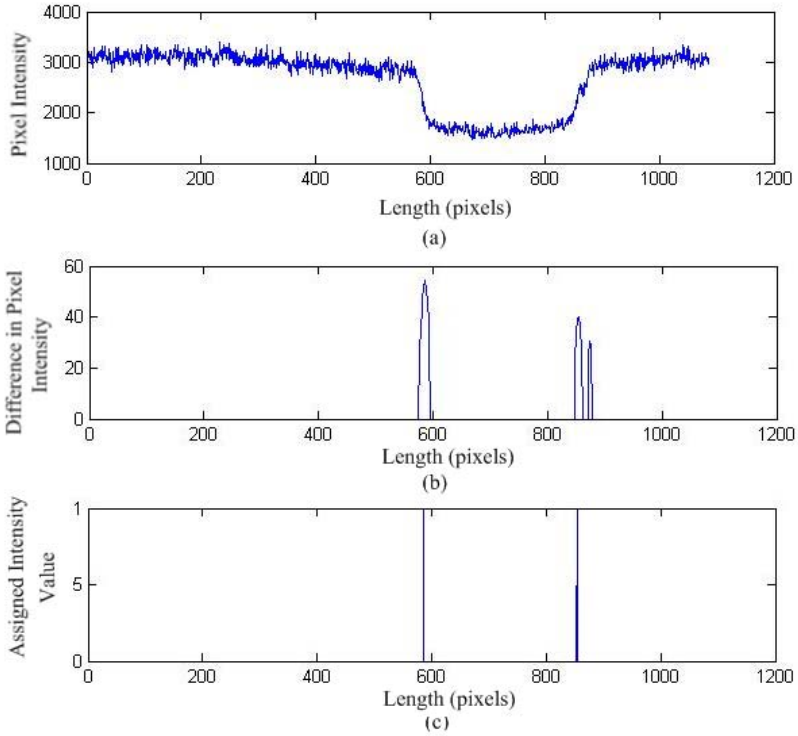


Figure 39: Example of algorithm to extract dowel boundary from connection images

Part (a) is a vertical intensity profile (i.e. the pixel intensity values down a single column of pixels in the connection). The intensity value of the PMMA pixels stays roughly constant around 3000 and the steel dowel is visible by the drop in intensity to less than 2000. In (b) the difference operation has been run on (a), values less than 10 have been set to zero, and only the maximum difference value and its partners within the given range ($p \pm p \times per$) are selected and shown. Finally, in (c) if there are two partner pixels associated with the maximum then the larger of the two partner pixels is selected. These two pixels are then assigned a value of 1, while all other pixels are assigned a value of 0. The indices of all the pixels of value 1 are then extracted and separated into the top and bottom edge, and then averaged together. The annotated Matlab ".m" file which performs this operation can be seen in Appendix C.

To verify that the boundary extraction from the method used previously on the simple beam tests (Otsu's method) and the newly developed method are consistent, the new method was run on the simple beam tests. Figure 40 shows a comparison of the boundaries extracted by the two methods, and also the calculated strains.

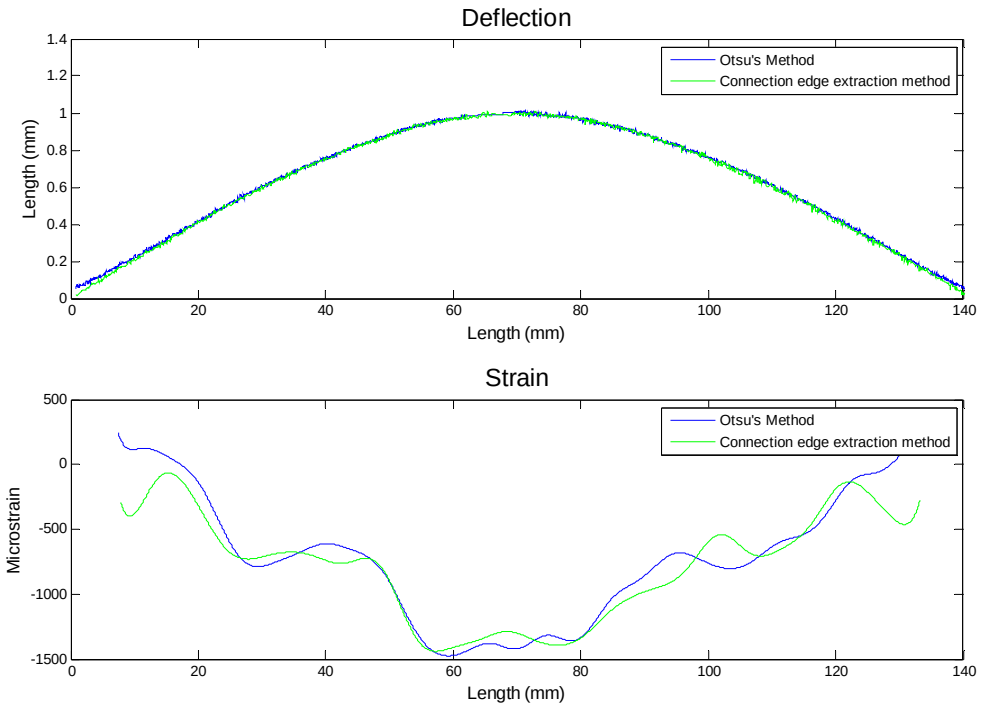


Figure 40: Comparison of *x-y* data and strain calculation from two extraction methods

While the deflection values in the two data sets in Figure 40 have many differences in their pixel to pixel appearance, the overall trends in the slopes and curvatures are very similar, which results in very similar calculations of strain values.

For the mock connections, much of the extracted bottom edge is not actually part of the dowel, but is either the strain gage or the strain gage wires. This is shown in Figure 41. The top edge is below the bottom edge due to the coordinate system of an image having an origin at the upper-left corner.

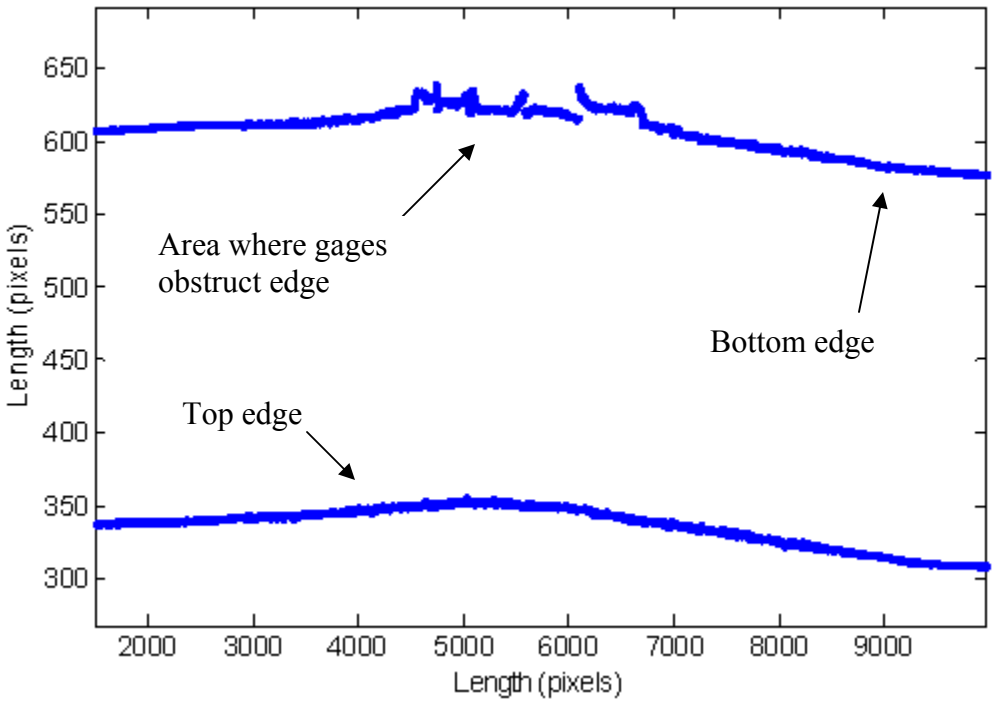


Figure 41: Extracted top and bottom edges for a deflected image

Averaging the top and bottom edges together would result in the gages and wires still having a significant impact on the dowel's calculated strain, while averaging only areas where the gages are not obstructing the edge would inevitably lead to small but problematic artifacts at the onset or offset of the averaging locations. For this reason, the unobstructed top edge was used to calculate the strain along the dowel.

6.3. Mock Connection Results

Two data sets were collected using a PMMA mock connection. Once again, the strain calculated from an image captured prior to loading when the strain gages were zeroed was subtracted from the rest of the images to calculate a strain to compare with the gage strain. Figure 42 and Figure 43 show the deflections and calculated strain distributions for all of the images from Data Set MC1.

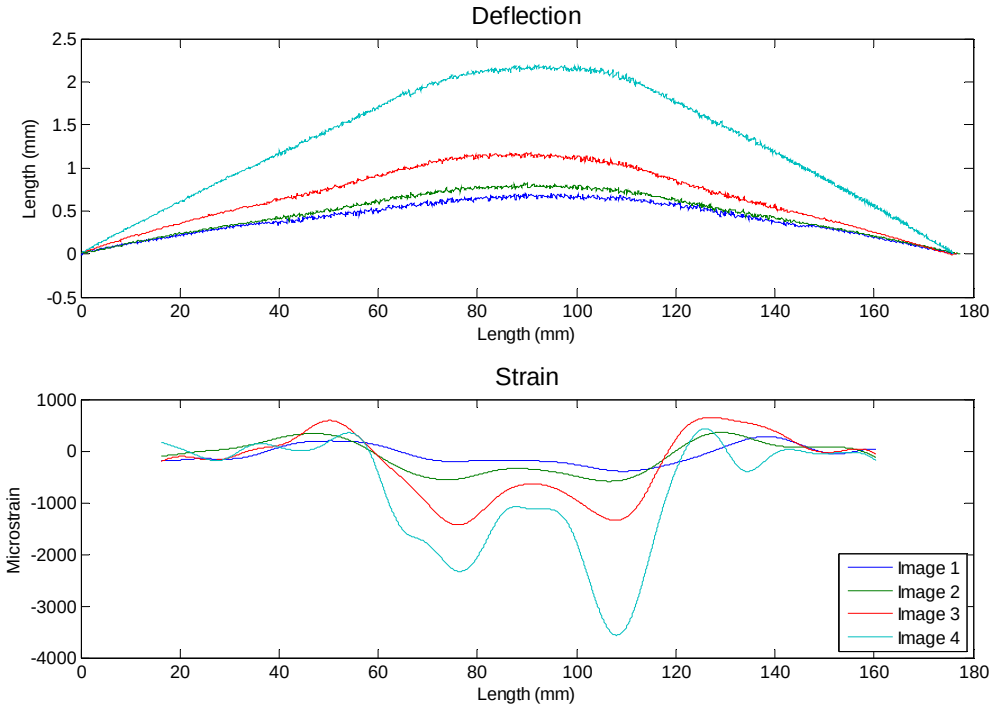


Figure 42: Deflection and calculated strains for Data Set MC1-Images 1-4

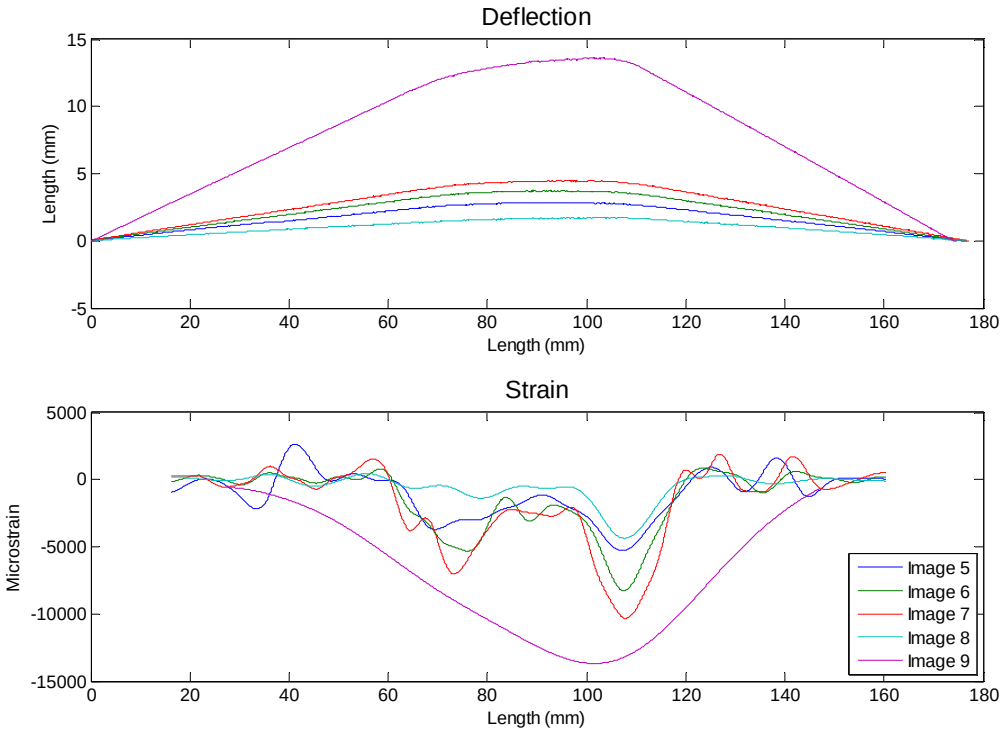


Figure 43: Deflections and strain calculations for Data Set MC1-Images 5-9

The progression of deflections and strains with the increase in applied load is clearly shown in Figure 42 and Figure 43. By even as early as Image 2 it is apparent where the yield points will occur. These large peaks in strain are consistent throughout the entire loading of the dowel, with the exception of Image 9, where the dowel was loaded to cause severe yielding. By this point, the small angle theory used to implement Equation 7 has been violated, and the curvature should be calculated by its true definition found in Equation 5.

The smaller oscillations in strain make determining minor trends in the strain data fruitless since it is not known whether they are truly reflecting the behavior, or are simply artifacts of the algorithm (where the calculated strain is oscillating about the true value). However, since these smaller oscillations appear to occur at different locations for each of the images it is acceptable to assume that they are simply artifacts of the strain extraction procedure.

Interpreting the results as having small oscillations which fluctuate about the true result and large oscillations which represent the true behavior gives the most meaningful and useful interpretation of the data. A figure showing the bounded oscillations for a mock connection test image is shown in Figure 44.

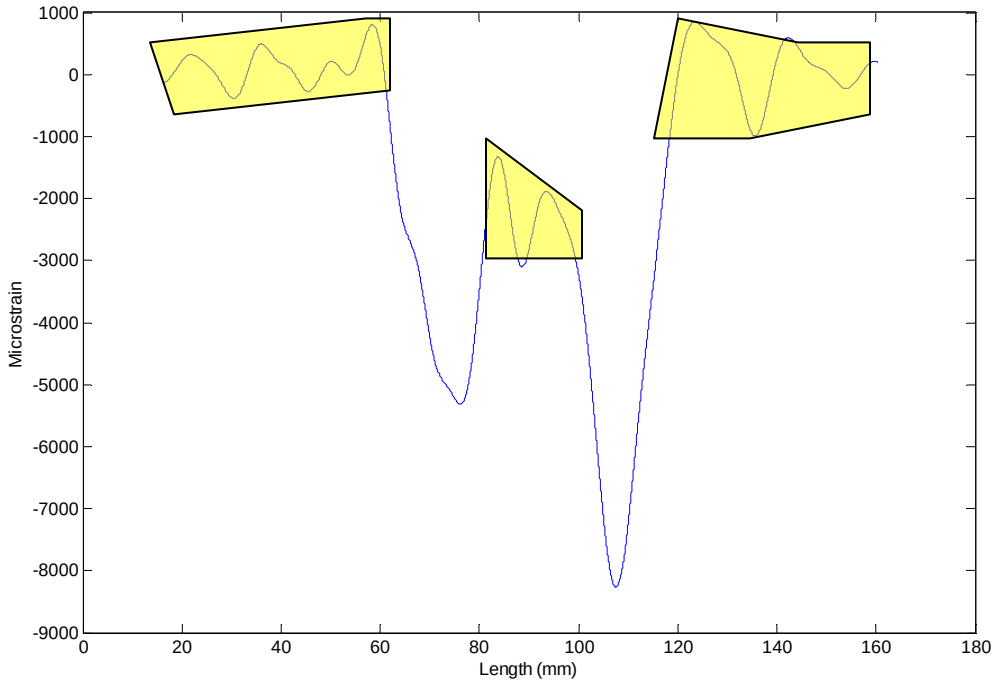


Figure 44: Bounded small oscillations for Data Set MC1- Image 6

It should be noted that examining the first derivative of the deflection can lead to some insight about the behavior of the strain as well, since the artifacts and oscillations are not as pronounced in the first derivative. In Figure 45, the first derivative of the deflection used to calculate the strain in Figure 44 is shown.

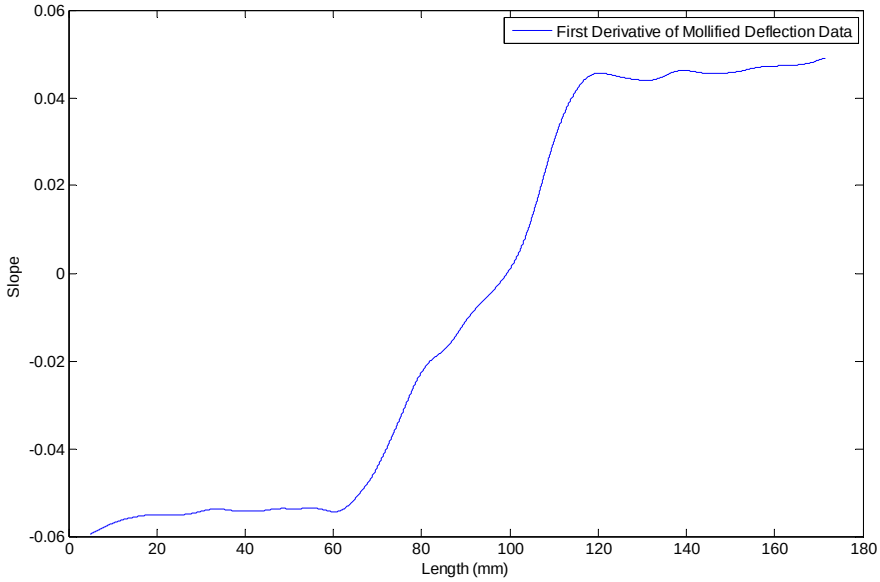


Figure 45: First derivative of mollified data from Data Set MC1- Image 6

All strains before 40 mm and after 140 mm should be equal to zero, since at these locations the dowel is outside of the connection where it is not being loaded or restricted from displacing. While the strain oscillates about zero from 20 mm to 60 mm, an inspection of the first derivative indicates that the general trend in the slope is horizontal and constant, and therefore the strain should be zero. It is impossible to discern information about the behavior right around 60 mm, where the slope decreases slightly before increasing rapidly. The slight decrease could be either another oscillation about the true value or could be the actual behavior. This would correspond to a positive valued strain before the sharp change to negative strain where the yield points are found.

For the calculated strains from the large deflection images (i.e. Image 9) it appears that the amount of smoothing which takes place suppresses the actual behavior of the strain in the dowel. Once again, by looking at the first derivative a better interpretation of the data can be made, as can be seen in Figure 46.

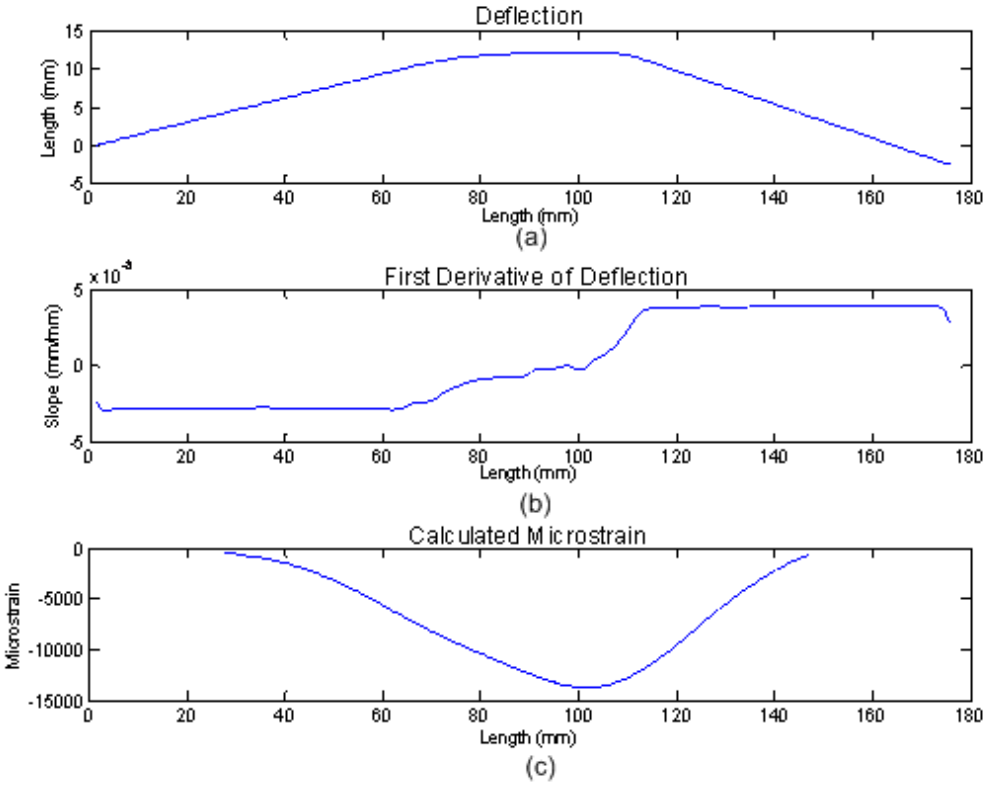


Figure 46: Deflection, first derivative and strain for Data Set MC1- Image 9

In part (b), the first derivative is extremely flat up to roughly 60 mm and after 120 mm. The smoothing of the mollification procedure causes the addition of significant slope in these regions, and this is reflected in the calculation of strains which are not equal to zero.

Figure 47 and Figure 48 show a comparison of the calculated strain with the measured strain for Data Set MC1- Image 7 and for Data Set MC2- Image 4, respectively. Comparisons for the other images from both Data Set MC1 and Data Set MC2 can be seen in Appendix B.

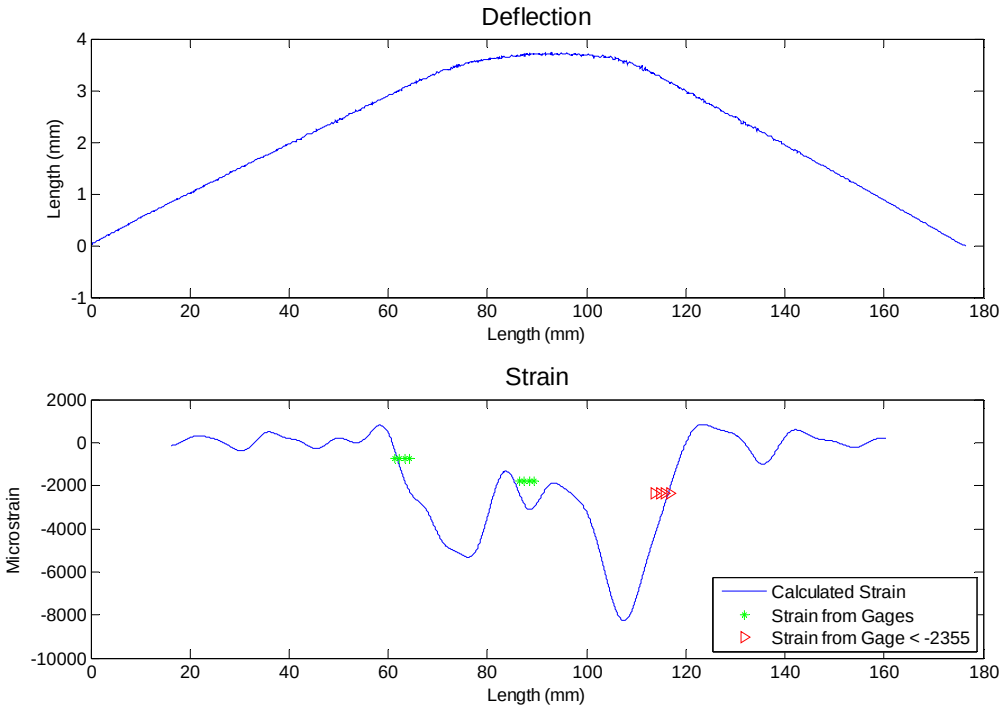


Figure 47: Deflection and strain calculation for Data Set MC1-Image 7, with comparison to measured strain

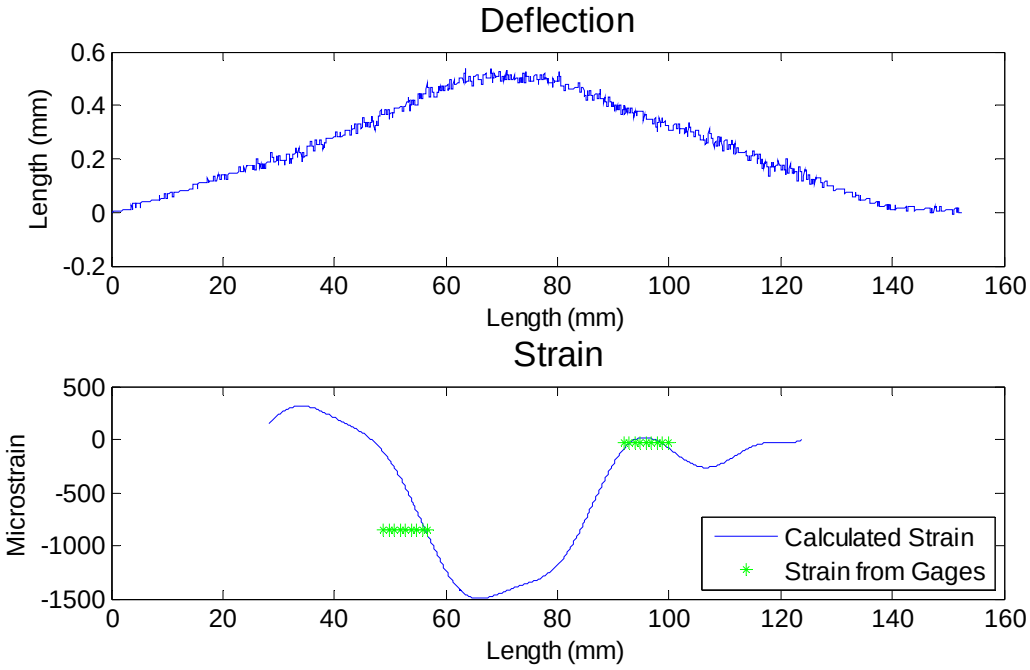


Figure 48: Deflection and strain calculation for Data Set MC2-Image 4, with comparison to measured strain

Despite the more complex geometry in the mock-connection when compared with the simply supported beam test, the calculated strains still compare well with values found from the gages. In many cases the calculated strain passes through or very near to the values found from the gages, indicating that the strains calculated from the image edge data is a good approximation to the actual strains measured on the dowel. In the other cases, the calculated strain correctly show the trends which were found from the gages and were consistent with the evolution of strain along the dowel between subsequent images. As an example, see Figure 49.

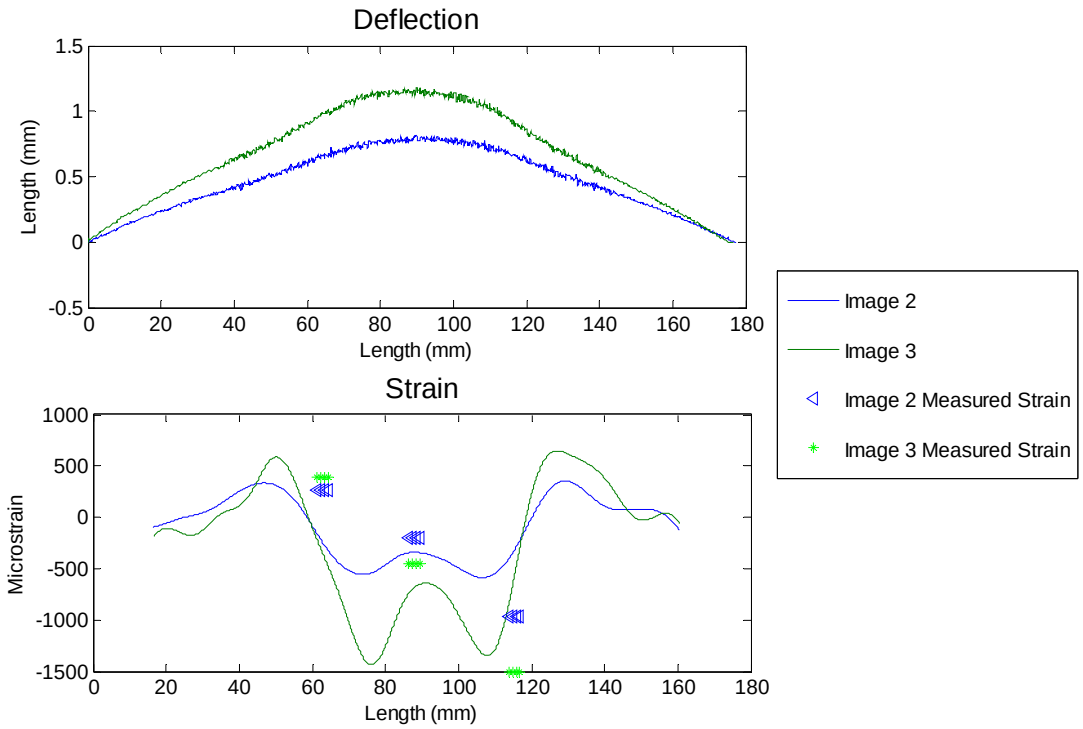


Figure 49: Comparison of measured and calculated strain for Data Set MC1- Image 2 and Image 3

The strains are significantly over or underestimated in each of the gage locations for both images. Despite this, the calculated strains correctly portray how the strains changed during the loading between the capturing of Image 2 and Image 3, both in direction and relative magnitude. The difference between the measured strains of the

center and right gages can be seen to be roughly equal to the change between the calculated strains. The situation is more complex for the left gage, where the gage is located almost directly on an inflection point. The discrepancies in calculated strain can most likely be attributed to the large amount of data smoothing required to calculate smooth second derivatives for small deflections. The smoothing procedure does not allow large changes in strain over short distances (i.e. large strain gradients). The effect of shear deformations on the behavior also contributes to the difference between the calculated and measured strains.

6.4. Developed Visualizations

The information calculated from the algorithm can be presented in numerous ways to help study the overall behavior of the strain in the dowel. Two noteworthy examples are shown in Figure 50 and Figure 51.

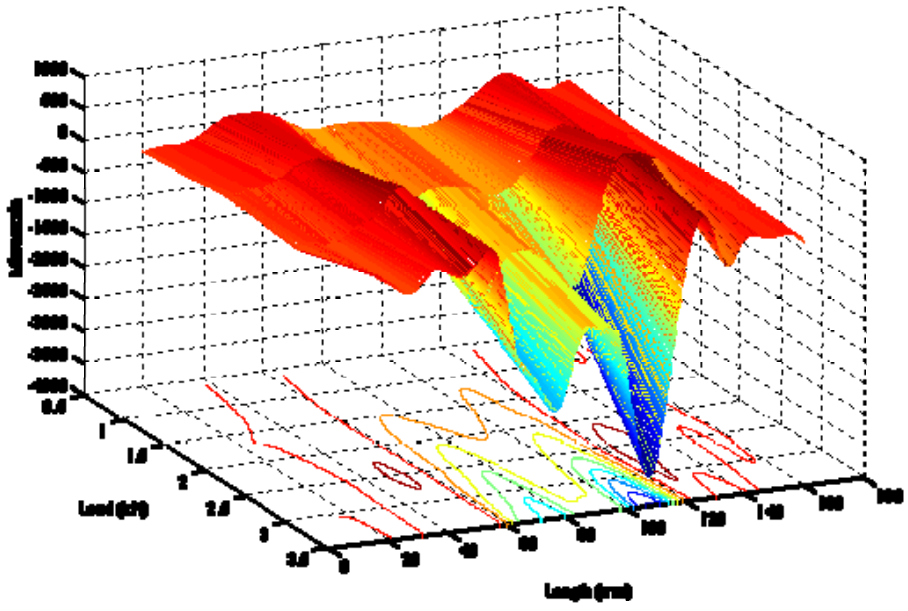


Figure 50: Mesh plot of calculated strain over the first 4 loaded images of Data Set MC1 (with 2-D contour plot)

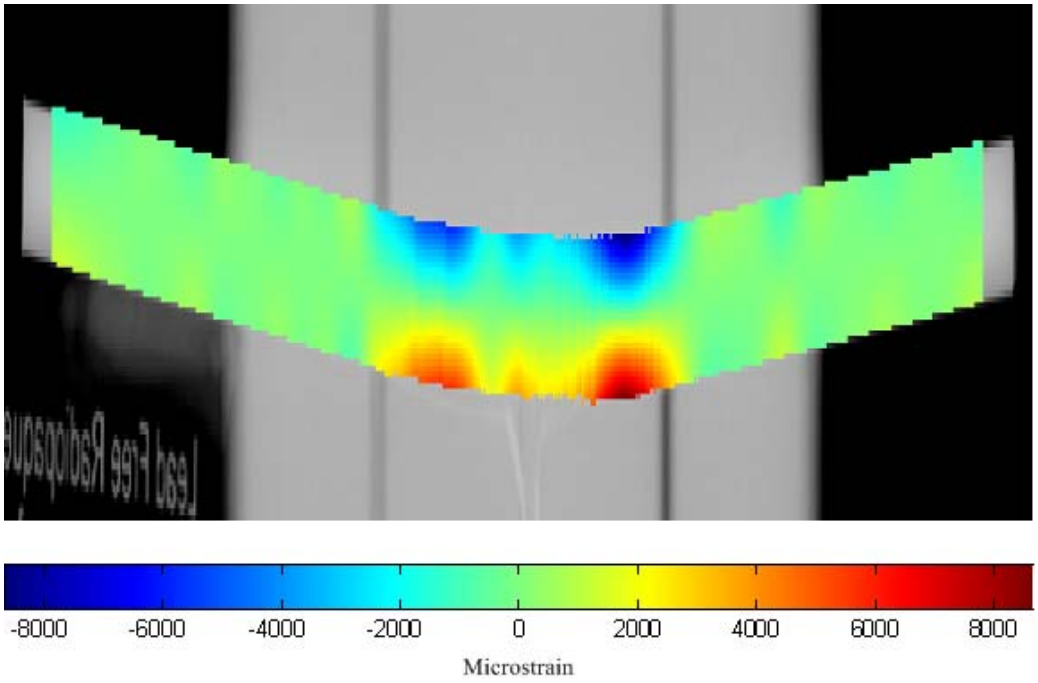


Figure 51: Overlay of microstrain values on the x-ray image for Data Set MC1-Image 6 [refer to Figure 43 for plotted strain data]

In Figure 50, the calculations of strain from Images 1 through 4 are combined with the load data to create a "mesh" plot. The change in strain in relation to the applied load can be seen for all locations along the dowel. A contour plot is also shown below the mesh.

Figure 51 shows an x-ray image with an overlay of the calculated strain values. The strain varies linearly with the depth, as in Equation 7 . It should be noted that the image has been stretched in the *y*-direction for better viewing. Also, a color bar is present to give quantitative information about the colors of the pixels. Plots like this would be ideal for visualizing the strain information calculated from a succession of images captured from digital video x-ray detectors. This could be done for both monotonic tests, and cyclic tests to see the changing strains caused by the load reversals.

Generating stress information is straightforward if the dowel is treated as having a bilinear stress-strain curve. Alternatively, if the entire stress-strain relationship were known, the stress values could be found by matching the correct stress values with the calculated strains. Examples are shown in Figure 52 and Figure 53 for an assumed Young's Modulus of 200000 MPa (29000 ksi) and an assumed yield stress of 250 MPa (36 ksi).

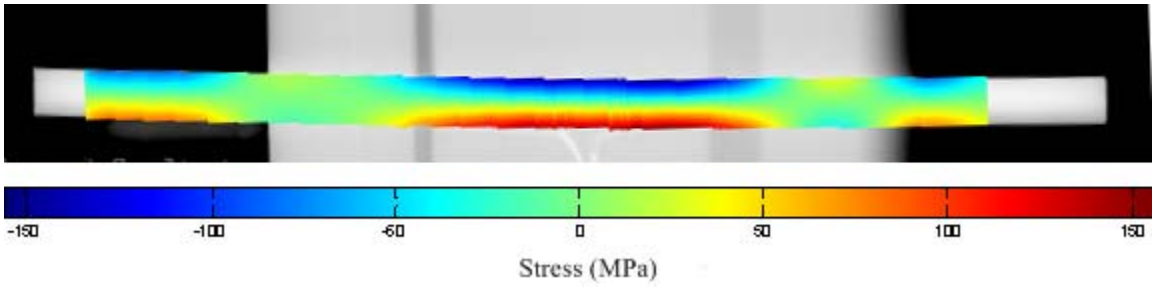


Figure 52: Overlay of stress data on the x-ray image for Data Set MC1- Image 1 [refer to Figure 42 for plotted strain data]

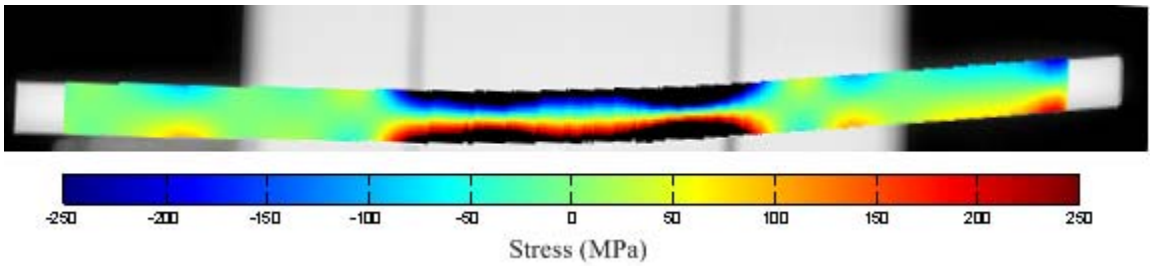


Figure 53: Overlay of stress data on the x-ray image for Data Set MC1- Image 4 [refer to Figure 42 for plotted strain data]

In Figure 52 all of the stresses are below the assigned yield stress, while in Figure 53 the pixels with an absolute value of strain which are greater than the yield stress are colored black. Since the neglected shear stresses will be additive to this calculated bending stress, the found yielding locations, especially over large areas as in Figure 53, are most likely truly yielded.

While these plots present the information in a less quantitative fashion (since exact calculated values are not shown), the artifacts due to the algorithm used to calculate

the strains are still present. They can be seen in locations outside of the connection where the dowel appears to have significant stresses or strains. The Matlab ".m" files used to produce these visualizations, as well as others, are presented in Appendix C.

Chapter 7. Demonstration of Usability for Wood Connections

Once a verified method was developed to obtain the strain distribution along the dowel in the mock connection, wood connections were tested to show this method's applicability to real-life structural components. While it was beyond the scope of this project to use this method to obtain any conclusions about joint behaviors based on such parameters as geometries, dowel size, etc, tests were run to demonstrate how this method could be used to do so. Two double shear wood connections, each with two fasteners, were tested. The geometries are described in Table 11 and Figure 54 shows photographs of Connection 2 prior to testing.

Table 11: Wood connection test parameters

	Member material	Fastener Diameter	Edge Distance	End Distance
Connection 1	Spruce-Pine-Fir	12.7 mm (0.5 in.)	44.45 mm (1.75")	50.8 mm (2")
Connection 2	Spruce-Pine-Fir	9.5 mm (0.375")	25.4 mm (1")	50.8 mm (2")

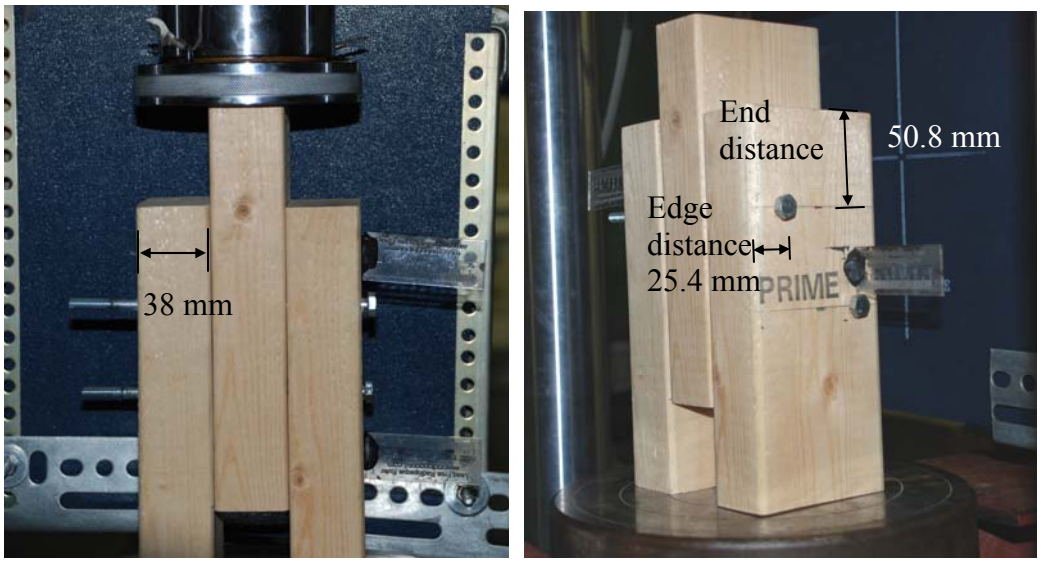


Figure 54: Connection 2 prior to testing

The edges were extracted using the same method used for the PMMA tests. Since both the top and bottom edges were unobstructed by gages or wires, they were averaged together to form the neutral axis data, which was used for the calculation of strains.

It was noted that the edge extraction procedure does not produce good results at very low levels of deflections for cylindrical dowels. When the dowel is unloaded, or even under slight load, the extracted deflection contains superharmonic oscillations which result in the calculation of significant strains, especially in cases when the dowel is not very slender. The extracted dowel edge for the bottom dowel of Connection 1 prior to loading is shown in Figure 55. The data points have been rotated so that the first and last data points have the same *y* value.

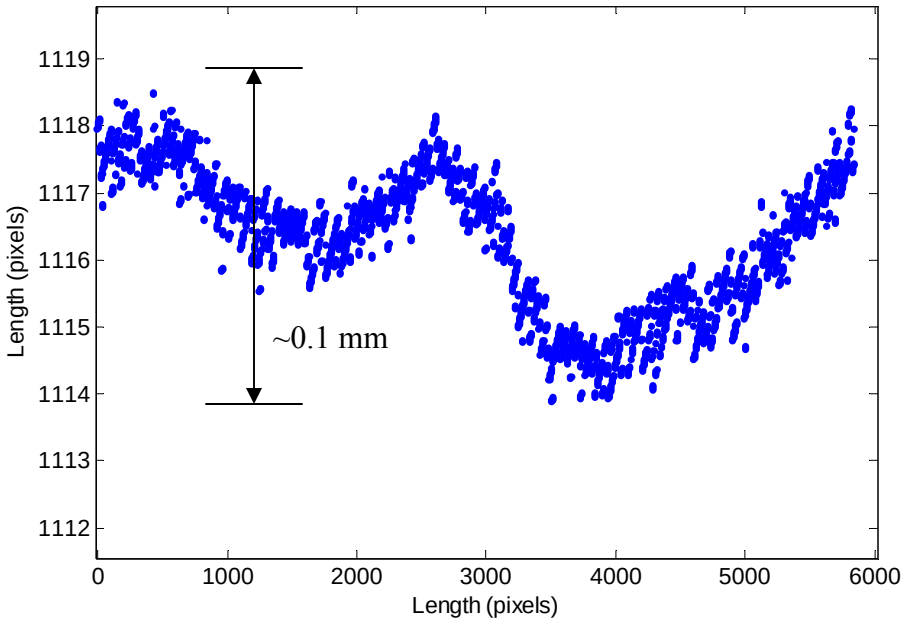


Figure 55: Averaged extracted top and bottom edges from the bottom dowel of Connection 1 prior to loading

The range in values is roughly 5 pixels, which corresponds to roughly 0.1 mm. While this is extremely small, the large oscillations over short distances result in a

maximum calculated strain of roughly 1000 microstrain. As a comparison to a more slender dowel, if the dowel were only 4.76 mm (3/16") in diameter instead of 12.7 mm (1/2"), then the calculated strain would be roughly 375 microstrain at most. The artifact could potentially be brought on by the fact that the cylindrical dowel does not have a distinct edge, and also the fact that wood is not homogeneous. This results in deviations in pixel intensity values based on the slight variations in density. This seems to be a main factor affecting the edge extraction, since the two local minimums (at roughly a location of 1500 and 4000 of the *x*-axis) in the *x*-*y* data shown in Figure 55 roughly correspond to the locations of the interfaces between the side and main members. Also, examining Figure 56 seems to be further proof.

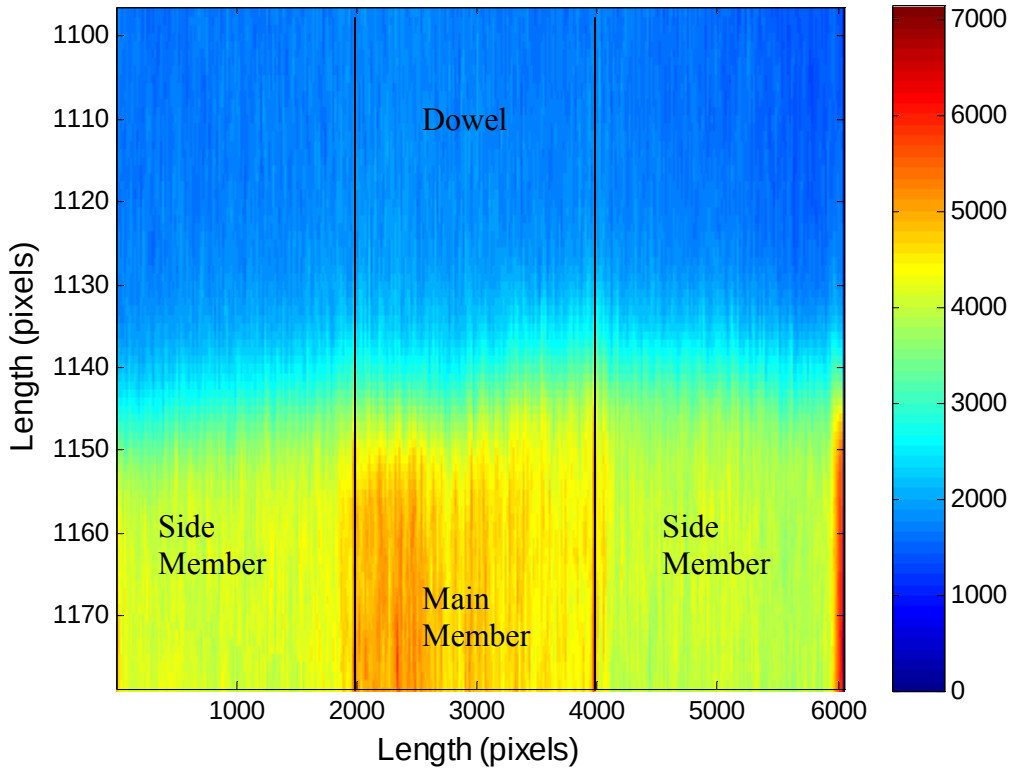


Figure 56: Scaled image of Connection 1 bottom edge of bottom dowel prior to loading

Figure 56 has been scaled to show roughly the whole length of the dowel inside the connection while zooming in on the bottom edge. The intensity values are plotted with a color scale, and with a color bar denoting intensity values, for better viewing of the changes in intensity. The higher intensity value pixels concentrated in the main member appear to have an influence on the edge position. While it is not simply the intensity value which defines an edge by the method developed for extracting dowel edges from connection images, it appears that the higher intensity values in the main member also have an affect on the first derivative in this region.

This occurrence of non-uniform density (and therefore radiograph intensity levels) is a known problem in edge detection of photographic images, where changes in illumination are analogous to the changes in density in a radiographic image. Many of the procedures to deal with these issues involve using morphological operations. For an introduction to these operations, and information on many other image processing topics see [25].

The following figures were all taken from images and calculated data of Connection 1 and Connection 2 at an overall joint displacement (slip) of roughly 2.52 mm (0.1") measured from the loading crosshead. By this displacement level, the small oscillations which resulted in large calculated strains (like those mentioned above) were no longer present in the data, since the overall deflection of the dowel was enough to make those artifacts inconsequential.

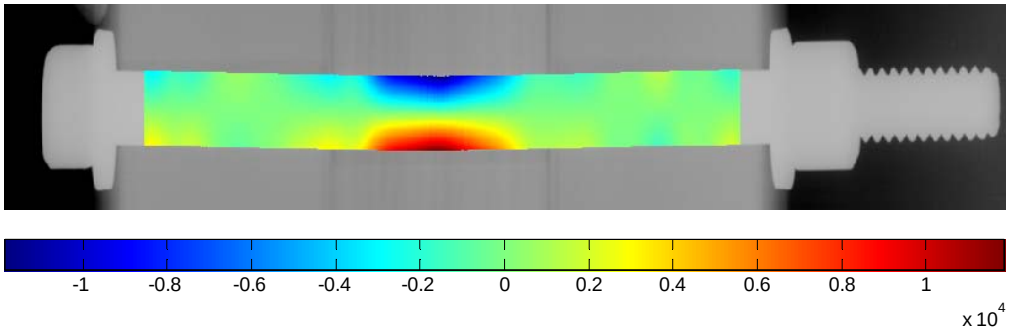


Figure 57: Connection 1 top dowel strains at a joint slip of 2.54 mm

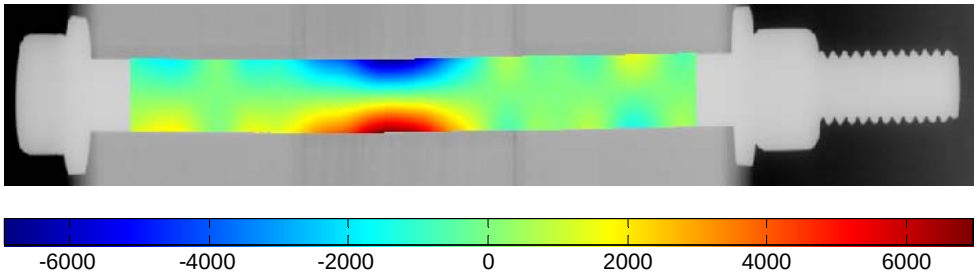


Figure 58: Connection 1 bottom dowel strains at a joint slip of 2.54 mm

As is typically the case for connections with multiple dowels, the load was not equally shared between dowels at different end distances. This can be seen in the different scales of the color maps for Figure 57 and Figure 58. The top dowel supports more of the load than the bottom dowel while under the sustained compressive loading. The opposite is the case for Connection 2, where, by the time 2.54 mm joint slip has been reached, the bottom dowel has been strained significantly more than the top dowel. This may be a result of the fabrication errors, as the top dowel can be seen to be considerably skewed from horizontal when compared with the bottom dowel.

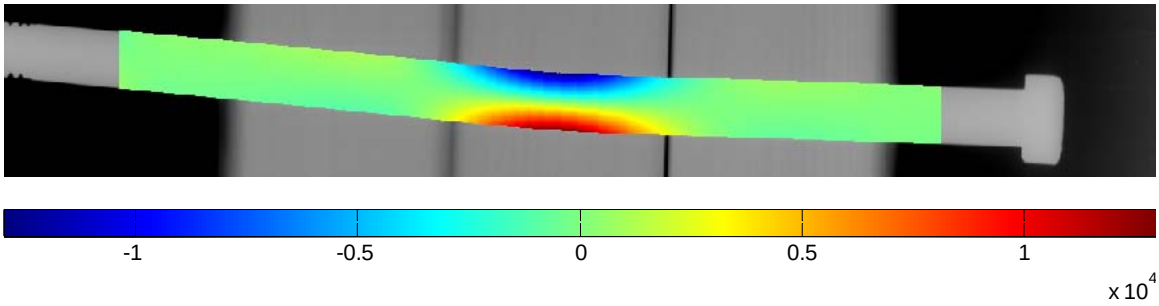


Figure 59: Connection 2 top dowel strains at a joint slip of 2.54 mm

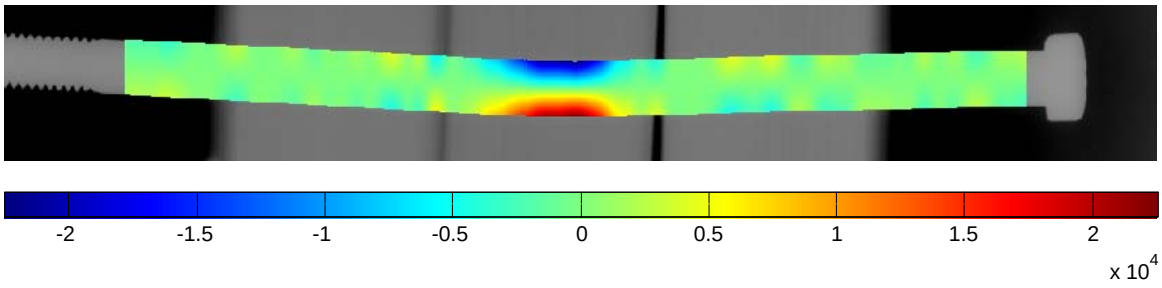


Figure 60: Connection 2 bottom dowel strains at a joint slip of 2.54 mm

Also, as expected, the greater depth of the Connection 1 dowels causes the overall strains in the Connection 1 dowels to be less than those in the Connection 2 dowels. Due to the fabrication errors, bolt clearances, and unequal load sharing the greater strained dowel in Connection 1 has roughly the same maximum strain as the lesser strained dowel in Connection 2.

Many other observations and comparisons could be made from the calculated strain data of the two connections at various deflection levels, and other procedures or visualizations (like the mesh plots and deflection/strain plots shown and discussed previously) could be used to view and make findings.

Chapter 8. Conclusions

As long as the extracted edge data correctly represents the edge of the deflected dowel, then basic qualitative findings can be made by simply observing the overall shape, and noting features of interest such as regions of maximum curvature and approximate inflection point locations. This work has demonstrated that these findings can be taken one step further, and strain information about the dowel can be determined by using basic beam theory principles to relate the curvature of the extracted edge data to the bending strain. The method developed is unable to produce *exact* results, meaning that it cannot exactly match the strain values measured by strain gages, and the calculated strains do not precisely match the expected strain distributions from beam theory principles. This inexactness is due to the discretization which occurs when capturing the image, the presence of noise in the system, the ill-posedness of the differentiation of discrete data, the effects of the smoothing procedures on the calculated derivatives, and the assumptions associated with applying basic beam theory principles to the deformations associated with dowels in connections.

Despite these factors, it was demonstrated in the feasibility tests that the methods presented can be used to extract dowel edges and obtain useful quantitative information which approximates the true strain in the dowel, a quantity that is otherwise not measurable for dowels in wood connections by non-contact procedures. For all of the types of specimens which were imaged, the edge extraction procedures produced edges with tolerances between roughly 0.05-0.1 mm. The higher values were noted especially to occur at regions of high curvature, where the extracted edge became a band of pixels, roughly 5 pixels thick at most. In simple bending tests it was found that the oscillations in

the calculated strains typically bounded the value found from the strain gage, although the oscillation amplitudes could range in value from 200 microstrain to 1000 microstrain because of the influence of the linearly interpolated data points at mid-span. For the mock connection tests, the calculated extreme fiber bending strains were found to typically pass through the values measured from the strain gages, especially at higher deflections values. In the cases when the calculated strain did not pass through the value from the gage, the calculated strain values could either over or underestimate the true strain value, by a maximum of approximately 1000 microstrain. Also, calculations of microstrain in regions of the dowel outside of the connection (where the calculated strain should have been zero) could be found to be roughly 1000 microstrain at most.

The fact that the results obtained from this method are not *exact* and contain artifacts from the procedure indicates that some interpretation is required when examining the results. The interpretation of the nuances associated with the inexactness of the method is not a subjective undertaking, but comes from knowledge of beam theory, basic calculus, and the understanding of both the ill-posedness of differentiation of noisy discrete data and the algorithms used to handle this ill-posedness.

The use of high resolution imaging for quantitative analysis of structural components is not new in any sense. Digital image processing of Moire patterns or using digital image correlation are two examples of procedures which have been used in the past to allow displacement and strain information to be found without the use of sensors which contact the specimens. The developed methodology from this work adds another way that information previously unattainable (without the use of sensors in contact with the component) can be found. There may be other useful applications of using beam

theory to relate geometrical curvature found from image processing to bending strains, but its application is surely limited by many factors. These would include examining slender components which carry load predominately in bending, cases where a high resolution image can be taken of a significant portion of the beam length, and cases where the beam edge can accurately be extracted from the image.

The limiting factors mentioned all deal with the imaging part of the developed method. To determine the strain from the geometrical curvature only requires that the edge data be obtained in some fashion which presents the edge data as a fairly large series of equally spaced data points. Therefore, methods other than imaging could be used for determining these data points, and could potentially extend the usefulness of the verified procedure developed to obtain strain information from noisy, discrete deflection data to larger beams.

The improvements which could be made to the developed method occur in two distinct sections: the extraction of the edge and the use of the deflected shape in determining strain values. Undoubtedly, more sophisticated edge extraction algorithms could be implemented which may better handle the extraction of edges, especially for more difficult types of edges like those found in radiographic images of cylindrical components in non-homogenous wood connections.

Regardless of the model used to attempt to calculate strains from the deflected shape, the calculation of discrete derivatives will be required. While most methods for extracting edges from images rely on techniques which have long been in practice, the differentiation of noisy discrete data is a research topic still undertaken by mathematicians today. Therefore, it is safe to say that in the future there will be

advancements in the treatment of noisy discrete data, especially for the calculation of derivatives. These advancements, when applied to supersede the mollification procedure, will certainly increase the accuracy and exactness of the resulting strain calculations. This could also potentially allow the accurate calculation of higher order derivatives, which could then be used with beam theory principles to relate the extracted deflection to the applied load on the dowel.

Possible verification projects which could be undertaken for this method include examining the accuracy of strain calculations combined from two images taken at orthogonal angles and the incorporation of higher order beam theory principles into the calculation of strains from the extracted deflections. Using a combination of two images would be of utmost importance for any type of study involving wood moment connections, and also when examining structural components in the field where there may be limitations on the placement of the source and detector. By implementing higher order beam theories, the effects of shear on the deformation of the dowels could be included and would potentially increasing the accuracy of the method, especially at the interfaces of the connection. This would also enable the calculation of combined shear and normal strains, which would allow for a better estimate of the true strain or stress state in the dowels. With the stress more accurately calculated, the assessment of yielding (or lack thereof) in dowels would be more reliable.

Overall, the created methodology and algorithms sufficiently meet the criteria desired at the onset of this work. It has been shown that by using the presented methods, radiographic images of connections (which are typically only used for the determination of qualitative information) can be processed and analyzed to yield useful strain data

which could not have otherwise been obtained. The Matlab programs developed and given in Appendix C give a verified and documented starting point from which there can be further development. These developments can be made to both the main algorithms (as was pointed out previously in this section) and/or to the finer programming details, as each user could make their own modifications to suit their own needs or programming style.

Chapter 9. References

[1] Humphrey, P.E., and Ostman, L.J. *Bolted Timber Connections: Part I. A Wafer Technique to Model Wood Deformation around Bolts.* Wood and Fiber Science 21.3 (1989). pp. 239-51.

[2] Humphrey, P.E., and Ostman, L.J. *Bolted Timber Connection: Part II. Bolt Bending and Associated Wood Deformation.* Wood and Fiber Science 21.4 (1989). pp. 354-66.

[3] Sjodin, J, et al. *Contact-free measurements and numerical analyses of the strain distribution in the joint area of steel-to-timber dowel joints.* Holz als Roh- und Werkstoff 64.6 (2006). pp. 497-506.

[4] Johansen, KW. *Theory of timber connections*. Publication 9. International Association of Bridge and Structural Engineering, Bern, (1949). pp. 249–262

[5] Aune, P. and Patton-Mallory, M. *Lateral Load-Bearing Capacity of Nailed Joints Based on the Yield Theory - Experimental Verification*. FPL Paper 470. (1986). PL Madison, WI.

[6] Heiduschke, A. *Seismic Behavior of Moment-Resisting Timber Frames with Densified and Textile Reinforced Connections*. Dissertation. Technische Universitat Dresden, (2004). Dresden, Germany.

[7] Pollock, David G, et. al. *Ultrasonic Detection of a Plastic Hinge in Bolted Timber Connections,* Materials Evaluation, 59.5 (2001). pp. 625-631

[8] Hart, D. *X-Ray Investigation of Buildings*. Bulletin of the Association for Preservation Technology, Vol. 5, No. 1. Association for Preservation Technology International (APT). (1973). pp. 9-21

[9] Wrenn, G. *Questions of Preservation and a New X-ray Investigation Technique* Proceedings of the North American Regional Conference, Williamsburg, VA, September 1976: pp. 10-12.

[10] Anthony, R.W. *Examination of Connections and Deterioration in Timber Structures Using Digital Radioscopy*. Technical Council on Forensic Engineering of the American Society of Civil Engineers, 21 Oct. 2003, San Diego, CA. Forensic Engineering. pp. 320-28.

[11] Lear, G. *Improving the assessment of* in situ *timber members with the use of nondestructive and semi-destructive testing techniques.* Thesis. North Carolina State University, (2005). Raleigh, NC.

[12] ASTM E 1742 – 08. *Standard Practice for Radiographic Examination.* Annual Book of ASTM Standards. ASTM International, 100 Barr Harbor Drive, West Conshohocken, PA

[13] ICM CP120B Zonig Les Plesesses. Welkeraedt. Belgium. 2006 (http://www.icm.be/)

[14] *Basics of IP.* Fuji Film Global. (http://www.fujifilm.com/products/life_science/si_imgplate/whatis01.html)

[15] ScanX 12. Air Techniques, Inc. Allpro Imaging. Melville, NY. 2007 (http://www.allproimaging.com)

[16] A/T Scan X Eraser. Air Tecniques, Inc. Hicksville, NY. 2002. (http://www.airtechniques.com/pdf/ScanX_Plate_Eraser_73961RevD.pdf)

[17] MATLAB software. The MathWorks, Inc.3 Apple Hill Drive, Natick, MA. 2008 (http://www.mathworks.com)
[18] N. Otsu. *A threshold selection method from gray-level histograms*., IEEE Transactions on Systems, Man and Cybernetics. 9.1 (1979). pp. 62–66.
[19] Huertas, A. and Medioni, G. *Detection of Intensity Changes with Subpixel Accuracy Using Laplacian-Gaussian Masks.* IEEE Transactions of Pattern Analysis and Machine Intelligence. 8.5. (1986). pp. 651-664
[20] Savitzky, A. and Golay, M. *Smoothing and Differentiation of Data by Simplified Least Squares Procedures.* Analytical Chemistry, 36. (1964). pp. 1627–1639
[21] Ahn, Soyoung, et al. *A Scheme for Stable Numerical Differentiation.* Journal of Computational and Applied Mathematics (2006). pp. 325-34.
[22] Lu, S. and Pereverzev, S.V. *Numerical Differentiation from a Viewpoint of Regularization Theory.* Mathematics of Computation, 75. (2006). pp. 1853-1870
[23] Murio, D.A., et al. *Discrete mollification and automatic numerical differentiation*, Computers and Mathematics with Applications. 5 (1998), pp. 1-16.
[24] Woodbury, Keith A. *Inverse Engineering Handbook*. New York: C R C P LLC, (2002).
[25] Gonzalez, Rafael C., and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Upper Saddle River: Prentice Hall, 2007.

Appendix A: Simple Beam Test Results

Method: Averaging Filters and Discrete Derivatives

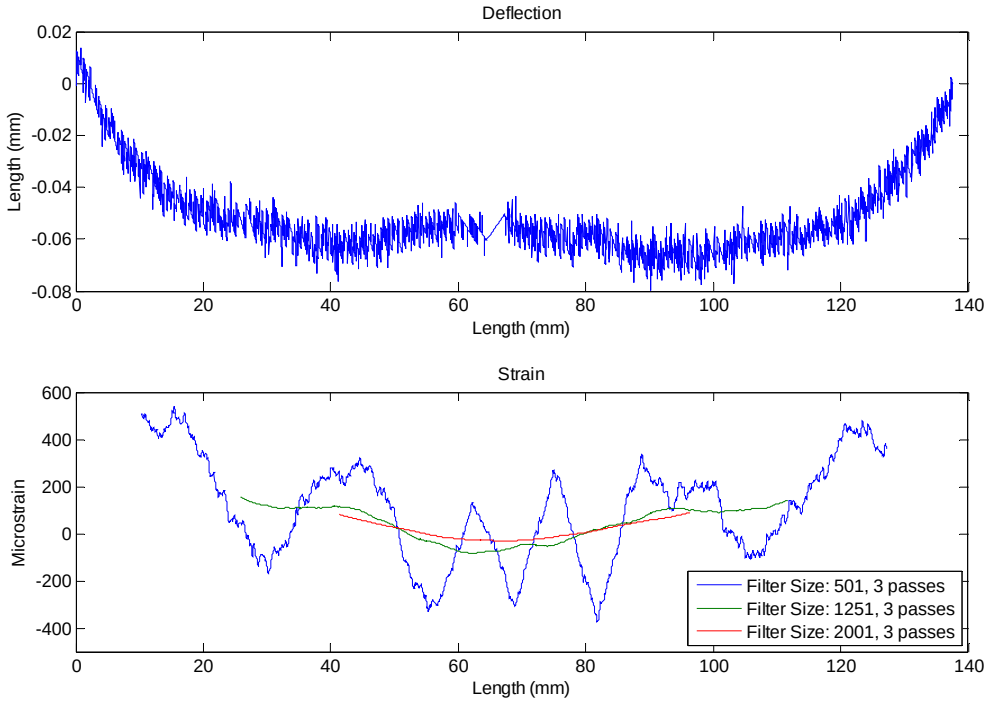


Figure: Data Set B1 – Image 04, Gage Microstrain: Zeroed

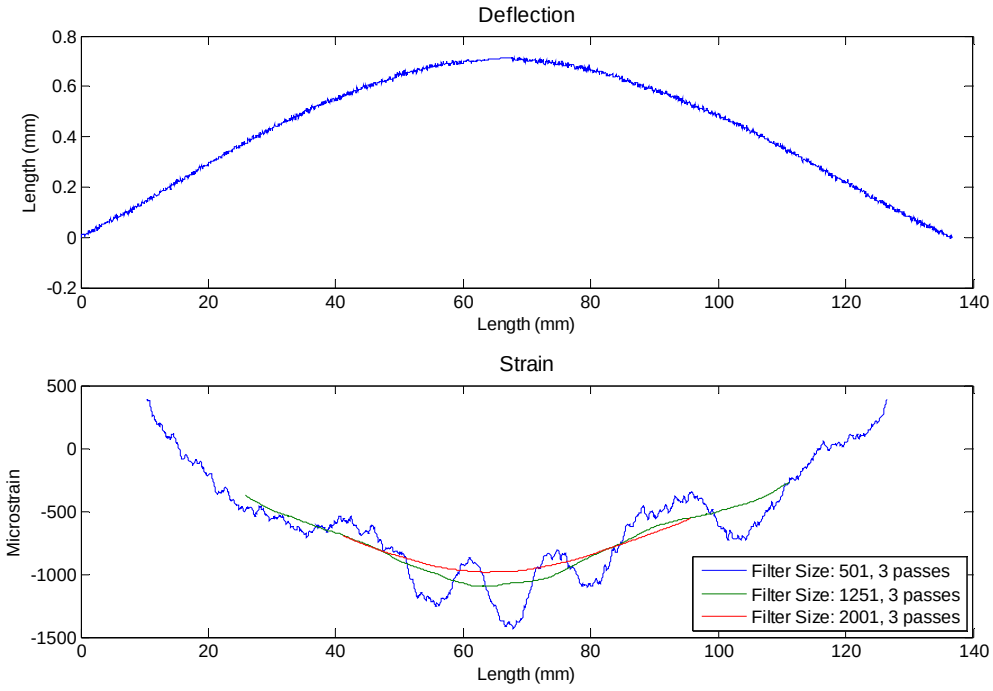


Figure: Data Set B1 – Image 05, Gage Microstrain: 805

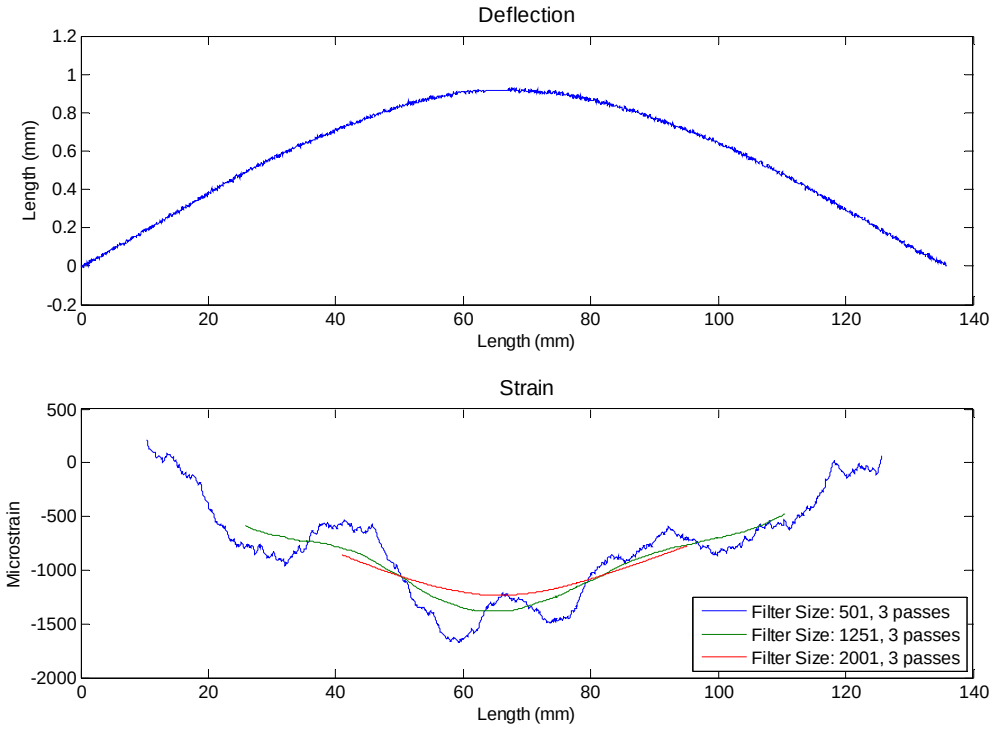


Figure: Data Set B1 – Image 06, Gage Microstrain: 1236

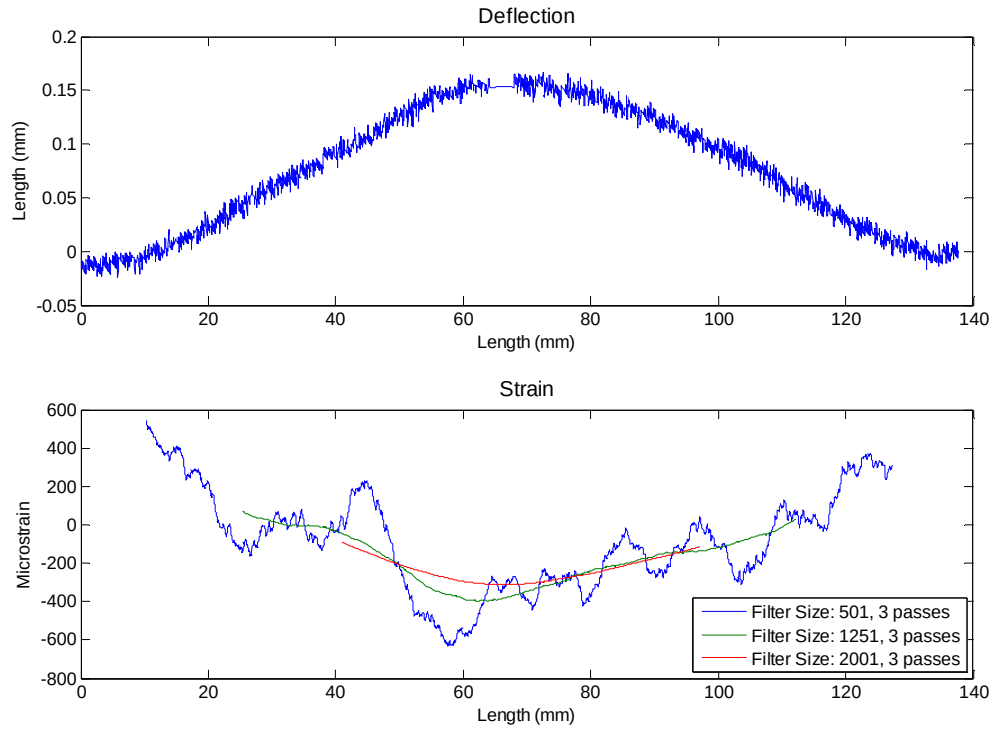


Figure: Data Set B1 – Image 07, Gage Microstrain: 265

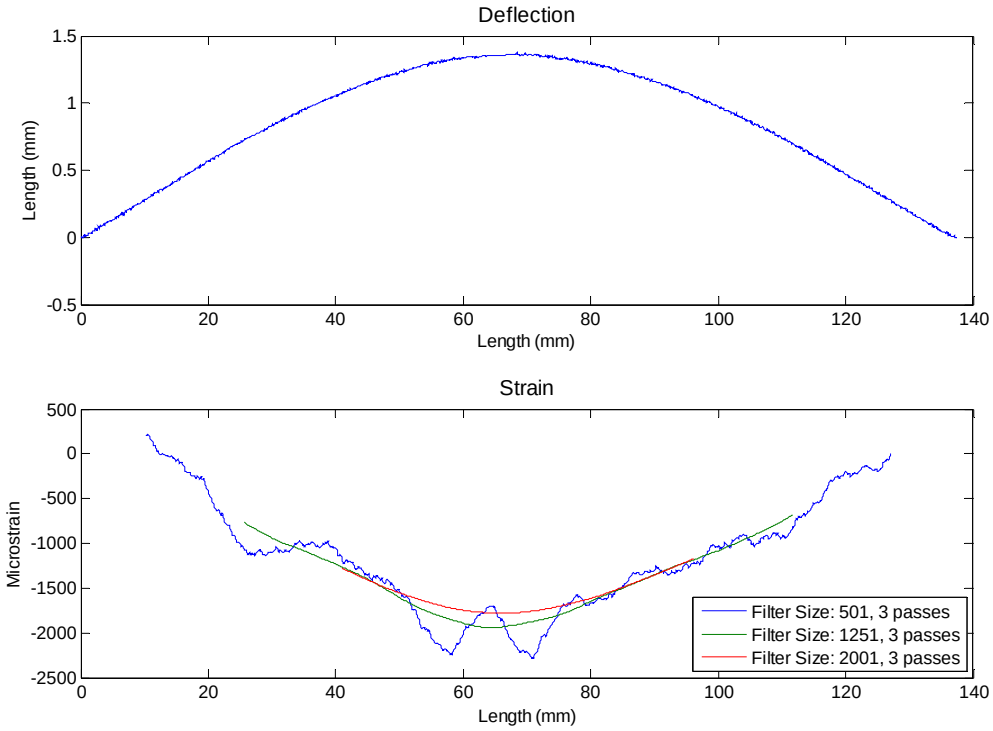


Figure: Data Set B1 – Image 08, Gage Microstrain: 1640

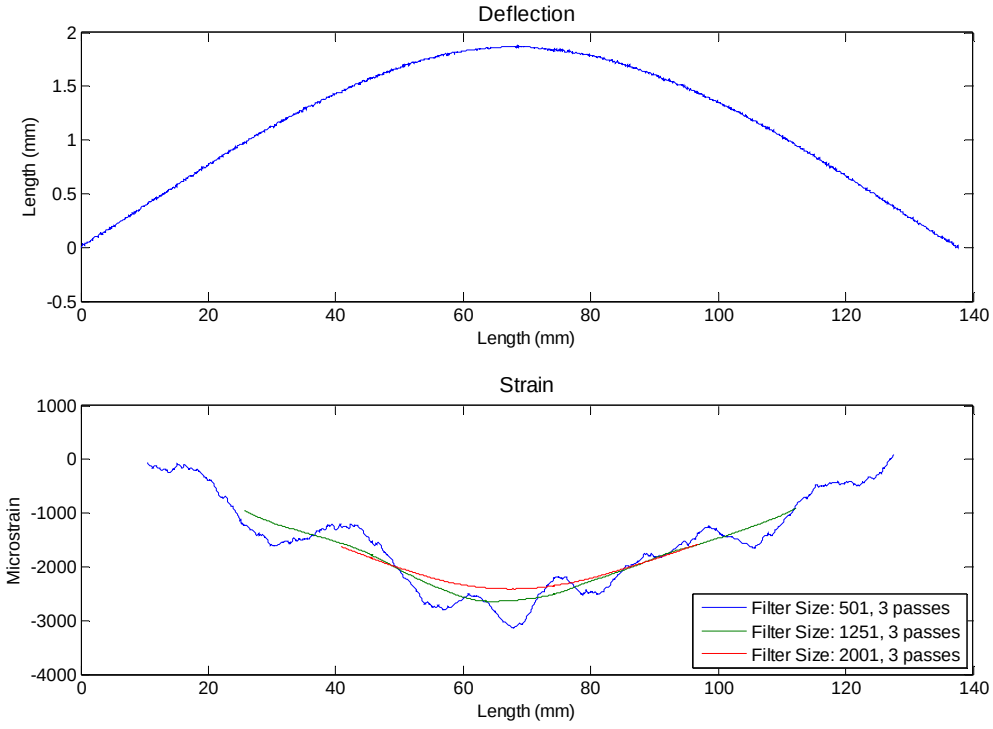


Figure: Data Set B1 – Image 09, Gage Microstrain: >2358 (limit of gage)

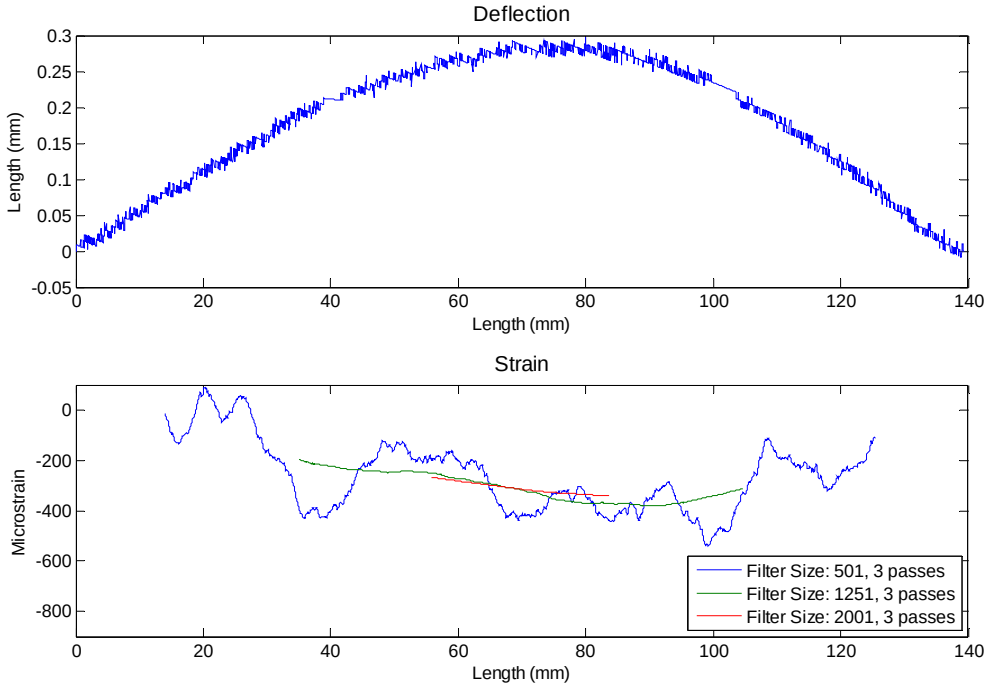


Figure: Data Set B2 – Image 01, Gage Microstrain: Zeroed

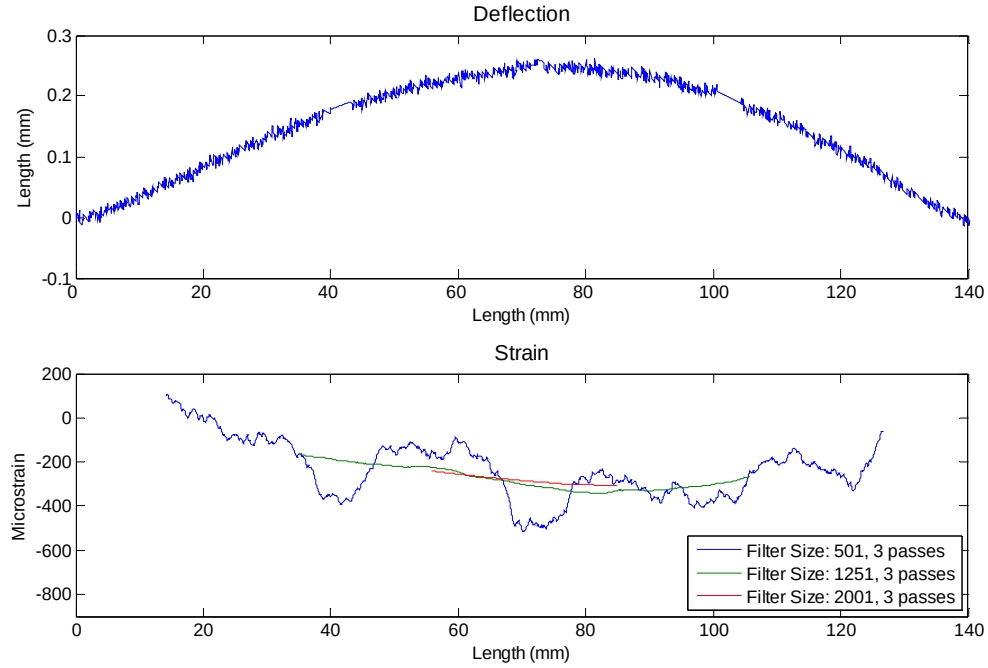


Figure: Data Set B2 – Image 02, Gage Microstrain: Zeroed

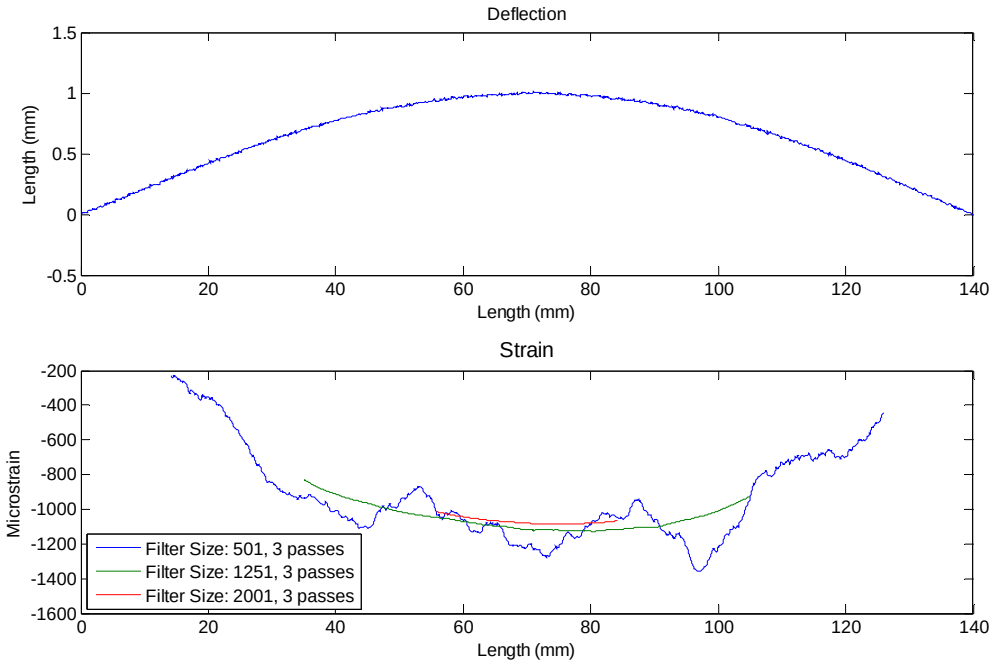


Figure: Data Set B2– Image 07, Gage Microstrain: 830

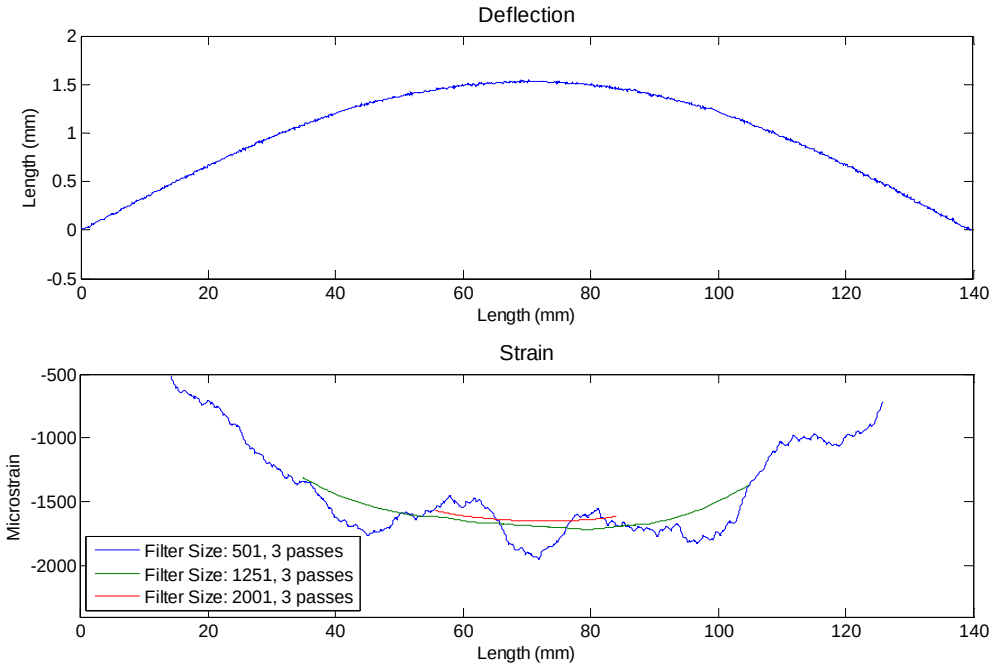


Figure: Data Set B2 – Image 09, Gage Microstrain: 1370

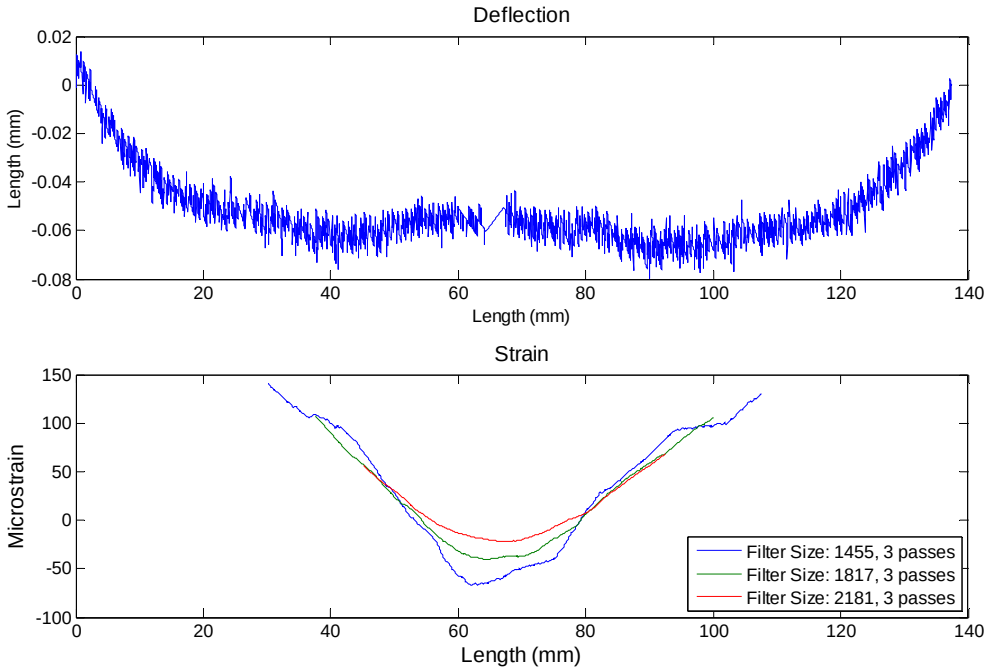


Figure: Data Set B1 – Image 04, Gage Microstrain: Zeroed

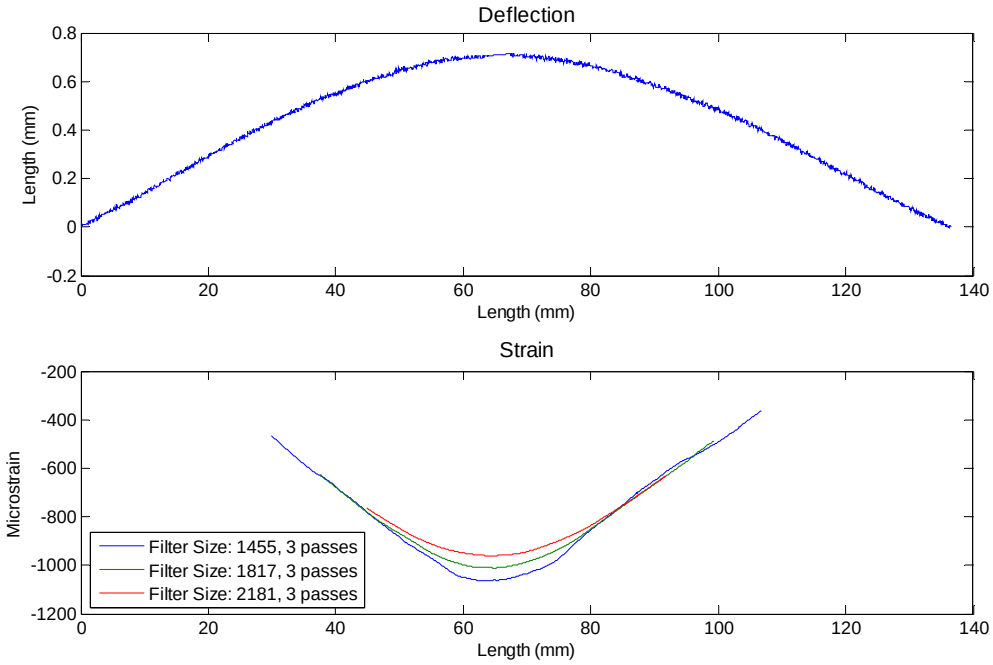


Figure: Data Set B1 – Image 05, Gage Microstrain: 805

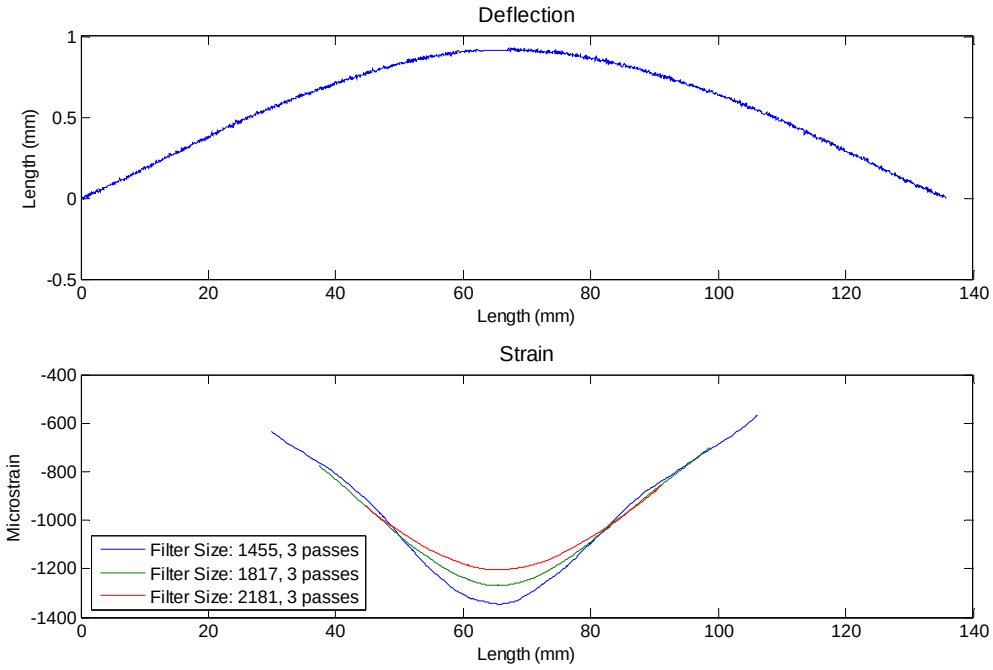


Figure: Data Set B1– Image 06, Gage Microstrain: 1236

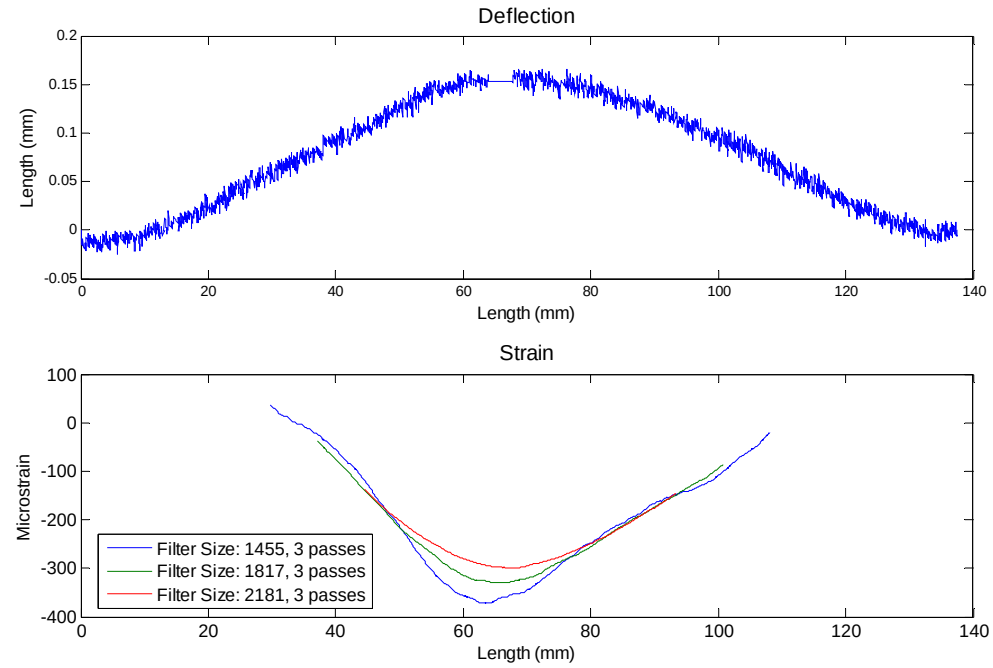


Figure: Data Set B1 – Image 07, Gage Microstrain: 265

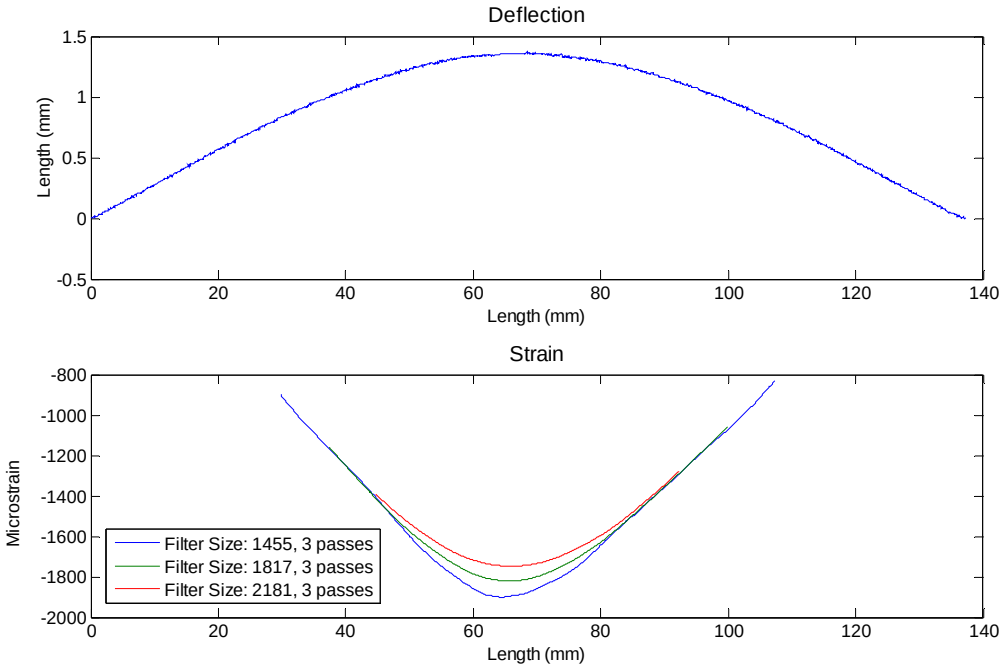


Figure: Data Set B1 – Image 08, Gage Microstrain: 1640

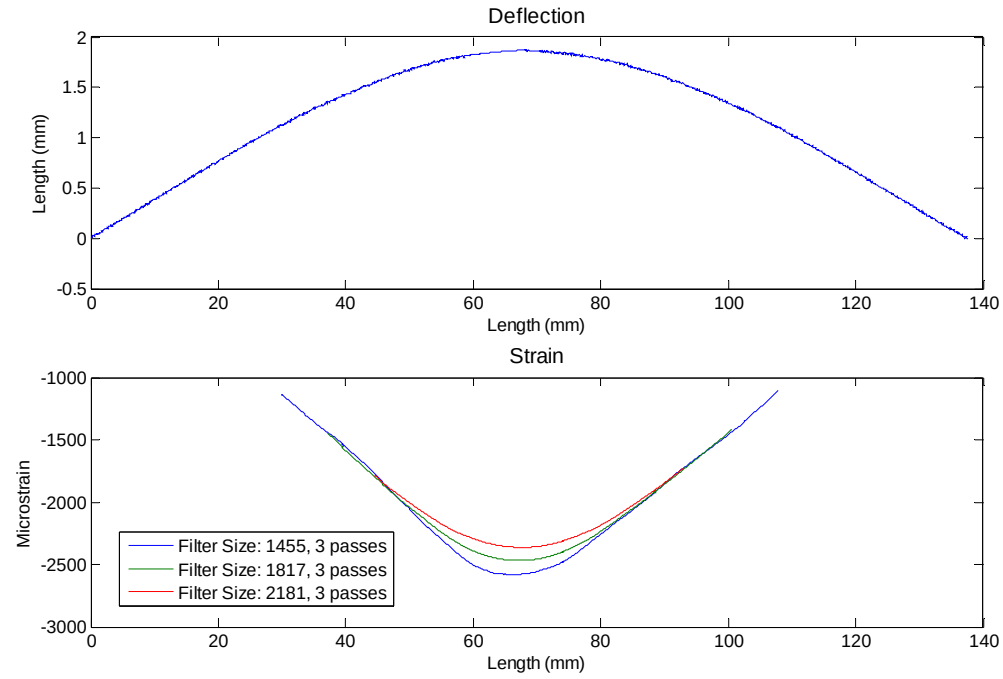


Figure: Data Set B1 – Image 09, Gage Microstrain: >2358 (limit of gage)

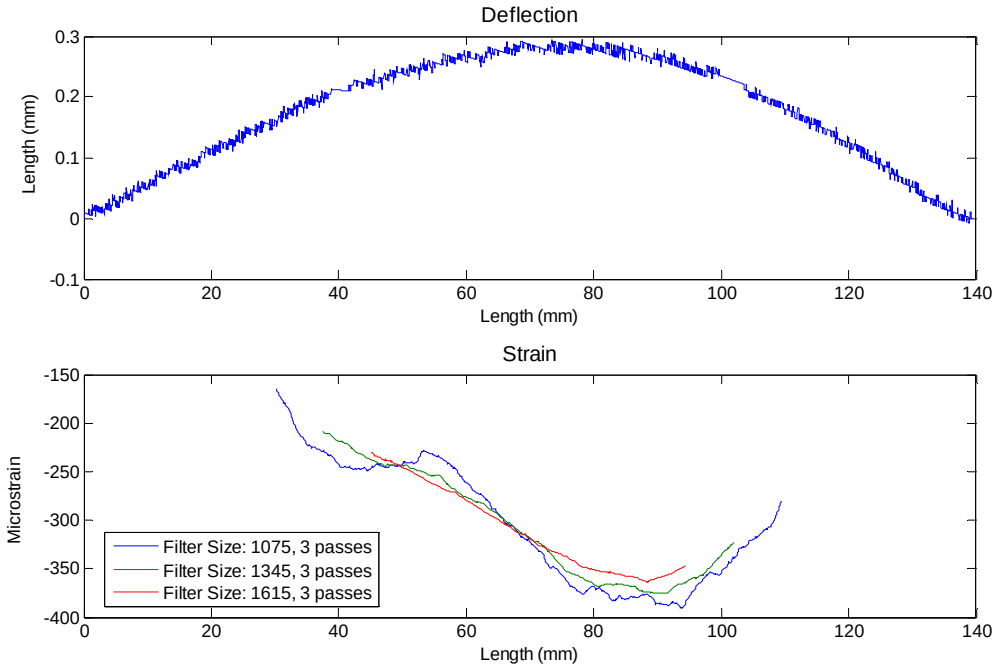


Figure: Data Set B2- Image 01, Gage Microstrain: Zeroed

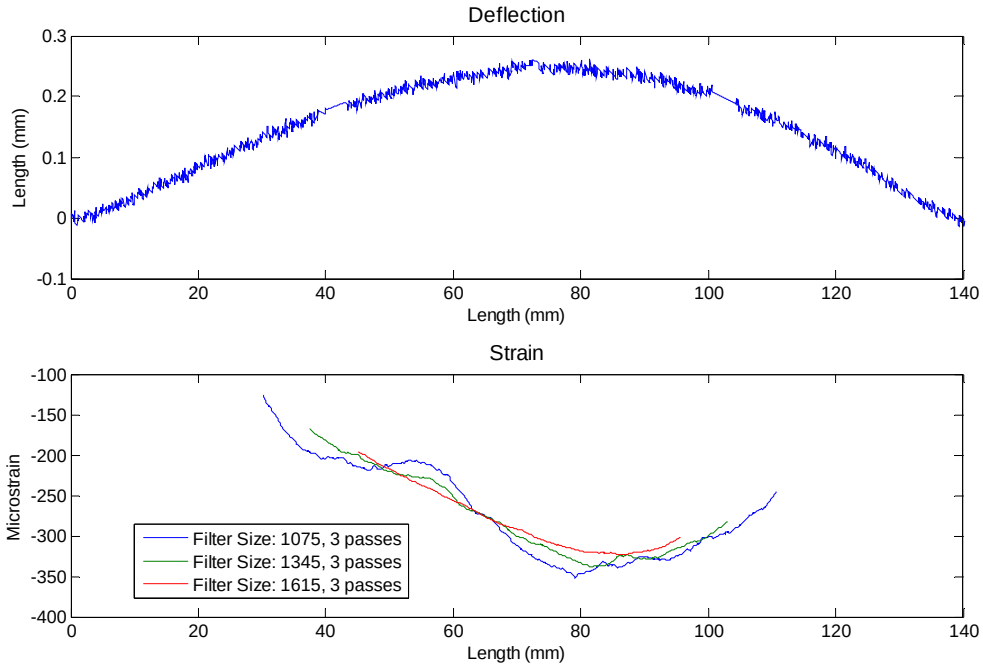


Figure: Data Set B2 – Image 02, Gage Microstrain: Zeroed

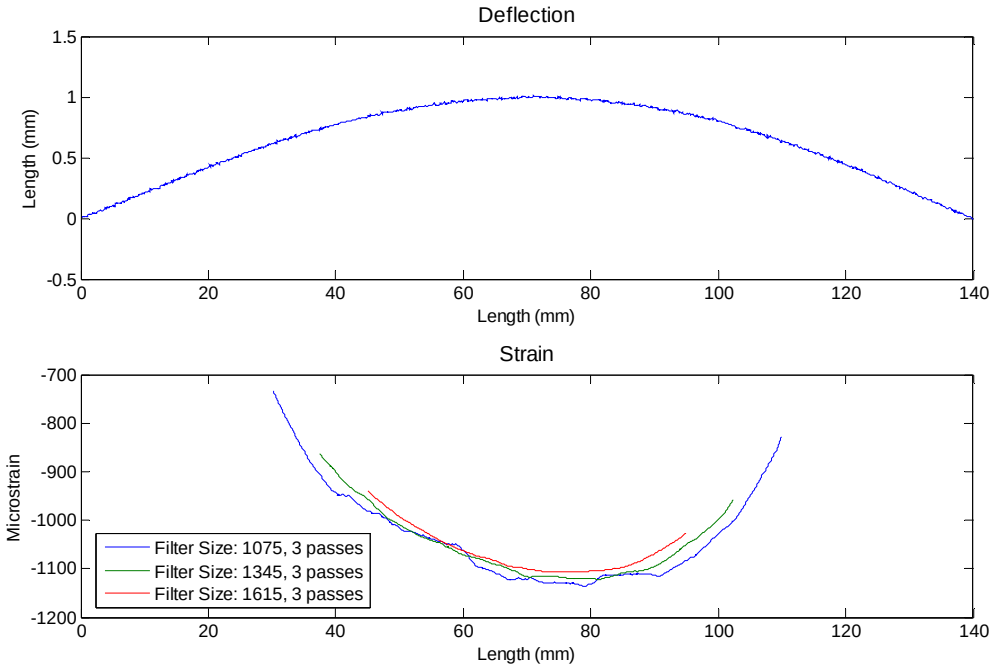


Figure: Data Set B2 – Image 07, Gage Microstrain: 830

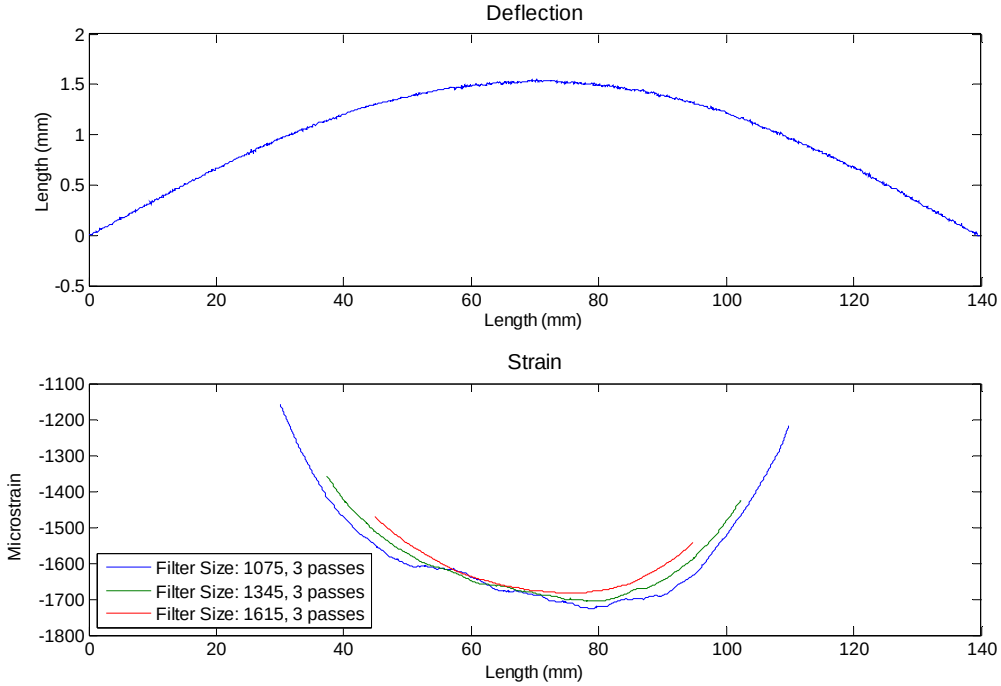


Figure: Data Set B2– Image 09, Gage Microstrain: 1370

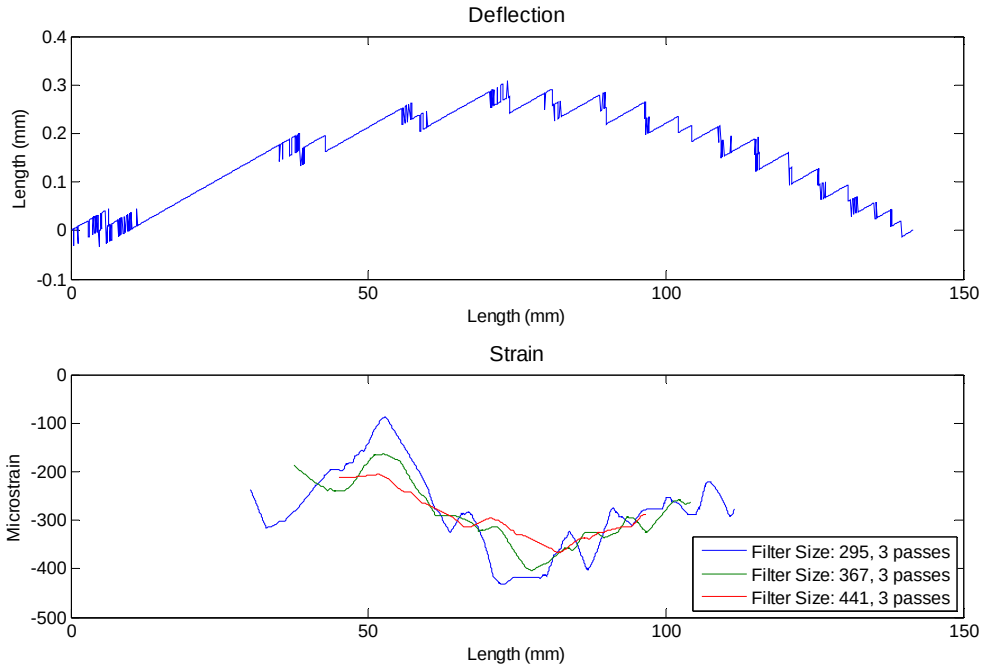


Figure: Data Set B3- Image 00, Gage Microstrain: Zeroed

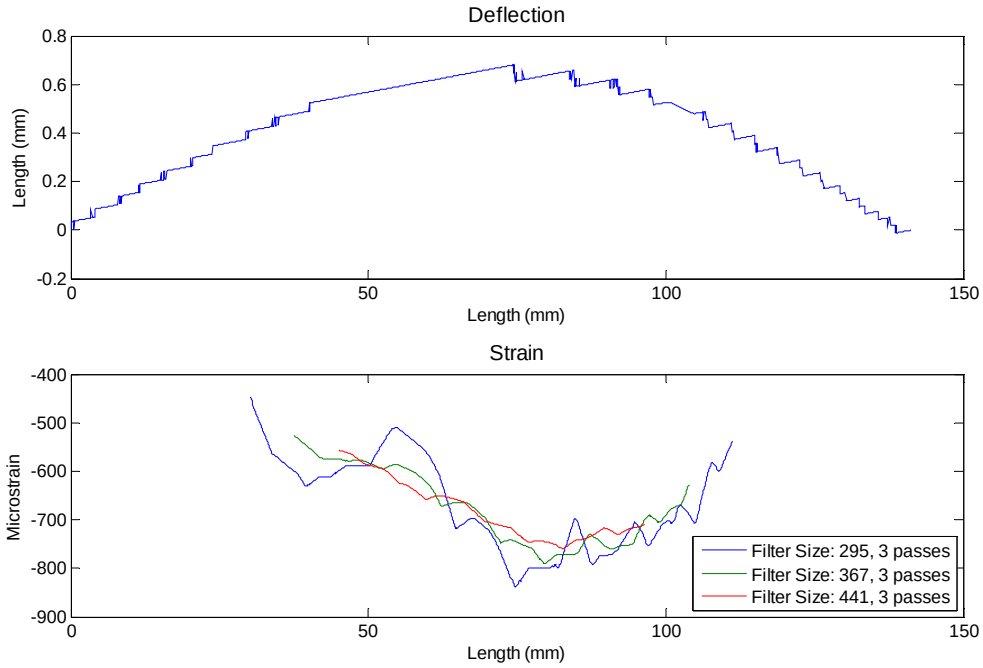


Figure: Data Set B3- Image 03, Gage Microstrain: 400

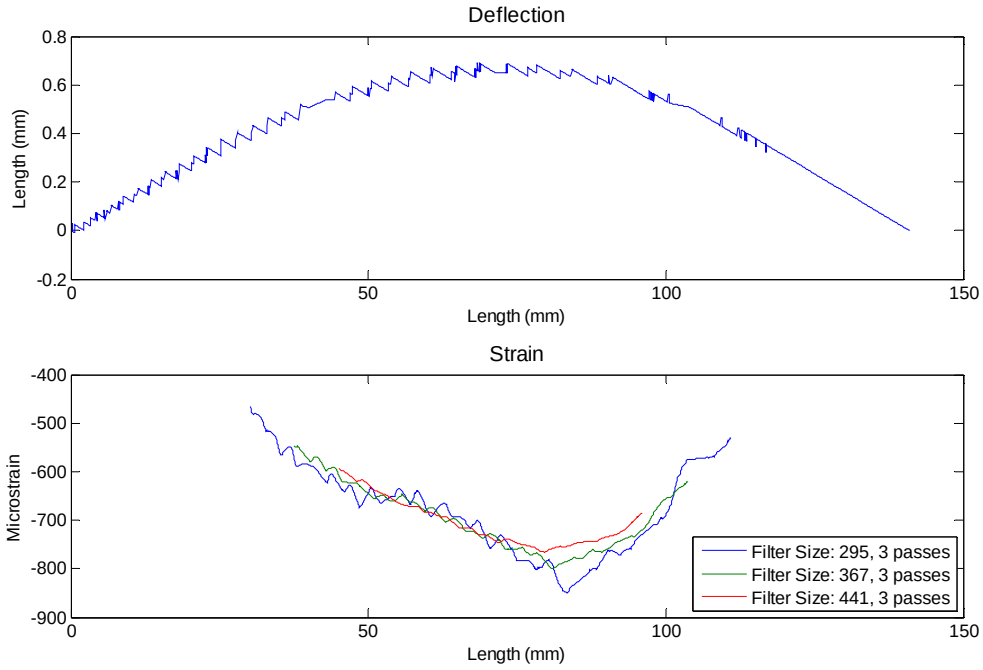


Figure: Data Set B3- Image 04, Gage Microstrain: 400

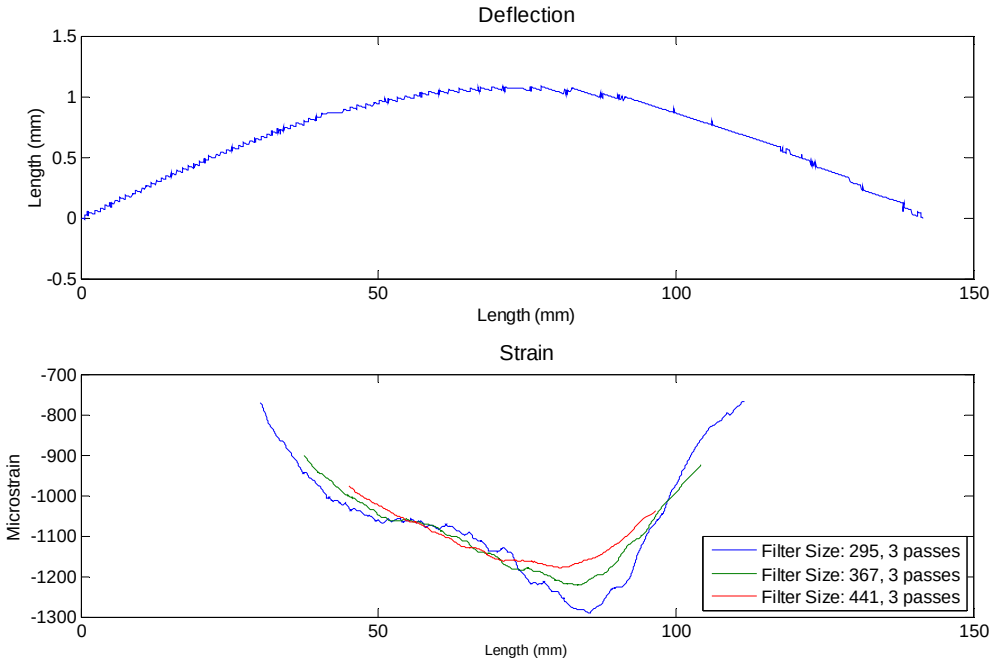


Figure: Data Set B3- Image 06, Gage Microstrain: 820

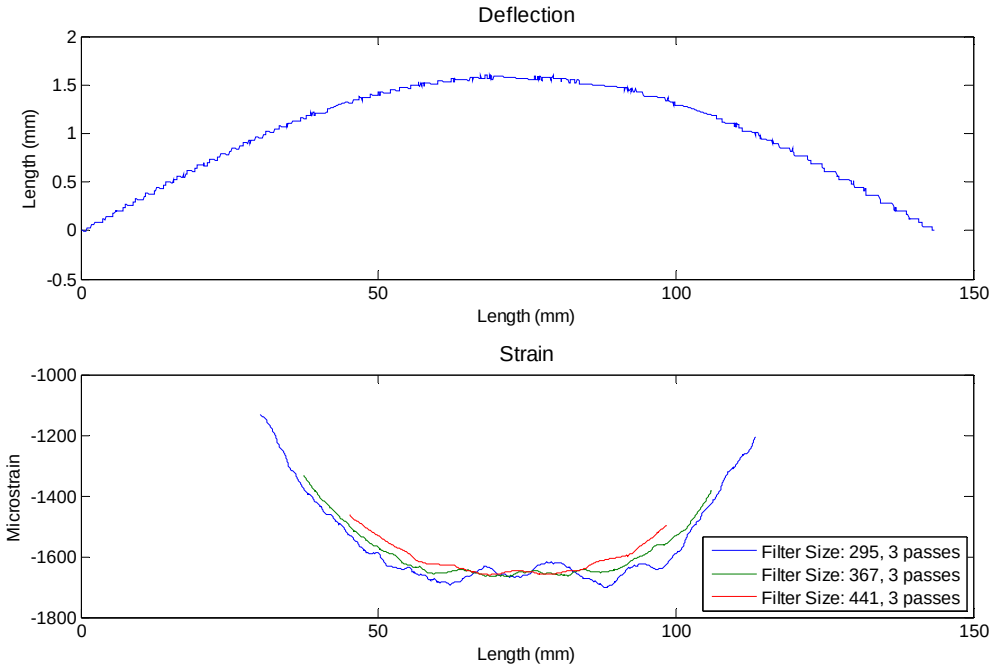


Figure: Data Set B3- Image 08, Gage Microstrain: 1370

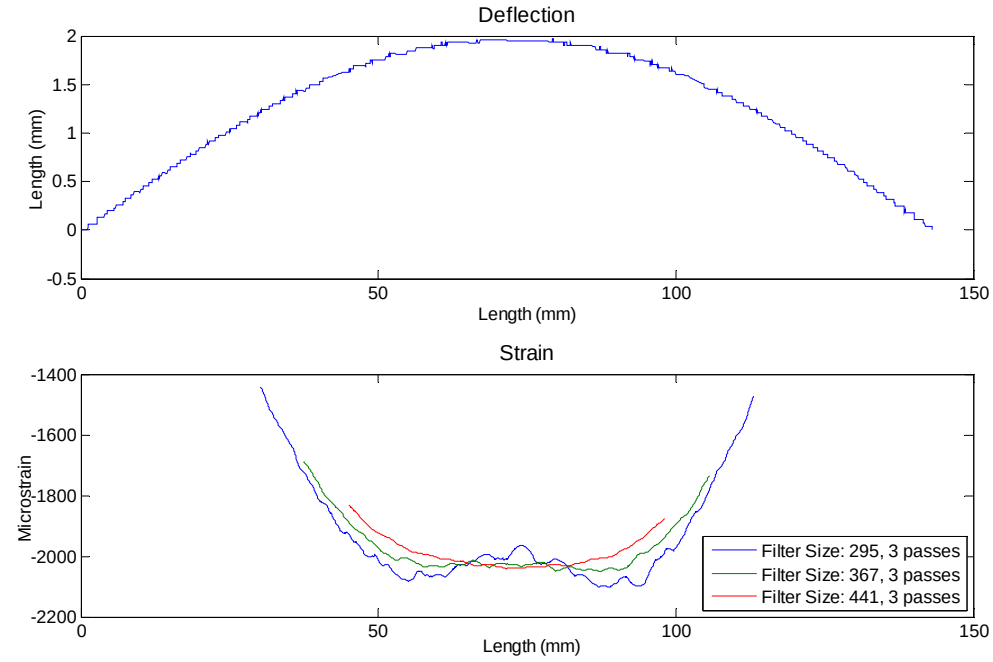


Figure: Data Set B3- Image 10, Gage Microstrain: 1760

Method: Mollification and Discrete Derivatives with subtraction of strain from images captured when the strain gage was zeroed

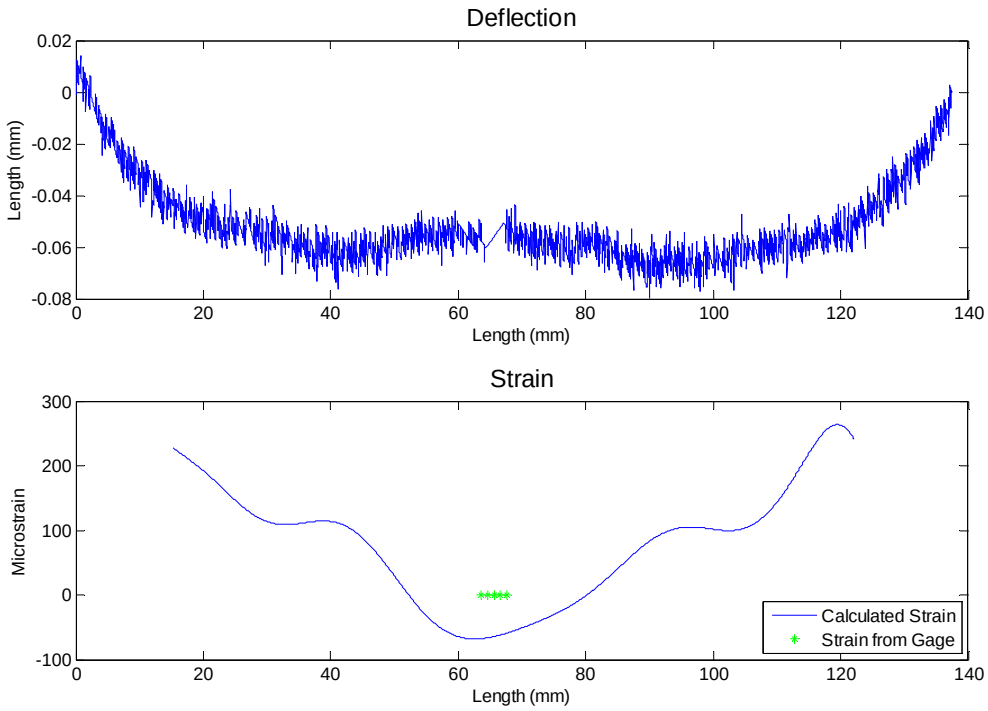


Figure: Data Set B1 – Image 04, Gage Microstrain: Zeroed

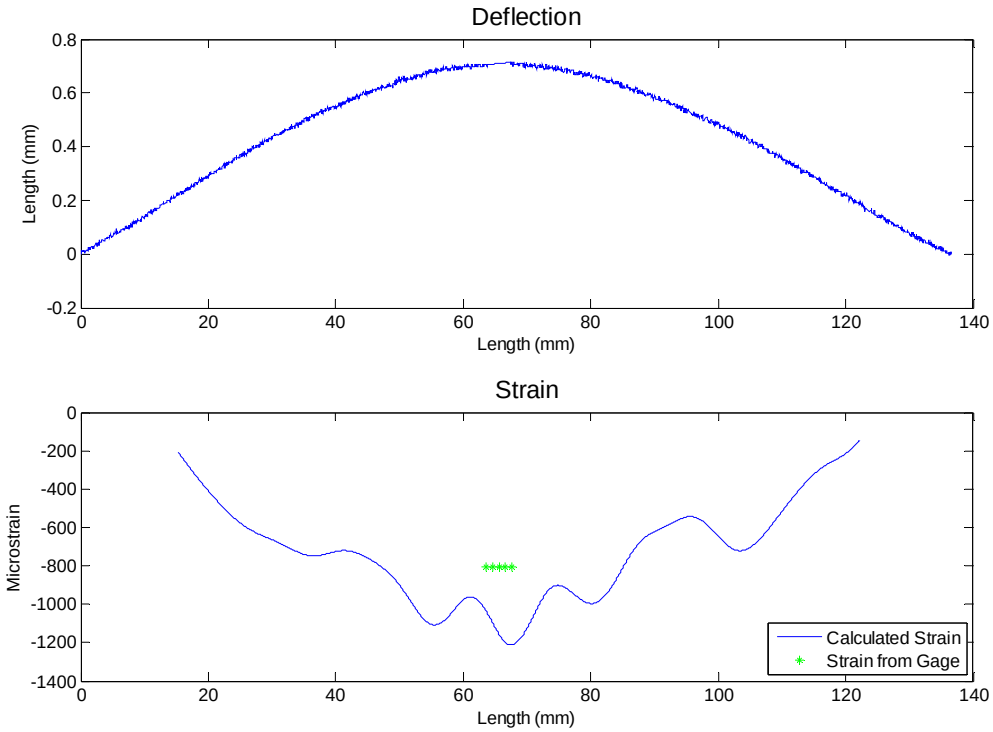


Figure: Data Set B1 – Image 05, Gage Microstrain: 805

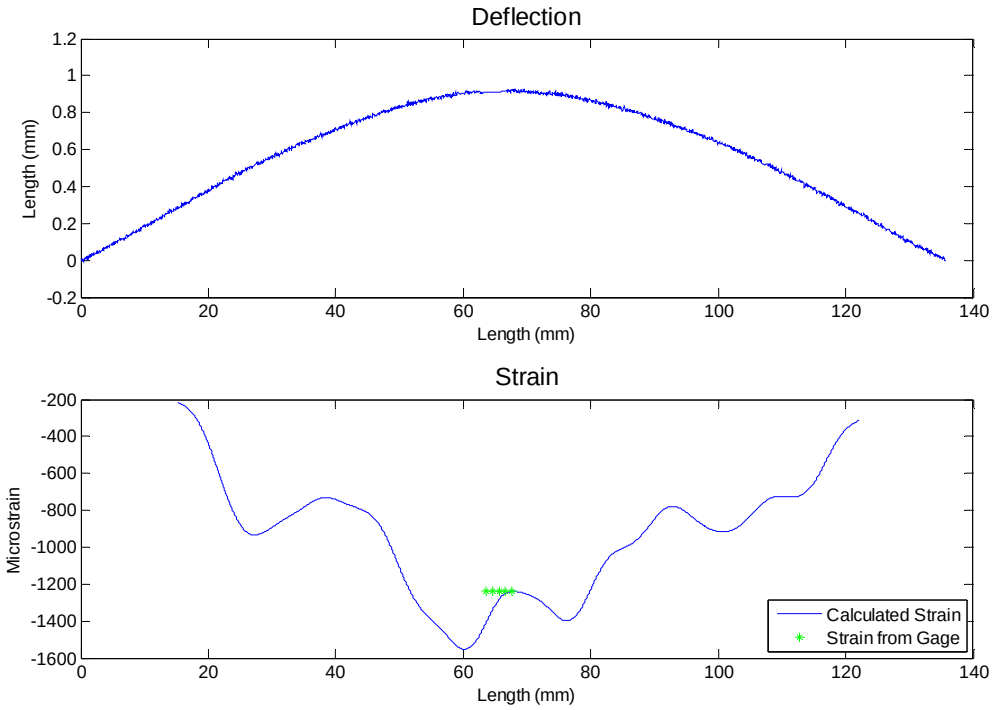


Figure: Data Set B1 – Image 06, Gage Microstrain: 1236

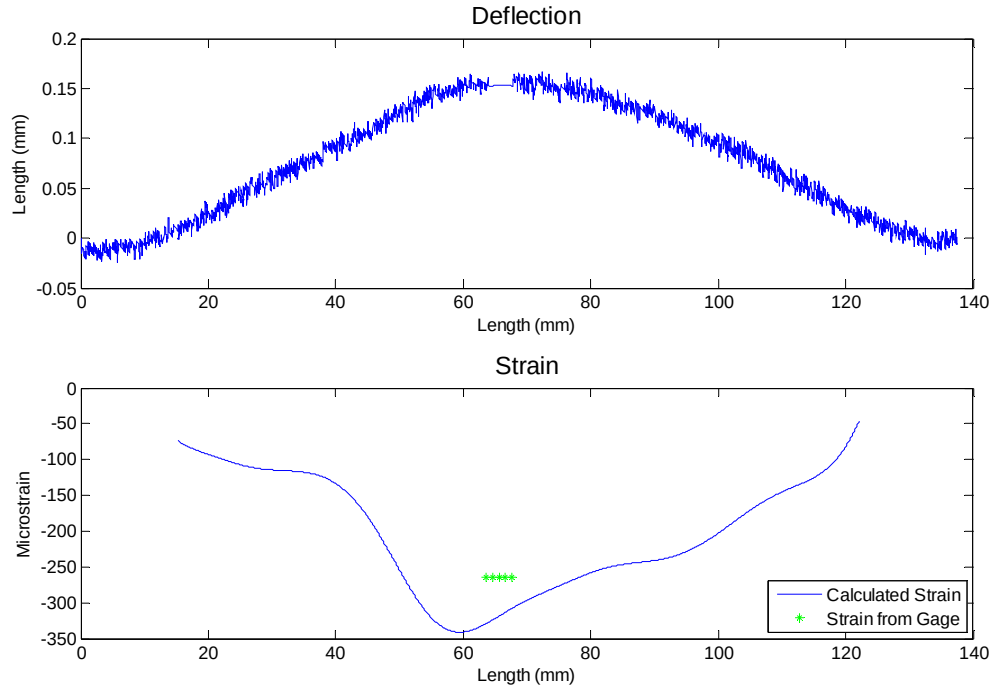


Figure: Data Set B1 – Image 07, Gage Microstrain: 265

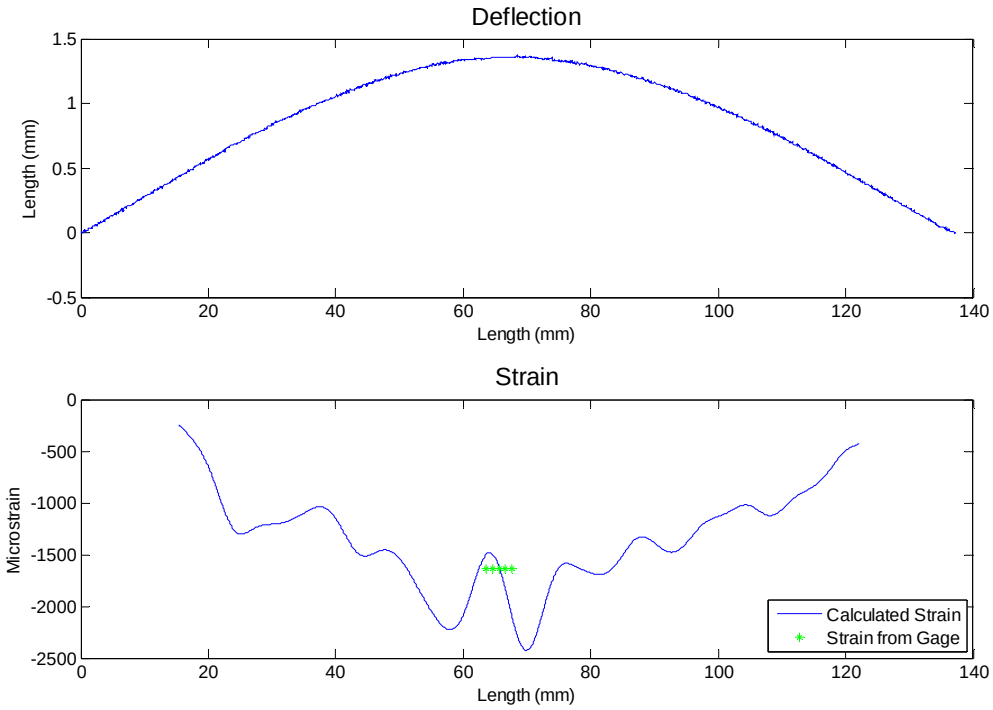


Figure: Data Set B1 – Image 08, Gage Microstrain: 1640

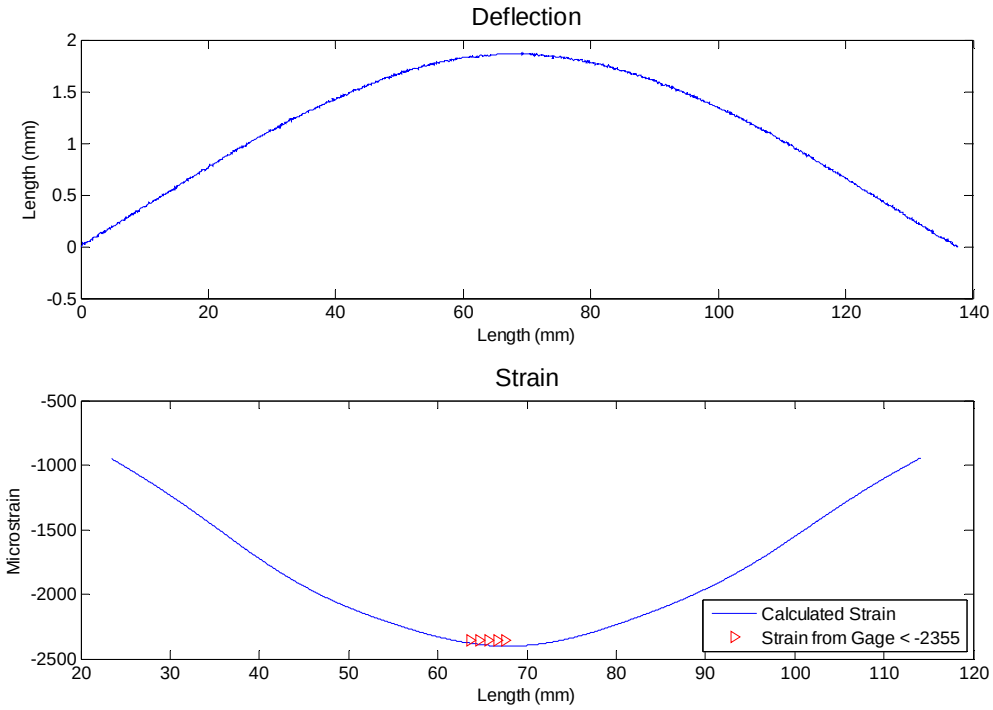


Figure: Data Set B1 – Image 09, Gage Microstrain: >2358 (limit of gage)

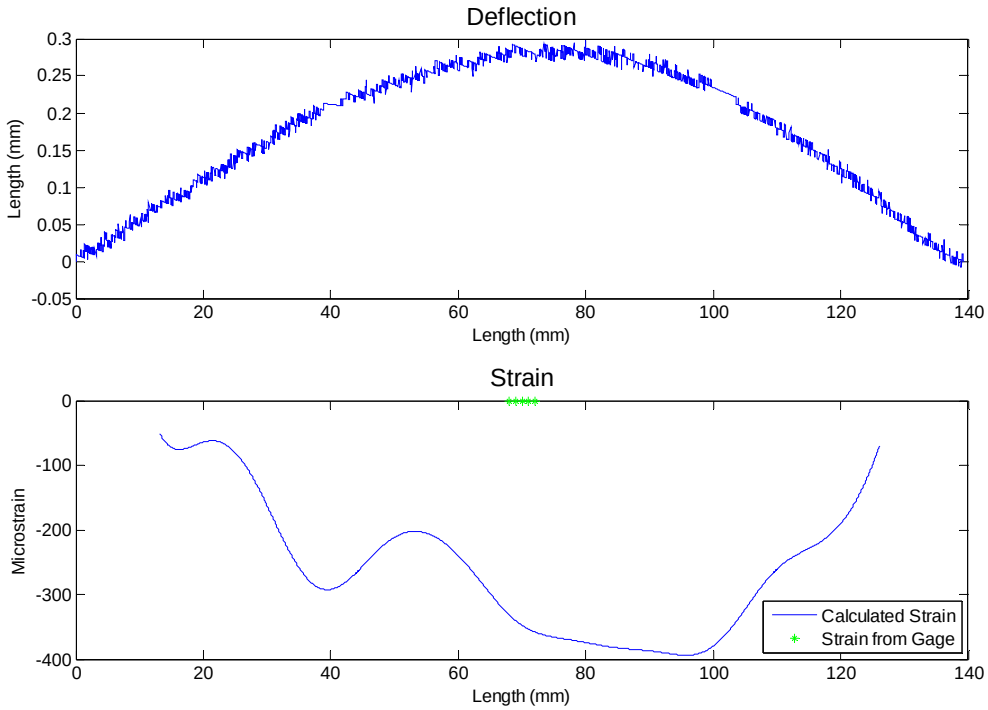


Figure: Data Set B2- Image 01, Gage Microstrain: Zeroed

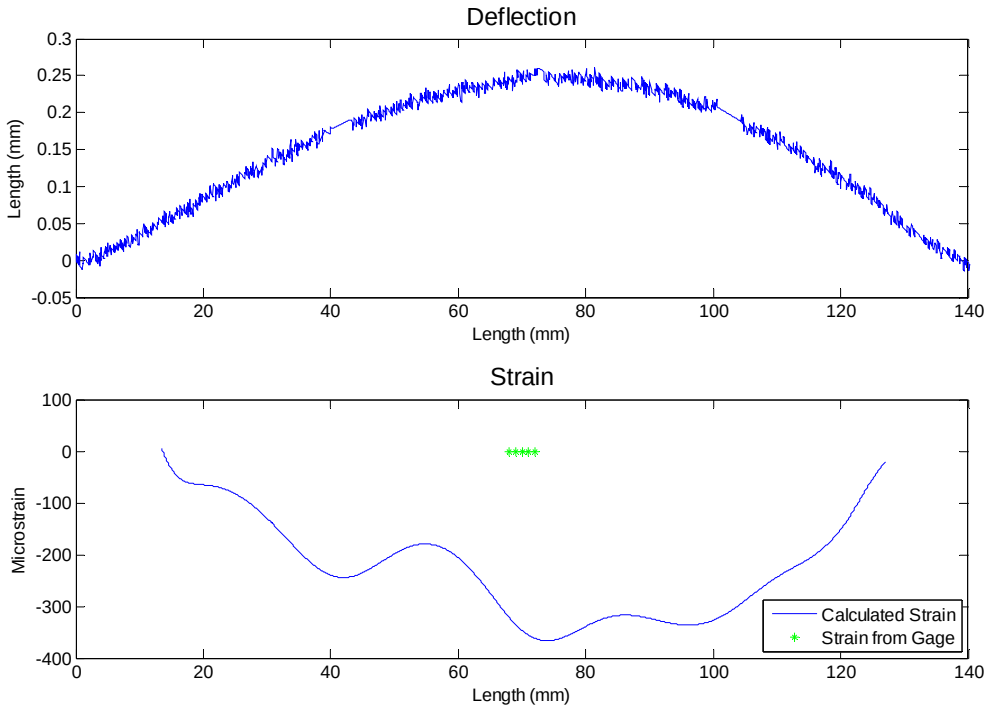


Figure: Data Set B2 – Image 02, Gage Microstrain: Zeroed

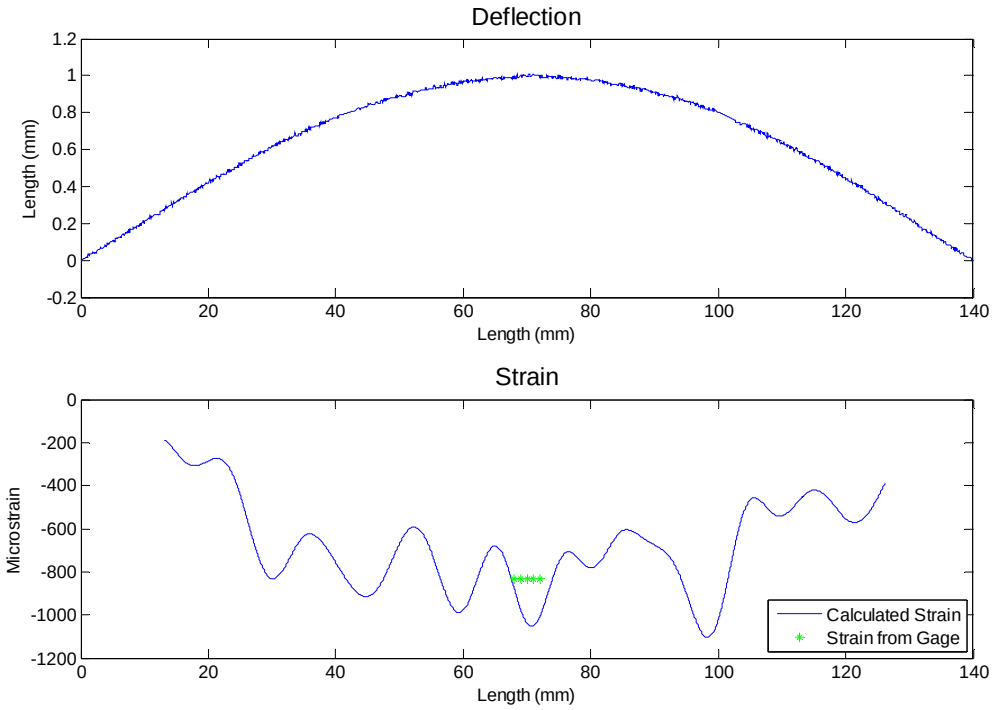


Figure: Data Set B2 – Image 07, Gage Microstrain: 830

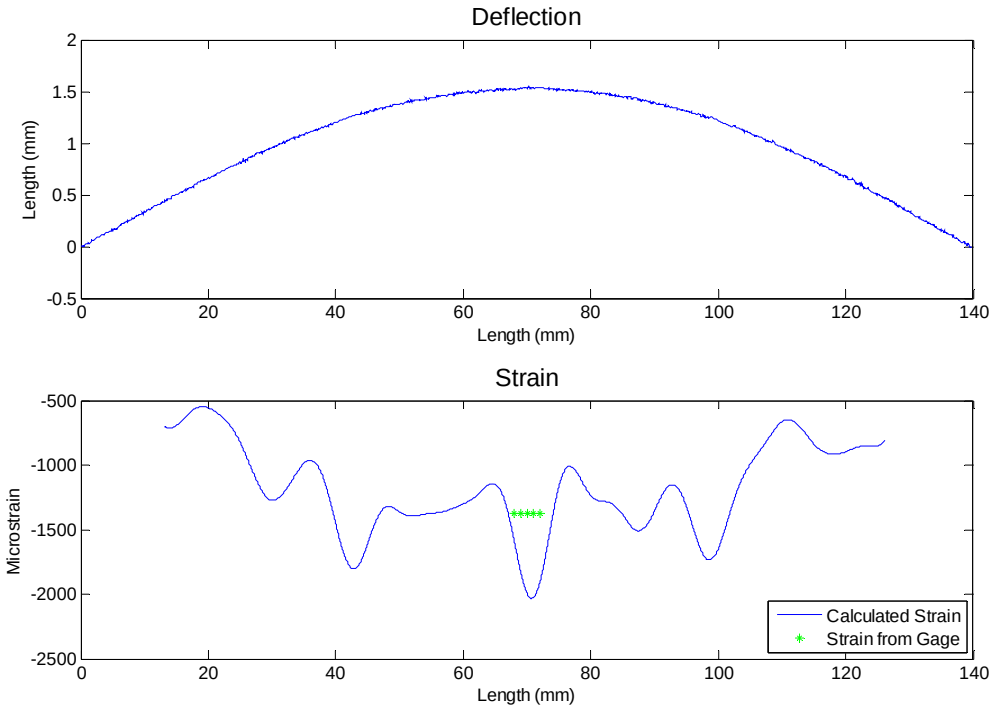


Figure: Data Set B2– Image 09, Gage Microstrain: 1370

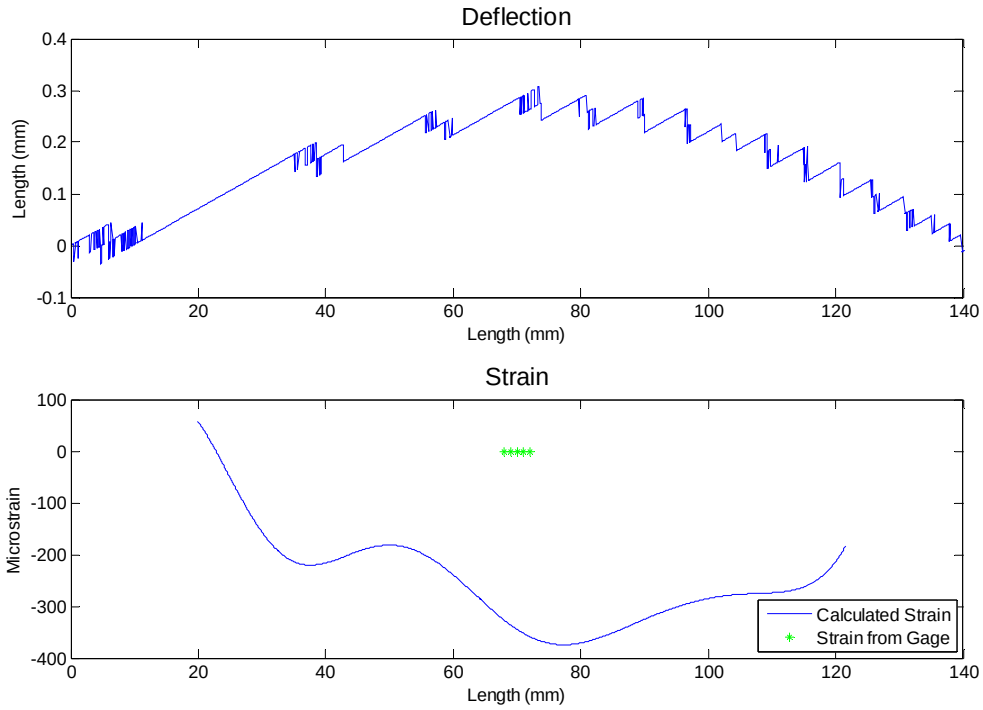


Figure: Data Set B3- Image 00, Gage Microstrain: Zeroed

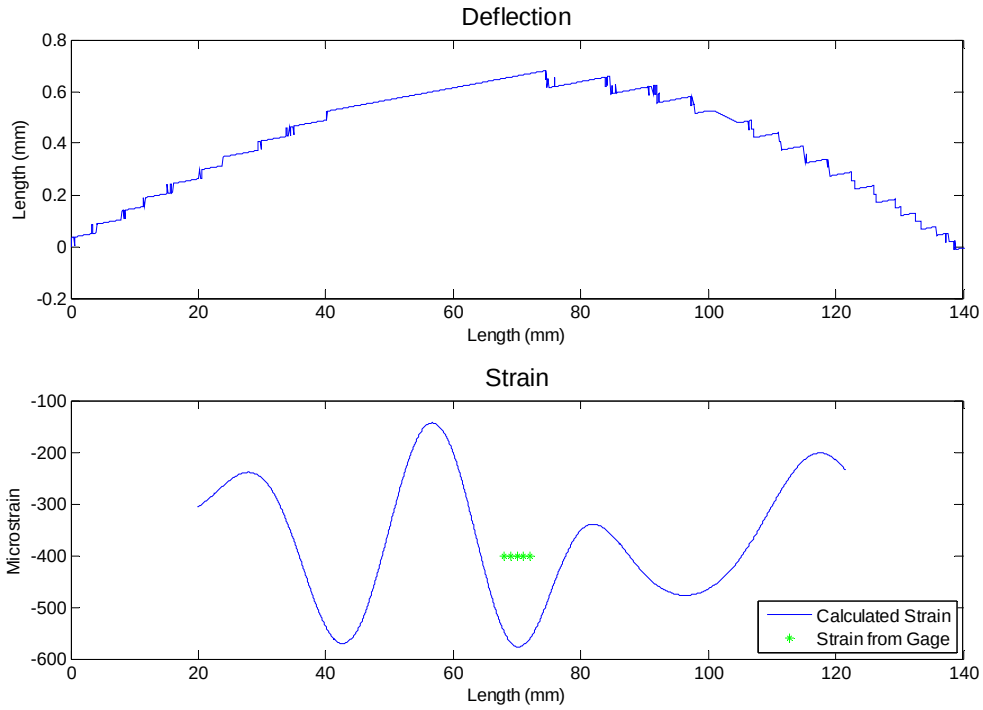


Figure: Data Set B3- Image 03, Gage Microstrain: 400

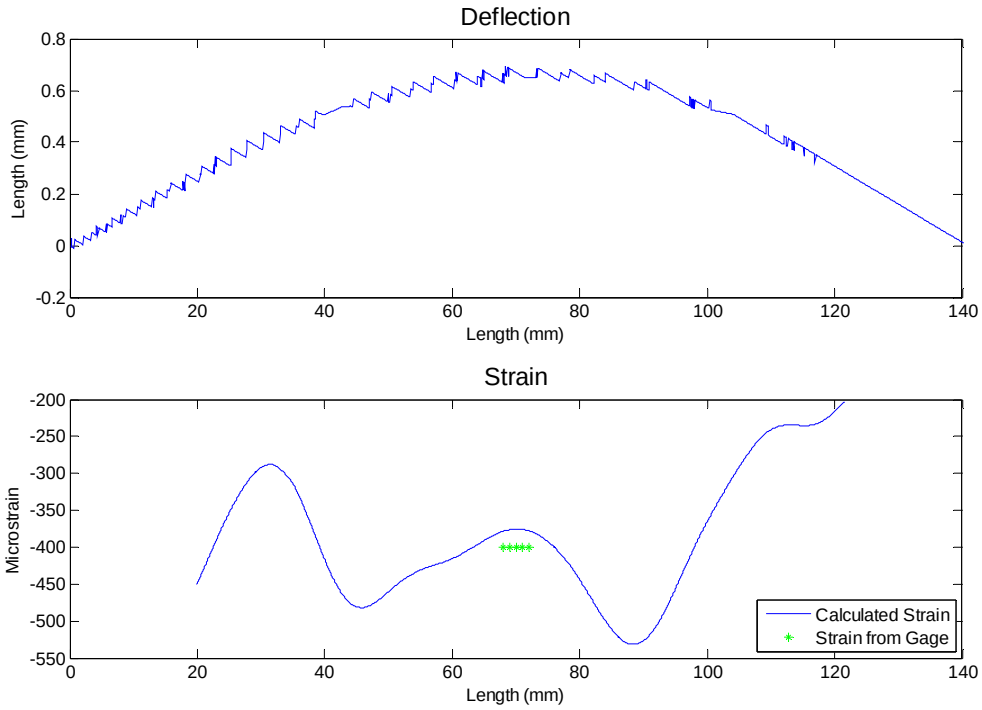


Figure: Data Set B3- Image 04, Gage Microstrain: 400

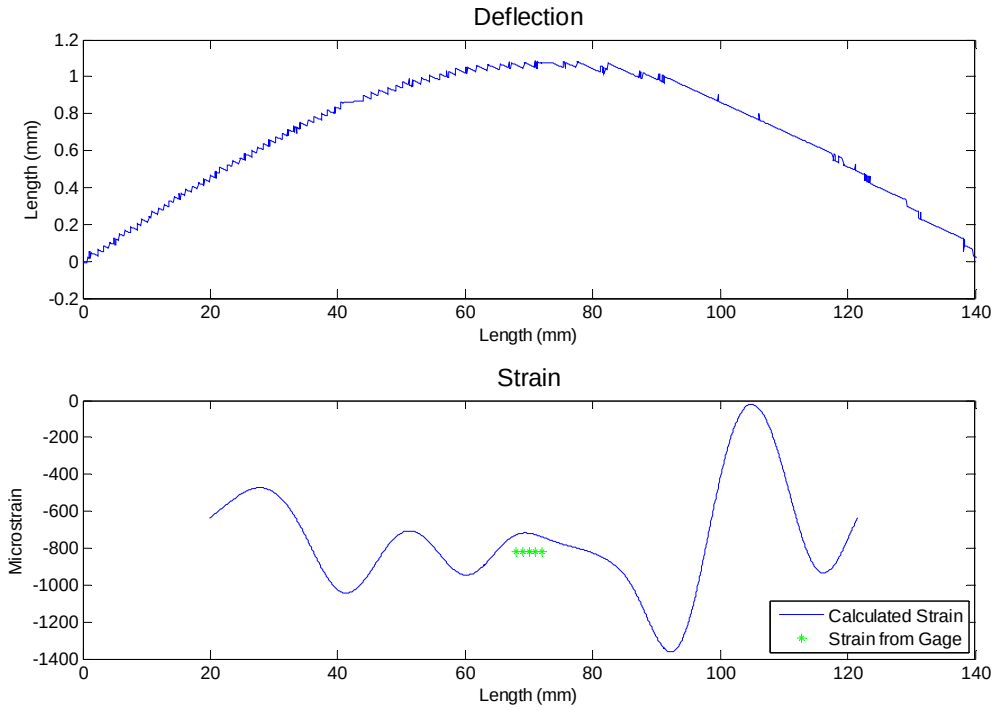


Figure: Data Set B3- Image 06, Gage Microstrain: 820

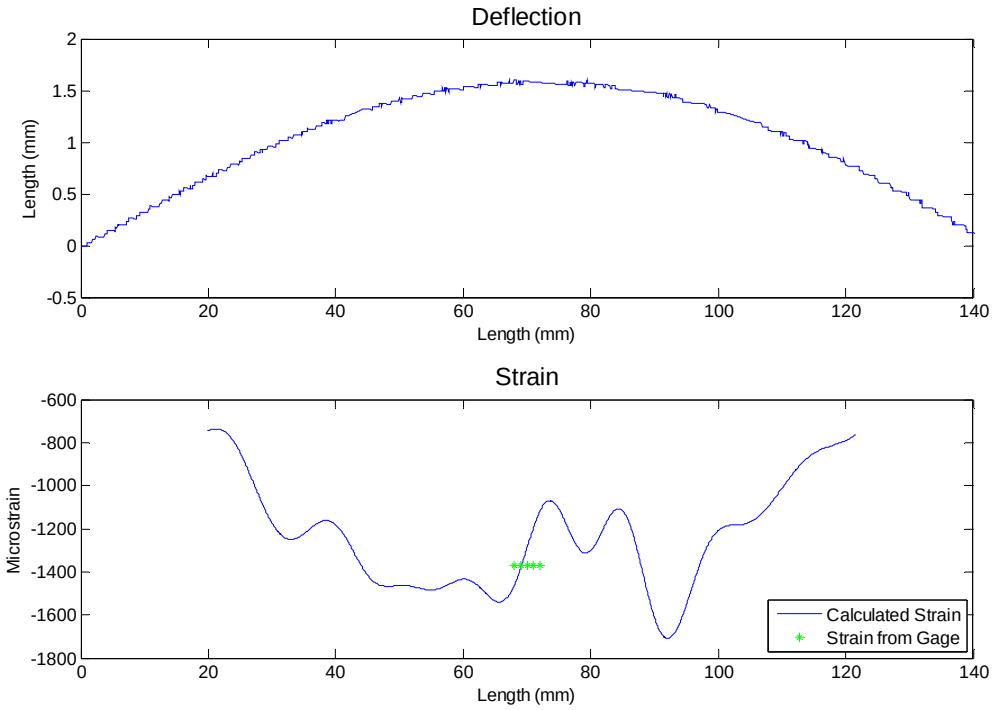


Figure: Data Set B3- Image 08, Gage Microstrain: 1370

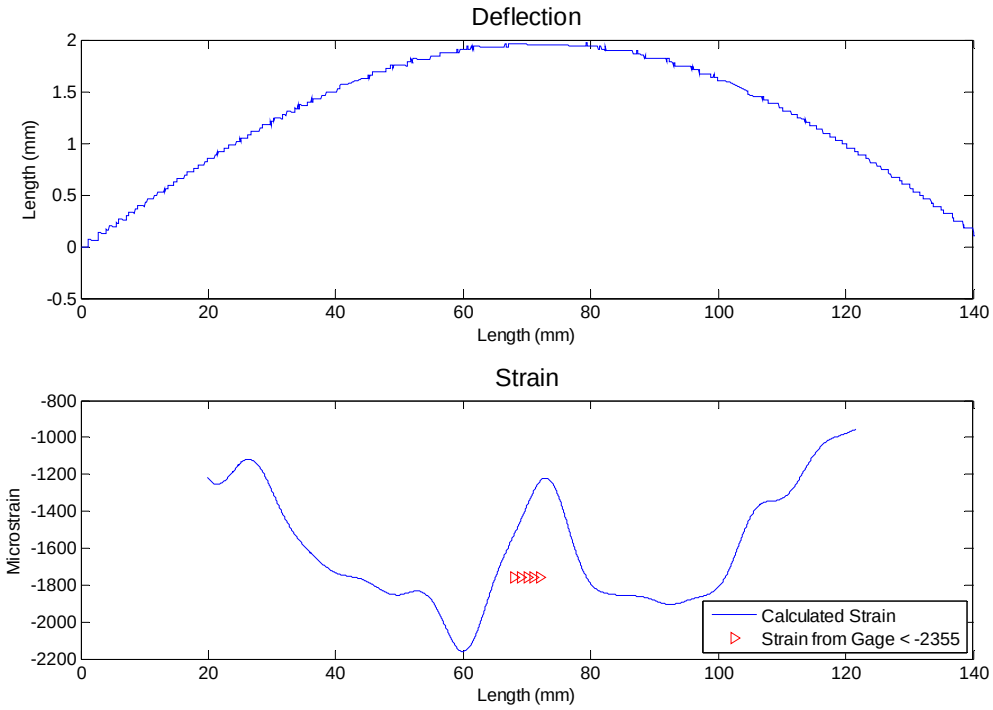


Figure: Data Set B3- Image 10, Gage Microstrain: 1760

Appendix B: Mock Connection Test Results
Method: Mollification and Discrete Derivatives

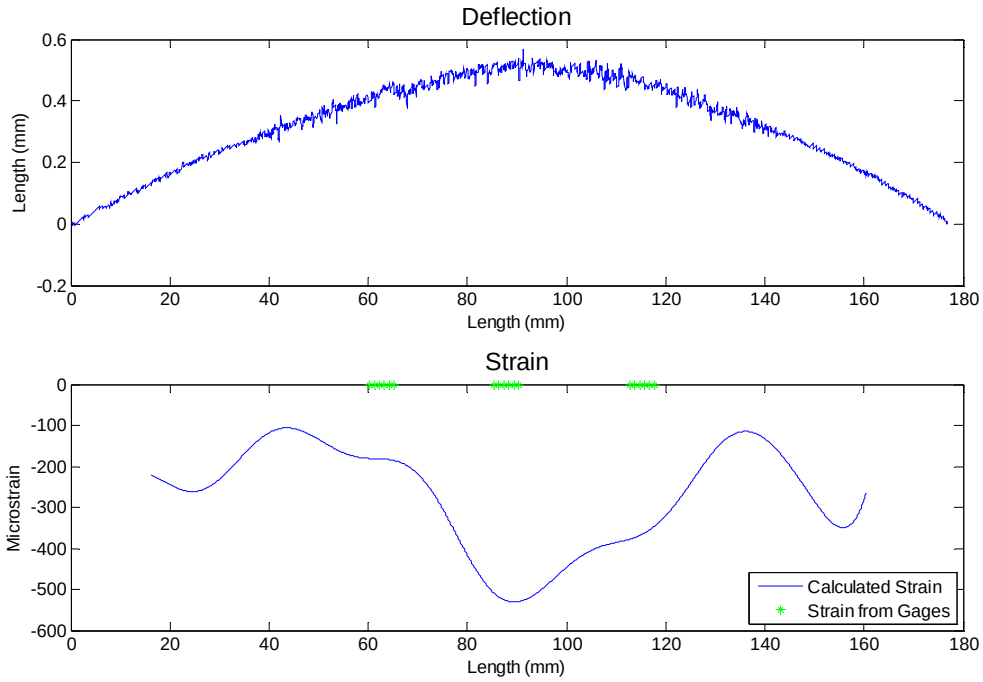


Figure: Data Set MC1- Image Zero (average of two zero strain images).
Gage Microstrain, Left: 0 Center: 0 Right: 0

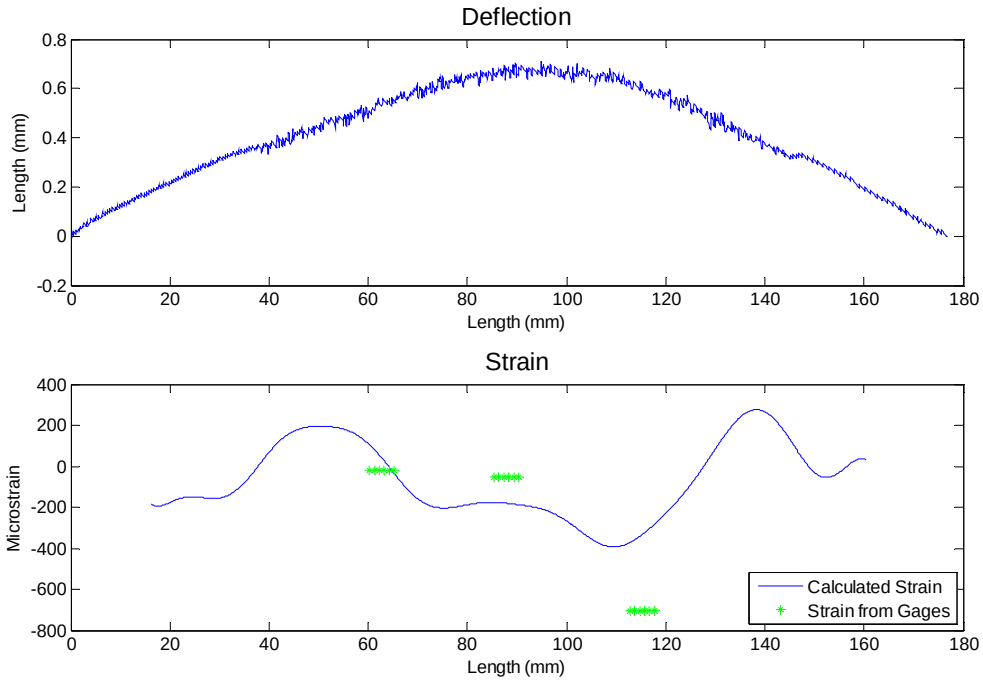


Figure: Data Set MC1- Image 1
Gage Microstrain, Left: -20 Center: -50 Right: -705

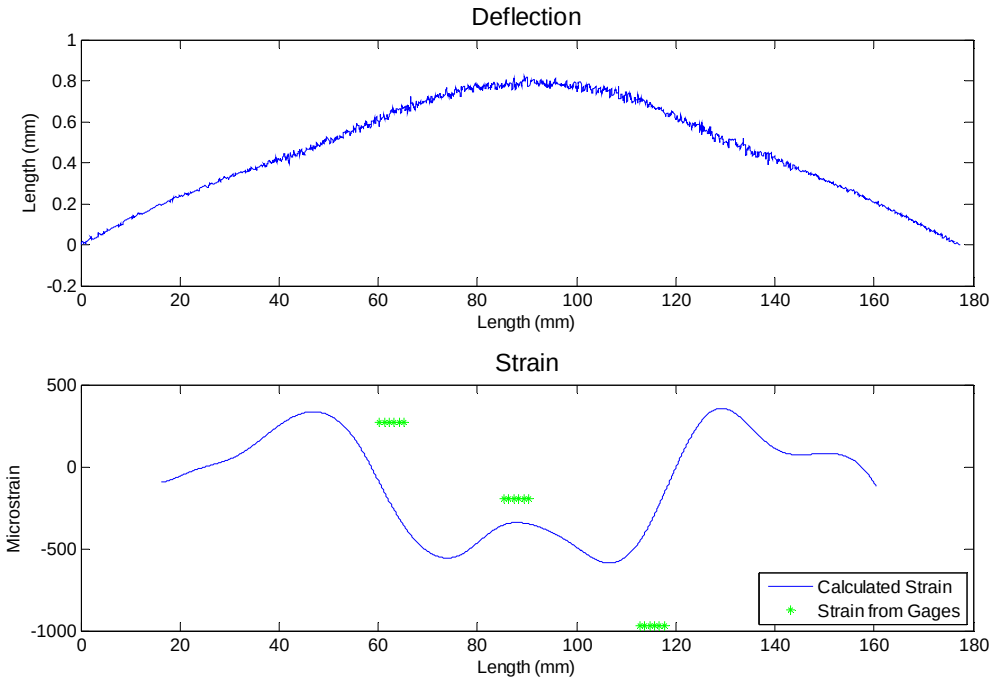


Figure: Data Set MC1- Image 2
Gage Microstrain, Left: 270 Center: -195 Right: -970

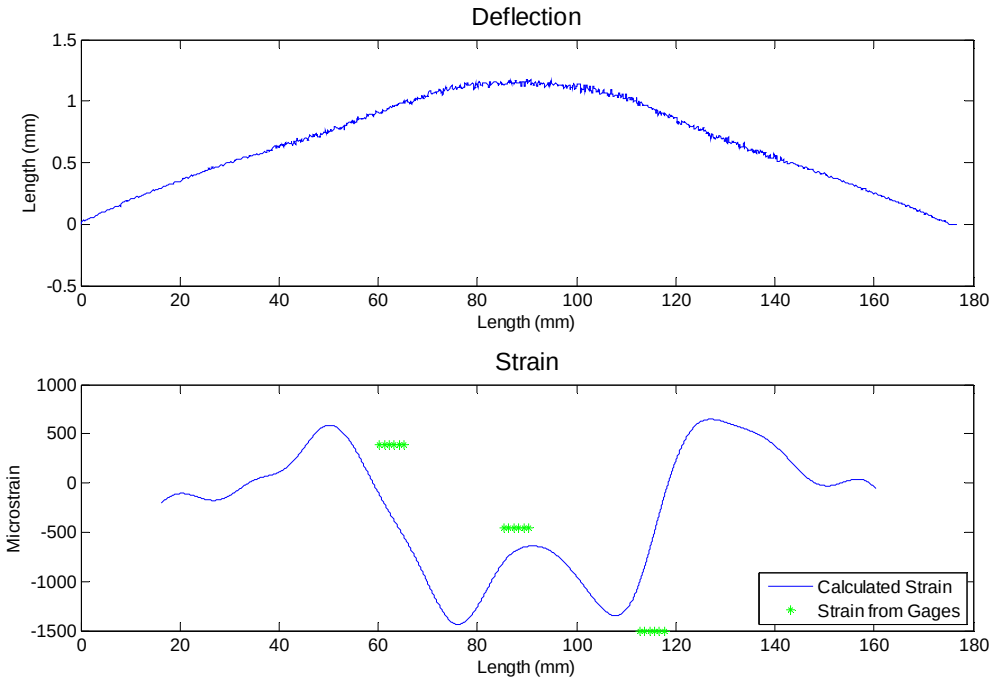


Figure: Data Set MC1- Image 3
Gage Microstrain, Left: 385 Center: -450 Right: -1500

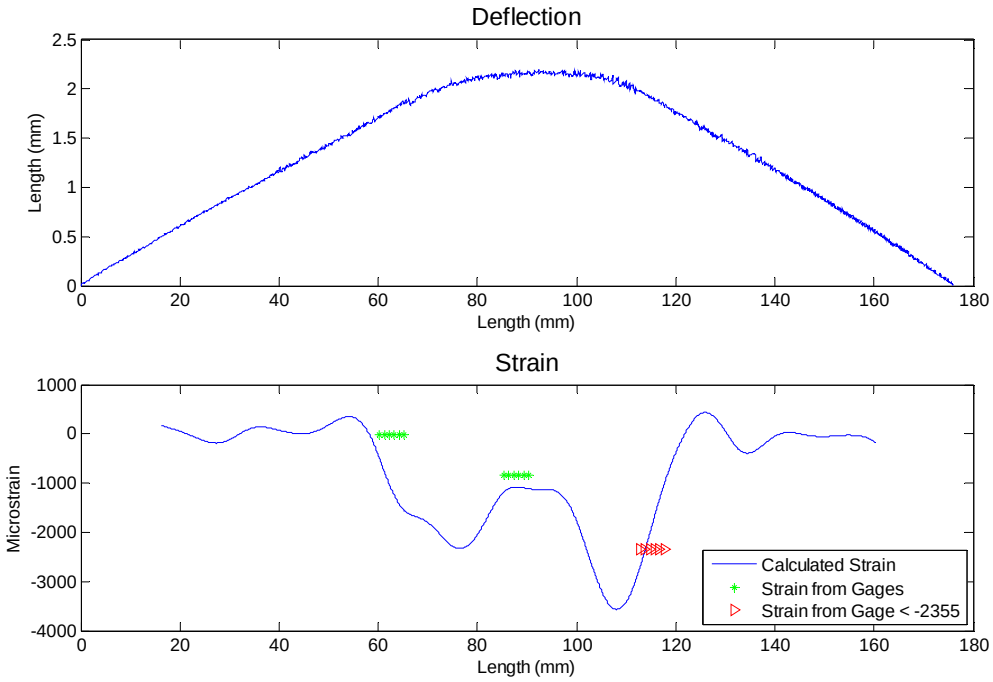


Figure: Data Set MC1- Image 4
Gage Microstrain, Left: 10 Center: -831 Right: <-2355

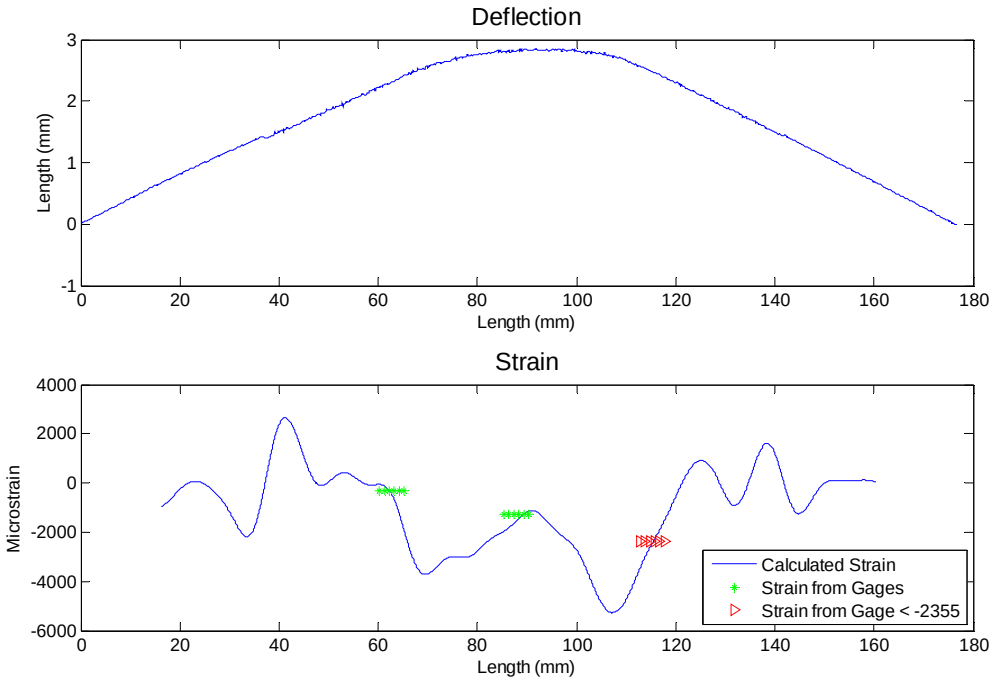


Figure: Data Set MC1- Image 5
Gage Microstrain, Left: -313 Center: -1275 Right: <-2355

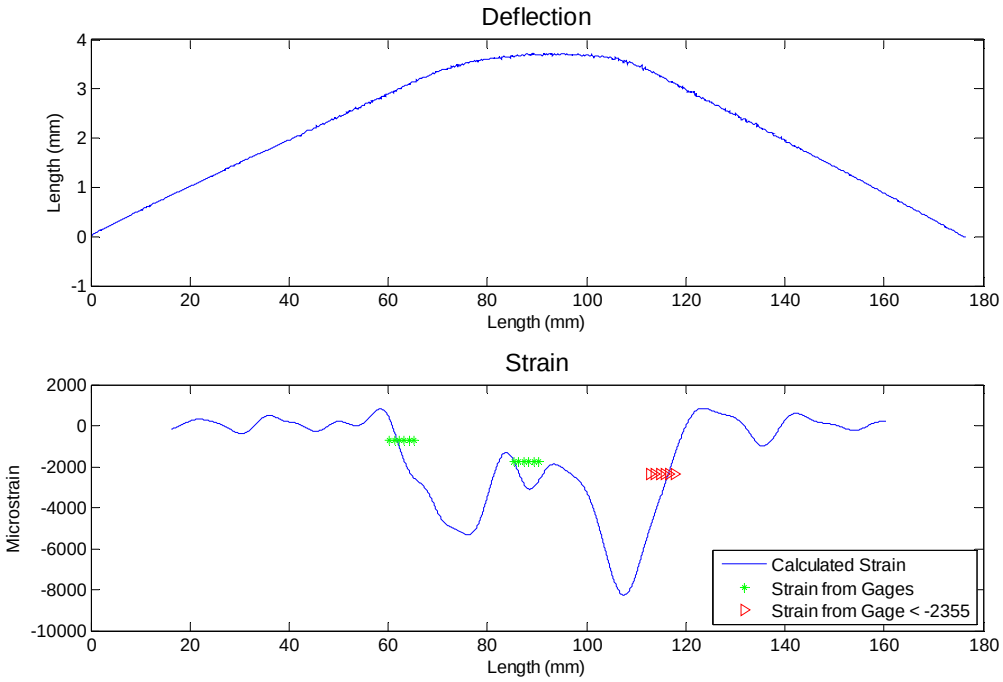


Figure: Data Set MC1- Image 6
Gage Microstrain, Left: -716 Center: -1765 Right: <-2355

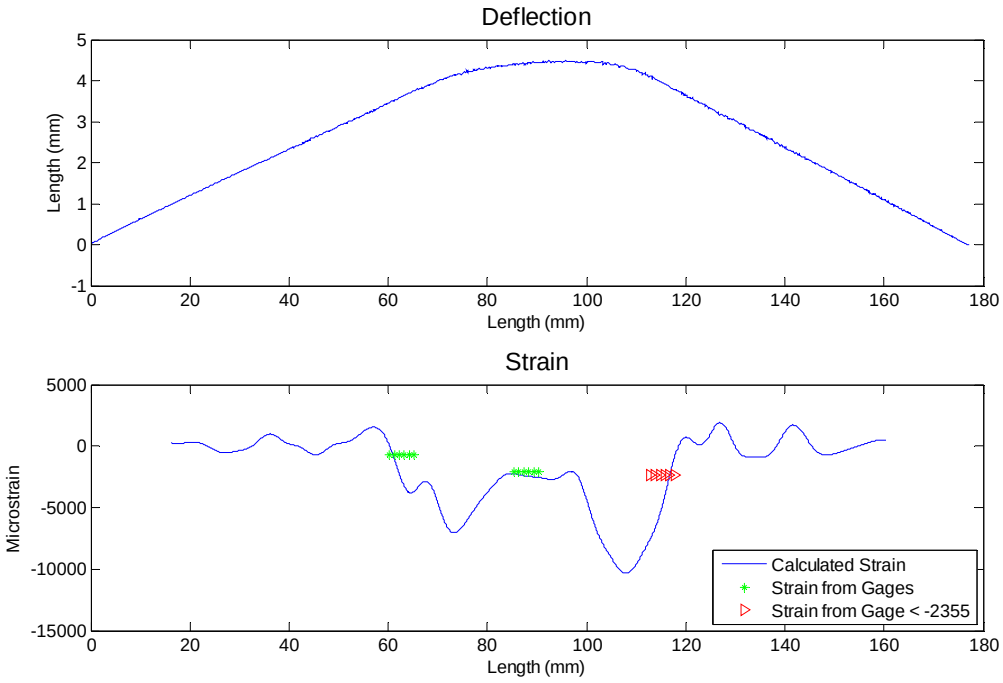


Figure: Data Set MC1- Image 7
Gage Microstrain, Left: -715 Center: -2080 Right: <-2355

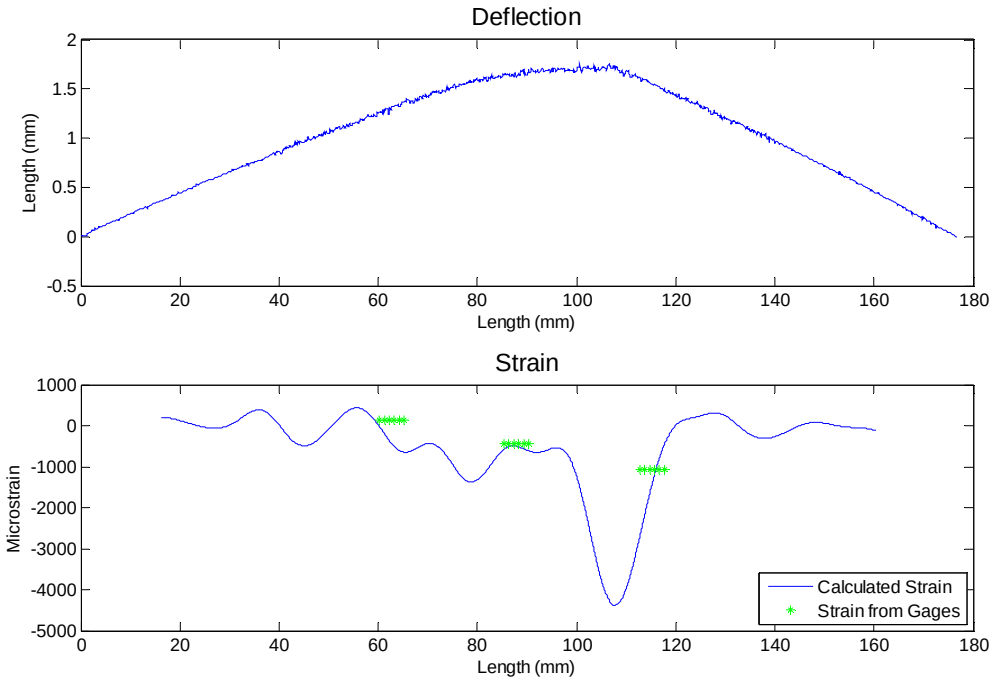


Figure: Data Set 1- Image MC8
Gage Microstrain, Left: 135 Center: -446 Right: -1078

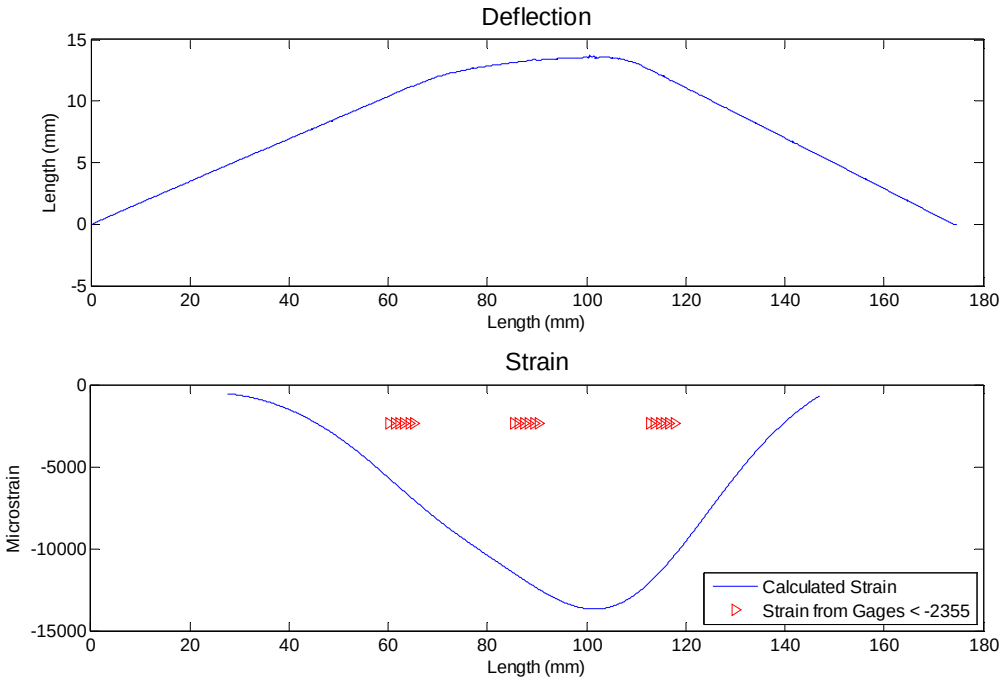


Figure: Data Set MC1- Image 9
Gage Microstrain, All gages: < -2355

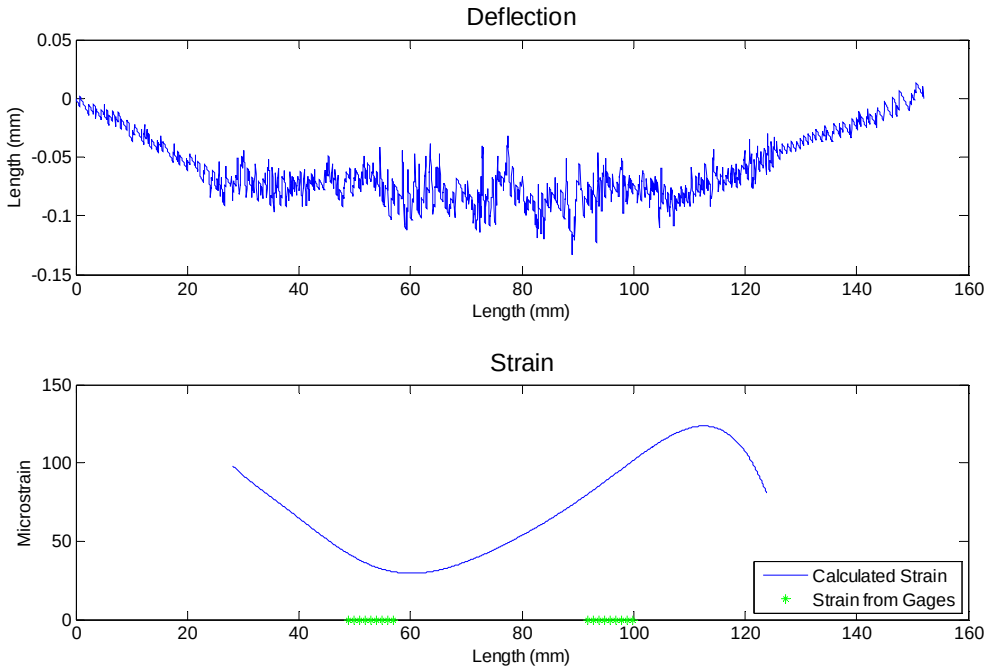


Figure: Data Set MC2- Image Zero
Gage Microstrain, Left: 0 Right: 0

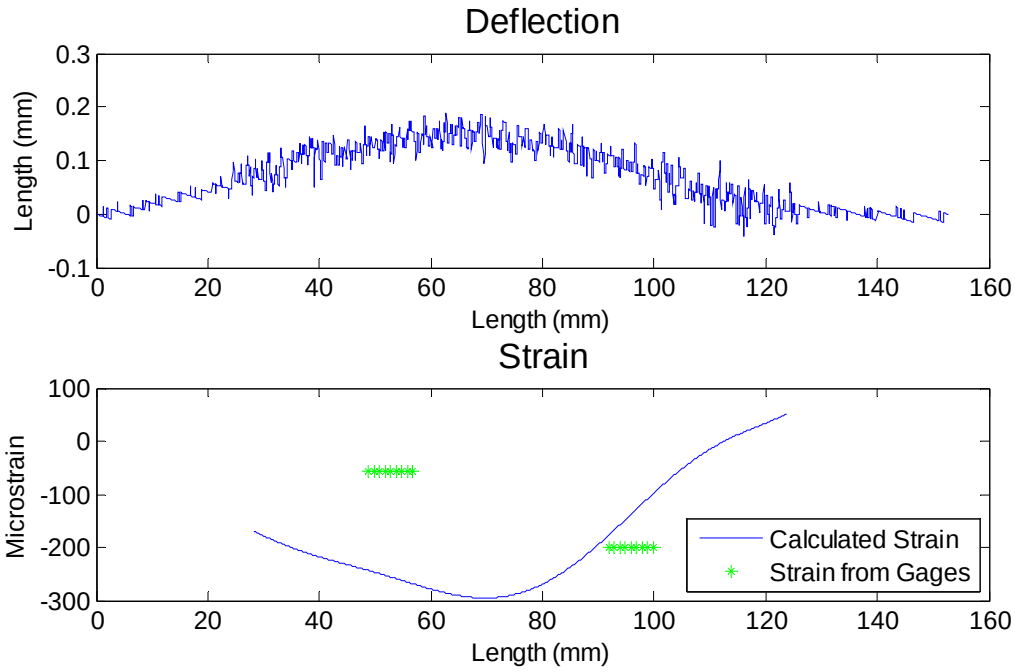


Figure: Data Set MC2- Image 1
Gage Microstrain, Left: -58 Right: -200

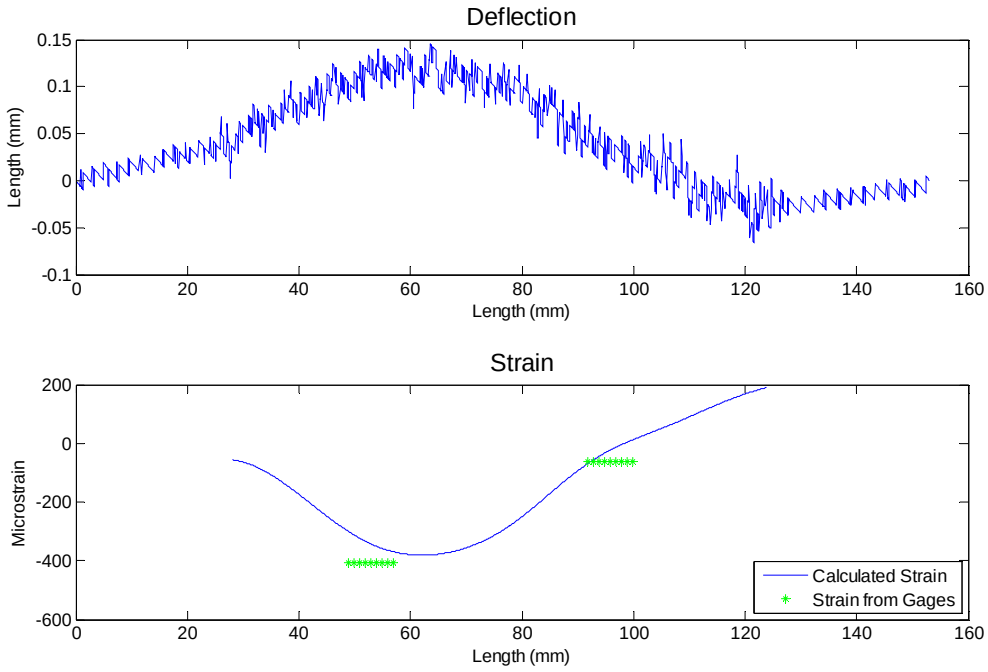


Figure: Data Set MC2- Image 2
Gage Microstrain, Left: -406 Right: -61

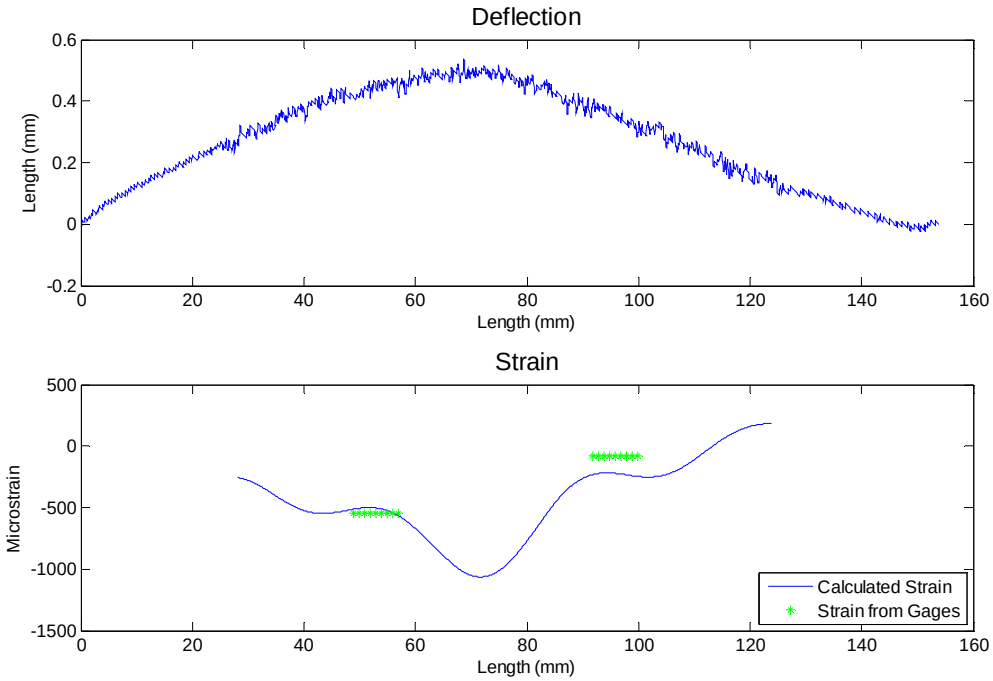


Figure: Data Set MC2- Image 3
Gage Microstrain, Left: -545 Right: -77

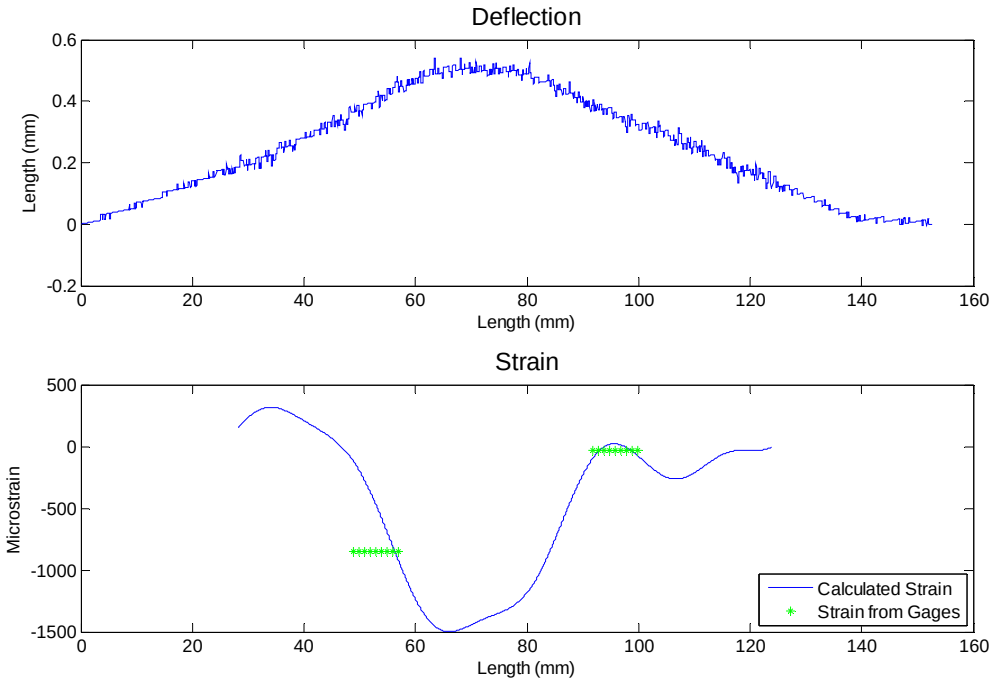


Figure: Data Set MC2- Image 4
Gage Microstrain, Left: -850 Right: -30

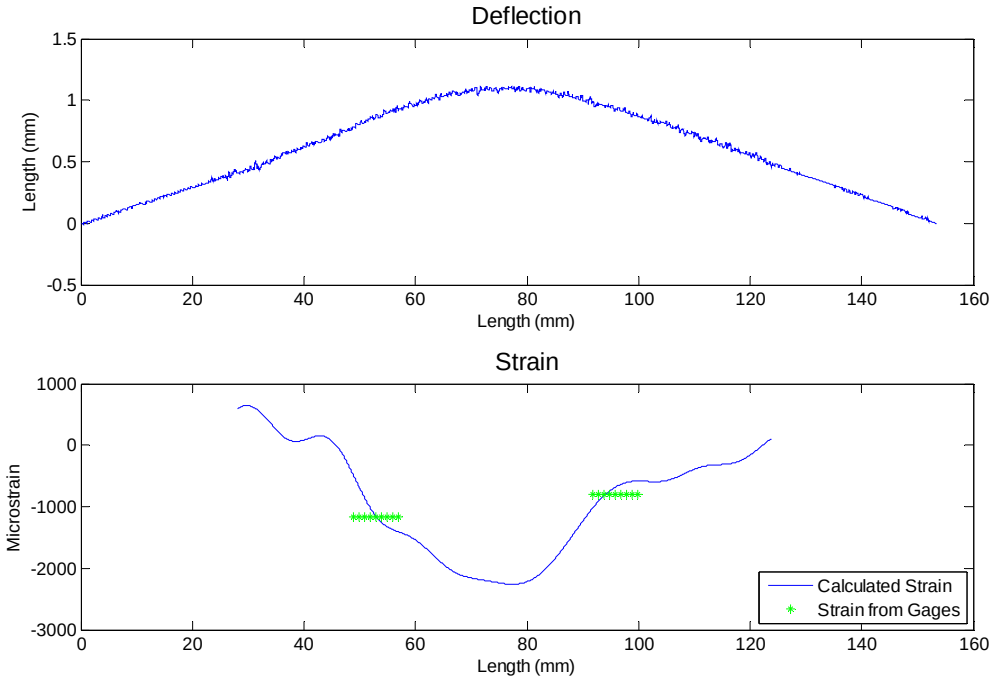


Figure: Data Set MC2- Image 5
Gage Microstrain, Left: -1160 Right: -790

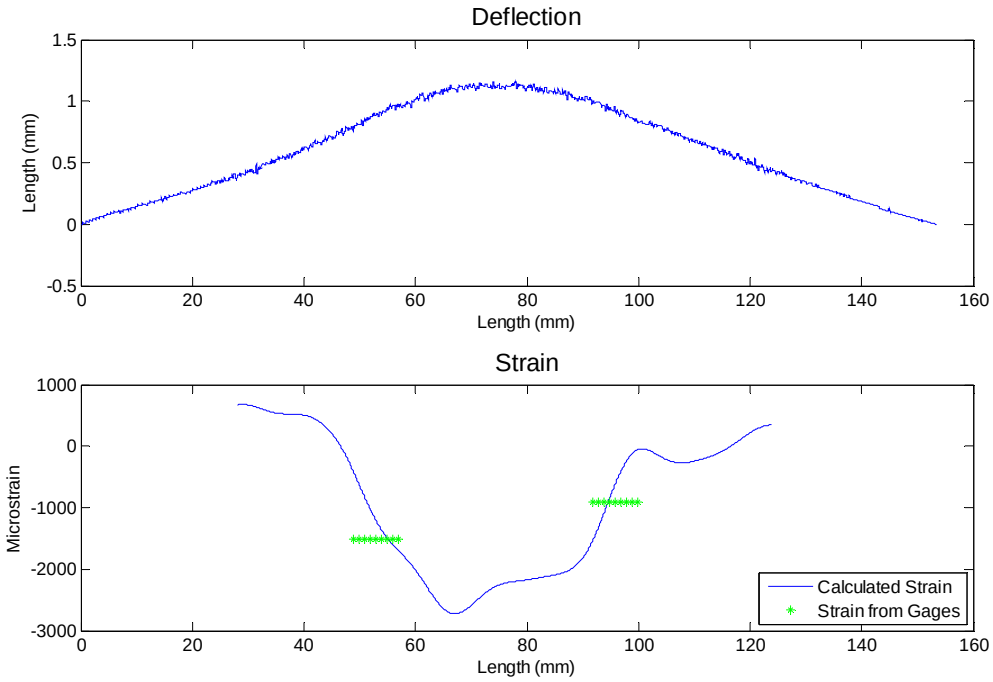


Figure: Data Set MC2- Image 6
Gage Microstrain, Left: -1510 Right: -902

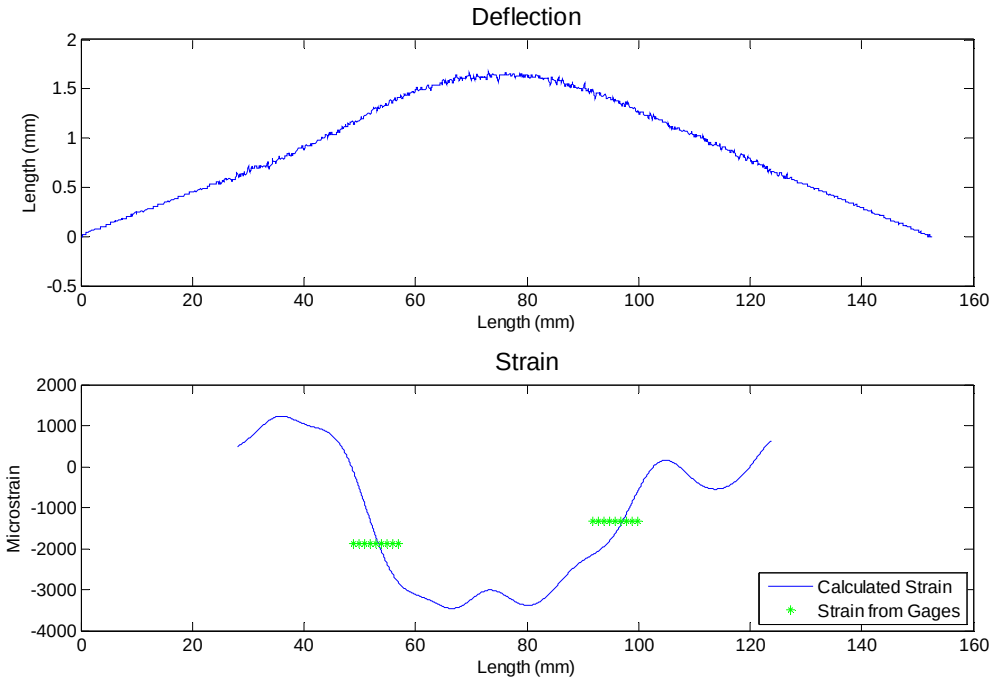


Figure: Data Set MC2- Image 7
Gage Microstrain, Left: -1877 Right: -1319

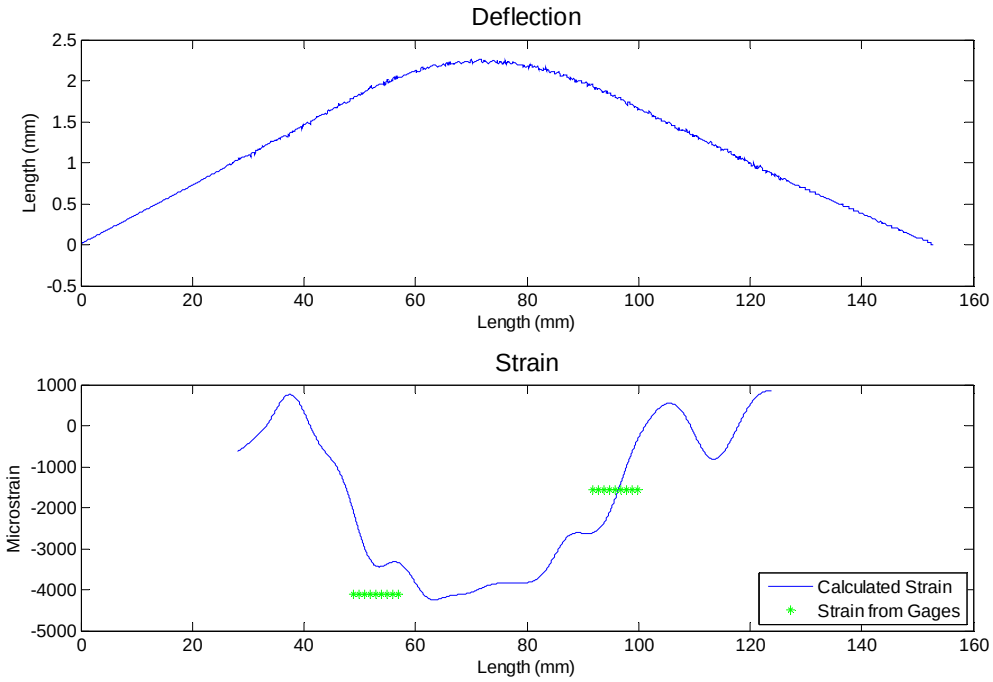


Figure: Data Set MC2- Image 8
Gage Microstrain, Left: -4100 Right: -1570

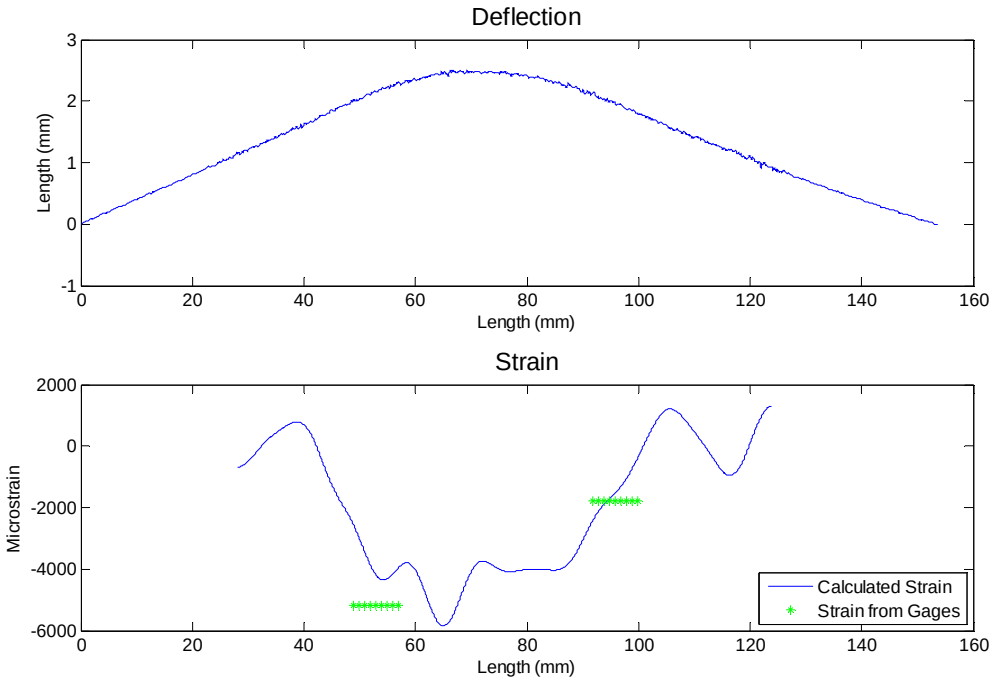


Figure: Data Set MC2- Image 9
Gage Microstrain, Left: -5195 Right: -1795

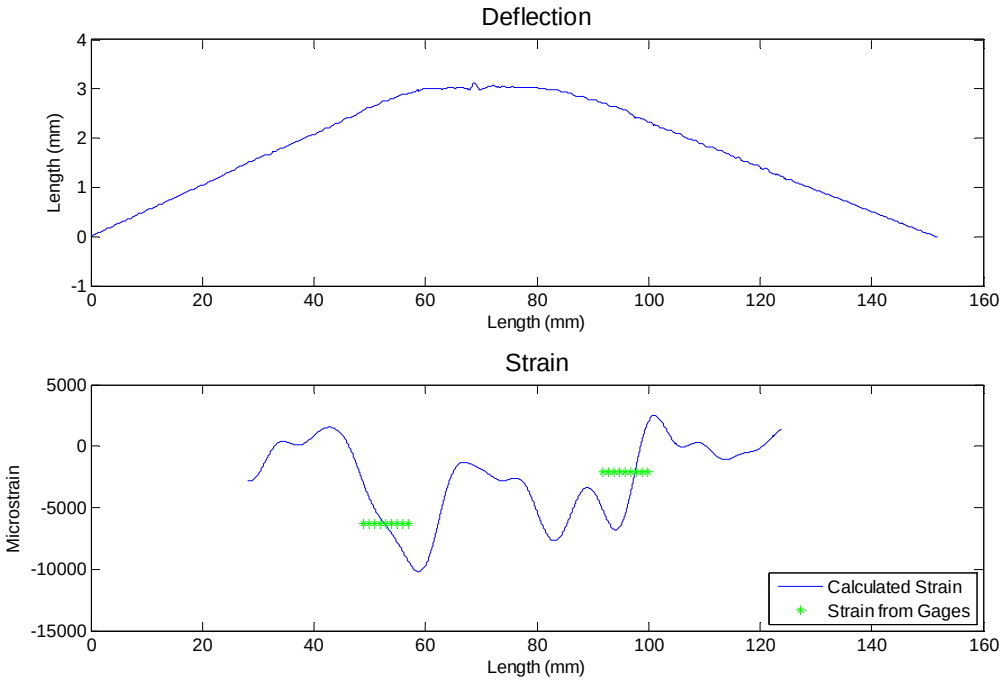


Figure: Data Set MC2- Image 10
Gage Microstrain, Left: -6278 Right: -2110

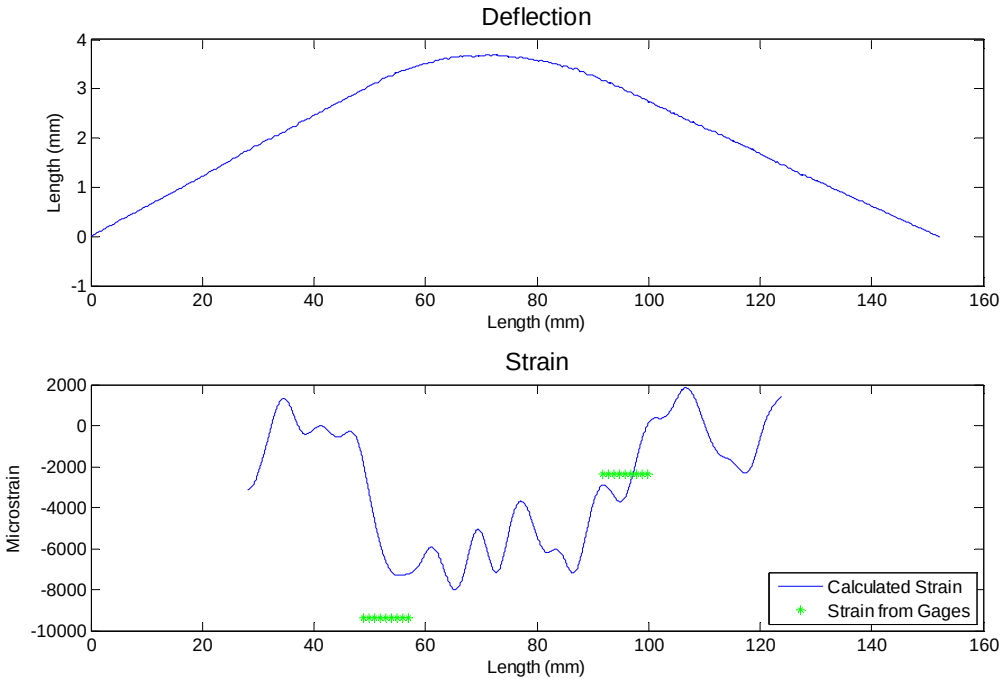


Figure: Data Set 2- Image 11. Gage Microstrain, Left: -9380 Right: -2371

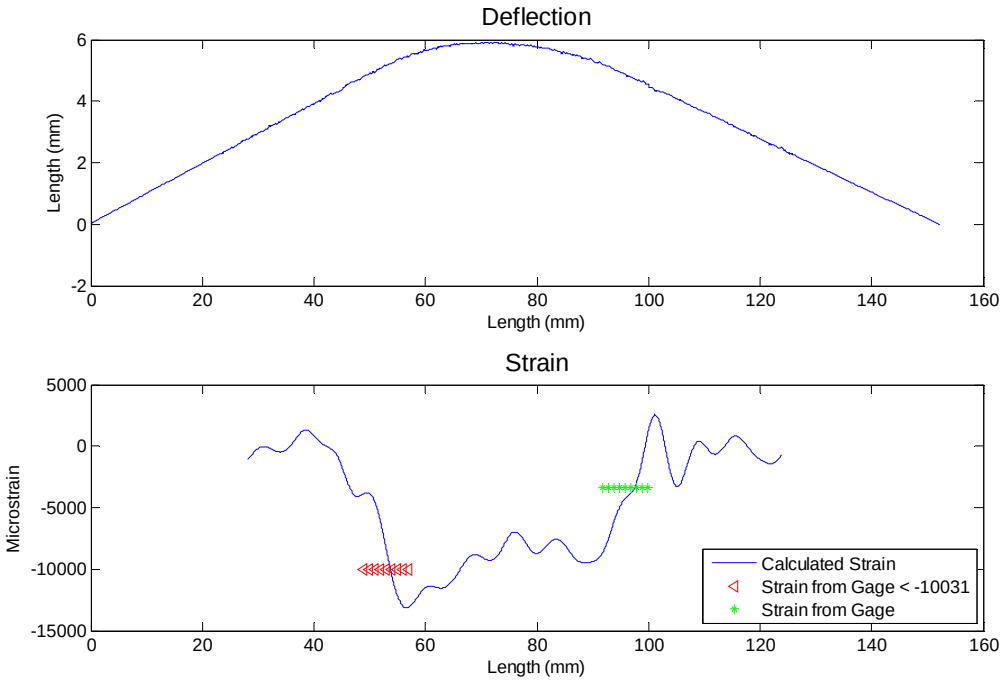


Figure: Data Set MC2- Image 12
Gage Microstrain, Left: <-10031 Right: -3355

Appendix C: Matlab Code

Main Programs and Scripts

```
%% Script file to extract edge data from dowel using Otsu's Method

%% Threshold image, and use image complement so dowel pixels have value
"1"
threshold = graythresh(I);
BW = im2bw(I,threshold);
BW=imcomplement(BW);

%% Get Top Edge Data
[r c]=size(BW);
col = 1;
row = find(BW(:,col), 1);
connectivity = 8;
num_points   = c+1000;
%num_points = the number of data points picked up
%1000 points are added to the width of the image to account for the
curving
%of the dowel and also problem areas such as the LVDT
contour = bwtraceboundary(BW, [row, col], 'N', connectivity,
num_points);
x=contour(:,2);
y=contour(:,1);
XYt(:,1)=x;XYt(:,2)=y;
imshow(BW);hold on;plot(x,y,'g');pause;
%shows image with edge data overlay. press "enter" to continue.
%at this point, remove points which are not on edge

%% Get Bottom Edge Data
[r c]=size(BW);
col = 1;
imshow(BW);pause;
%zoom in on left edge of bottom edge. press "enter" and then select
first point
[d e]=getpts;
row = round(e);
connectivity = 8;
num_points   = c+1000;
contour = bwtraceboundary(BW, [row, col], 'E', connectivity,
num_points,'counterclockwise');
x=contour(:,2);
y=contour(:,1);
imshow(BW);hold on;plot(x,y,'g');
XYb(:,1)=x;XYb(:,2)=y;
%shows image with edge data overlay. press "enter" to continue.
%at this point, remove points which are not on edge

%% Get averaged top and bottom edge data
avgXY=ones(size(XYt));
for i=1:size(XYt,1)
    for j=1:size(XYb,1)
    if XYt(i,1)== XYb(j,1)
        avgXY(i,1)=XYt(i,1);
```

```
        avgXY(i,2)=(XYt(i,2)+XYb(j,2))/2;
    end
    end
end
plot(XYt(:,1),XYt(:,2));hold
on;plot(XYb(:,1),XYb(:,2),'g');plot(avgXY(:,1),avgXY(:,2),'y');
```

```
function [XY,XYt,XYb,avgXY]=getpointsconnection(I,d,ppi,per)
%[XY,XYt,XYb,avgXY]=getpointsconnection(I,d,ppi,per)
%Get x-y data from dowel image
%INPUT:
%       I = grayscale image of dowel (cropped close around single
dowel)
%       d = depth of dowel (in inches)
%       ppi = pixels per inch
%       per = percent of d*ppi to use for window in search for edges
%OUTPUT:
%       XY = n by 2 matrix of x and y values from both edges
%       XYt = n by 2 matrix of x and y values from the top edge
%       XYb = n by 2 matrix of x and y values from the bottom edge
%       avgXY = n by 2 matrix of the averaged x and y values from the
top
%               and bottom edge

%Notes:
%The last part of the algorithm removes some data which isn't
%actually on the dowel but was picked up. Points still present but not
%a part of the dowel must be removed manually

%If data is missing from parts of the dowel, try increasing "per" value

%sz can be changed, although 11 works well to smooth out small local
%maximums

%If there are difficulties generating data from image, use commented
code
%at the bottom of the function to examine columns of data

%Andrew R. Adams, 2009
%ara149@psu.edu

sz=11; %can be changed before running procedure (must be odd)
havg=fspecial('average',sz);
II=imfilter(double(I),havg,'replicate');
diffI=diff(II,1);
sdiffI=imfilter(diffI,havg,'replicate');
vdiffI=abs(sdiffI);
vdiffI(vdiffI<10)=0;
%value "10" can be changed; higher values require less computational
effort
%but will not work for low contrast images.
xmax=ceil((1+per/100)*d*ppi);
xmin=floor((1-per/100)*d*ppi);
%This section takes each column, finds local maximums which are the
%specified distance apart, and then out of that group keeps the one
with
%the max value and its partner with the max value.
bw=zeros(size(vdiffI));
for i=1:size(vdiffI,2)
    [dd gg]=lmax(vdiffI(:,i),2);

    xx=[];
```

```
    yy=[];
    tmpxx=[];
    tmpyy=[];
    for j=1:size(gg,2)
        for k=2:size(gg,2)
            if ((gg(k)-gg(j))>=xmin && (gg(k)-gg(j))<=xmax)
                tmpxx=[gg(j) gg(k)];
                tmpyy=[dd(j) dd(k)];
            else tmpxx=[];tmpyy=[];
            end
            xx=[xx;tmpxx];
            yy=[yy;tmpyy];
        end
    end
    if size(yy,1)>1
        [c d]=find(yy==max(max(yy)));
        yy=yy(sort(c),:);
        xx=xx(sort(c),:);
        [c d]=max(sum(yy,2));
        xx=xx(d,:);
    end
    bw(xx,i)=1;
end
%Get x-y data
[y x]=find(bw==1);

XY(:,1)=x;XY(:,2)=y;
XYt=zeros(size(XY,1)/2,2);
XYb=zeros(size(XYt));
v=1:2:size(XY,1);
for i=1:size(v,2)
    XYt(i,:)=XY(v(i),:);
    XYb(i,:)=XY(v(i)+1,:);
end
% Get Avg XY data.
avgXY=zeros(size(XYt,1),1);
for i=1:size(XYt,1)
    avgXY(i,1)=XYt(i,1);
    avgXY(i,2)=(XYt(i,2)+XYb(i,2))/2;
end

plot(avgXY(:,1),avgXY(:,2),'.');
%after running code, some points may need to be removed from edges

%%TROUBLESHOOTING CODE (to use, uncomment rest of code once)
% sz=input('input size of averaging filter [11]: ');
% if isempty(sz);
%     sz=11;
% end
% havg=fspecial('average',sz);
% II=imfilter(double(I),havg,'replicate');
% diffI=diff(II,1);
% sdiffI=imfilter(diffI,havg,'replicate');
% vdiffI=abs(sdiffI);
% vdiffI(vdiffI<25)=0;
% clear II diffI sdiffI havg sz
```

```
% %run pack
%
%%next section can be used to look at random columns to determine
%%appropriate ppi or to look at random columns to see how the how the
%%pixel intensities in the region of the dowel are behaving
% col=3000; %change this value to look at different columns
% figure(1);
%
subplot(2,1,1);plot(I(:,col),'.');subplot(2,1,2);plot(vdiffI(:,col),'.'
);
```

```
function [XYt,XYb,avgXY]=getpointslog(Is)
%function [x,yavg]=getpointslog(Is)
%Work in progress method to get x-y data from dowel image using LoG
filtering %and second derivative zero crossings
%INPUT:
%       I = grayscale image of dowel (cropped close around single
dowel)
%OUTPUT:
%       XYt = n by 2 matrix of x and y values from the top edge
%       XYb = n by 2 matrix of x and y values from the bottom edge
%       avgXY = n by 2 matrix of the averaged x and y values from the
top
%               and bottom edge


s=55; %value can be adjusted
yedge1=zeros(size(Is,2),length(s));
yedge2=yedge1;
xedge=[1:size(Is,2)]';
h=fspecial('log',s,s/6); %use standard deviation of s/6
III=imfilter(double(Is),h);
for j=1:size(III,2)
    %break column data into top and bottom by finding a point in the
middle
    %of the dowel
%     [cx,dx]=sort(III(:,j));
%     cut=median(dx(1:501));
    %value used is 301. must be odd and close to but less than the
width of
    %the dowel
    cut=floor(size(III,1)/2);
    col1=III(ceil(s/2):cut,j);
    col2=III(cut+1:end-floor(s/2),j);
    col=[col1;col2];
    [maxc1,indmax1]=max(col1);
    [minc1,indmin1]=min(col1);
    [maxc2,indmax2]=max(col2);
    [minc2,indmin2]=min(col2);
    %find location closest to zero between max and min values
    if indmin1<indmax1
        [appzero1,indzero1]=min(abs(col1(indmin1:indmax1)));
        indzero1=indzero1+indmin1-1;
        x1=[indzero1-1 indzero1 indzero1+1];
        if indzero1==1
            indzero1=2;
        end
        y1=[col(indzero1-1) col(indzero1) col(indzero1+1)];
        p1=polyfit(x1,y1,1);
        yedge1(j,1)=roots(p1)+ceil(s/2);
    else yedge1(j,1)=NaN;
    end
    if indmax2<indmin2
        [appzero2,indzero2]=min(abs(col2(indmax2:indmin2)));
        indzero2=indzero2+indmax2+length(col1)-1;
        x2=[indzero2-1 indzero2 indzero2+1];
        if indzero2==1
            indzero2=2;
```

```
        end
        y2=[col(indzero2-1) col(indzero2) col(indzero2+1)];
        p2=polyfit(x2,y2,1);
        yedge2(j,1)=roots(p2)+ceil(s/2);
    else yedge2(j,1)=NaN;
    end
end

XYt(:,1)=xedge;
XYt(:,2)=yedge1;
XYb(:,1)=xedge;
XYb(:,2)=yedge2;
% XYt(isnan(yedge1),:)=[];
% XYb(isnan(yedge2),:)=[];
% yavg=(yedge1+yedge2)./2;
% x=xedge;
% x(isnan(yavg))=[];
% yavg(isnan(yavg))=[];
% avgXY(:,1)=x;
% avgXY(:,2)=yavg;
%

diffXYtb=zeros(size(XYt,1),1);
for i=1:size(XYt,1)
    diffXYtb(i)=abs(XYt(i,2)-XYb(i,2));
end
avgXY=zeros(size(diffXYtb));
for i=1:size(XYt,1)
    if isnan(XYt(i,2)) && isnan(XYb(i,2))
        avgXY(i,1)=NaN;
        avgXY(i,2)=NaN;
    end
    if isnan(XYt(i,2))
        avgXY(i,1)=XYt(i,1);
        avgXY(i,2)=XYb(i,2)-mean(diffXYtb);
    end
    if isnan(XYb(i,2))
        avgXY(i,1)=XYt(i,1);
        avgXY(i,2)=XYt(i,2)+mean(diffXYtb);
    end
    if abs(XYt(i,2)-XYb(i,2))<(mean(diffXYtb)+3) %value can be adjusted,
        avgXY(i,1)=XYt(i,1);                     %3 works well
        avgXY(i,2)=(XYt(i,2)+XYb(i,2))/2;
    else
       avgXY(i,1)=XYt(i,1);
       avgXY(i,2)=XYt(i,2)+mean(diffXYtb)/2;
    end
end
```

```
function [xmm,strain,delta] = getstrain(XY,d,ppi)
%[xmm,strain,delta] = getstrain(XY,d,ppi)
%Get strain from data, using beam theory with small angle
approximations
%Algorithm uses curvature = k = d2y/dx2
%Instead of the actual curvature, k = (d2y/dx2)/((1+(dy/dx)^2))^0.5
%For use with data with small deflections

%INPUT:
%       XY = n by 2 matrix of x and y values (typically avgXY output
from
%       "getpointsconnection.m"
%       d = the depth of the dowel
%       ppi = the relationship between the dowel and pixel size
%           (ex. for d in inches, ppi would be pixels per inch)
%
%OUTPUT:
%       xmm = x values which correspond to strain values
%       strain = strain values from algorithm (in microstrain)
%       delta = a 1 by 2 vector of delta values from the mollification
%               procedure (to compare with min/max values input into
%               mollification algorithm)

%Andrew R. Adams, 2009
%ara149@psu.edu

C=d/2*10^6*ppi;
D=1/ppi*25.4;

x=XY(:,1);y=XY(:,2);
x=x-x(1);y=y-y(1);
[x,y]=nodups(x,y);
[moly,delta1]=molgcv1(y',length(y)-1);
num1=round(delta1*length(y));
moly=moly(num1:end-num1);
x=x(num1:end-num1);
[xs,ys]=secderiv(x,moly',3);
[molysecd,delta2]=molgcv1(ys',length(ys)-1);
num2=round(delta2*length(ys));
xmm=xs(num2:end-num2)*D;
strain=molysecd(num2:end-num2)*C;
delta=[delta1,delta2];
```

```
function [xmm,strain,delta] = getstrain2(XY,d,ppi)
%[xmm,strain,delta] = getstrain2(XY,d,ppi)
%Get strain from data, not simplifying curvature for small angle theory
%Curvature = k = (d2y/dx2)/((1+(dy/dx)^2))^0.5
%For use with data with large deflections
%INPUT:
%       XY = n by 2 matrix of x and y values (typically avgXY output
from
%       "getpointsconnection.m")
%       d = the depth of the dowel
%       ppi = the relationship between the dowel and pixel size
%           (ex. for d in inches, ppi would be pixels per inch)
%
%OUTPUT:
%       xmm = x values which correspond to locations of calc'd strain
%             values
%       strain = strain values from algorithm (in microstrain)
%       delta = a 1 by 2 vector of delta values from the mollification
%               procedure (to compare with min/max values input into
%               mollification algorithm)

%Andrew R. Adams, 2009
%ara149@psu.edu

C=d/2*10^6*ppi;
D=1/ppi*25.4;

x=XY(:,1);y=XY(:,2);
x=x-x(1);y=y-y(1);
[x,y]=nodups(x,y);
[moly,delta1]=molgcv1(y',length(y)-1);
num1=round(delta1*length(y));
x=x(num1:end-num1);
moly=moly(num1:end-num1);
[dmx,dmy]=firstderiv(x,moly,3);
[ddmx,ddmy]=secderiv(x,moly,3);
[molyfirstd,delta2]=molgcv1(dmy',length(dmy)-1);
num2=round(delta2*length(dmy));
dmx=dmx(num2:end-num2);
molyfirstd=molyfirstd(num2:end-num2);
[molysecd,delta3]=molgcv1(ddmy',length(ddmy)-1);
num3=round(delta3*length(ddmx)-1);
ddmx=ddmx(num3:end-num3);
molysecd=molysecd(num3:end-num3);
[molyfirstd,molysecd]=diffL(molyfirstd,molysecd);
[dmx,ddmx]=diffL(dmx,ddmx);
strain=(molysecd./((1+(molyfirstd).^2).^1.5)).*C;
delta=[delta1,delta2,delta3];
xmm=ddmx*D;
```

Mollification Code

```
function[molda,delta]=molgcv1(da,n)
%function[molda,delta]=molgcv1(da,n)
%
%This function calls mol.m and mol1.m
% Taken from Inverse Engineering Handbook [24]
%
% Slight modifications made by Andrew Adams, 2009
% 1. tol, deltamin, and deltamax values changed (original values are
noted)
% 2. Lines 103 to 108 added to return original input value if selected
delta
%    value is greater than deltamax or less than deltamin

% INPUT:
% Data vector da of dimension n+1.
% Number of points minus one, n
% OUTPUT:
% Mollified data vector, molda.
% Radius of mollification, delta.
%
% OPTION:
% The values of the GCV functional
% and/or the number of iterations
% can be activated or not. Simply
% comment (or not) the last two lines.
%
% tol=0.001;
% deltamin=0.001;
% deltamax=0.1   %original values
tol=0.0001;
deltamin=0.0001;
deltamax=0.165;
delta=0.04;
maxniter=40;
gr=0.5*(sqrt(5.0)-1);
cc=1-gr;
x0=deltamin;
x3=deltamax;
if(abs(deltamax-delta)>abs(delta-deltamin))
    x1=delta;
    x2=delta+cc*(deltamax-delta);
else
    x2=delta;
    x1=delta-cc*(delta-deltamin);
end
[iwtmax,wt]=mol(x1,n);
[molda,gcv]=mol1(iwtmax,n,wt,x1,da);
f1=gcv;
[iwtmax,wt]=mol(x2,n);
[molda,gcv]=mol1(iwtmax,n,wt,x2,da);
f2=gcv;
counter=0;
test1=abs(x3-x0);
test2=tol*(abs(x1)+abs(x2));
while((test1>test2)&&(counter<maxniter))
```

```
    counter=counter+1;
    if(f2<f1)
        x0=x1;
        x1=x2;
        x2=gr*x1+cc*x3;
        f1=f2;
        [iwtmax,wt]=mol(x2,n);
        [molda,gcv]=mol1(iwtmax,n,wt,x2,da);
        f2=gcv;
    else
        x3=x2;
        x2=x1;
        x1=gr*x2+cc*x0;
        f2=f1;
        [iwtmax,wt]=mol(x1,n);
        [molda,gcv]=mol1(iwtmax,n,wt,x1,da);
        f1=gcv;
    end
    test1=abs(x3-x0);
    test2=tol*(abs(x1)+abs(x2));
end
if(f1<f2)
    golden=f1;
    minx=x1;
else
    golden=f2;
    minx=x2;
end
delta=minx;
[iwtmax,wt]=mol(delta,n);
[molda,gcv]=mol1(iwtmax,n,wt,delta,da);
% ------------------------------------------------
% OPTION
if maxniter <= counter
    fprintf('Warning: maximum number of interations used.\n');
end
if delta-2*tol <= deltamin
    fprintf('Warning: delta value is deltamin.\n')
    fprintf('No smoothing performed.\n');
    fprintf('To smooth, try decreasing deltamin search value\n')
end
if delta+2*tol >= deltamax
     fprintf('Warning: delta value used is delta max.\n')
     fprintf('No smoothing performed.\n');
     fprintf('To smooth, try increasing deltamax search value\n')
end
fprintf('Number of iterations: %4.0f\n',counter)
fprintf('GCV functional: %12.5e\n',gcv)
% ------------------------------------------------

if delta>=deltamax
    molda=da;
end
if delta<=deltamin
    molda=da;
end
function [iwtmax,wt]=mol(delta,n)
```

```
%
% This function is called by molgcv1.m
%
% INPUT:
% Radius of mollification, delta.
% Number of points minus one, n.
%
% OUTPUT:
% Length (in step size units) of numerical
% "support" of Gaussian kernel, iwtmax.
% Gaussian kernel, wt.

a=1/(n*delta*sqrt(pi));
m=round(3*delta*n)+3;
iwtmax=2*m;
x=-m/n+1/n:1/n:m/n;
wt=a*exp(-x.^2/(delta*delta));
%plot(x,wt)
```

```
function[molda,gcv]=mol1(iwtmax,n,wt,delta,da)
%
% This function is called by molgcv1.m.
%
% INPUT:
% Length (in step size units) of numerical
% support of Gaussian kernel, iwtmax.
% Number of points minus one, n.
% Gaussian kernel, wt.
% Radius of mollification, delta.
% Data vector to be mollified, da.
%
% OUTPUT:
% Generalized Cross Validation functional value, gcv.
% Mollified data vector, molda.
%
% Calculation of the constants ca and caa.
sn1=0;
sn2=0;
sd=0;
itmin=iwtmax/2;
for i=1:iwtmax-1
    dr(i)=da(2+n-i);
end
wt(iwtmax)=0; %added by ARA
for i=1:itmin
    sk=0;
    for k=itmin+i:iwtmax
        sk=sk+wt(k);
    end
    sj1=0;
    sj2=0;
    for j=1:itmin+i-1
        sj1=sj1+wt(j)*da(itmin+i-j);
        sj2=sj2+wt(j)*dr(itmin+i-j);
    end
    sn1=sn1+(da(i)-sj1)*sk;
    sn2=sn2+(dr(i)-sj2)*sk;
    sd=sd+sk*sk;
end
ca=sn1/sd;
caa=sn2/sd;
for i=1:itmin-1
    one(i)=ca;
    three(i)=caa;
end
big=[one,da,three];
wtt=wt(1:(iwtmax-1));
c=conv(big,wtt);
for i=1:n+1
    molda(i)=c(iwtmax-2+i);
end
gcv=2*((da-molda)*(da-molda)')/iwtmax;
```

Graphs and Visualizations

```
function []=strainplots(XY,xs,strain,ppi)
%function []=strainplots(XY,xs,strain,ppi)
%Create a 2 by 1 plot, with the top being the dowel deflection and the
%bottom being the dowel strain
%Input:
%       XY = n by 2 matrix of x and y deflection values
%       xs = x values which correspond to strain values
%       strain = strain values from getstrain.m
%       ppi = pixels per inch
%
%Notes:
%   If ppi is not in pixels per inch, then the algorithm will need to
be
%   changed to produce charts with units of mm.

%Andrew R. Adams, 2009
%ara149@psu.edu

D=1/ppi*25.4;
x=XY(:,1);y=XY(:,2);
x=x-x(1);
y=y-y(1);
[x,y]=rot(x,y);
figure();
subplot(2,1,1);
plot(x.*D,y.*D);
title('Deflection','fontsize',16);
xlabel('Length (mm)','fontsize',12);
ylabel('Length (mm)','fontsize',12);
subplot(2,1,2);
plot(xs,strain);
title('Strain','fontsize',16);
xlabel('Length (mm)','fontsize',12);
ylabel('Microstrain','fontsize',12);
```

```
function []=strainimage(I,XY,xm,strain,ppi)
%function []=strainimage(I,XY,strain,ppi);
%Shows the original x-ray image with the dowel overlaid with colors
%corresponding to the strain values at the extreme fiber.
%INPUT:
%   I = grayscale image
%   XY = XY data from top and bottom edge of dowel (XY found from
%        getpointsconnection.m procedure)
%   xm = x values which correspond to values of strain in the vector
%        "strain"
%   ppi = pixels per inch (or other unit). Used to convert "xm" values
back
%         to pixels
%OUTPUT:
%   Program displays the image with the strains at the extreme fiber
%   overlaid

%Notes:
%Line 92 has been commented. Uncommenting it will use the complement of
the
%x-ray image for the background. This can in cases yield a better
overall
%image.

%This program requires sc.m, a function which can be used for
displaying
%one image with two or more different color maps.  It can be found on
the
%Matlab File Exchange.

%The colormaps used are "gray" for the background and "jet" for the
strain
%information.  These can be changed to suit different purposes. See
line
%100.

%The value "c" below represents the amount of image to size down to (in
%percent).  This is necessary due to the extreme size of the
%x-ray images, but can be changed to any value from 1-100.

%Andrew R. Adams, 2009
%ara149@psu.edu

c=10;
sz=1/c;

D=1/ppi*25.4;
xm2=xm./D;
%separate top and bottom data from XY
XYt=zeros(size(XY,1)/2,2);
XYb=zeros(size(XYt));
v=1:2:size(XY,1);
for i=1:size(v,2)
    XYt(i,:)=XY(v(i),:);
```

```
    XYb(i,:)=XY(v(i)+1,:);
end
%get "strain" and x-y data to be same size (using diffL function)
strain2=strain;
xt=XYt(:,1);yt=XYt(:,2);
xmnt=xm2+xt(1);
[nxt,nxmt]=diffL(xt,xmnt);
[nyt,strain2]=diffL(yt,strain2);
nxmt=round(nxmt);
XYt2(:,1)=nxmt;XYt2(:,2)=nyt;
xb=XYb(:,1);yb=XYb(:,2);
xmnb=xm2+xb(1);
[nxb,nxmb]=diffL(xb,xmnb);
[nyb,strain2]=diffL(yb,strain2);
nxmb=round(nxmb);
XYb2(:,1)=nxmb;XYb2(:,2)=nyb;

%remove data points in x-y and strain accoring to
ind=1:length(XYt2(:,1));
XYt2=XYt2(mod(ind,c)==0,:);
XYt2=round(XYt2./c);
ind=1:length(XYb2(:,1));
XYb2=XYb2(mod(ind,c)==0,:);
XYb2=round(XYb2./c);
ind=1:length(xm2);
xmn2=xm2(mod(ind,c)==0);
xmn2=round(xmn2./c);
ind=1:length(strain);
strain2=strain2(mod(ind,c)==0);
nxmb=XYb2(:,1);nyb=XYb2(:,2);
nxmt=XYt2(:,1);nyt=XYt2(:,2);

v=zeros(length(nxmt)+1,1);
h=v;
for i=1:length(nxmt)
    %stopbar=progressbar(i/length(nxmt));
    v(i+1)=floor((nyb(i)-nyt(i)))+1;
    h(i+1)=sum(v(1:i+1));
    ny(h(i)+1:h(i+1))=nyt(i):nyb(i);
    nx(h(i)+1:h(i+1))=nxmt(i);
    nstrain(h(i)+1:h(i+1))=strain2(i);
end

II=imresize(I,sz);
II=imadjust(II);
%II=imcomplement(II);
III=-double(II)+min(nstrain);

for i=1:length(ny)
    %stopbar=progressbar(i/length(ny));
    III(ny(i),nx(i))=nstrain(i);
end
figure();sc(III,[min(nstrain)
max(nstrain)],'jet',sc(III,'gray'),III<min(nstrain));
colorbar('Location','SouthOutside');
```

```
function []=strainimage1(I,XY,xm,strain,ppi)
%function []=strainimage1(I,XY,strain,ppi);
%Shows the original x-ray image with the dowel overlayed with colors
%corresponding to the absolute values of strain the extreme fiber.
%INPUT:
%   I = grayscale image
%   XY = XY data from top and bottom edge of dowel (XY found from
%        getpointsconnection.m procedure
%   xm = x values which correspond to values of strain in the vector
%        "strain"
%   ppi = pixels per inch (or other unit). Used to convert "xm" values
back
%         to pixels
%OUTPUT:
%   Program displays the image with the strains at the extreme fiber
%   overlaid

%Notes:
%Line 93 has been commented. Uncommenting it will use the complement of
the
%x-ray image for the background. This can in cases yield a better
overall
%image.

%This program requires sc.m, a function for displaying one image with
two
%different color maps.  It can be found on the Matlab File Exchange.

%The colormaps used are "gray" for the background and "jet" for the
strain
%information.  These can be changed to suit different purposes. See
line
%100.

%The value "c" below represents the amount of image to size down to (in
%percent).  This is necessary due to the extreme size of the
%x-ray images, but can be changed to any value from 1-100.

%Andrew R. Adams, 2009
%ara149@psu.edu

c=10;
sz=1/c;

D=1/ppi*25.4;
xm2=xm./D;
%separate top and bottom data from XY
XYt=zeros(size(XY,1)/2,2);
XYb=zeros(size(XYt));
v=1:2:size(XY,1);
for i=1:size(v,2)
    XYt(i,:)=XY(v(i),:);
    XYb(i,:)=XY(v(i)+1,:);
end
%get "strain" and x-y data to be same size (using diffL function)
```

```
strain2=strain;
xt=XYt(:,1);yt=XYt(:,2);
xmnt=xm2+xt(1);
[nxt,nxmt]=diffL(xt,xmnt);
[nyt,strain2]=diffL(yt,strain2);
nxmt=round(nxmt);
XYt2(:,1)=nxmt;XYt2(:,2)=nyt;
xb=XYb(:,1);yb=XYb(:,2);
xmnb=xm2+xb(1);
[nxb,nxmb]=diffL(xb,xmnb);
[nyb,strain2]=diffL(yb,strain2);
nxmb=round(nxmb);
XYb2(:,1)=nxmb;XYb2(:,2)=nyb;

%remove data points in x-y and strain accoring to
ind=1:length(XYt2(:,1));
XYt2=XYt2(mod(ind,c)==0,:);
XYt2=round(XYt2./c);
ind=1:length(XYb2(:,1));
XYb2=XYb2(mod(ind,c)==0,:);
XYb2=round(XYb2./c);
ind=1:length(xm2);
xmn2=xm2(mod(ind,c)==0);
xmn2=round(xmn2./c);
ind=1:length(strain);
strain2=strain2(mod(ind,c)==0);
nxmb=XYb2(:,1);nyb=XYb2(:,2);
nxmt=XYt2(:,1);nyt=XYt2(:,2);

v=zeros(length(nxmt)+1,1);
h=v;
for i=1:length(nxmt)
    %stopbar=progressbar(i/length(nxmt));
    v(i+1)=floor((nyb(i)-nyt(i)))+1;
    h(i+1)=sum(v(1:i+1));
    ny(h(i)+1:h(i+1))=nyt(i):nyb(i);
    nx(h(i)+1:h(i+1))=nxmt(i);
    nstrain(h(i)+1:h(i+1))=strain2(i);
end


II=imresize(I,sz);
II=imadjust(II);
%II=imcomplement(II);
III=-double(II);

for i=1:length(ny)
    %stopbar=progressbar(i/length(ny));
    III(ny(i),nx(i))=abs(nstrain(i));
end
figure();sc(III,[0 max(abs(nstrain))],'jet',sc(III,'gray'),III<0);
colorbar;
```

```
function []=strainimage2(I,XY,xm,strain,ppi)
%function []=strainimage2(I,XY,strain,ppi);
%Shows the original x-ray image with the dowel overlayed with colors
%corresponding to the strain values, which vary from the calculated
value
%linearly to zero at the neutral axis (located at mid-depth for a
symmetric
%cross-section) and then to the opposite of the calculated value.

%INPUT:
%   I = grayscale image
%   XY = XY data from top and bottom edge of dowel (XY found from
%        getpointsconnection.m procedure
%   xm = x values which correspond to values of strain in the vector
%        "strain"
%   ppi = pixels per inch (or other unit). Used to convert "xm" values
back
%         to pixels
%OUTPUT:
%   Program displays the image with the strains at the extreme fiber
%   overlaid

%Notes:
%Line 102 has been commented. Uncommenting it will use the complement
of
%thex-ray image for the background. This can in cases yield a better
%overall image.

%This program requires sc.m, a function which can be used for
displaying
%one image with two different color maps.  It can be found on the
Matlab
%File Exchange.

%Compression strain is shown as negative values, while tension strain
is
%shown at positive values. To reverse this, simply change line 88 to
have
%"linspace(-1,1,v(i+1)" instead of "linspace(1,-1,v(i+1)"

%The colormaps used are "gray" for the background and "jet" for the
strain
%information.  These can be changed to suit different purposes. See
line
%108.

%The value "c" below represents the amount of image to size down to (in
%percent).  This is necessary due to the extreme size of the
%x-ray images, but can be changed to any value from 1-100.

%Andrew R. Adams, 2009
%ara149@psu.edu

c=10;
sz=1/c;
```

```
D=1/ppi*25.4;
xm2=xm./D;
%separate top and bottom data from XY
XYt=zeros(size(XY,1)/2,2);
XYb=zeros(size(XYt));
v=1:2:size(XY,1);
for i=1:size(v,2)
    XYt(i,:)=XY(v(i),:);
    XYb(i,:)=XY(v(i)+1,:);
end
%get "strain" and x-y data to be same size (using diffL function)
strain2=strain;
xt=XYt(:,1);yt=XYt(:,2);
xmnt=xm2+xt(1);
[nxt,nxmt]=diffL(xt,xmnt);
[nyt,strain2]=diffL(yt,strain2);
nxmt=round(nxmt);
XYt2(:,1)=nxmt;XYt2(:,2)=nyt;
xb=XYb(:,1);yb=XYb(:,2);
xmnb=xm2+xb(1);
[nxb,nxmb]=diffL(xb,xmnb);
[nyb,strain2]=diffL(yb,strain2);
nxmb=round(nxmb);
XYb2(:,1)=nxmb;XYb2(:,2)=nyb;

%remove data points in x-y and strain accoring to
ind=1:length(XYt2(:,1));
XYt2=XYt2(mod(ind,c)==0,:);
XYt2=round(XYt2./c);
ind=1:length(XYb2(:,1));
XYb2=XYb2(mod(ind,c)==0,:);
XYb2=round(XYb2./c);
ind=1:length(xm2);
xmn2=xm2(mod(ind,c)==0);
xmn2=round(xmn2./c);
ind=1:length(strain);
strain2=strain2(mod(ind,c)==0);
nxmb=XYb2(:,1);nyb=XYb2(:,2);
nxmt=XYt2(:,1);nyt=XYt2(:,2);


v=zeros(length(nxmt)+1,1);
h=v;
for i=1:length(nxmt)
    %stopbar=progressbar(i/length(nxmt));
    v(i+1)=floor((nyb(i)-nyt(i)))+1;
    h(i+1)=sum(v(1:i+1));
    ny(h(i)+1:h(i+1))=nyt(i):nyb(i);
    nx(h(i)+1:h(i+1))=nxmt(i);
    nstrain(h(i)+1:h(i+1))=strain2(i).*linspace(1,-1,v(i+1));
end

II=imresize(I,sz);
II=imadjust(II);
%II=imcomplement(II);
```

```
III=-double(II)+min(nstrain);

for i=1:length(ny)
    %stopbar=progressbar(i/length(ny));
    III(ny(i),nx(i))=nstrain(i);
end
figure();sc(III,[min(nstrain)
max(nstrain)],'jet',sc(III,'gray'),III<min(nstrain));
colorbar('Location','SouthOutside');
```

```
function []=stressimage(I,XY,xm,strain,ppi,E,fy)
%function []=stressimage(I,XY,strain,ppi,E,fy);
%Shows the original x-ray image with the dowel overlayed with colors
%corresponding to the stress values, which vary from the calculated
value
%linearly to zero at the neutral axis (located at mid-depth for a
symmetric
%cross-section) and then to the opposite of the calculated value.
Values
%greater than abs(fy) are shown as black (can be changed by changing
the
%the colormap

%INPUT:
%   I = grayscale image
%   XY = XY data from top and bottom edge of dowel (XY found from
%        getpointsconnection.m procedure
%   xm = x values which correspond to values of strain in the vector
%        "strain"
%   ppi = pixels per inch (or other unit). Used to convert "xm" values
back
%         to pixels
%   E = modulus of elasticity
%   fy = yield stress (use consistent units for E and fy)
%OUTPUT:
%   Program displays the image with the streses which vary linearly
from
%   the top to bottom, with values greater than fy (or less than -fy)
%   clipped to fy.

%Notes:
%Line 104 has been commented. Uncommenting it will use the complement
of the
%x-ray image for the background. This can in cases yield a better
overall
%image.

%This program requires sc.m, a function which can be used for
displaying
%one image with two different color maps.  It can be found on the
Matlab
%File Exchange.

%Compression strain is shown as negative values, while tension strain
is
%shown at positive values. To reverse this, simply change line 88 to
have
%"linspace(-1,1,v(i+1)" instead of "linspace(1,-1,v(i+1)"

%The colormaps used are "gray" for the background and "jet" and
"colorcube"
%for the stress information.  These can be changed to suit different
%purposes. See line 115.

%The value "c" below represents the amount of image to size down to (in
%percent).  This is necessary due to the extreme size of the
```

```
%x-ray images, but can be changed to any value from 1-100.

%Andrew R. Adams, 2009
%ara149@psu.edu

c=10;
sz=1/c;

D=1/ppi*25.4;
xm2=xm./D;
%separate top and bottom data from XY
XYt=zeros(size(XY,1)/2,2);
XYb=zeros(size(XYt));
v=1:2:size(XY,1);
for i=1:size(v,2)
    XYt(i,:)=XY(v(i),:);
    XYb(i,:)=XY(v(i)+1,:);
end
%get "strain" and x-y data to be same size (using diffL function)
strain2=strain;
xt=XYt(:,1);yt=XYt(:,2);
xmnt=xm2+xt(1);
[nxt,nxmt]=diffL(xt,xmnt);
[nyt,strain2]=diffL(yt,strain2);
nxmt=round(nxmt);
XYt2(:,1)=nxmt;XYt2(:,2)=nyt;
xb=XYb(:,1);yb=XYb(:,2);
xmnb=xm2+xb(1);
[nxb,nxmb]=diffL(xb,xmnb);
[nyb,strain2]=diffL(yb,strain2);
nxmb=round(nxmb);
XYb2(:,1)=nxmb;XYb2(:,2)=nyb;

%remove data points in x-y and strain accoring to
ind=1:length(XYt2(:,1));
XYt2=XYt2(mod(ind,c)==0,:);
XYt2=round(XYt2./c);
ind=1:length(XYb2(:,1));
XYb2=XYb2(mod(ind,c)==0,:);
XYb2=round(XYb2./c);
ind=1:length(xm2);
xmn2=xm2(mod(ind,c)==0);
xmn2=round(xmn2./c);
ind=1:length(strain);
strain2=strain2(mod(ind,c)==0);
nxmb=XYb2(:,1);nyb=XYb2(:,2);
nxmt=XYt2(:,1);nyt=XYt2(:,2);


v=zeros(length(nxmt)+1,1);
h=v;
for i=1:length(nxmt)
    %stopbar=progressbar(i/length(nxmt));
    v(i+1)=floor((nyb(i)-nyt(i)))+1;
    h(i+1)=sum(v(1:i+1));
    ny(h(i)+1:h(i+1))=nyt(i):nyb(i);
```

```
    nx(h(i)+1:h(i+1))=nxmt(i);
    nstress(h(i)+1:h(i+1))=(strain2(i).*linspace(1,-1,v(i+1)))*10^-6*E;
end
nstress(nstress>fy)=fy+10;
nstress(nstress<-fy)=fy+10;

II=imresize(I,sz);
II=imadjust(II);
%II=imcomplement(II);
III=-double(II)+min(nstress);

for i=1:length(ny)
    %stopbar=progressbar(i/length(ny));
    III(ny(i),nx(i))=nstress(i);
end
figure();sc(III,[-fy fy],'jet',sc(III,'gray'),III<-fy,sc(III,'-
colorcube'),III>fy);
colorbar('Location','SouthOutside');
```

```
function []=meshstrainplot(load,xm,strain)
%function []=meshstrainplot(load,xm,strain)
%Creates a mesh plot of load, length, and strain.  Program performs
%operations so that even if the input data is not the appropriate size,
%corrections will be made (using diffL function) and the data will be
%plotted
%INPUT:
%   load = n by 1 vector of load values, where n is the number of
length
%          and strain data sets (xm and strain, respectively) in each
of
%          the cell structures
%   xm = cell containing "n" sets of xm values which correspond to
strain
%        values in their cells
%   strain = cell containing "n" sets of strain values
%OUTPUT:
%   Mesh plot

%Example of how to create cells with data
% xm = {xm0 xm03 xm14}; %Note the use of {} and not ()
% strain = {strain0 strain03 strain14};

%This allows the xm and strain data to be combined into a single
variable
%even though they are of different lengths

%Andrew R. Adams, 2009
%ara149@psu.edu

lengthstrain=length(strain);
L=zeros(length(xm));
for i=1:length(xm)
    L(i)=length(xm{i});
    [minL,ind]=min(L);
end
xmm=xm{ind};
vals=1:length(xm);vals(vals==ind)=[];

nstrain=cell(1,lengthstrain);
nstrain(1,ind)=strain(1,ind);
for i=1:length(vals);
    N=strain{ind};
    n1=strain{vals(i)};
    [nstrain{vals(i)},dum]=diffL(n1,N);
end

S=zeros(length(load),length(xmm));
for i=1:length(nstrain)
    S(i,:)=nstrain{i};
end
mesh(xmm,load,S)
xlabel('Length (mm)');ylabel('Load (kN)');zlabel('Microstrain')
set(gca,'YDir','reverse')
```

Required Helper Functions

```
function [nx ny]=rot(x,y,theta)
%[nx ny]=rot(x,y,theta)
%Rotate data by angle theta
%If theta is not provided, data is rotated so that the end points have
%equal y values
%INPUT:
%   x = x data values
%   y = y data values
%   theta = angle of rotation (radians)
%OUTPUT:
%   nx = rotated x values
%   ny = rotated y values

%Andrew R. Adams, 2009
%ara149@psu.edu


if ~exist('theta','var') || isempty(theta) || ...
      numel(theta) ~= 1
    g=x(1);
    h=y(1);
    t=x(end);
    u=y(end);
    theta=-(u-h)/(t-g);
end
nx=x*cos(theta)-y*sin(theta);
ny=x*sin(theta)+y*cos(theta);
```

```
function [xx,firdy]=firstderiv(x,y,nn)
%[xx,secdy]=firstderiv(x,y,nn)
%Compute first derivative of discrete data and crops x input to same be
%the size as secdy
%INPUT:
%   x = x data values
%   y = y data values
%   nn = size of finite difference weights filter to calculate first
%        derivative (must be an odd positive integer)
%OUTPUT:
%   xx = x values which correspond to calculated second derivative
values
%   firdy = first derivative values of y with respect to x
%
%Notes:
%nn must be and odd postive integer
%x values must be equally spaced with unit intervals

%requires function ufdwt.m from the Matlab File Exchange

%Andrew R. Adams, 2009
%ara149@psu.edu

filt=ufdwt(1,nn,1);
filt=filt(ceil(nn/2),:);
h=x(2)-x(1);
firdy=conv(y,filt)./h^2;
if size(firdy,1)<size(firdy,2)
        firdy=firdy';
end
g=length(filt);
firdy(1:g)=[];
firdy(end-g+1:end)=[];
xx=x;
xx(1:ceil(g/2))=[];
xx(end-floor(g/2):end)=[];
```

```
function [xx,secdy]=secderiv(x,y,nn)
%[xx,secdy]=secderiv(x,y,nn)
%Compute second derivative of discrete data and crops x input to same
be
%the size as secdy
%INPUT:
%   x = x data values
%   y = y data values
%   nn = size of finite difference weights filter to calculate second
%        derivative (must be an odd positive integer)
%OUTPUT:
%   xx = x values which correspond to calculated second derivative
values
%   secd = second derivative values of y with respect to x
%
%Notes:
%nn must be an odd positive integer (default nn=3)
%x values must be equally spaced with unit intervals

%The function ufdwt.m from the Matlab File Exchange is required to run
this
%code

%Andrew R. Adams, 2009
%ara149@psu.edu

filt=ufdwt(1,nn,2);
filt=filt(ceil(nn/2),:);
h=x(2)-x(1);
secdy=conv(y,filt)./h^2;
if size(secdy,1)<size(secdy,2)
    secdy=secdy';
end
g=length(filt);
secdy(1:g)=[];
secdy(end-g+1:end)=[];
xx=x;
xx(1:ceil(g/2))=[];
xx(end-floor(g/2):end)=[];
```

```
function [xs,ys] = avgfilt(x,y,s,n)
% [xs,ys]=avgfilt(x,y,s,n);
%Filters data with an averaging filter of the form [1/s 1/s...1/s]
where
% "s" is equal to the number of terms in the filter
%INPUT:
%   x = x data values
%   y = y data values
%   s = size of filter (values of coefficients equals 1/s). Must be an
odd
%       number
%   n = number of passes of filter through data
%OUTPUT:
%   xs = x values to correspond with filtered y values (ys)
%   ys = filtered y values
%
%Notes
% "s" must be an odd positive integer
% Generally it is better to use a smaller filter and more passes than a
large % filter and one pass (n=1)
%
%Andrew R. Adams, 2009
%ara149@psu.edu

avgx=x;
avgy=y;
h=ones(s,1)./s;
for i=1:n
    avgy=conv(h,avgy);
    avgy(1:floor(s/2))=[];
    avgy(end-floor(s/2)+1:end)=[];
    avgx(1:floor(s/2))=[];
    avgy(1:floor(s/2))=[];
    avgx(end-floor(s/2)+1:end)=[];
    avgy(end-floor(s/2)+1:end)=[];
end

ys=avgy;
xs=avgx;
```

```
function [X] = avgdiffL(X1,X2)
%[X] = avgdiffL(X1,X2)
%Take two vectors of differing length, crop them to be the same size
and
%average together
%INPUT:
%   X1 = n by 1 vector
%   X2 = nn by 2 vector
%OUTPUT:
%   X = cropped and averaged vector
%
%Notes:
%   Vectors can be input in any order, regardless of size

%Andrew R. Adams, 2009
%ara149@psu.edu

if length(X2)==length(X1);
    X=(X1+X2)./2;
else
    if length(X1)>length(X2)
        x=X1;
        y=X2;
    end
    if length(X2)>length(X1)
            x=X2;
            y=X1;
    end
diffL=length(x)-length(y);

if mod(diffL,2)==0;
    x(1:diffL/2)=[];
    x(end-diffL/2+1:end)=[];
else x(1:floor(diffL/2))=[];
    x(end-floor(diffL/2):end)=[];
end
X=(x+y)./2;
end
```

```
function [XX1,XX2] = diffL(X1,X2)
%[XX1,XX2] = diffL(X1,X2)
%take two vectors of differing length and crop them to be the same size
by
%removing an equal number of points from each end of the longer vector
%INPUT:
%       X1 = n by 1 vector number 1
%       X2 = nn by 1 vector number 2
%
%Notes:
%   Vectors can be input in any order, regardless of size

%Andrew R. Adams, 2009
%ara149@psu.edu

if length(X2)==length(X1);
    XX1=X1;XX2=X2;
else
    if length(X1)>length(X2)
        x=X1;
        y=X2;
    end
    if length(X2)>length(X1)
        x=X2;
        y=X1;
    end
    diffL=length(x)-length(y);

    if mod(diffL,2)==0;
        x(1:diffL/2)=[];
        x(end-diffL/2+1:end)=[];
    else x(1:floor(diffL/2))=[];
        x(end-floor(diffL/2):end)=[];
    end
    if length(X1)>length(X2)
        XX1=x;
        XX2=y;
    end
    if length(X2)>length(X1)
        XX1=y;
        XX2=x;
    end
end
```

```
function [xy]=cropxy(avgXY)
%function [xy]=cropxy(avgXY)

%Input:
%       avgXY = n by 2 matrix of x and y values
%
%Output:
%       xy = nn by 2 matrix of x and y values cropped between the
points
%            selected
%
%Procedure: After each step press "Enter" to move to the next step
%1. When plot appears zoom in on region where the data should begin
%2. Select first data point which should be included in the output data
set
%3. Zoom in on the region where the data set should end
%4. Select last data point which should be included in the output data
set


%Andrew R. Adams, 2009
%ara149@psu.edu

plot(avgXY(:,1),avgXY(:,2));
pause;
[g,h]=getpts;
pause;
[i,j]=getpts;
close all;
g=round(g);
i=round(i);
avgXY(avgXY(:,1)<g,:)=[];
avgXY(avgXY(:,1)>i,:)=[];
xy=avgXY;
```

```
function [xx,yy]=nodups(x,y)
%[xx,yy]=nodups(x,y)
%Finds and removes points with the same x value
%The first y value associated with the x value is kept
%INPUT:
%       x = x values
%       y = y values
%OUTPUT:
%       xx = x values with multiple x values removed
%       yy = y values with the y values corresponding to multiple x
values
%            removed

%Andrew R. Adams, 2009
%ara149@psu.edu

xx=x;yy=y;
for i=xx(1):xx(end)
    d=find(xx==i);
    if length(d)>1
        d=d(2:end);
        xx(d)=[];
        yy(d)=[];
    end
end
```

```
function [X]=subtractzero(X1,z)
%[X]=subtractzero(X1,z)
%Subtracts zeroed strain data from other strain data by first using
"diffL"
%function which crops the two vectors to be the same size.

%INPUT:
%   X1 = strain data
%   z = strain data from image when strain gage was zeroed
%OUTPUT:
%   X = zeroed strain data

%Andrew R. Adams, 2009
%ara149@psu.edu

diffL=abs(length(X1)-length(z));
if size(z,2)<size(z,1)
    z=z';
end
if size(X1,2)<size(X1,1)
    X1=X1';
end
if length(X1)==length(z);
    X=X1-z;
else
    if length(X1)>length(z)
        x=X1;
        if mod(diffL,2)==0;
            x(1:diffL/2)=[];
            x(end-diffL/2+1:end)=[];
        else x(1:floor(diffL/2))=[];
            x(end-floor(diffL/2):end)=[];
        end
        X=x-z;
    end
    if length(z)>length(X1)
        x=z;
        if mod(diffL,2)==0;
            x(1:diffL/2)=[];
            x(end-diffL/2+1:end)=[];
        else x(1:floor(diffL/2))=[];
            x(end-floor(diffL/2):end)=[];
        end
        X=X1-x;
    end
end
```

```
 function [p]=ppi(XYt,XYb,d)
%[p]=ppi(XYt,XYb,d)
%Finds the pixels per inch in the image by finding the distance between
the
%top and bottom edges and using the known depth
%INPUT:
%   XYt = n by 2 matrix of x and y values for the top edge
%   XYb = n by 2 matrix of x and y values for the bottom edge
%   d = depth of bar
%OUTPUT:
%   p = pixels per inch (or other unit depending on units of "d" input)
%
%Notes:
%   XYt and XYb do not be be the same length

%Andrew R. Adams, 2009
%ara149@psu.edu

if length(XYt)==length(XYb)
    diff=abs(XYt(:,2)-XYb(:,2));
end
if length(XYt)>length(XYb)
    diff=zeros(1,length(XYb));
    for i=1:length(XYb)
        for j=1:length(XYt)
            if XYb(i,1)==XYt(j,1)
                diff(i)=abs(XYb(i,2)-XYt(j,2));
            end
        end
    end
end

if length(XYt)<length(XYb)
    diff=zeros(1,length(XYt));
   for i=1:length(XYt)
        for j=1:length(XYb)
            if XYt(i,1)==XYb(j,1)
                diff(i)=abs(XYt(i,2)-XYb(j,2));
            end
        end
   end
end

p=mean(diff)*1/d;
```

Other required functions from the Matlab File Exchange

"Local min, max, nearest neighbour" by Sergei Koptenko
http://mathworks.com/matlabcentral/fileexchange/31700

"Finite Difference Weights" by Greg von Winckel
http://www.mathworks.com/matlabcentral/fileexchange/5269

"SC – powerful image rendering" by Oliver Woodford
http://www.mathworks.com/matlabcentral/fileexchange/16233

"Keep" by David Yang
http://www.mathworks.com/matlabcentral/fileexchange/181