Guidelines for Implementing MODEM:
An Open-Source, MATLAB-Based Digital Image Correlation Software

By
Ryan Schwartz

Submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Architectural Engineering at California Polytechnic State University – San Luis Obispo, 2020.

San Luis Obispo, California
10/2020

Senior Project Advisor:
Anahid A. Behrouzi, Ph.D.

ACKNOWLEDGEMENTS

I would start by thanking everyone who has helped me during the course of my research and, through their assistance and support, have helped me to create the document you see before you today:

Thank you to Dr. Anahid Behrouzi for always being a sounding board for my ideas and helping give me direction. The realm of research is new to me and your constant support and guiding hand were invaluable in my undertaking of this project. Without your words of wisdom this project would not be where it is today. Your gracious help and work ethic will always serve as a source of inspiration to me. Thank you for all that you have done for me. I will never forget it.

Thank you to Nicole Buck for being my partner-in-crime for a few months this year. When you joined the DIC team the project really took off. Your continually positive attitude and your incredible personality always put a smile on my face. With your help this research finally reached its potential and I could not be more grateful for your help throughout this entire process. For your insight, creative thinking, and countless hours in lab with me I will always remember and appreciate. Thank you.

Thank you to Dr. Lucas Hogan for introducing me to this new world of research and for helping me start this journey into digital image correlation. It was your vision for your lab in Auckland that brought me to a country halfway around the world and started a summer adventure that I will never forget. Thank you for your continual support over the years and for all of your ideas and insight along the way.

Thank you to Dr. Tom Allen, Dr. Rick Henry, and Hafez Heravi for showing and discussing your research with me. It was with your help that I was able to truly begin this research and be able to unravel its mysteries and intricacies. Thank you for answering my questions, posing ones I had never even though of before, and bringing a whole new world into light.

Thank you to Rory de Sevilla and Tommy Sidebottom for being my traveling companions during our two months in New Zealand. It was with your friendship and support that I was able to make it through those two months across the Pacific. I will always remember and cherish our time together and am excited to finally able to share with you the work that I have been working on since we met.

Thank you to Rory de Sevilla and Jerry Leong for allowing me to work their full-scale wall test. This experience gave me a true idea as to what real structural testing at Cal Poly was like and how, with my researchers, I could help future Cal Poly students further their experimental capabilities.

Thank you to ZGF Building Performance Evaluation Studies Fund, the College of Architecture and Environmental Design, Architectural Engineering Department for granting me the funding to conduct this research. This grant gave me the ability to purchase equipment and assemble a DIC system for Cal Poly.

Thank you to Marga Lamoreaux, Gabriele Granello, the researchers at University of Canterbury's Structural Engineering Laboratory, Brandy Alger, and Holmes Consulting and for the amazing experiences I had when I went to Christchurch during my internship in New Zealand. Learning how earthquakes not only effect the physical infrastructure of a place, but also its social and cultural matrix, was eye-opening. Then, seeing how a community can come together to rebuild and begin to thrive again was truly a sight to behold. Thank you for your warm welcomes and your guides through the city. My trip to Christchurch is something I will fondly remember thanks to you all.

Thank you to Vince Pauschek for being the best lab tech and the best help anyone could ask for in a lab. Your endless wisdom, helpful insight, and hearty laugh were always something I looked forward to when going down into lab. Without you I would have been stumped a long time ago. Thank you for all of your help and inspiration. It meant the world.

Thank you to my friends Tracy Doan, Anisha Datta, and Kiana Underwood for being my rocks during this entire process. It was your friendship and understanding that got me though this process and it was your continual and unending support that got me where I am today. I cannot put into words how happy I have been to have you all in my life and to have been there for me when I had a problem to work through or simply needed someone with which to vent. For your friendship I will be forever grateful. Thank you all from the bottom of my heart.

Thank you to my family for being a constant reservoir of support and a never-ending parade of joy and happiness. You all are like a chaotic group of fun and entertainment and without you this research would have never been completed. For every time you have listened to my long and boring explanations and for every time you give me perspective and clarity, I will always be grateful. You all are the light of my life and I cannot that you enough for everything that you have done for me. From the bottom of my heart I want to say thank you. I will love you always.

ABSTRACT

MODEM is an open-source, MATLAB-based digital image correlation (DIC) program that was developed at the University of Auckland for small-scale testing of flexible materials. Structural engineering researchers at both the University of Auckland and Cal Poly – San Luis Obispo wanted to expand the uses of the program to study the seismic response of large-scale test specimens. This guide document describes how to implement DIC using MODEM, including the hardware and software needed to run an experiment as well as data collection and post-processing procedures for the program. Additionally, this document includes a case study focusing on a DIC test program consisting of several aluminum coupons subjected to pure tension. A summary of MODEM's output from one of these tests informs future users of the benefits and pitfalls that can occur while running DIC experiments and prepares them to use this program in their experiments. Furthermore, this work demonstrates that researchers can accurately quantify the full-field deformation of structures at a localized scale and utilize this data to corroborate traditional instrumentation like strain gages and linear potentiometers as well as to calibrate computational finite element models.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

TABLE OF EQUATIONS

LIST OF TERMS

DIC TERMS

DAQ – an acronym for data acquisition system.

DIC – an acronym for “digital image correlation”. Digital image correlation is a non-contact optical imaging technique that allows users to track the deformation of a surface and, from that tracking, produce strain and displacement fields.

Features (Coords) – unique portions of a speckle pattern identified within a region of interest by DIC software that are tracked throughout an experiment to help determine the strain and displacements within the region of interest. MODEM refers to these points as “coords” in its output.

Nodes – intersection of a user-defined grid placed above a region of interest. It is at these points the DIC software computes the strain and displacements of the specimen based off the deformation of the features.

MODEM (or TRIDENT) – an open-source, MATLAB-based DIC software developed at the University of Auckland for mechanical and materials engineering purposes.

Region of Interest (Tracking Region) – a region of a specimen that a user wishes to analyze with a DIC software.

Speckle Pattern – a high-contrast, black and white pattern applied to a specimen that a DIC system tracks throughout an experiment to determine the specimen’s deformation and compute strain and displacement fields.

Subsets – regions placed around nodes. The features located within these are utilized to determine the strain and displacement of the node they surround.

CAMERA/IMAGE SETTINGS TERMS

Bit Depth – refers to the amount of color information stored in an image. Specifically, the bit depth refers to how many bits are used to represent the RGB color for a given pixel in an image.

f-Stop - the ratio between the lens’ focal length and the diameter of the camera’s entrance pupil. It is the relationship between the distance at which the camera lens is focused and the amount of light that the camera is allowing into its sensor

ISO Speed - the image sensor’s sensitivity to light. The higher the ISO value, the more sensitive the sensor is and the more light it registers. A lower ISO value prevents overexposing the sensor to light while also providing a greater contrast in the image.

JPEG – an acronym for “joint photographic experts group”. This image format compresses the data acquired from a camera’s image sensor when a photograph is taken, removing pixels the compression algorithm deems unimportant.

RAW – image formats that are exact copies of the data generated by a camera’s image sensor when a photograph is taken. This image format can come in many different file extensions, like .NEF, .RW2, or .RAW.

Shutter Speed - the amount of time the shutter of a camera stays open, allowing light to hit the camera’s sensor.

TIFF – an acronym for “tagged image file format”. This image format compresses the data acquired from a camera’s image sensor when a photograph is taken without losing any of the image data.

White Balance - the camera’s attempt to convert images into true color (the colors human eyes see) by using the white tones in the photo as a reference. The camera takes the white tones of an image and, based on the white balance setting on the camera, adds different color tones to the image.

MATLAB VARIABLES TERMS

Array – in this document, an array is a MATALB variable that contains floating point numbers in a 2D matrix where one dimension of the matrix is one.

Matrix – in this document, a matrix is a MATLAB variable that contains floating-point numbers in an 2D or 3D matrix orientation.

1 INTRODUCTION

The objective of the research described in this report is to provide researchers with guidelines for using a digital image correlation software developed at the University of Auckland known as MODEM.

MODEM is an open-source MATLAB-based digital image correlation (DIC) software developed at the University of Auckland's Materials Engineering Department. DIC is a non-contact optical measurement technique that tracks the unique visual features (speckle pattern or set of targets) on specimens in photographs taken throughout the course of an experiment. DIC software, like MODEM, enables researchers to analyze the changes in this pattern to generate strain and displacement fields. DIC has been a popular tool for mechanical and materials engineering for many decades but has been growing in use in structural engineering due to its utility and flexibility. DIC systems can be used to track deformations in real-world infrastructure, including: bridge deflection and pipework vibration [1], masonry cracking [2] and even to determine the angular velocity of massive structures like the London Eye [1].

It is for this reason that structural engineering researchers at both the University of Auckland and Cal Poly – San Luis Obispo sought out a DIC system to use in large-scale tests investigating the seismic response of various structural components. Unlike many DIC systems that are proprietary, costing tens of thousands of dollars, and prohibit modifications to fit the needs of a given experiment, MODEM was chosen since it is an open-source software. Previously known as TRIDENT, it was utilized in the University of Auckland's Materials Engineering Department as an additional instrumentation system for their experiments on flexible materials such as polymeric foam cores [3].

At the onset of the project detailed in this document, MODEM had only been previously used in experiments testing extremely flexible materials and did not have an accompanying user manual. Without which, an experimentalist had to invest months to understand the software and conduct parametric studies to determine the proper settings to use in structural engineering experimental testing.

The experimental tests described in this report, along with those conducted by Buck [4], inform the guidelines recommended for the use of MODEM.

1.1 GUIDELINE CHAPTERS

Chapter 1 includes an introduction to the document including a brief overview of its contents and a basic overview of how a DIC system operates. This chapter also includes a list of references that may be used to gain deeper understanding of different DIC topics relevant to the use of MODEM.

Chapter 2 includes lists of required and recommended computer software, hardware, and accessories needed to run MODEM, along with explanations for each item. This section can be used to generate a list of materials needed for an experiment implementing MODEM.

Chapter 3 includes various guides on how to install programs needed to run MODEM such as MATLAB, MODEM, and a camera control software known as digicamControl. For each software mentioned, a brief overview of each software is given along with a procedure on how to install it.

Chapter 4 includes recommendations for camera settings and experimental parameters used in a DIC experiment. The chapter begins by summarizing all the recommendations made for camera settings for a DIC experiment, followed explanations for each. Guidelines are also provided for experimental test setup including the data acquisition system for traditional instrumentation. Chapter 4 concludes, with supplemental materials needed for a MODEM experiment.

Chapter 5 includes user guides explaining various functions of MODEM, including speckle pattern application as well as conduct of a DIC calibration, 2D MODEM data analysis, and post-processing.

Chapter 6 includes the settings, procedures, and results for an example experiment conducted on an aluminum coupon subjected to pure tension. This experimental case study is meant to be used as a reference for researchers who want to utilize MODEM for their experiments.

The appendices include a summary of the output folders and variables created during every MODEM analysis, a list of equipment to purchase to prepare for conducting any experiment using MODEM, and checklists to be used when conducting MODEM experiments to help prevent simple, common mistakes.

1.2 PRINCIPLES OF DIC

1.2.1 Definition of a DIC System

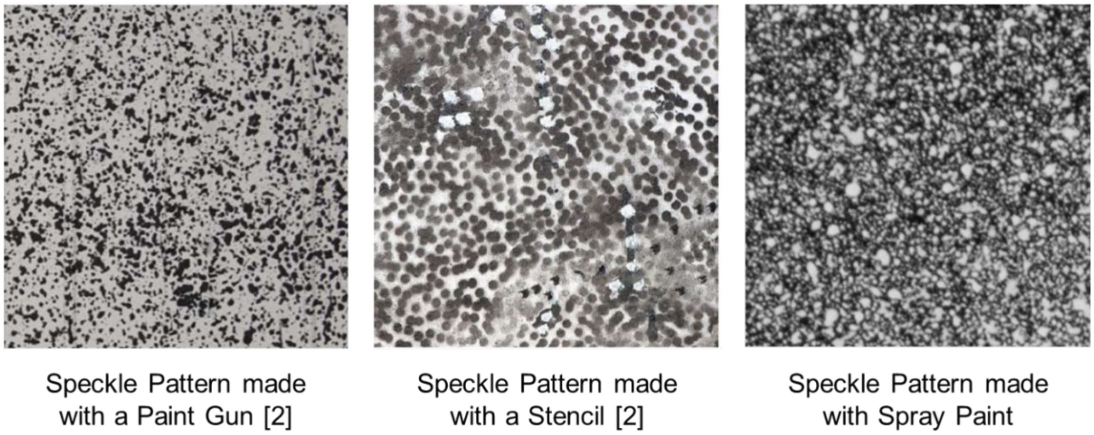
Digital Image Correlation, more commonly known an DIC, is an incredibly powerful non-contact optical imaging technique that allows users to track the deformation of practically any surface. DIC has been used in numerous ways, from tracking the movement of a beating heart [5] to the deformations of a concrete wall. This tracking ability enables these systems to generate strain and displacement fields over an entire region of interest, data that would be incredibly laborious to acquire using traditional forms of instrumentation. DIC is low-cost compared to other sensor systems as it requires only a camera, speckle pattern (or targets), and software and is relatively simply to assemble for any experiment.

1.2.2 Basics of a DIC Analysis

DIC analyses consist of three steps: specimen and experiment setup, capturing images while the specimen is deforming, and processing these images with a computer program to acquire strain and displacement fields [6]. These three basic steps will be briefly elaborated in this section and will reference other sections within this document that expand upon these topics.

1.2.2.1 SPECIMEN AND EXPERIMENT SETUP

Whenever a DIC analysis is being conducted, the first step is always creating the a unique visual pattern. A speckle pattern is a randomized, high-contrast pattern that can be applied to the test specimen [2]. This pattern can be made in various ways, from stencils to spray paint (some examples of which are shown in Figure 1.3) and it is this pattern that the DIC system tracks. Speckle patterns often cover a specimen so researchers can gain a full-field understanding of specimen behavior; alternatively, speckle patterns can be used on small, discrete targets to track specific points more precisely (see Section 5.2). More information about speckle patterns and how they are created is explained in Section 5.1.


Speckle Pattern made with a Paint Gun [2]

Speckle Pattern made with a Stencil [2]

Speckle Pattern made with Spray Paint

Figure 1.1: Examples of Speckle Patterns

After the speckle pattern is applied, the specimen is situated in the testing apparatus, and the optical system that captures the specimen's deformations is assembled. This consists of a camera, tripod, light sources, and other materials that are specified in Section 2.2. The specifications and recommendations for how the optical system should be set up are explained in Sections 4.2.1 and 4.2.3.

1.2.2.2 CONDUCTING EXPERIMENTS USING DIC AND OTHER INSTRUMENTATION

After the experiment and specimen are prepared, then the specimen can be tested. Normally, DIC is not the only form of instrumentation used in an experiment. Other forms of traditional instrumentation are used in conjunction with DIC such as strain gages and linear potentiometers. These other forms of data need to be correlated to the images taken during the experiment for them to be compared. This can be done by using the approach stated in Sections 4.2.2 and 4.2.4.

1.2.2.3 PROCESSING EXPERIMENTAL IMAGES USING DIC SOFTWARE

After the experiment is completed, the images collected need to be processed via DIC software. Although many proprietary programs exist which use similar computing techniques, in this document the program being investigated is MODEM. Once the images are input, MODEM will ask for a region of interest to be selected in the reference image, which is typically the first image taken of the undeformed specimen. This region of interest (or "tracking region") is the region the user wants to analyze. The region will be subdivided by a user defined virtual grid [6]. At the grid intersections MODEM generates strain and displacement fields. MODEM defines these points as "nodes" (explained in Section 5.7.2.2) and the process for selecting this spacing is detailed in Section 5.6.

After defining the placement of the nodes, MODEM will automatically define regions around each node know as subsets [6].

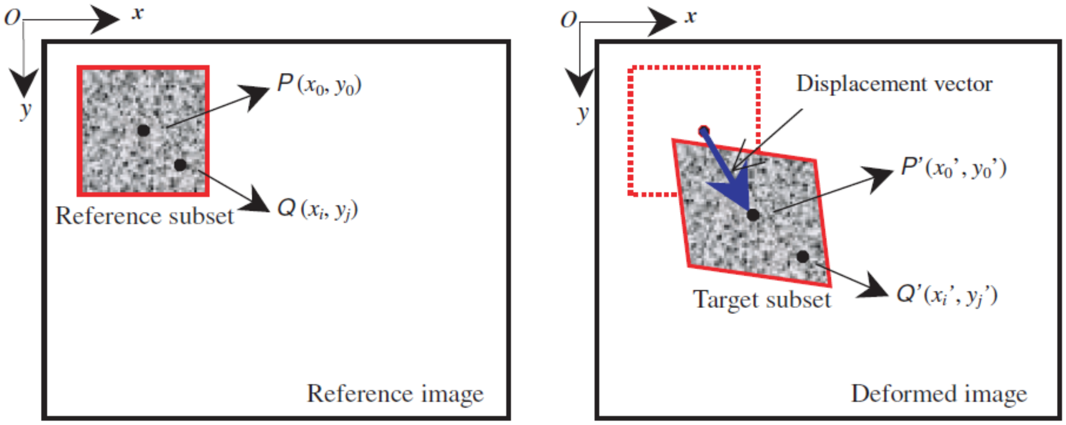


Figure 1.2: Visual Depiction of a Subset throughout an Experiment [6]

Within each subset, MODEM will select points called "coords" but most literature refers to these as features (explained in Section 5.7.2.1). These features are a 3x3 grid of pixels with a unique set of tonal values as seen in Figure 1.3. The eight pixels that surround the central pixel act as a QR code, allowing the central pixel to be identified from the rest of the image. These points are tracked in conjunction with the nodes to help determine their location throughout an experiment. The more coords MODEM identifies that can be used to define a subset, the more unique each subset becomes improving the ease of MODEM to identify and track the subsets.

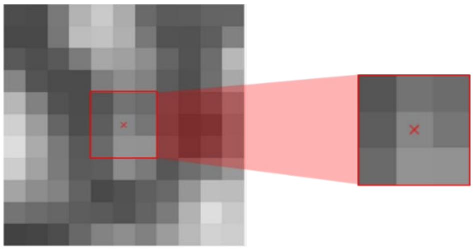


Figure 1.3: Example of a Feature

After tracking the coords and nodes throughout the experiment, their strains and displacements are determined (Sections 5.7.2.4 and 5.7.2.3 respectively). Initially displacements are quantified in pixels but, if a calibration process is executed, these values can be converted into measurements units like inches or centimeters (see Section 5.4). From here, MODEM can generate strain contour plots (Section 5.8.1) and videos (Section 5.8.2) to visualize test results. Additionally, data can be extracted from MODEM and post-processed in another software such as MATLAB or Excel. Explanations of which files are needed from MODEM and how to extract these for post-processing can be found in Sections 5.7.2 and 5.7.3.

1.2.3 Other References for DIC Explanations

Some useful articles or papers that may be helpful to a researcher interested in learning more about MODEM or DIC in general are located in Table 1.1. Citations for these documents can be found in References.

Table 1.1: Useful DIC Articles and Papers

Researcher (Date)	Year	Document Type	Document Content
R. M. Stubbing [3]	2013	Dissertation	Detailed description of the research behind and alterations made to MODEM (TRIDENT)
M. A. Sutton, J. Orteu, and H. W. Schreier [7]	2009	Textbook	Detailed description of the basic concepts, theory, and applications of DIC
N. McCormick and J. Lord [1]	2012	Journal Article	General overview of DIC and its applications in civil engineering
Y.L. Dong and B. Pan [8]	2017	Journal Article	Comprehensive review of various speckle pattern applications

A.H. Salmanpour and N. Mojsilovic [2]	2013	Conference Paper	Description of the preparing, testing, and analyzing of masonry structural walls
N. McCormick and J. Lord [9]	2010	Journal Article	General overview of digital image correlation
B. Pan, K. Qian, H. Nagel, and A. Asundi [6]	2009	Journal Article	Description of the tracking process and mathematics behind DIC

2 REQUIRED MATERIALS

To properly run and utilize MODEM, the following materials specified in Section 0 and Section 2.2 will be required for the system to function properly. The materials are listed by category.

2.1 COMPUTER SOFTWARE

The software specified in this section only deals with that required for operating MODEM. All other systems required for an experiment, such as traditional instrumentation, need to be considered separately.

2.1.1 MATLAB R2017a or Newer

Versions of MATLAB R2017a or newer contain all the necessary functions required to run MODEM and accept all of the useful toolboxes that can be used in conjunction with MODEM.

2.1.2 Windows Operating System/Operating Systems

The MODEM code references file paths and information using Windows nomenclature. It is possible, but time intensive, to convert these references so the code works on a MAC OS. If the user’s computer runs only a MAC OS, it would be best to partition the computer to run both Windows and Mac operating systems. To select the correct Windows OS to support the given version of MATLAB being used, look up the computer and system requirements for that version of MATLAB on the MathWorks website.

2.1.3 Camera Tethering/Control Software (Optional)

A camera tethering or control software regulates the rate of image capture during an experiment. There is proprietary software created by camera manufacturers that guarantee proper control of given brand’s camera, but can be rather costly, like Nikon Camera Control Pro [10]. There are also free, open-source camera control software, like digiCamControl from MIT [11], that can control numerous camera makes and models. Yet software like this is prone to bugging and does not work consistently across different computers. Another alternative is to use a camera’s internal continuous triggering capabilities to consistently take successive images at a given time increment; however, this confines the user to the capabilities of the camera’s triggering system.

2.1.4 Screen Capture Software (Optional)

In the absence of an automated approach to accurately sync images with sensor data via a timestamp, screen capture software can be helpful for recording sensor data displayed on a computer monitor. This video can be selectively edited to coincide with the capture rate of the images from a camera. Windows 8 and newer contains a built-in screen recorder contained within its Xbox app, accessed by pressing the Windows button + G and selecting the record function. For older Windows versions, a user can download a free screen capture software such as VideoProc [12] and ShareX13.1.0 [13]. Note that free software often has a maximum recording time, requiring the purchase of a software upgrade to increase recording time. If the data acquisition system (DAQ) assigns a timestamp to all recorded data, screen capture software is not required. Utilizing this type of DAQ system in conjunction with MODEM is described in Section 4.2.2.

2.2 HARDWARE AND ACCESSORIES

2.2.1 Camera

The camera type used with MODEM is not manufacturer specific but only depends on the types of images that it can generate. Any camera that meets the camera criteria listed below can be utilized:

- Manual control of camera settings (ISO, white balance, shutter speed, F-stop) and lens focusing
- Monochromatic image capture (Monochromatic sensor preferred)
- Ability to save in both JPEG and RAW file types (TIFF format also helpful)
- Continuous image capture

2.2.2 Tripod or Camera Mount

A sturdy tripod that can be weighted down, or fixed to a camera mount, is a critical part of any DIC system. For the entire length of an experiment the camera needs to remain stationary since movement can lead to significant data loss when processing the images in MODEM.

2.2.3 Light Sources

Light sources must be positioned to fully illuminate the desired region of the test specimen. The light source type and required irradiance varies based on the experimental set up as detailed in Section 4.2.1.

2.2.4 Memory Storage

Large data sets consisting of tens to hundreds of photos can be created when utilizing a DIC system. Thus, it is necessary to have an adequate amount of data storage while running an experiment as well as when processing its data. To approximate the amount of memory storage needed for an experiment, Equation 1 can be used which is based on an average photo size of around 6000 by 4000 pixels:

$$S = r_{image} * s_{image} * 10^6 * c_{rate} * t \tag{1}$$

S = data storage required (gigabytes, GB)

r_{image} = ratio of file size to image size for an image type. Common ratios are:
  - .jpg: r_{image} = 0.30 bytes/pixel
  - .tiff: r_{image} = 3.00 bytes/pixel
  - .raw: r_{image} = 1.20 bytes/pixel

s_{image} = size of the image the camera produces (megapixels)

c_{rate} = image capture rate of the experiment (images/sec)

t = length of the experiment (sec)

Equation 1 can be tailored to a specific camera by taking an image with a camera in a given file format and dividing the image file size in megabytes by the size of this image in megapixels. Substitution of this ratio for r_{image} provides a more accurate representation of the data storage needed for a given experiment.

Camera memory cards range from 16 to 256 GB. Due to the unknowns of experimentation, it is recommended to select a memory card that provides more data storage than needed. If there is not a memory card large enough to capture images for the entire experiment it may be prudent to decrease the capture rate or save photos directly to a computer or portable hard drive with greater storage capacity. Replacing a full memory card during a test is unadvisable as the camera can shift and lead to data loss.

A portable hard drive, online cloud, and/or local server storage are useful tool for additional computer storage to back up or share data. Specifically, portable hard drives can vary in size from around 250 GB to 10 TB or more and can be used to save images directly from a camera during the course of an experiment.

2.2.5 Tethering Cord (Optional)

A tether is a cord that connects a camera to a computer, allowing the user to remotely control a camera and its functions with a given software. The tether must have plugs that allow it to interface with the both the camera and computer model as well as have sufficient length to reach from the experimental setup to the computer. 5ft, 10ft, and 20ft tethers can be easily and cheaply purchased online. A tether cord is not necessary when utilizing continuous image capture capabilities of the camera.

2.2.6 Power Cord (Optional)

A power cord connects the camera to an external power supply, rather than using an internal battery. A power cord can be advantageous as it can power a camera for durations much longer than a battery without having to recharge. This avoids the need for can allow battery replacement during long experiments that may result in camera movement and result in data loss.

3 INSTALLATION GUIDES

This section describes how to install the various software required to conduct MODEM analyses including MATLAB (Section 3.1), MATLAB's Computer Vision System Toolbox (Section **Error! Reference source not found.**), MODEM (Section 3.3), and digiCamControl (Section 3.4). MATLAB and its toolboxes can be purchased directly from the MathWorks website. MODEM can be obtained for free on request from researchers at the University of Auckland where the software was developed. DigiCamControl is an optional camera triggering software that is free to download.

3.1 INSTALLING MATLAB

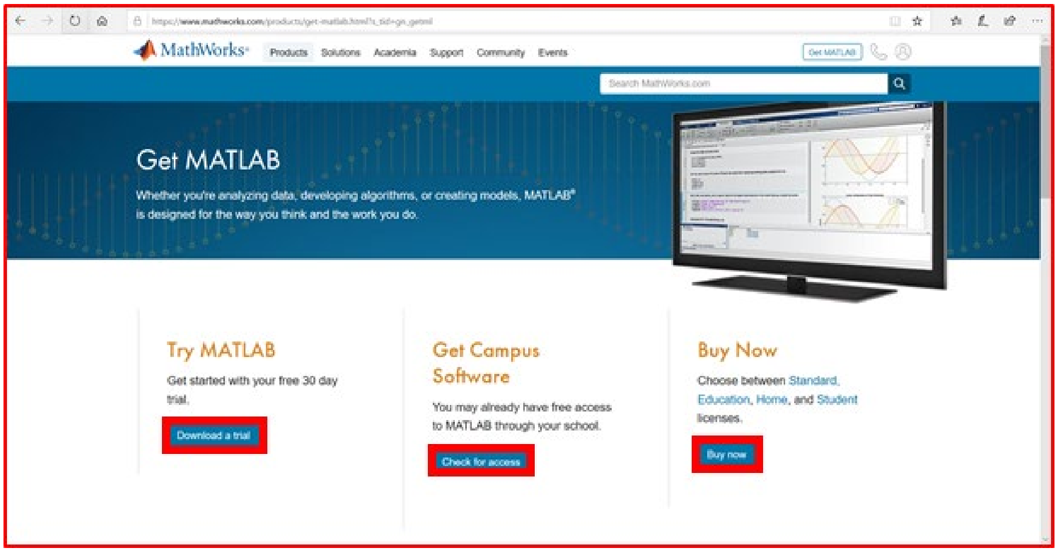
3.1.1 Overview

This section will describe how to download MATLAB and the Computer Vision System Toolbox from the MathWorks website.

3.1.2 Downloading Procedure

1. Navigate to https://www.mathworks.com/ for the MathWorks website.
2. Select the "Get MATLAB" button.

3. Obtain MATLAB either by purchasing a license, accessing MATLAB through an academic license, or getting a free trial. Most users of this manual will be using an academic license so that is what will be described below (by selecting the “Check for access” button).



Get MATLAB

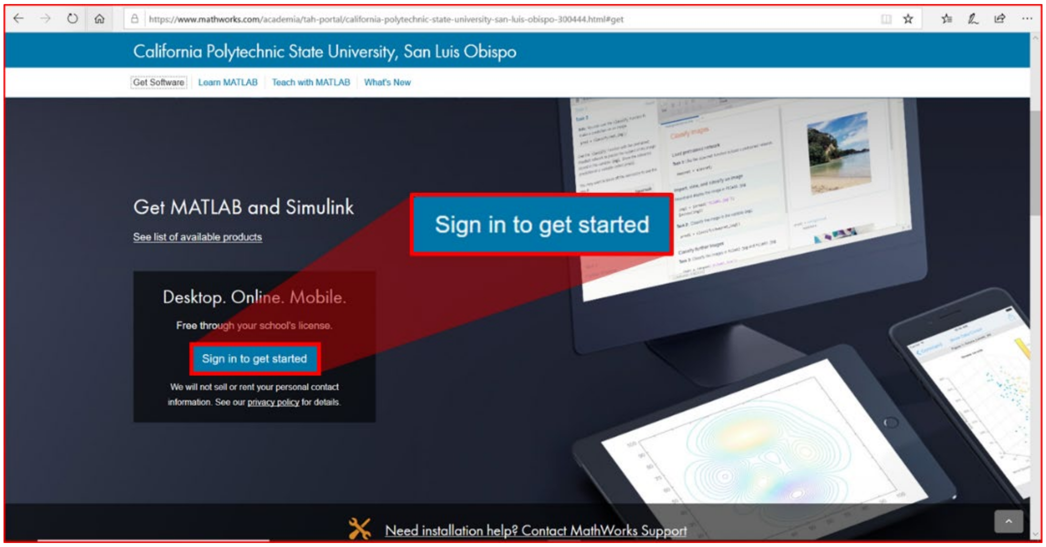
Whether you're analyzing data, developing algorithms, or creating models, MATLAB® is designed for the way you think and the work you do.

Get started with your free 30 day trial.

You may already have free access to MATLAB through your school.

Choose between Standard, Education, Home, and Student licenses.

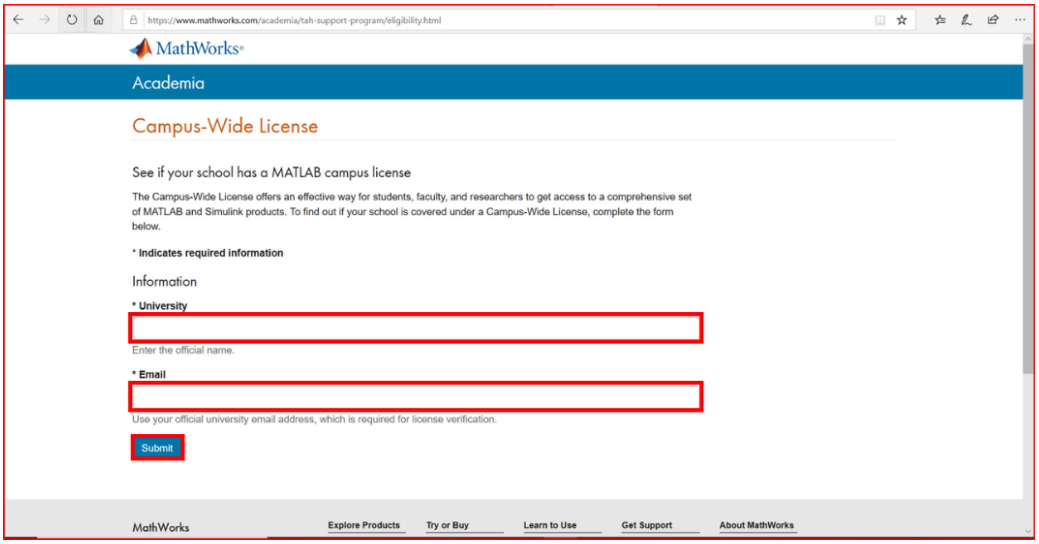
4. For obtaining an academic version of MATLAB, type the name of the university whose license is being accessed in the upper dialog box. If the university has a relationship with MathWorks, the name of the university should appear in a drop-down menu. Select the name of the university, then enter a valid university email into the lower dialog box and select the “Submit” button.
5. After selecting the “Submit” button, the email address used should receive an email from MathWorks containing a link. Follow the link back to the MathWorks site.
6. Scroll down the page and select the “Sign in to get started” button.



Free through your school's license.

We will not sell or rent your personal contact information. See our privacy policy for details.

7. Sign into the licensed university's portal using a valid university username and password. If the user has an existing MathWorks account, they will now be asked to login to this as well.



See if your school has a MATLAB campus license

The Campus-Wide License offers an effective way for students, faculty, and researchers to get access to a comprehensive set of MATLAB and Simulink products. To find out if your school is covered under a Campus-Wide License, complete the form below.

* Indicates required information

Information

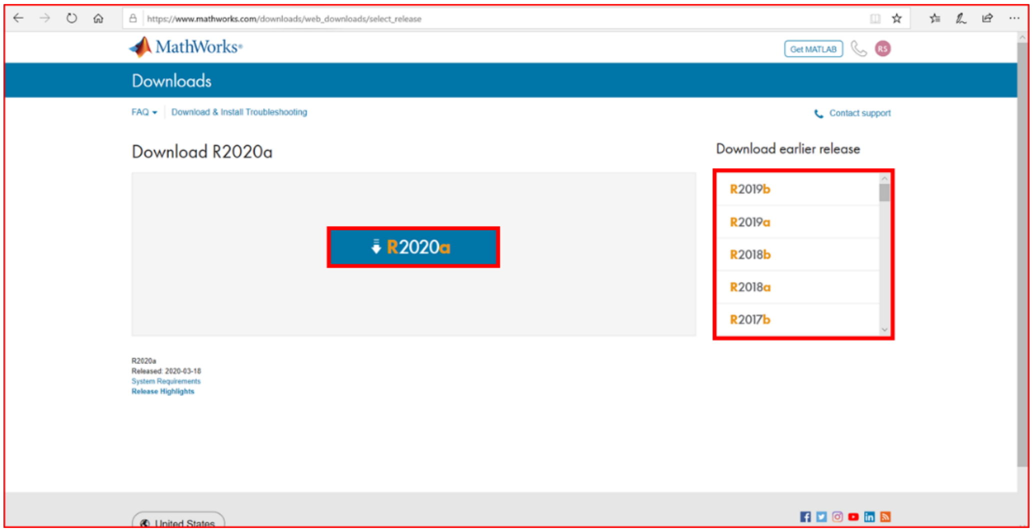
* University

Enter the official name.

* Email

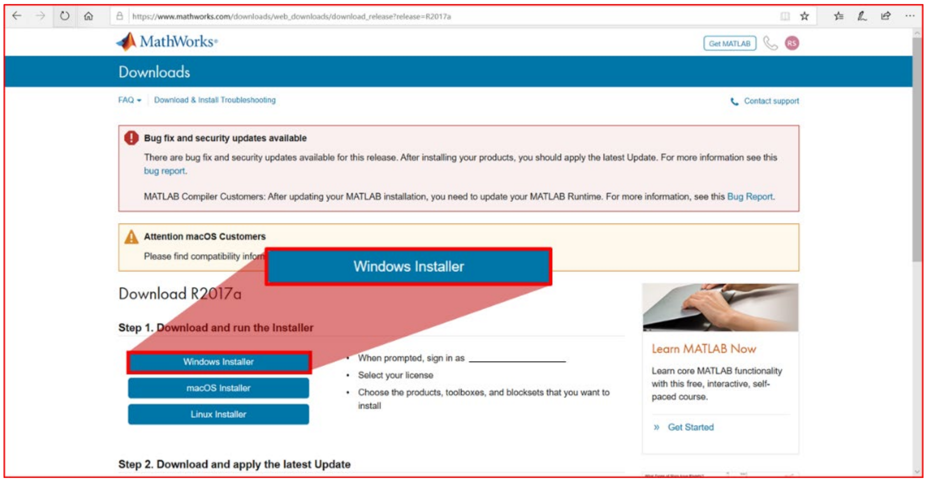
Use your official university email address, which is required for license verification.

8. From here, scroll through the versions of MATLAB and select the desired version to download. It is recommended that MATLAB R2017a or a newer version is installed.



R2020a
Released: 2020-03-18

9. Now select the “Windows Installer” button to begin downloading and installing MATLAB. Then, when a popup appears, select “Run”. The MATLAB installer will download and ask if changes can be made to the downloading device. Select “Yes”.



Bug fix and security updates available

There are bug fix and security updates available for this release. After installing your products, you should apply the latest Update. For more information see this bug report.

MATLAB Compiler Customers: After updating your MATLAB installation, you need to update your MATLAB Runtime. For more information, see this Bug Report.

Attention macOS Customers

Please find compatibility infor

Download R2017a

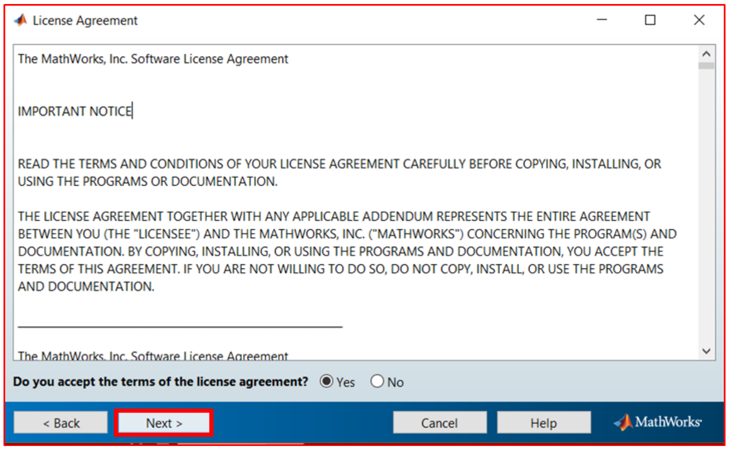
Step 1. Download and run the Installer

- When prompted, sign in as ______________
- Select your license
- Choose the products, toolboxes, and blocksets that you want to install

Learn core MATLAB functionality with this free, interactive, self-paced course.

Step 2. Download and apply the latest Update

10. Select the desired method by which to install MATLAB and then click “Next”. This guide will continue by logging into a valid MathWorks account.
11. The license agreement will then appear. Select “Next”.



The MathWorks, Inc. Software License Agreement

IMPORTANT NOTICE

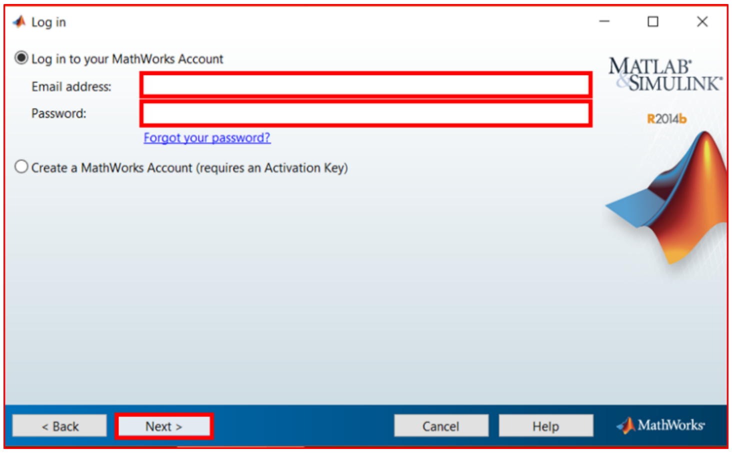
READ THE TERMS AND CONDITIONS OF YOUR LICENSE AGREEMENT CAREFULLY BEFORE COPYING, INSTALLING, OR USING THE PROGRAMS OR DOCUMENTATION.

THE LICENSE AGREEMENT TOGETHER WITH ANY APPLICABLE ADDENDUM REPRESENTS THE ENTIRE AGREEMENT BETWEEN YOU (THE "LICENSEE") AND THE MATHWORKS, INC. ("MATHWORKS") CONCERNING THE PROGRAM(S) AND DOCUMENTATION. BY COPYING, INSTALLING, OR USING THE PROGRAMS AND DOCUMENTATION, YOU ACCEPT THE TERMS OF THIS AGREEMENT. IF YOU ARE NOT WILLING TO DO SO, DO NOT COPY, INSTALL, OR USE THE PROGRAMS AND DOCUMENTATION.

The MathWorks, Inc. Software License Agreement

Do you accept the terms of the license agreement? ● Yes ○ No

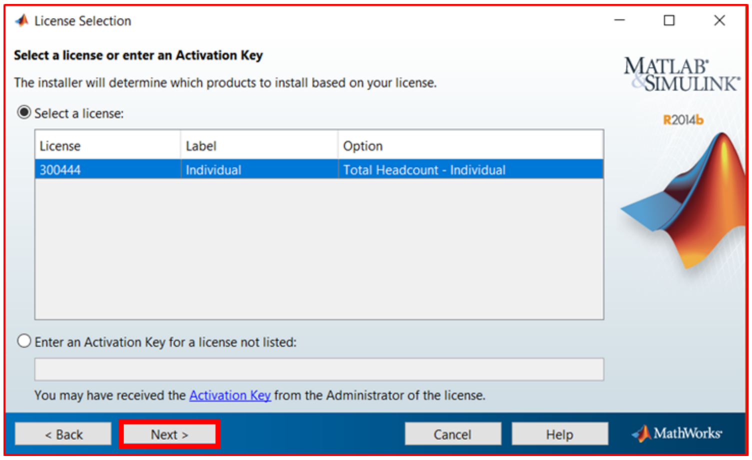
12. Log into a valid MathWorks account and then select “Next”.



● Log in to your MathWorks Account

○ Create a MathWorks Account (requires an Activation Key)

13. Select the license that MATLAB provided and select the “Next” button.



Select a license or enter an Activation Key

The installer will determine which products to install based on your license.

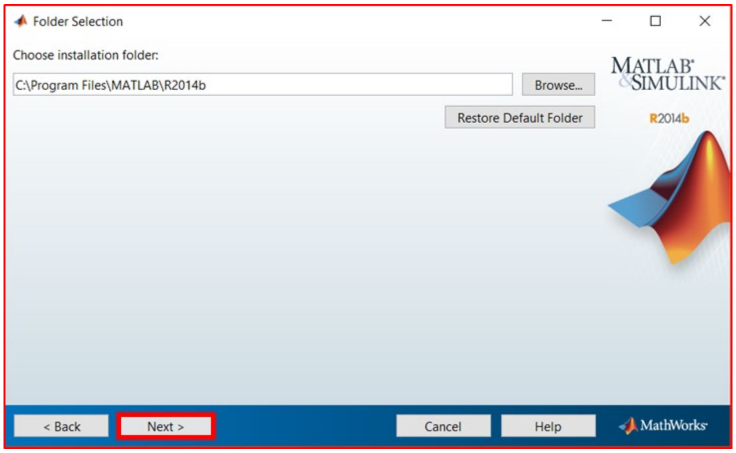
● Select a license:

License	Label	Option
300444	Individual	Total Headcount - Individual

○ Enter an Activation Key for a license not listed:

You may have received the Activation Key from the Administrator of the license.

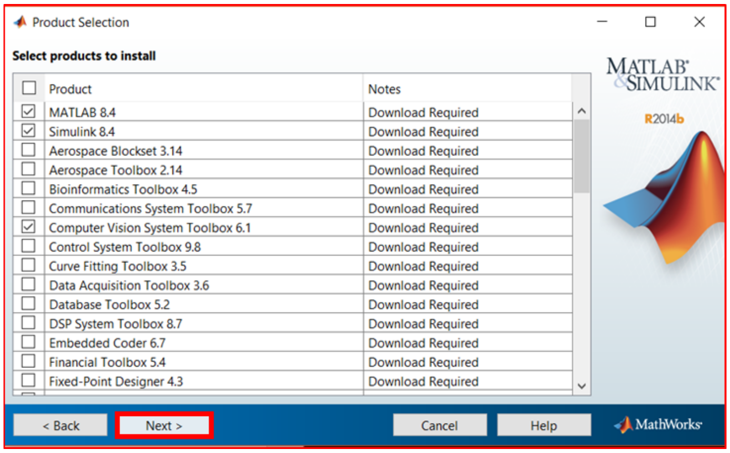
14. MATLAB will then ask for a folder in which to save the program. Simply select the “Next” button.



Choose installation folder:

C:\Program Files\MATLAB\R2014b

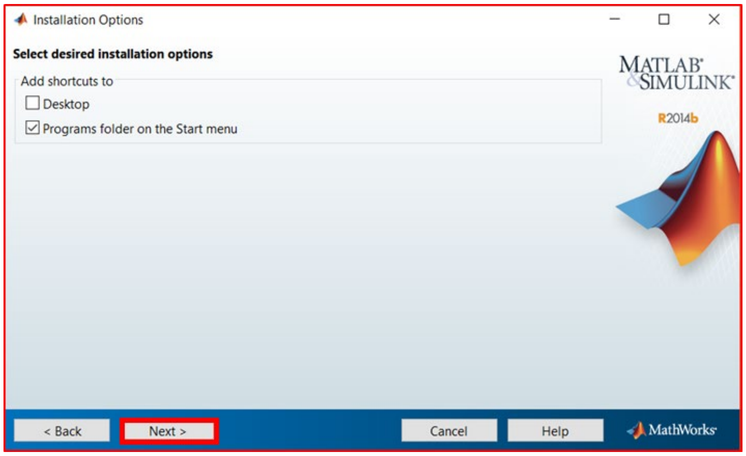
15. Select the MATLAB and MATLAB toolboxes that should be download. In order to run MODEM effectively, MATLAB and the Computer Vision Toolbox should be included. After selecting all the MathWorks products to be downloaded, select the “Next” button.



Select products to install

☐	Product	Notes
☑	MATLAB 8.4	Download Required
☑	Simulink 8.4	Download Required
☐	Aerospace Blockset 3.14	Download Required
☐	Aerospace Toolbox 2.14	Download Required
☐	Bioinformatics Toolbox 4.5	Download Required
☐	Communications System Toolbox 5.7	Download Required
☑	Computer Vision System Toolbox 6.1	Download Required
☐	Control System Toolbox 9.8	Download Required
☐	Curve Fitting Toolbox 3.5	Download Required
☐	Data Acquisition Toolbox 3.6	Download Required
☐	Database Toolbox 5.2	Download Required
☐	DSP System Toolbox 8.7	Download Required
☐	Embedded Coder 6.7	Download Required
☐	Financial Toolbox 5.4	Download Required
☐	Fixed-Point Designer 4.3	Download Required

16. Now select the desired program accessibility options and then select the “Next” button.



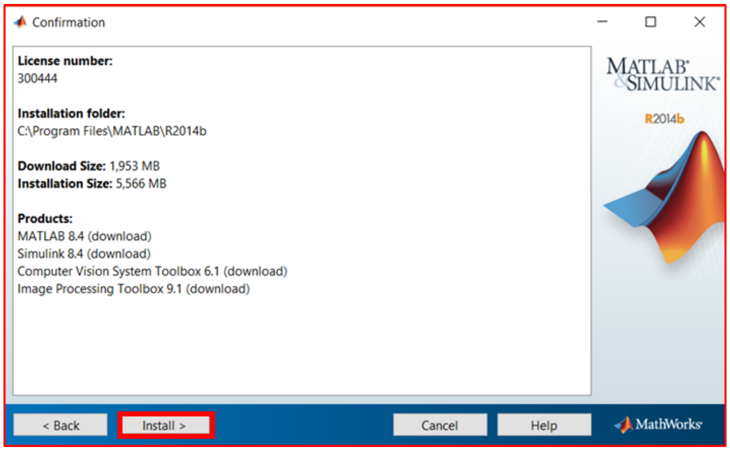
Select desired installation options

Add shortcuts to

☐ Desktop

☑ Programs folder on the Start menu

17. Confirm the installation parameters previously entered and then select the “Install” button. From here, MATLAB and the other MathWorks products and toolboxes that were selected will be installed onto the downloading computer. This process may take a while.



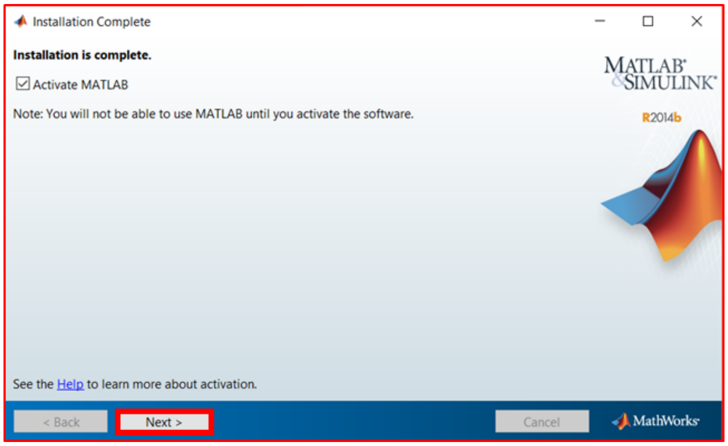
License number:
300444

Installation folder:
C:\Program Files\MATLAB\R2014b

Download Size: 1,953 MB
Installation Size: 5,566 MB

Products:
MATLAB 8.4 (download)
Simulink 8.4 (download)
Computer Vision System Toolbox 6.1 (download)
Image Processing Toolbox 9.1 (download)

18. Select if MATLAB should reopen after it is installed and then select “Next”.



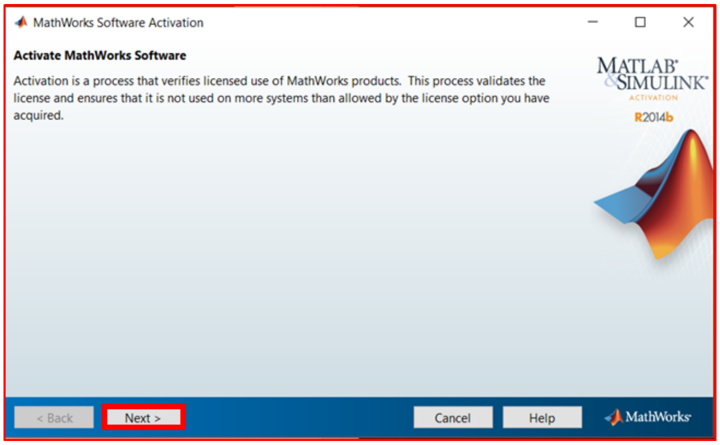
Installation is complete.

☑ Activate MATLAB

Note: You will not be able to use MATLAB until you activate the software.

See the Help to learn more about activation.

19. Select the “Next” button.

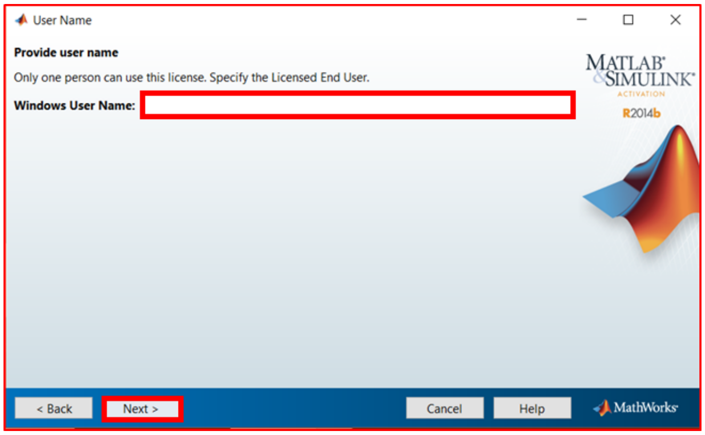


MathWorks Software Activation

Activate MathWorks Software

Activation is a process that verifies licensed use of MathWorks products. This process validates the license and ensures that it is not used on more systems than allowed by the license option you have acquired.

20. Enter the username that should be used while utilizing MATLAB. A full, given name is always a good choice and it may autofill as such. After entering a username, select the “Next” button.



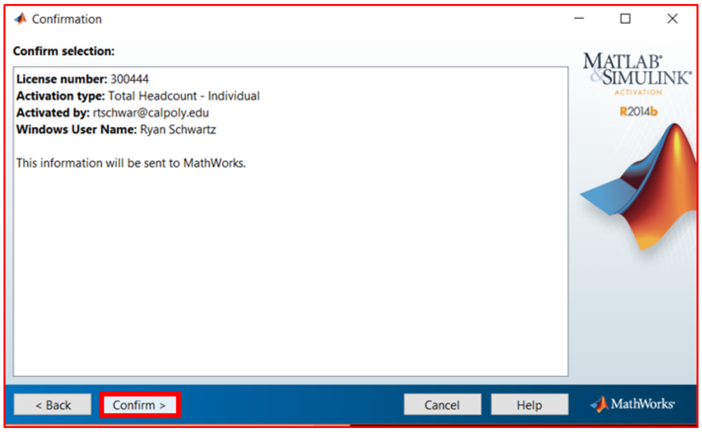
User Name

Provide user name

Only one person can use this license. Specify the Licensed End User.

Windows User Name:

21. Read over the MATLAB license information and then select the “Confirm” button.



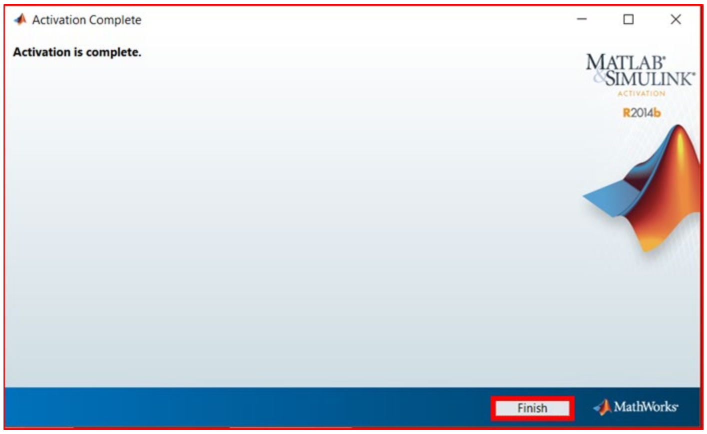
Confirmation

Confirm selection:

License number: 300444
Activation type: Total Headcount - Individual
Activated by: rtschwar@calpoly.edu
Windows User Name: Ryan Schwartz

This information will be sent to MathWorks.

22. Now MATLAB is installed. Simply click the “Finish” button.


Activation Complete

Activation is complete.

3.2 INSTALLING THE COMPUTER VISION SYSTEM TOOLBOX

3.2.1 Overview

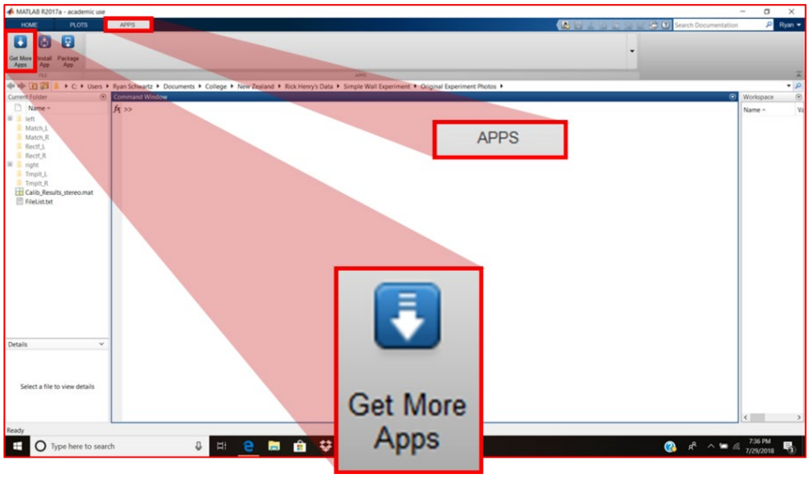
MATLAB’s Computer Vision System Toolbox will be used when creating calibration files for a given experiment, allowing for the conversion of displacement values into real world units. If this toolbox is already downloaded when MATLAB was downloaded, then this section can be ignored.

3.2.2 Checking if Toolbox is Downloaded

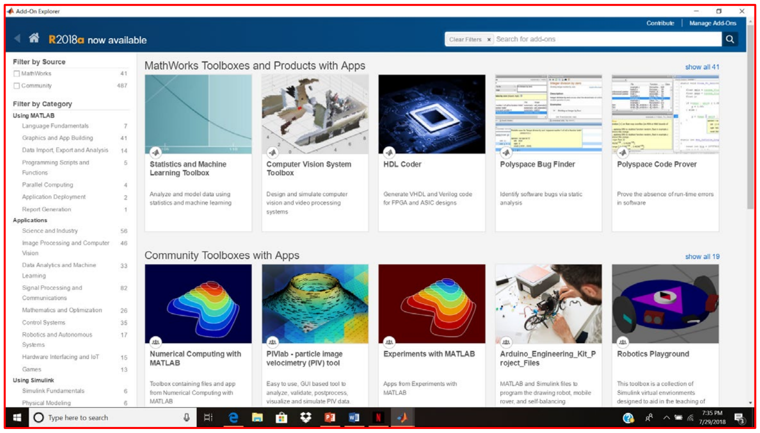
To check if this toolbox is already downloaded, select the “Add-Ons” drop-down menu in the “Home” tab and select “Manage Add-Ons”. Once the Add-On Manager is open, scroll through the downloaded toolboxes and see if the Computer Vision System Toolbox is downloaded. If not, continue to Section 3.2.3.

3.2.3 Downloading Procedure

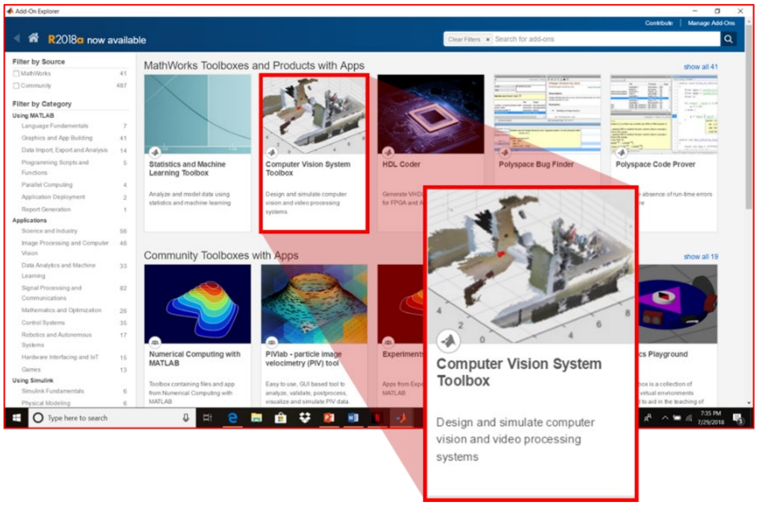
1. Open MATLAB and go to the “Apps” Tab.



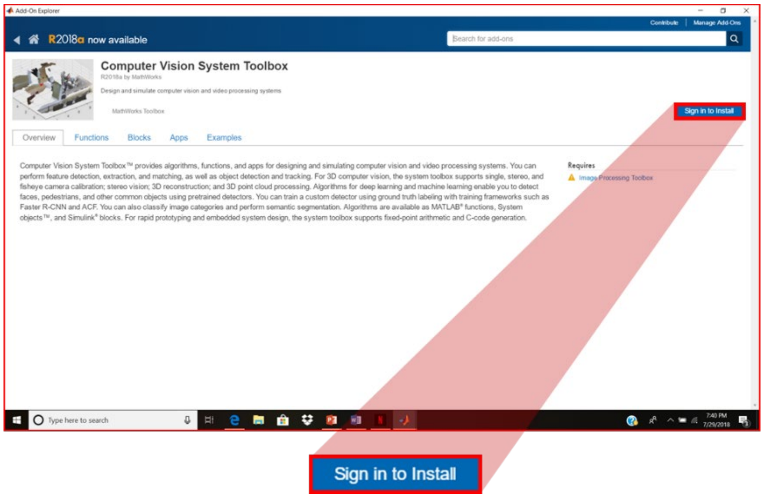
2. Select the “Get More Apps” button. The Add-On Explorer will then appear.



3. Select the Computer Vision System Toolbox.

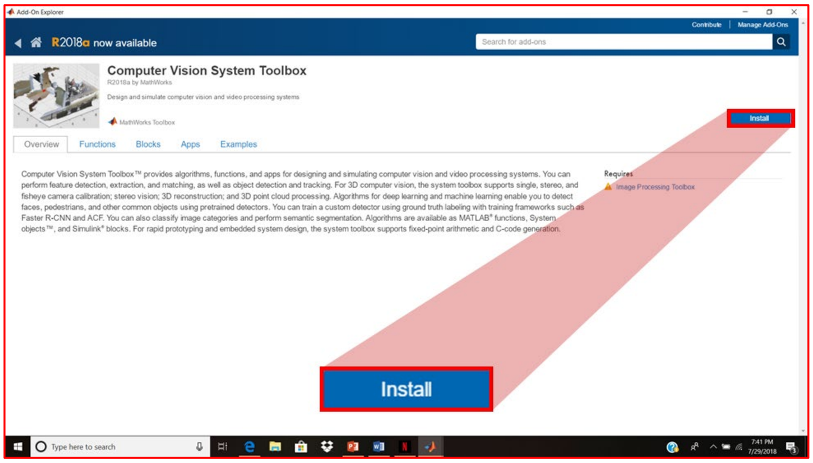


4. Select the “Sign in to Install” button or the “Install” button if MATLAB is already signed into a valid MathWorks account.



Computer Vision System Toolbox™ provides algorithms, functions, and apps for designing and simulating computer vision and video processing systems. You can perform feature detection, extraction, and matching, as well as object detection and tracking. For 3D computer vision, the system toolbox supports single, stereo, and fisheye camera calibration; stereo vision; 3D reconstruction; and 3D point cloud processing. Algorithms for deep learning and machine learning enable you to detect faces, pedestrians, and other common objects using pretrained detectors. You can train a custom detector using ground truth labeling with training frameworks such as Faster R-CNN and ACF. You can also classify image categories and perform semantic segmentation. Algorithms are available as MATLAB® functions, System objects™, and Simulink® blocks. For rapid prototyping and embedded system design, the system toolbox supports fixed-point arithmetic and C-code generation.

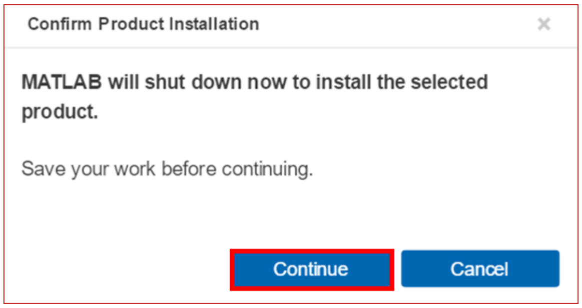
5. If needed, sign into a valid MathWorks account by using a licensed university with a university email address. After this data is entered, select the “Install” button.



Design and simulate computer vision and video processing systems

Computer Vision System Toolbox™ provides algorithms, functions, and apps for designing and simulating computer vision and video processing systems. You can perform feature detection, extraction, and matching, as well as object detection and tracking. For 3D computer vision, the system toolbox supports single, stereo, and fisheye camera calibration; stereo vision; 3D reconstruction; and 3D point cloud processing. Algorithms for deep learning and machine learning enable you to detect faces, pedestrians, and other common objects using pretrained detectors. You can train a custom detector using ground truth labeling with training frameworks such as Faster R-CNN and ACF. You can also classify image categories and perform semantic segmentation. Algorithms are available as MATLAB® functions, System objects™, and Simulink® blocks. For rapid prototyping and embedded system design, the system toolbox supports fixed-point arithmetic and C-code generation.

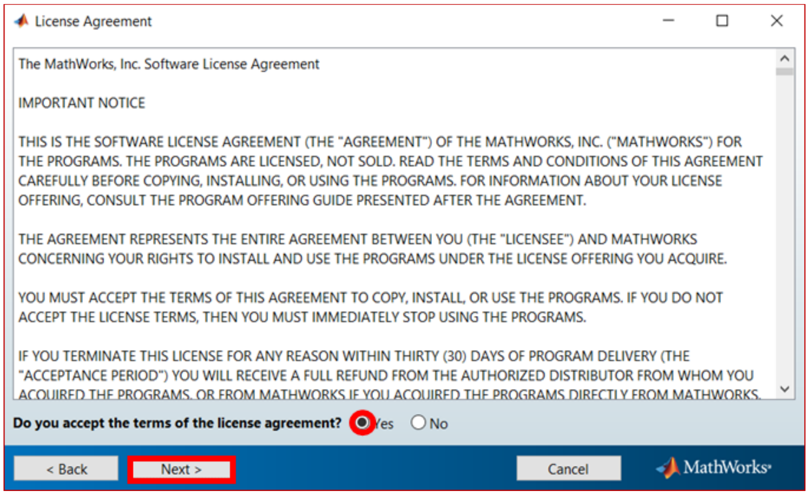
6. Select the “Continue” Button.



MATLAB will shut down now to install the selected product.

Save your work before continuing.

7. Agree to the License Agreement and select the “Next” button.



The MathWorks, Inc. Software License Agreement

IMPORTANT NOTICE

THIS IS THE SOFTWARE LICENSE AGREEMENT (THE "AGREEMENT") OF THE MATHWORKS, INC. ("MATHWORKS") FOR THE PROGRAMS. THE PROGRAMS ARE LICENSED, NOT SOLD. READ THE TERMS AND CONDITIONS OF THIS AGREEMENT CAREFULLY BEFORE COPYING, INSTALLING, OR USING THE PROGRAMS. FOR INFORMATION ABOUT YOUR LICENSE OFFERING, CONSULT THE PROGRAM OFFERING GUIDE PRESENTED AFTER THE AGREEMENT.

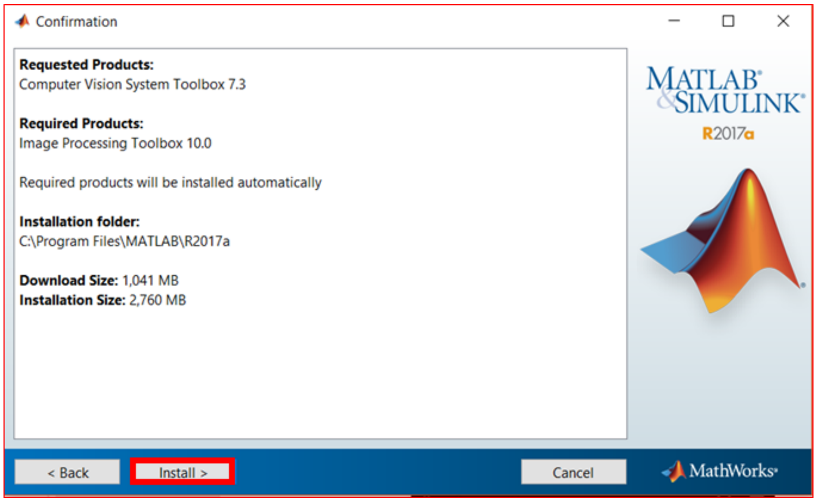
THE AGREEMENT REPRESENTS THE ENTIRE AGREEMENT BETWEEN YOU (THE "LICENSEE") AND MATHWORKS CONCERNING YOUR RIGHTS TO INSTALL AND USE THE PROGRAMS UNDER THE LICENSE OFFERING YOU ACQUIRE.

YOU MUST ACCEPT THE TERMS OF THIS AGREEMENT TO COPY, INSTALL, OR USE THE PROGRAMS. IF YOU DO NOT ACCEPT THE LICENSE TERMS, THEN YOU MUST IMMEDIATELY STOP USING THE PROGRAMS.

IF YOU TERMINATE THIS LICENSE FOR ANY REASON WITHIN THIRTY (30) DAYS OF PROGRAM DELIVERY (THE "ACCEPTANCE PERIOD") YOU WILL RECEIVE A FULL REFUND FROM THE AUTHORIZED DISTRIBUTOR FROM WHOM YOU ACQUIRED THE PROGRAMS, OR FROM MATHWORKS IF YOU ACQUIRED THE PROGRAMS DIRECTLY FROM MATHWORKS.

Do you accept the terms of the license agreement? ● Yes ○ No

8. Select the “Install” button.


Confirmation

Requested Products:
Computer Vision System Toolbox 7.3

Required Products:
Image Processing Toolbox 10.0

Required products will be installed automatically

Installation folder:
C:\Program Files\MATLAB\R2017a

Download Size: 1,041 MB
Installation Size: 2,760 MB

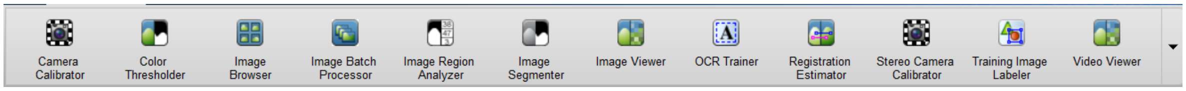
9. Select the “Finish” button.


Installation Complete

Your setup is complete.
Restart MATLAB to use your products.

☑ Launch MATLAB

From here MATLAB will open again and the Computer Vision System Toolbox will now be visible under the Apps button.



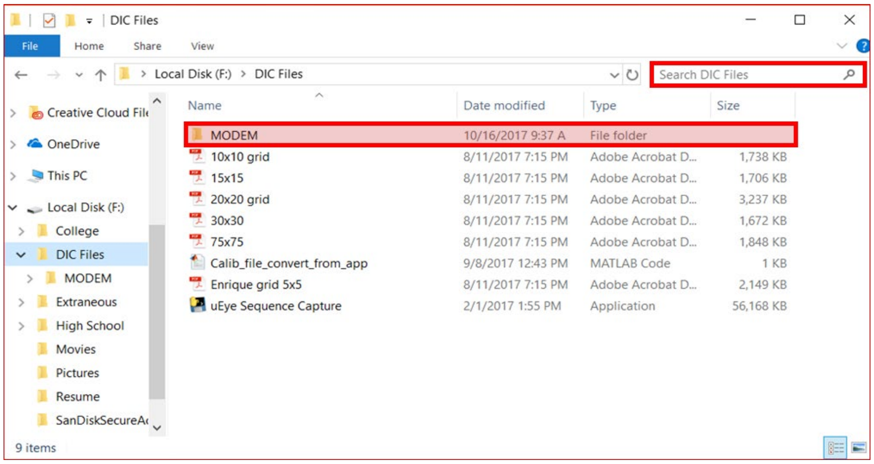
3.3 INSTALLING MODEM

3.3.1 Overview

MODEM is a series of MATLAB scripts and functions organized into various subfolders, enclosed in a folder labeled “MODEM”. This software package is maintained at the University of Auckland on portable memory devices and can be acquired by requesting access from Dr. Lucas Hogan (lucas.hogan@auckland.ac.nz). When installing MODEM, never alter the original version of the software, only copies on a user’s computer should be edited in order to maintain the original source code.

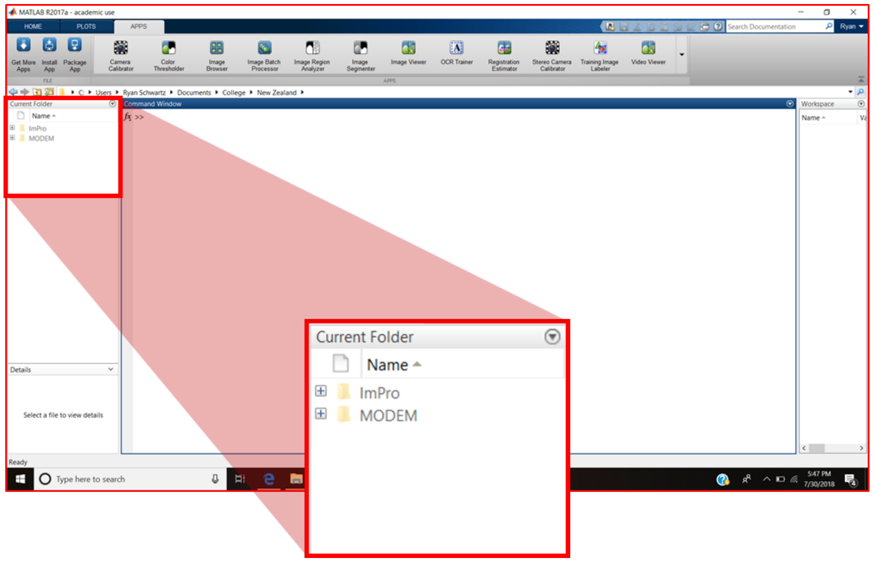
3.3.2 Downloading Procedure

1. Download MODEM from a portable memory device, or online shared folder, onto a computer.
2. Locate the MODEM folder in the file structure by typing MODEM into the Search Bar.

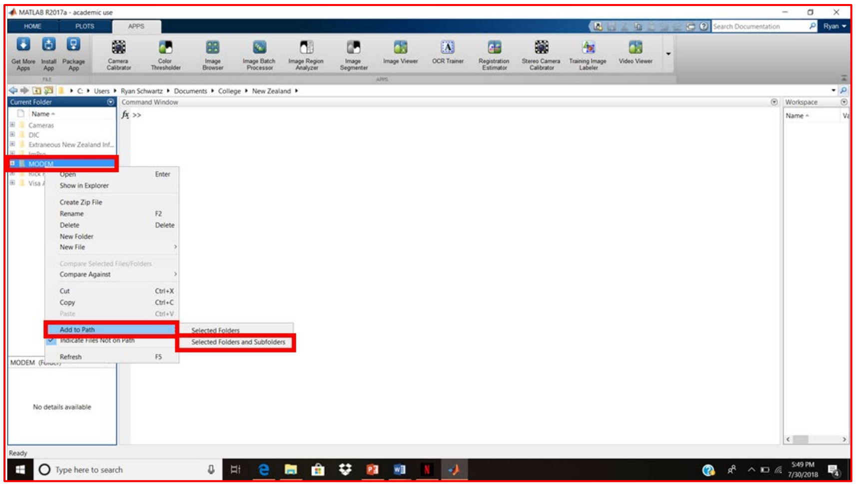


Name	Date modified	Type	Size
MODEM	10/16/2017 9:37 A	File folder	
10x10 grid	8/11/2017 7:15 PM	Adobe Acrobat D...	1,738 KB
15x15	8/11/2017 7:15 PM	Adobe Acrobat D...	1,706 KB
20x20 grid	8/11/2017 7:15 PM	Adobe Acrobat D...	3,237 KB
30x30	8/11/2017 7:15 PM	Adobe Acrobat D...	1,672 KB
75x75	8/11/2017 7:15 PM	Adobe Acrobat D...	1,848 KB
Calib_file_convert_from_app	9/8/2017 12:43 PM	MATLAB Code	1 KB
Enrique grid 5x5	8/11/2017 7:15 PM	Adobe Acrobat D...	2,149 KB
uEye Sequence Capture	2/1/2017 1:55 PM	Application	56,168 KB

3. Copy the MODEM folder onto the computer.
4. Open MATLAB.
5. Find the file location of the MODEM folder in the Current Directory window within MATLAB.



6. Now right click on the MODEM folder and select “Add to Path” in the drop-down window. Then, in the drop-down window, select “Selected Folders and Subfolders”.



7. Now MODEM can be run in MATLAB by simply typing “MODEM” into the Command Window. “MODEM” must be typed in all caps. If, at any time, entering “MODEM” generates an error, simply close MATLAB and restart it. MODEM should then be able to run.

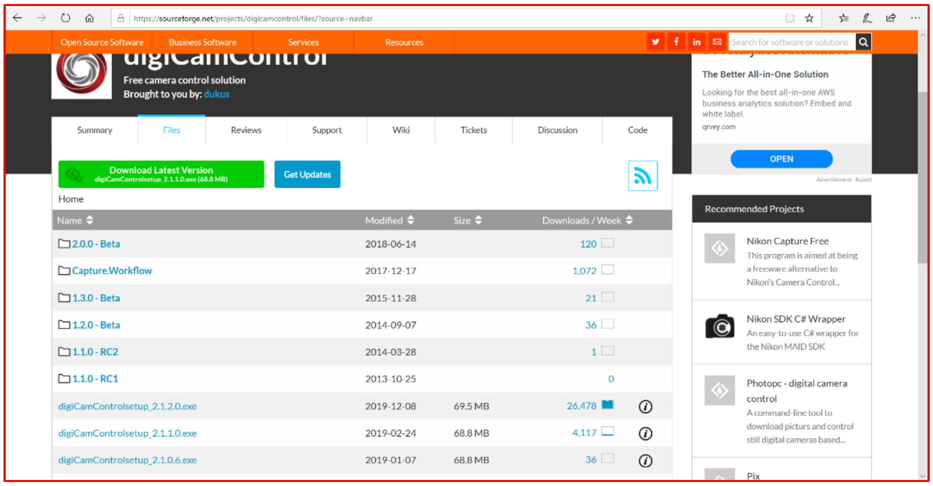
3.4 INSTALLING DIGICAMCONTROL (OPTIONAL)

3.4.1 Overview

DigiCamControl is an open-source camera control software developed by MIT that can control numerous camera makes and models from a personal computer [11]. Older versions of digiCamControl are recommended as they contain fewer bugs and are generally more stable. Performance of digiCamControl varies based on computer, so if it is not functioning properly select another version.

3.4.2 Downloading Procedure

1. Open https://sourceforge.net/projects/digicamcontrol/files/?source=navbar in the web browser.



Free camera control solution
Brought to you by: dukus

Name	Modified	Size	Downloads / Week
2.0.0 - Beta	2018-06-14		120
Capture.Workflow	2017-12-17		1,072
1.3.0 - Beta	2015-11-28		21
1.2.0 - Beta	2014-09-07		36
1.1.0 - RC2	2014-03-28		1
1.1.0 - RC1	2013-10-25		0
digiCamControlsetup_2.1.2.0.exe	2019-12-08	69.5 MB	26,478
digiCamControlsetup_2.1.1.0.exe	2019-02-24	68.8 MB	4,117
digiCamControlsetup_2.1.0.6.exe	2019-01-07	68.8 MB	36

The Better All-in-One Solution

Looking for the best all-in-one AWS business analytics solution? Embed and white label.

qrvey.com

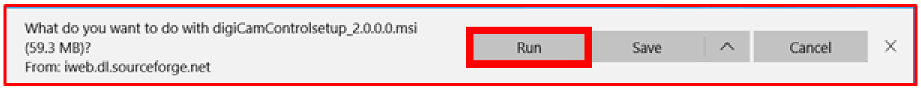
Nikon Capture Free
This program is aimed at being a freeware alternative to Nikon's Camera Control...

Nikon SDK C# Wrapper
An easy-to-use C# wrapper for the Nikon MAID SDK

Photopc - digital camera control
A command-line tool to download picturs and control still digital cameras based...

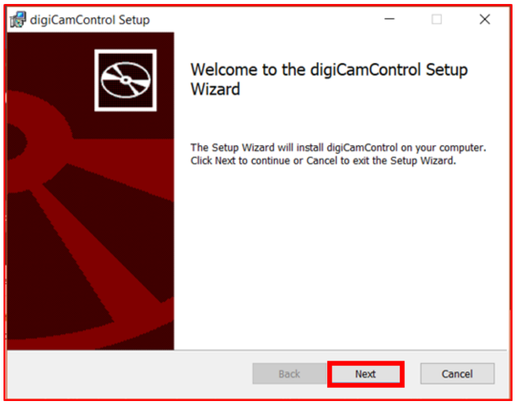
Pix

2. Scroll though and select the version of digiCamControl to be downloaded. Once selected, digiCamControl will begin downloading.
3. After digiCamControl has begun downloading, a popup window will appear in the bottom of the web browser. Select the "Run" button. DigiCamControl will then begin loading.



What do you want to do with digiCamControlsetup_2.0.0.0.msi (59.3 MB)?
From: iweb.dl.sourceforge.net

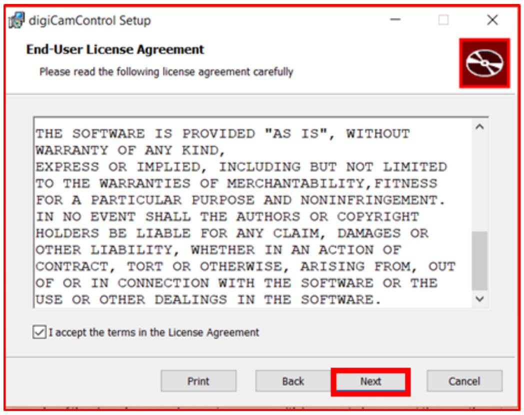
4. When the following window appears, select the "Next" button.



Welcome to the digiCamControl Setup Wizard

The Setup Wizard will install digiCamControl on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

5. Accept the terms of the License Agreement and then select the "Next" button.



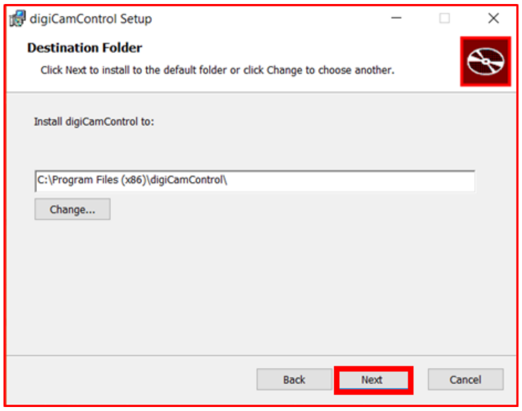
End-User License Agreement

Please read the following license agreement carefully

```
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT
WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED
TO THE WARRANTIES OF MERCHANTABILITY,FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION OF
CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT
OF OR IN CONNECTION WITH THE SOFTWARE OR THE
USE OR OTHER DEALINGS IN THE SOFTWARE.
```

☑ I accept the terms in the License Agreement

6. Now Digicam will ask for a memory location at which to save the program. Either select the “Next” button or select another location to which digiCamControl can be saved.



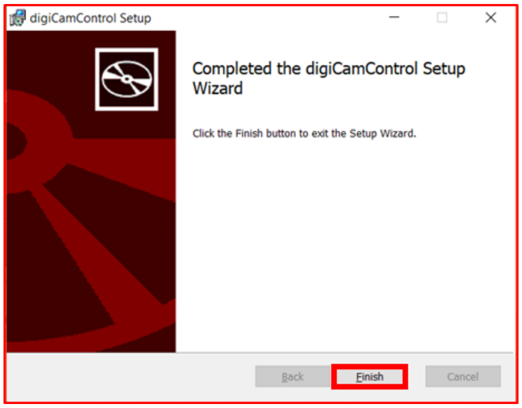
Destination Folder

Click Next to install to the default folder or click Change to choose another.

Install digiCamControl to:

C:\Program Files (x86)\digiCamControl\

7. Select the “Install” button.
8. DigiCamControl will then ask permission to make changes to the downloading computer’s hard drive. Allow the changes.
9. Once digiCamControl has finished installing, select the “Finish” button. Now digiCamControl has been installed.



Completed the digiCamControl Setup Wizard

Click the Finish button to exit the Setup Wizard.

4 BEST PRACTICE GUIDELINES

The following chapter provides guidelines for how to implement MODEM based on findings of an experimental study conducted by the author.

4.1 CAMERA SETTINGS

Table 4.1 contains recommended camera parameters to use when conducting MODEM analyses. These camera parameters are explained in Section 4.1.1 to Section 4.1.8 in the order of Table 4.1.

Table 4.1: Recommended Camera Settings

Settings	Recommended DIC Setting
Image File Format	TIFF/RAW
Image Coloring	Monochromatic
ISO	100 - 800
F-Stop	F/5.6 – F/8
White Balance	• Never Use Auto White Balance • Select Incandescent for Incandescent or Warmer Lighting • Select Florescent for Florescent, LED, or Cooler Lighting • Maintain Constant White Balance Setting throughout the Experiment
Zoom/Camera Distance	• Set Maximum Zoom • Orient Camera Orthogonal to Specimen • Position Whole Tracking Region within View
Manual Settings	• Set Focus to Manual • Set Camera Exposure to Manual • Turn Auto-Stabilization On
Shutter Speed	1/100 – 1/10 sec

4.1.1 Image File Format

<u>RECOMMENDATION:</u> Utilize a .tiff or .raw file format when saving photos during an experiment and use a .tiff file format when running data sets through MODEM.

Most modern digital cameras save photos in three main file types: RAW, FINE JPEG, and NORMAL JPEG. For DIC, RAW files are the best of these file types because they include all of the information that the camera's image sensor collects without compressing it. In contrast, JPEG images loses some of the data (or pixel information) the camera sensor collects in the compression process. This information can never be recovered since it was lost before the image was saved. This degradation in image quality can lead to inaccuracies in MODEM 's subset tracking because pixels within the original image were deleted, averaged, and edited to compress the file. The only problem with using RAW image types is that they cannot be read by all DIC software, MODEM included. RAW image files can also require proprietary software to read and process the image. However, a work-around can be to use TIFF images which can be read by practically all DIC image software (including MODEM) and compress images without losing data, making them an ideal file type to use in DIC. Most cameras cannot produce TIFF images so it is recommended to select a camera that can produce TIFF images. If not, save experimental images in a RAW file format and convert them into TIFF files before uploading them into MODEM. A photo processing software like Adobe Photoshop should be used for this task. The distinction between TIFF and JPEG images is shown in Figure 4.1.

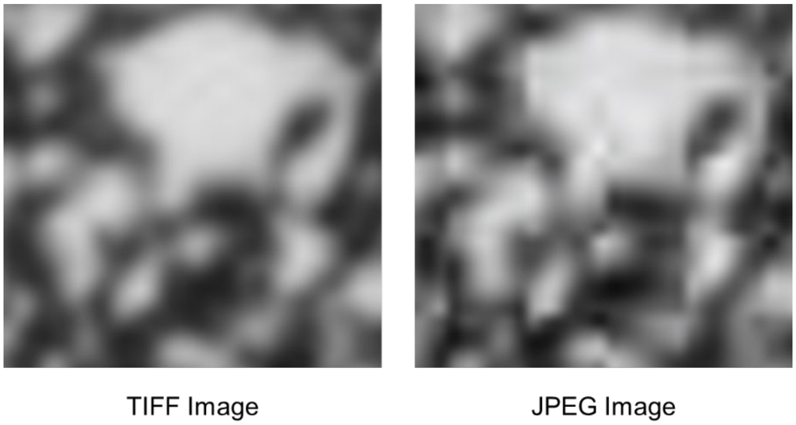


Figure 4.1: Comparing TIFF and JPEG Image Quality

4.1.2 Image Coloring (RBG vs. Monochrome)

<u>RECOMMENDATION:</u> *Use monochromatic photos in MODEM analysis by saving images as monochromatic with the camera's internal settings. If not possible, convert the images to monochromatic before they are uploaded to MODEM. Calibration images can be either monochrome or colored.*

Monochromatic images are easier for MODEM to analyze and produce more accurate results. This is due to the way MODEM identifies tracking subsets through transitions from light to dark tones. This tonal difference (or contrast) is difficult to determine in colored images because their tones change in three different channels (red, green, and blue) at once. One change in one color channel might not correspond to the same tonal change in a different color channel which can confuse the program as it tracks the subsets. With monochrome images, only one color channel is analyzed, excluding the data collected by the other color channels from the analyses due to the configuration of digital camera sensors. Thus, the best method for analyzing images in MODEM is using monochromatic images.

4.1.3 ISO Speed

<u>RECOMMENDATION:</u> *ISO speed settings should be between 100 to 800 to prevent overexposure and highlight clipping (where image highlights obscure image details) as defined in Section 4.2.1.1.1.*

ISO speed is the image sensor's sensitivity to light. The higher the ISO value, the more sensitive the sensor is and the more light it registers. A lower ISO value prevents overexposing the sensor to light while also providing a greater contrast in the image. Selection of a particular ISO value is dependent on the lighting in an experiment and should be determined by taking test photos at various ISO speeds to selecting one that does not create an image that is too dark or washed out with good contrast within the speckle pattern. In taking test images, zoom in on the speckle pattern to verify the ISO selection.

4.1.4 f-Stop

RECOMMENDATION: The f-stop should be around f/5.6 to f/8 to limit image brightness, preventing overexposure, and increasing the camera's field of view to create a buffer for the camera's focus.

The f-stop of a camera is the ratio between the lens' focal length and the diameter of the camera's entrance pupil [14]. It is the relationship between the distance at which the camera lens is focused and the amount of light that the camera is allowing into its sensor. A photographer can utilize f-stop to control the brightness of the image as well as the depth at which an image is in focus. The f-stop of a camera is directly correlated to the size of the camera's aperture, or the region of the camera lens that allows light to reach the interior of the camera. The greater the f-stop fraction (f/1.25 = 1/25) the greater the lens aperture and the brighter the image becomes. F-stop is also inversely proportional to the depth of field of the image. If the f-stop of two images are f/1.25 and f/8, the image with f/8 will have a greater field of view as shown in Figure 4.2 (right). For the purposes of DIC, limiting the amount of light to prevent overexposure of the image and creating good contrast is key. It is also useful to have a greater depth of field in case the camera is not precisely focused on the specimen. The specific f-stop from the recommended range of f/5.6 to f/8 should be selected based on effects of other camera parameters [14].

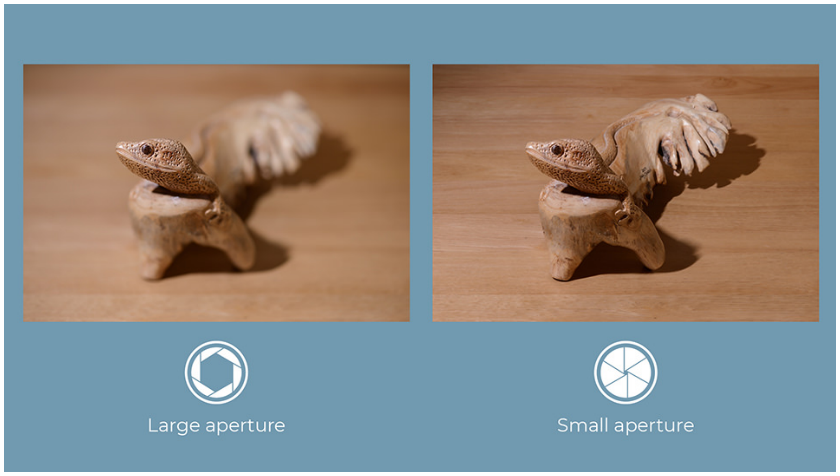


Figure 4.2: Depth of Field and Camera Aperture [14]

4.1.5 White Balance

RECOMMENDATION: It is not recommended to use an auto white balance setting. Instead, the white balance of the camera should be based on the lighting color. For lighting that is incandescent or warmer, use the incandescent white balance setting. For lighting that is florescent, LED, or cooler, use the florescent white balance setting. Make sure this setting remains constant throughout the experiment.

The white balance of a photograph is the camera's attempt to convert images into true color (the colors human eyes see) by using the white tones in the photo as a reference. The camera takes the white tones of an image and, based on the white balance setting on the camera, adds different color tones to the image. In DIC, the manual setting for white balance is important, only in that it remains constant throughout an experiment so there is consistent contrast between different points on the specimen within the entire image set. The automatic setting for white balance is problematic, and should be avoided, since the camera will add different tones to each image to correct the color. These color changes will be inconsistent across the image, altering the appearance of tracking subsets and potentially resulting in data loss [15].

4.1.6 Zoom/Camera Distance

<u>RECOMMENDATION:</u> The zoom of the lens should be at its maximum to allow for the most amount of detail possible to be seen by the camera's sensor. The camera should be placed perpendicular to the specimen face to capture a flat surface, and far enough away so that the entire portion of the specimen being tracked stays within the camera's field of view, even when it is deforming.

The lens should be set to its maximum zoom to force the camera sensor to only capture the region of the interest for the specimen and maximize the amount of light it captures this area. The camera lens should also be perpendicular the surface being tracked to only capture 2-D deformations.

4.1.7 Manual Settings

<u>RECOMMENDATION:</u> Both lens focus and camera exposure settings should be set to manual. Automatic lens focus can be used to initially focus on the specimen and while the experiment is being set up but should never be used during testing. All other automatic settings should be turned off except lens auto-stabilization.

Auto-focus can be used during the initial set up of an experiment to assist with focusing on a specimen and selecting the field of view. However, this should be the extent of its use. While an experiment is running all automatic settings should be turned off. This includes flash settings, auto exposure, Active-D Lighting, and other default camera functions. Manual settings for lens focus and camera settings should be used during an experiment to maintain image quality and settings. Automatic settings may cause the camera to autofocus for each photo, changing the camera settings and the properties of the resulting images. This can lead to tracking subsets shifting during an experiment and data loss.

4.1.8 Shutter Speed

<u>RECOMMENDATION:</u> Shutter speed should be determined in conjunction with ISO speed and should be around 1/100th to 1/10th of a second.

Shutter speed is the amount of time the shutter of a camera stays open, allowing light to hit the camera's sensor. Along with ISO speed and f-stop, shutter speed controls the amount of light that is used to generate an image. Using a long shutter speed (1/100th of a second or larger) can cause images to become overexposed and motion to blur. The recommended shutter speeds should be used cautiously to avoid these effects. For quasi-static structural tests, it may be appropriate to use a longer shutter speed since the motion being observed is quite slow. Note that shortening the shutter speed can cause an image to be underexposed and result in detail loss in the black regions. For these reasons a shutter speed of 1/100th - 1/10th of a second is recommended, although the ISO speed, f-stop, and testing rate should be considered when deciding the shutter speed [16].

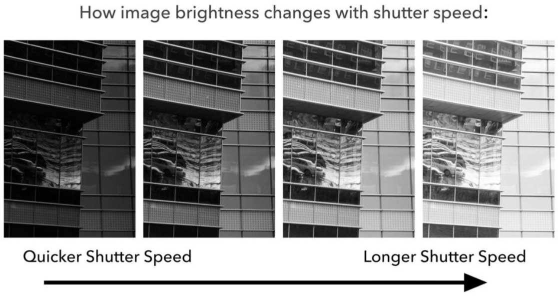


Figure 4.3: Shutter Speed's Effect on Image Lighting [16]

4.2 EXPERIMENTAL SETTINGS

4.2.1 Experimental Lighting

When setting up the lighting for a given experiment, the primary focus should be that the specimen is entirely illuminated and does not vary throughout the entire experiment. The best way to do this is by illuminating the specimen from two opposing directions, shown in Figure 4.4. Lighting should not oversaturate the specimen and thus overexpose the image.

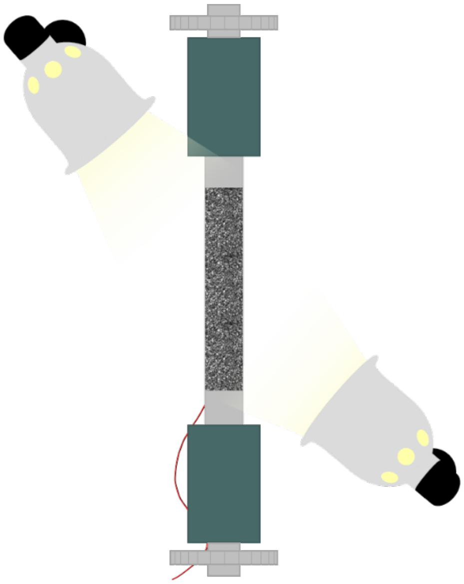


Figure 4.4: Experimental Lighting from Experiment 1M

Lighting must remain constant and stationary; if it changes then the appearance of the tracking subsets may vary across the image set, resulting in data loss. These changes are often due to ambient lighting in a lab or from exterior windows. To mitigate these affects, it is recommended that diffused light sources be used. It is also recommended that ambient light sources around any DIC experiment should be covered, removed, or otherwise dealt with. Methods for do so can be:

- Turning off the lights in the room containing the experiment
- Covering the windows with a material that blocks sunlight
- Construct a tarp over the experiment to block out ambient light

The type of lighting utilized in an experiment is not of the utmost concern for any researcher because, as long as it is constant and unchanging, the use of one type of lighting compared to another will not change the way the speckle pattern appears in a data set. Therefore, it is up to the experimentalist to determine the best light source for the structural tests being conducted. Lighting should also be placed far enough from the camera and the specimen so as to not heat up the specimen or induce heat waves that could obscure the image [4]. It is up to the experimentalist's judgement where light sources will be placed that meet the discussed criteria while also not interfering with the experiment being conducted.

4.2.1.1 PHOTOGRAPHY PRINCIPLES TO HELP WITH LIGHTING

Knowledge of histogram and image metadata is useful when finalizing the lighting for a DIC experiment to ensure image quality.

4.2.1.1.1 *HISTOGRAMS*

Histograms are a useful graphic representation of the tonal values of an image. Tones in black and white photography signify the brightness of a given image at a given location within that image (a pixel). These tones vary throughout the photograph and increase with the complexity of the object or scene being imaged [17].

A histogram plots tonal values of an image along the horizontal axis and number of pixels containing that tonal value along the vertical axis. These tonal values can be then broken down into five ranges: blacks, shadows, midtones, highlights, and whites for black and white images as shown in Figure 4.5.

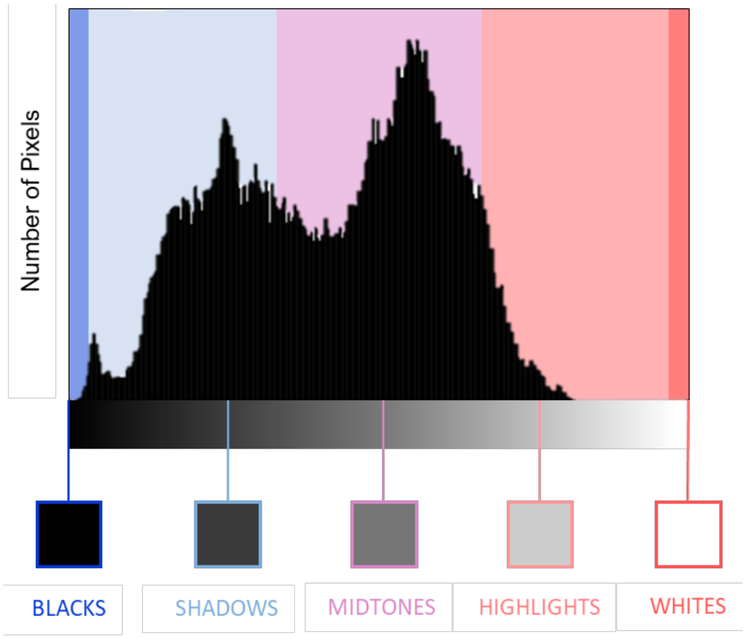


Figure 4.5: Breakdown of a Histogram

In DIC it is important to have an even balance of tones within the images. An even distribution of tones means the images, or the speckle pattern within the images, are very contrasted. Some examples of images with high and low contrast are shown in Figure 4.6.

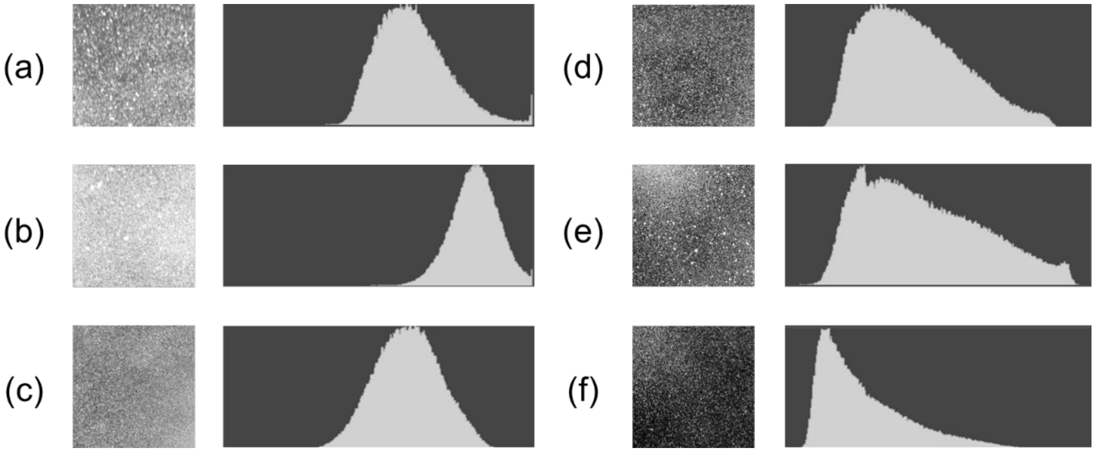


Figure 4.6: Examples of Speckle Patterns and their Histograms

Figure 4.6 shows that high contrast images (Figure 4.6d and Figure 4.6e) consist of a greater range of tones and do not have many pixels concentrated in either the black or white regions. This wider range of tones allows MODEM to more easily differentiate one grouping of pixels from another since more tones make up the image. This allows for more unique tracking subsets to be identified. Tonal ranges that are smaller but still centralized can still provide good results but, with less tonal values, will be less likely to create unique tracking subsets (Figure 4.6a and Figure 4.6c).

When an image lacks contrast such that it too dark or too light, MODEM can lose tracking subsets in the homogeneity of black and white tonal values. This can be prevented by using the process described in Section 4.2.1.1.1.1 by taking a reference image of the speckle pattern applied to the test specimen and examining the histogram to identify if the images has histogram with shadow or highlight clipping. If the histogram peaks on the left, the image is experiencing shadow clipping where the details of the image are getting lost in the black tones (Figure 4.6f). While if a histogram peaks to the right, the image is experiencing highlight clipping where image details are washed out in the whites tones (Figure 4.6b).

4.2.1.1.1.1 UTILIZING HISTOGRAMS IN MODEM ANALYSES

Histograms can be used as a tool to prevent data loss due to poorly contrast in speckle patterns. By determining how a speckle pattern is represented in a histogram, some important analyses can be made before the pattern is ever used. To utilize a histogram in a MODEM analysis:

1. Take an image of the specimen’s speckle pattern before it is placed in the experimental setup.
2. Open the image in a photo-editing software like Adobe Photoshop.
3. Generate a histogram from the image taken in Step 1.
4. See if it the image exhibits highlight or shadow clipping as explained in Section 4.2.1.1.1.
5. If highlight or shadow clipping is observed, follow the procedure specified in Section 5.1 to recreate the specimen’s speckle pattern.
6. Place the specimen in the experimental setup (including lighting).
7. Take another image of the speckle pattern in the experimental setup.
8. Generate a histogram from the image acquired in Step 7.
9. Determine if any highlight or shadow clipping is present as discussed in Section 4.2.1.1.1.
    a. If highlight clipping is present, try reducing the ISO (explained in Section 4.1.3) and/or the shutter speed (explained in Section 4.1.8) to reduce the amount of light in the image [17].
    b. If shadow clipping is present, try increasing the ISO and/or the shutter speed to increase the amount of light in the image [17].
    c. Continue taking images, producing histograms, and altering camera parameters until a desirable histogram is achieved.
10. Begin the experiment.

4.2.2 Data Acquisition

After a DIC experiment is completed, it is necessary to relate data from the traditional instrumentation (strain gages, extensometers, etc.) to the DIC images. One approach is to continuously sample sensor data at a given rate and each sample be given a timestamp. The timestamp for each DIC photograph could then be used to automatically sync with the sensor data. This method, however, requires that both the traditional instrumentation and the camera be initiated at the same time and that the sample rate for those sensors to have at least one matching least common multiple with the image capture rate to have corresponding timestamps. The timestamps for the images within the image set can be extracted from photograph metadata as explained in Section 4.2.2.1.

If the traditional instrumentation used cannot include a timestamp with its data streams, then the method specified in Section 6.2.5 can be used instead. This method records the streams of traditional instrumentation data with a screen capture software and an on-screen stopwatch while the images are taken at a given interval. After the experiment is completed, the streams of instrumentation data are examined to align recorded data to the same interval as the images were taken (indicated by the stopwatch). This method was found to be tedious and lacking accuracy, so it is not recommended.

4.2.2.1 METADATA

Metadata is descriptive information embedded in images files when they are generated and can include: ISO speed, shutter speed, focal length, f-stop, camera model, file type, time image was taken, etc. [15]. Metadata can be used to record or verify camera parameters used in a given experiment as well as add comments to given photos to note specific events in an experiment. This data can be accessed through photo-editing software like Adobe Photoshop or more readily accessed in Windows computers by right-clicking on a given image and selecting the “Properties” button from the dropdown menu. The *Properties* window that opens, and all of the tabs it contains, is filled with the metadata for that given image. Some of the metadata cannot be edited, like the ISO speed and f-stop. Other parameters, like the author, date acquired, and comments, can be edited. Metadata, in the case of MODEM, may be best way to cross-reference the timestamp to relate sensor data with DIC images.

4.2.3 Test Setup

When setting up a DIC experiment, the items/components specified in Table 4.2 will need to be acquired or determined beforehand.

Table 4.2: Materials Required to Set Up a DIC Experiment

Experimental Components	Accompanying Document Section
Camera	Section 0
Tripod or Camera Mount	Section 2.2.2
Computer	Section 2.1.2
Selected Software	Section 2.1.3 & 2.1.4
Tethering Cord (Optional)	Section 2.2.5
Power Cord (Optional)	Section 2.2.6
Memory Storage	Section 2.2.4
Light Sources	Section 2.2.3
Preliminary Camera Settings	Section 4.1
Speckle Pattern	Section 5.1

Once all of these components are assembled, the structural component or system test specimen should be set up as normal without regards to the camera equipment for DIC. To finalize the camera setup as shown in Figure 4.7, complete the following steps:

1. Place the camera perpendicular to the specimen’s surface with targets and/or speckle pattern to measure 2D deformations. The camera should be as close to the specimen as possible with the desired tracking region in the camera’s field of view. The distance between the camera and the

specimen, as well as the camera's field of view, should also account for the anticipated displacement of the tracking region during the experiment.

2. Attach the tethering cord (if desired) between the camera and the computer with image triggering software. Make sure that the cord, camera, and computer are out of the way so that nothing is disconnected or shifted during the experiment, risking data loss. The camera's tripod can be weighted down and a zone around it can be demarcated with tape to designate it as off limits.
3. Set up the camera power source. If the battery can last the whole experiment it is possible to use the internal battery, otherwise use a cable to power the camera externally.
4. Place light sources according Section 4.2.1. Verify the lighting conditions and the camera settings by using histograms as described in Section 4.2.1.1.1.
5. Take a few photos of the tracking region to make sure the specimen, targets, and/or speckle pattern are in focus, checking there are no odd changes in camera settings or camera view between photos.
6. If possible, before starting the full experiment, test that all of the instrumentation is working, as well as the DIC setup, by loading the specimen in the elastic region.

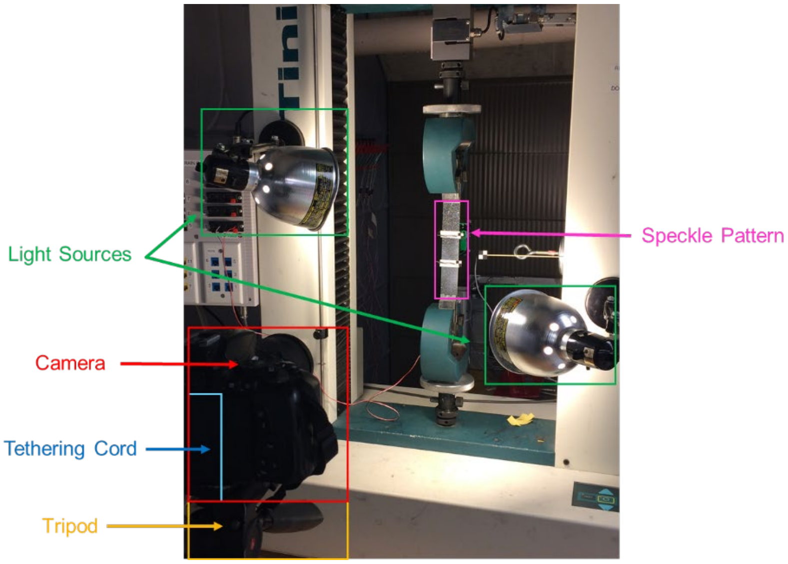


Figure 4.7: Example of DIC Experiment Test Setup

4.2.4 Comparing Traditional Instrumentation to MODEM

It is recommended that MODEM should not be used as the sole instrument in an experiment, rather as an additional tool to supplement traditional systems. It is necessary to compare the results between the systems, and this can be accomplished by creating an diagrammatic instrumentation plan cataloging the locations of all traditional instrumentation as well as the regions or targets that utilized DIC. Thus, targets can be placed at the ends of instruments, like linear potentiometers, and nodes can be generated at points where strain gages exist. This deliberate placement of DIC instrumentation can allow MODEM to blend seamlessly with existing sensor plans for structural engineering experiments.

5 USER GUIDES

The following section contains simple guides to help new learners of programs such as MODEM easily navigate its interfaces and utilize its capabilities.

5.1 HOW TO PREPARE A SPECKLE PATTERN

Speckle patterns are the "sensors" in any DIC system. These patterns are important to DIC analysis and pose some of the biggest challenges when utilizing DIC systems [8]. Although DIC systems can utilize the natural patterns found on a specimen, it is common to apply a speckle pattern to a specimen instead so that the software can identify numerous unique tracking points across its surface.

A good speckle pattern, as described by Dong & Pan (2017), is

- **High contrast:** black and white pattern with little to no grey
- **Random:** no repeating patterns
- **Isotropic:** there is no directionality in the pattern causing speckles and the gaps between them to be approximately the same size
- **Stable**: tightly adheres to the specimen, even under deformation, with the pattern shape and color remaining constant throughout loading

A common material used to create speckle patterns is aerosolized paint or matte spray paint available at any hardware or craft store.

Figure 5.1 demonstrates how to create a good speckle pattern using spray paint for an aluminum tensile coupon. First, mark off the intended patterned region with painter's tape. Apply a layer of black spray paint to this region, spraying evenly so that none of the specimen's natural surface shows through. After the black paint has dried, apply white speckles by spraying white matte paint through a mesh screen onto the specimen. Placing a mesh screen approximately one foot between the spray paint canister and the specimen can capture larger droplets of spray paint preventing them from being deposited on the surface of the specimen. This is necessary since larger speckles can reduce the number of tracking subsets MODEM generates due to the large areas of color uniformity.

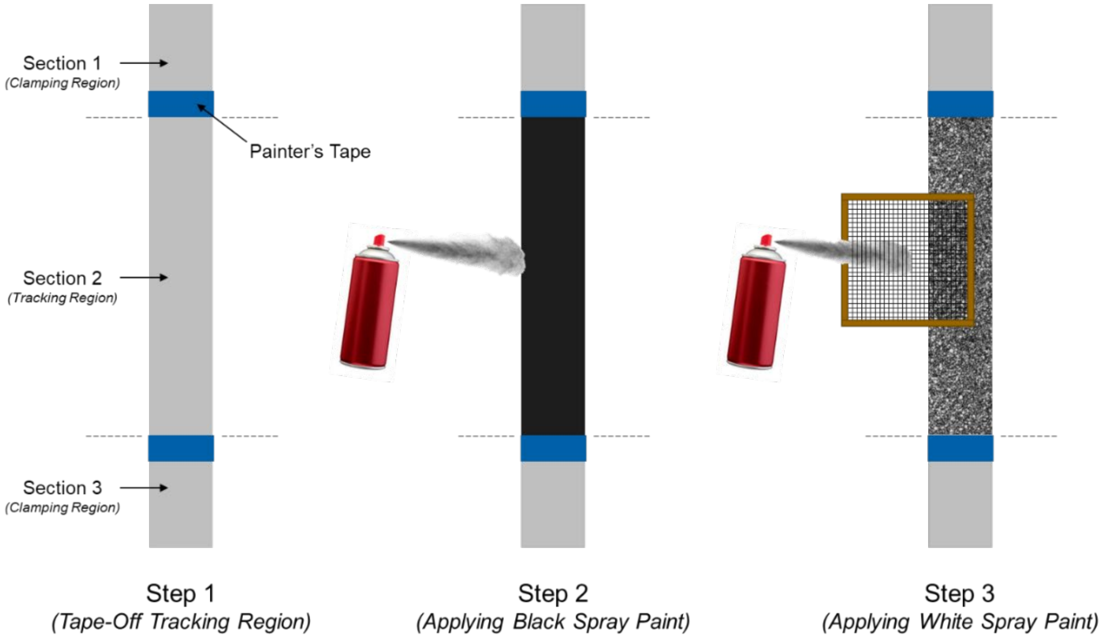
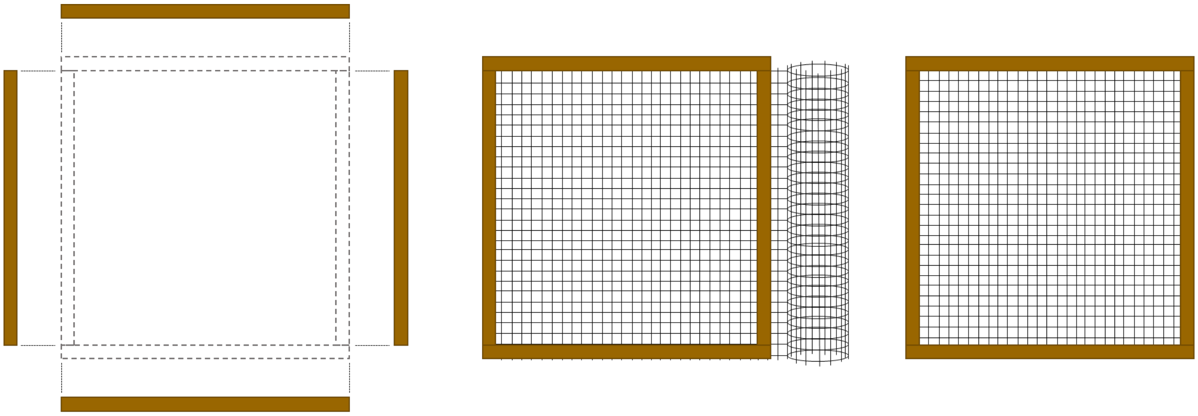


Figure 5.1: Diagram of Speckle Pattern Application

As shown in Figure 5.2, this mesh screen consists of a 10” by 10” wooden frame of ¾” members wrapped in mesh and stretched until taut. The mesh (a 3/64 in or 1.2 mm square grid) is secured with some sort of fastener, like staples.



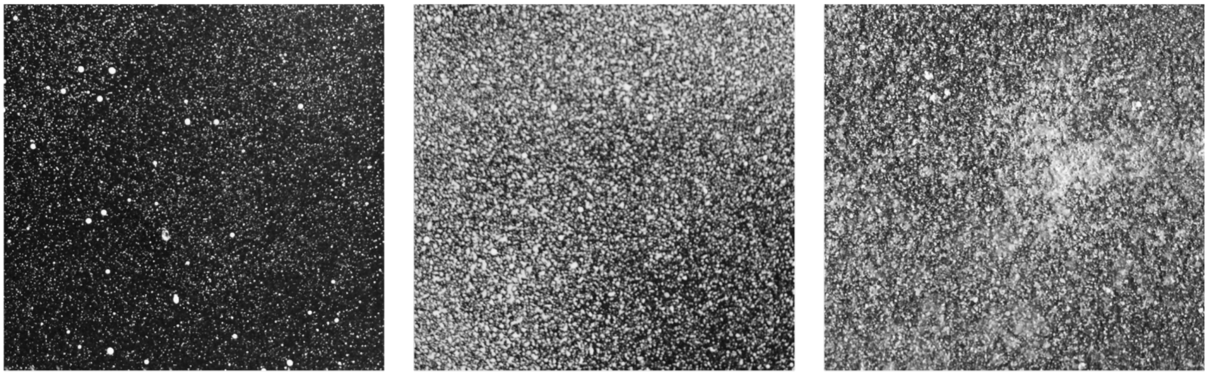
Step 1
(Assemble Frame)

Step 2
(Stretch Mesh over Frame)

Step 3
(Fix Mesh to Frame)

Figure 5.2: Mesh Screen Layout and Construction

Creating speckles with this mesh screen can be achieved by quickly spraying the white paint over the specimen with the screen in between the paint canister and the specimen, moving the screen with the spray can. If the speed with which the spray can is moved is too quick, the region being sprayed might need to be sprayed multiple times to create a dense enough speckle pattern. It would be helpful to practice on a piece of scrap material first to achieve the proper technique. Some examples of good and poor speckle patterns are shown in Figure 5.3.



(a) *Poor*

(b) *Good*

(c) *Poor*

Figure 5.3: Examples of Good and Poor Speckle Patterns

Figure 5.3a has too few speckles on its surface, resulting in large sections of one color which might reduce the number of tracking subsets identified by MODEM. The same can be said for large areas of the white speckle color in Figure 5.3c. Figure 5.3b represents a good speckle pattern that incorporated a random but thorough mix of both black and white in which the speckles themselves are, on average, the same size as the gaps between them. Conveniently, if an error is made in creating a speckle pattern, simply spray over the desired tracking region with another base layer and try again.

Speckle patterns can utilize black speckle on a white base or vice versa, but in either case the patterns must be dense enough to be isotropic.

When using spray paint to create speckle patterns it is important to keep the spray can's nozzle clean. If not, paint can build up on the nozzle and make the paint aerosolize unevenly or even clog the nozzle. To prevent this, use rubbing alcohol and a q-tip to remove excess paint off the spray nozzle after each use.

5.2 TARGETS

Targets allow users to track specific locations on a specimen throughout an experiment. They can be used in numerous ways, whether it is to create a grid to compare a specimen to a finite element model (FEM) or to track if an experimental setup is moving or vibrating. Using targets also creates a simpler DIC setup while still utilizing a powerful non-contact measurement system.

From numerous tests, it was determined that the best targets to use with MODEM are small squares covered in a speckle pattern. These targets can be made by applying a speckle pattern (detailed in Section 5.1) to a small square of cardstock, cardboard, or any other rigid material. These can then be adhered to a specimen with a putty or adhesive.

Tracking targets is the exact same process as conducting the regular 2D DIC (detailed in Section 5.3). When conducting this analysis, it is recommended that each target be its own tracking region for simplicity, organization, and flexibility in target placement. This recommendation is displayed in Figure 5.4 where each tracking region is outlined in a unique color. After an analysis is complete, the data on the locations of the targets can be extracted from MODEM (using the process described in Section 5.7.3).

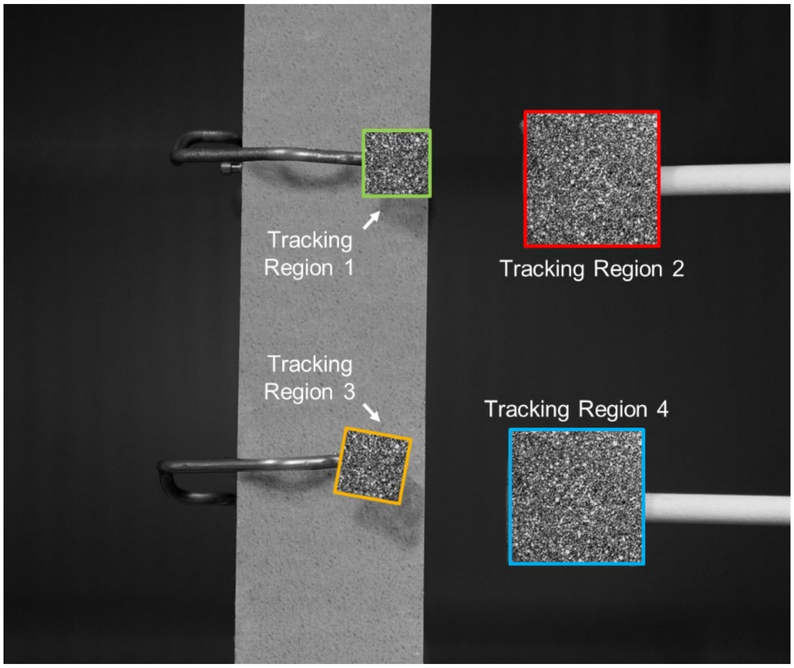
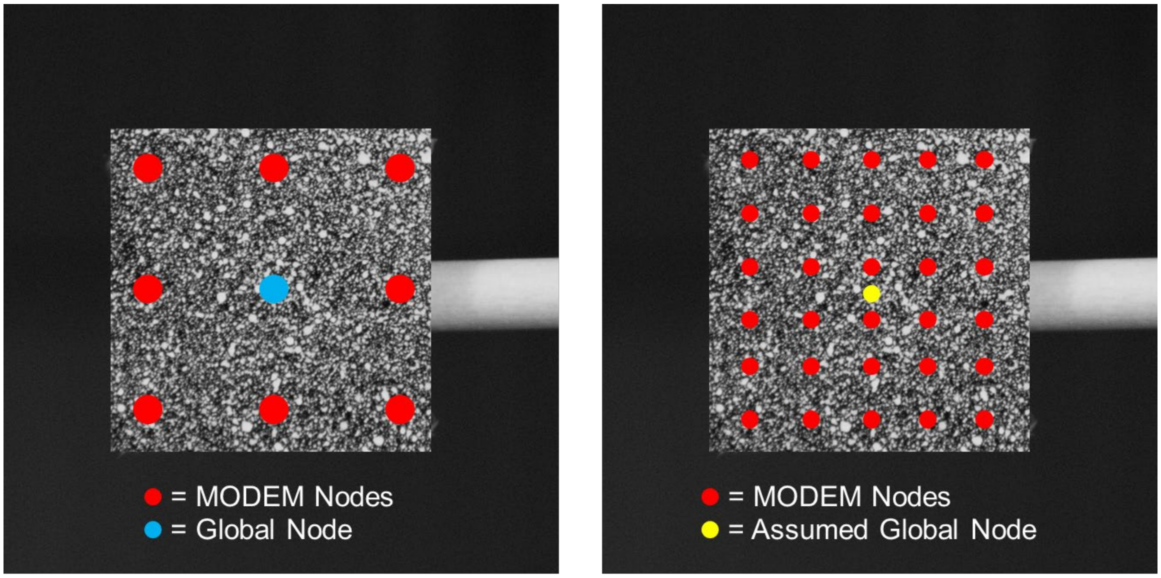


Figure 5.4: Example of Targets and Accompanying Tracking Regions in MODEM

Using targets instead of applying a speckle pattern to the entirety of a specimen can reduce MODEM's computation time but does forgo full-field measurements. However, a speckle pattern can still be applied to the specimen in addition to the targets to also acquire the specimen's full-field behavior. When using targets in MODEM the user needs to verify that the patterns applied to the targets are still visible when imaged and meet the criterion specified in Section 5.1 so the tracking subsets can be identified.

5.2.1 Utilizing Targets in MODEM

When utilizing targets in MODEM it is important to be aware that MODEM will generate a series of nodes (explained in Section 5.7.2.2) within the bounds of each target. Multiple nodes within each target and multiple targets in each experiment can make the numerous nodes generated in each experiment hard to manage and utilize. Therefore, after the data has been processed and extracted from MODEM's output, the user must select a centrally located node on each target to act as the "global node" for the entire target (shown with the cyan marker in Figure 5.5a and the yellow marker in Figure 5.5b). This global node will be the only node the user utilizes from the nodes generated for each tracking region, minimizing the number of nodes being dealt with and reducing data post-processing efforts. Based on the node spacing there may not be a centered global node on a target so a global node needs to be assumed (shown with a yellow marker in Figure 5.5b). The location and displacements of this node needs to be averaged from the location and displacements of the nodes adjacent to the assumed central node. Strain values for these targets should be ignored since the targets themselves are not deforming.



(a) Target Nodes with Global Node

(b) Target Nodes with Assumed Global Node

Figure 5.5: Target Nodes and Global Node Examples

5.2.2 Duplicating Traditional Instrumentation

One application for targets in MODEM is utilizing them as redundant systems for traditional instrumentation. Targets can be placed at the same points at which instruments are attached to a specimen. These targets can then be tracked over the course of the experiment and their locations can be used to calculate and corroborate the strains or displacements recorded from traditional instrumentation.

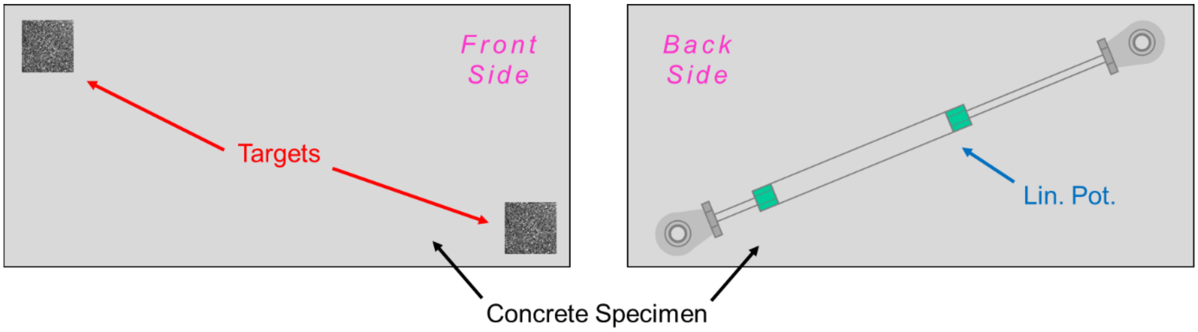


Figure 5.6: Utilizing Targets to Duplicate Traditional Instrumentation

5.2.3 Comparing MODEM to FEM Models Using Targets

Targets can be placed in a grid to create a matrix of known points on a specimen that can then be compared to other software such as FEM models discretized to match the locations of the targets. Four adjacent targets that create a square or rectangle can be used to mimic quadrilateral elements when comparing MODEM to FEM analyses. One drawback to this method is that, since the targets must be placed far enough apart to identify individual targets, the elements themselves will have to be quite large. This will cause the resulting strain fields generated from the targets (which extrapolate internal strain values using shape functions) to be course, resulting in a loss of some localized behavior.

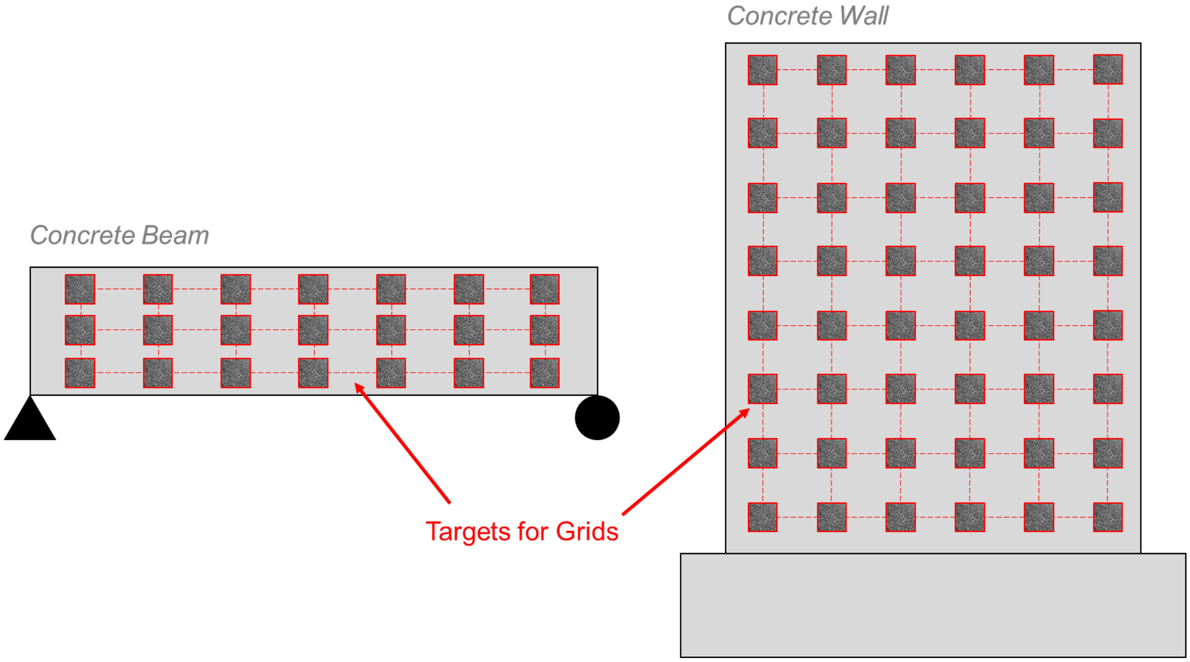


Figure 5.7: Example Target Placements for Grids

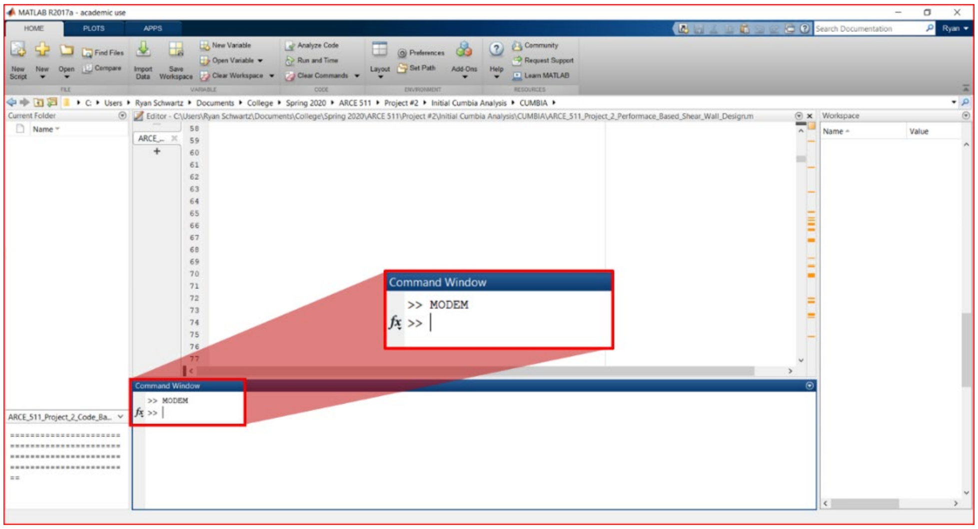
5.2.4 Identifying Movement in a System

Targets can be applied to loading equipment to track any movement in the experimental setup due to external forces such as vibration or accidental movement of the camera. This displacement can then be subtracted globally from all nodes to obtain the true behavior of a specimen.

5.3 2D IMAGE PROCESSING IN MODEM

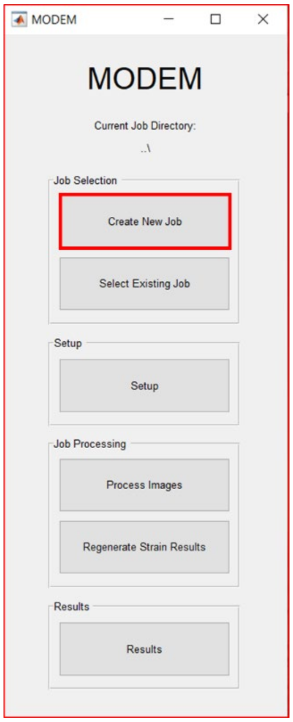
To run an 2D MODEM analysis, do the following:

1. Open the file path containing MODEM in MATLAB's *Current Folder* window.
2. Run MODEM by typing the command "MODEM" into the MATLAB's *Command Window.*

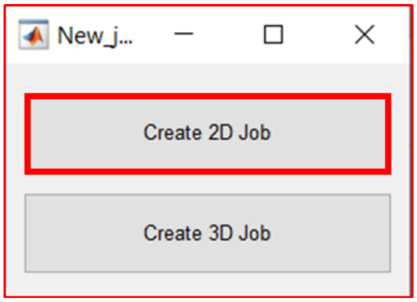


```
>> MODEM
>>
```

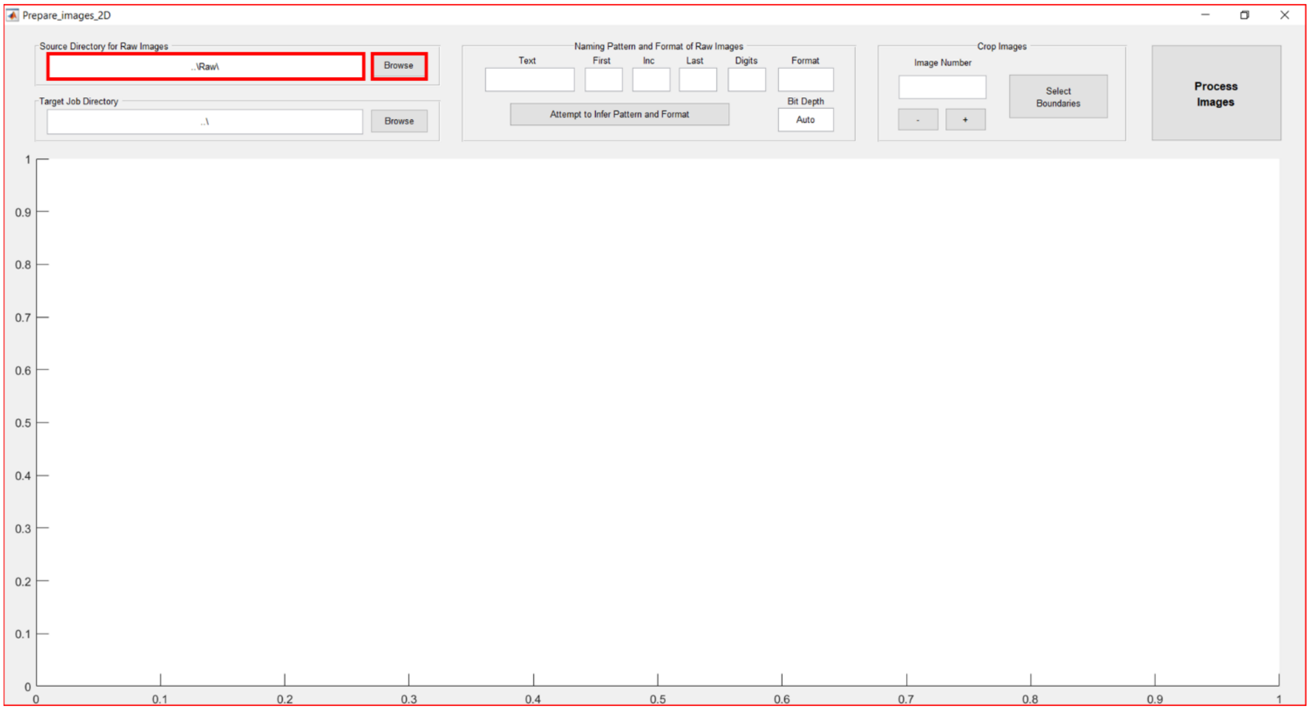
3. The main *MODEM* window will appear. If MATLAB does not register the MODEM command, close and restart MATLAB or repeat Step 1. The command should then work. To run a new analysis in MODEM, select the "Create New Job" button.



4. The New Job Window will appear. Select the “Create 2D Job” button.



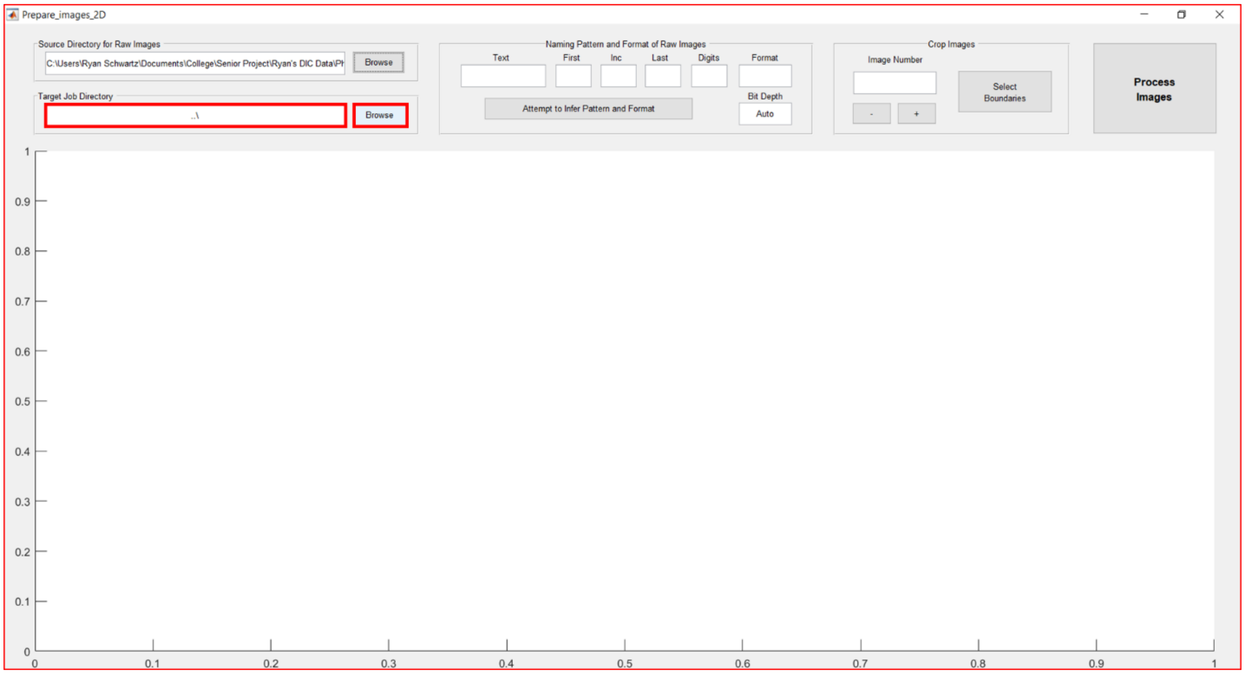
5. In the *Prepare Images 2D* window, input the file path of the folder containing the images being used in the 2D analysis into the bar beneath “Source Directory for Raw Images” or select the “Browse” button next this bar and open the folder containing the image file to be used and press the “Select Folder” button.


Source Directory for Raw Images: ..\Raw\

Target Job Directory: ..\

Bit Depth: Auto

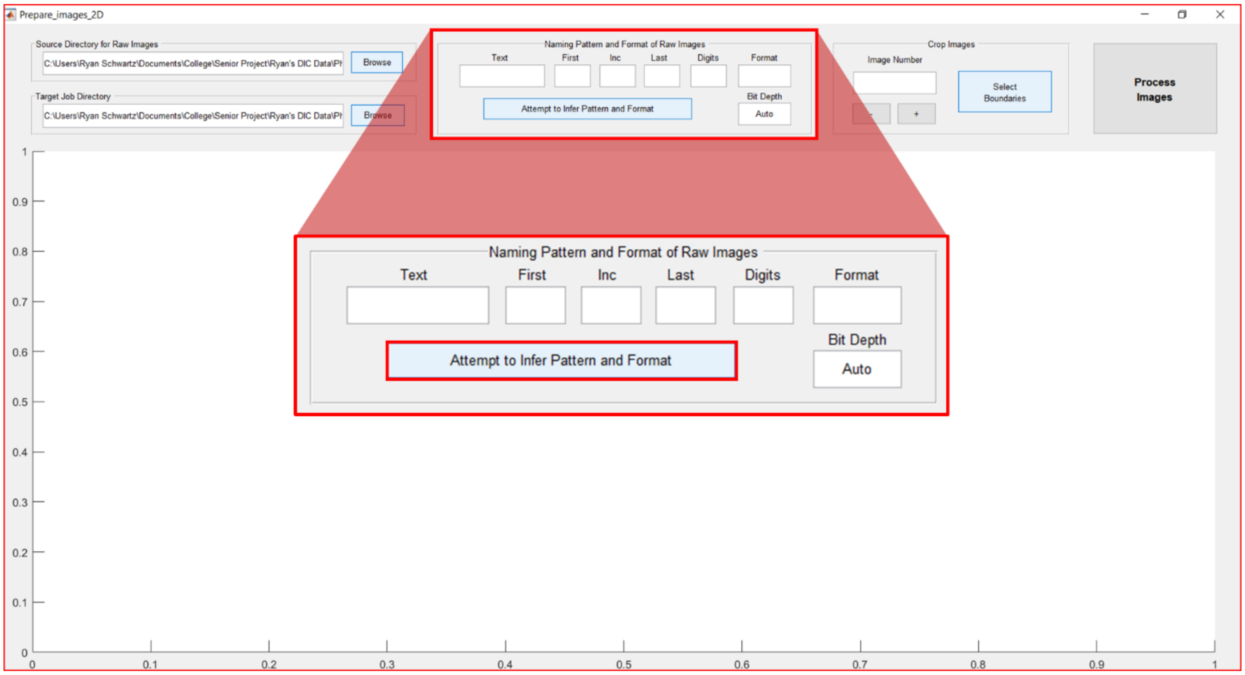
6. Now input the file path of the folder where MODEM should save the results of its 2D analysis into the bar beneath “Target Job Directory” or select the “Browse” button next to this bar and open the folder in which MODEM should save its results and press the “Select Folder” button.



Source Directory for Raw Images: C:\Users\Ryan Schwartz\Documents\College\Senior Project\Ryan's DIC Data\P

Target Job Directory: ..\

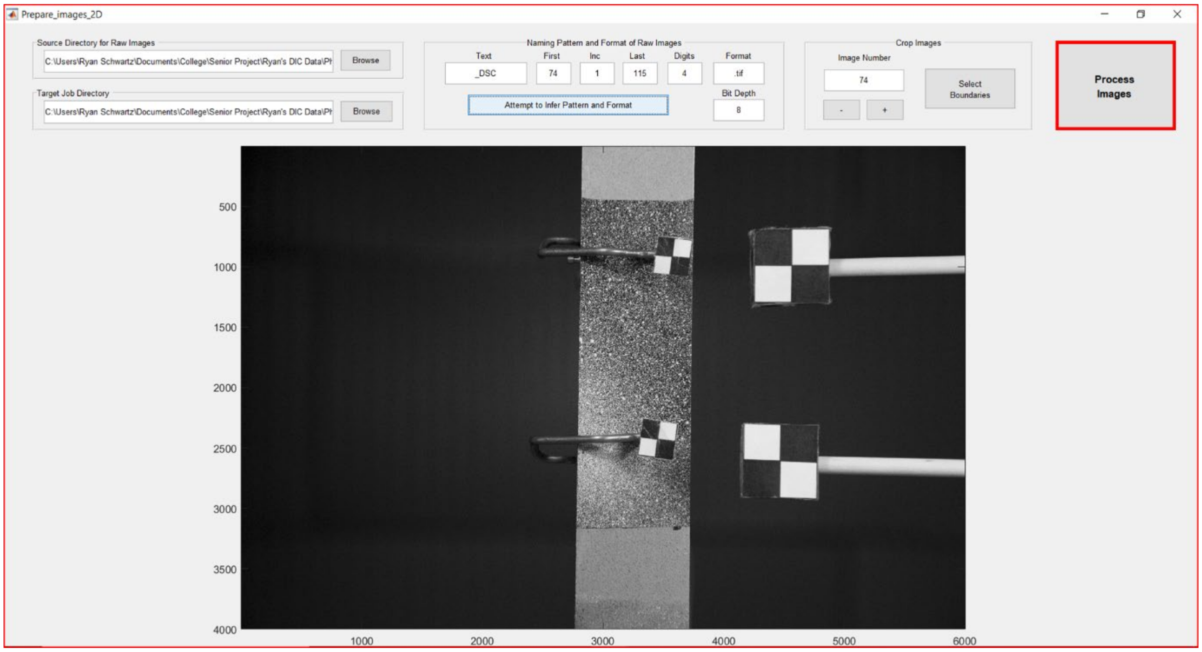
7. Now, select the “Attempt to Infer Pattern and Format” button. This will infer the naming pattern that exists within the set of photos selected and identify all of the photos that are within the folder containing the photos. The naming convention MODEM identifies will autofill the adjacent boxes around the button and the first image within the data set will appear in the plotting area.



Source Directory for Raw Images: C:\Users\Ryan Schwartz\Documents\College\Senior Project\Ryan's DIC Data\P

Target Job Directory: C:\Users\Ryan Schwartz\Documents\College\Senior Project\Ryan's DIC Data\P

8. Select the “Process Images” button.

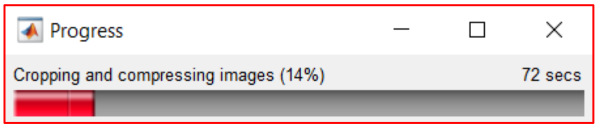


Source Directory for Raw Images: C:\Users\Ryan Schwartz\Documents\College\Senior Project\Ryan's DIC Data\Ph

Target Job Directory: C:\Users\Ryan Schwartz\Documents\College\Senior Project\Ryan's DIC Data\Ph

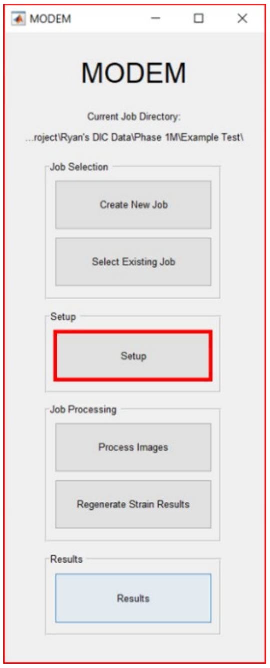
Text: _DSC
First: 74
Inc: 1
Last: 115
Digits: 4
Format: .tif
Bit Depth: 8
Image Number: 74

9. A progress bar should then appear. The commands being run while this window is up should take a minute or two, but possibly more depending on the number of photos in the data set.



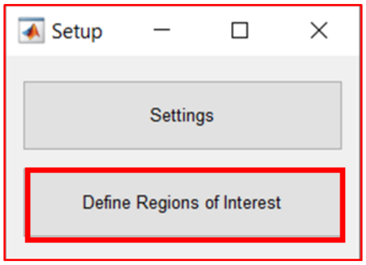
Cropping and compressing images (14%) 72 secs

10. After the computations are completed, the main *MODEM* window will appear again. Select the “Setup” button.

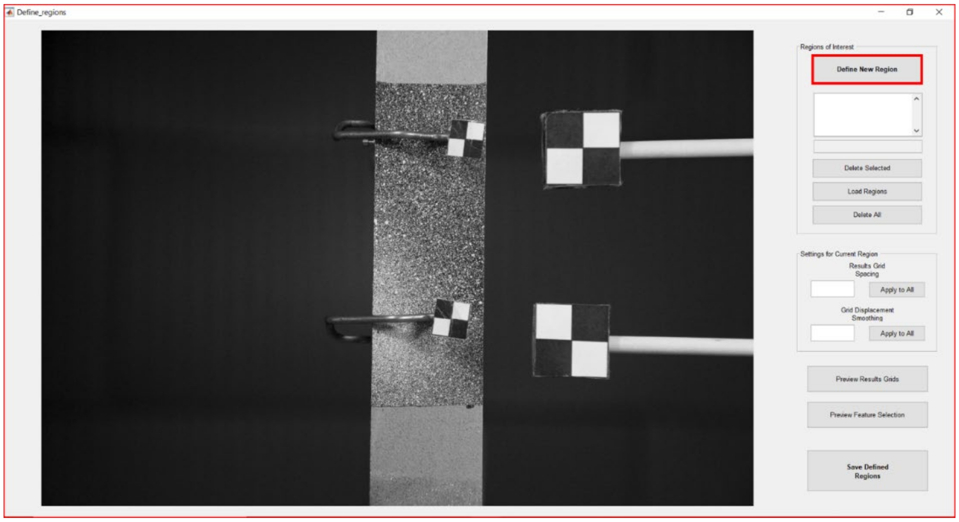


Current Job Directory:
...roject\Ryan's DIC Data\Phase 1M\Example Test\

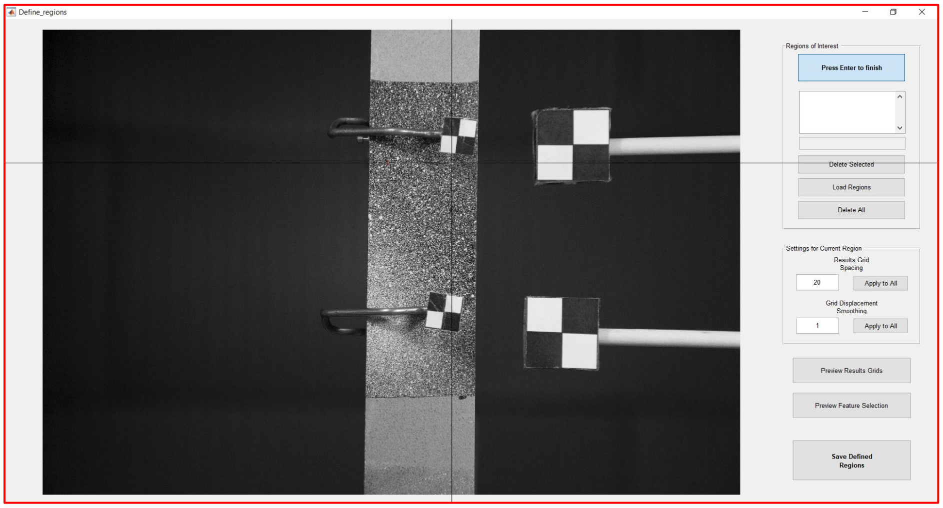
11. Once the *Setup* window appears, press the “Define Regions of Interest” button.



12. After this button is pressed the *Define Regions* window should appear displaying the first photo in the image set. This window may take a minute to appear. After it does, select the “Define New Region” button to start drawing out the area MODEM will analyze.



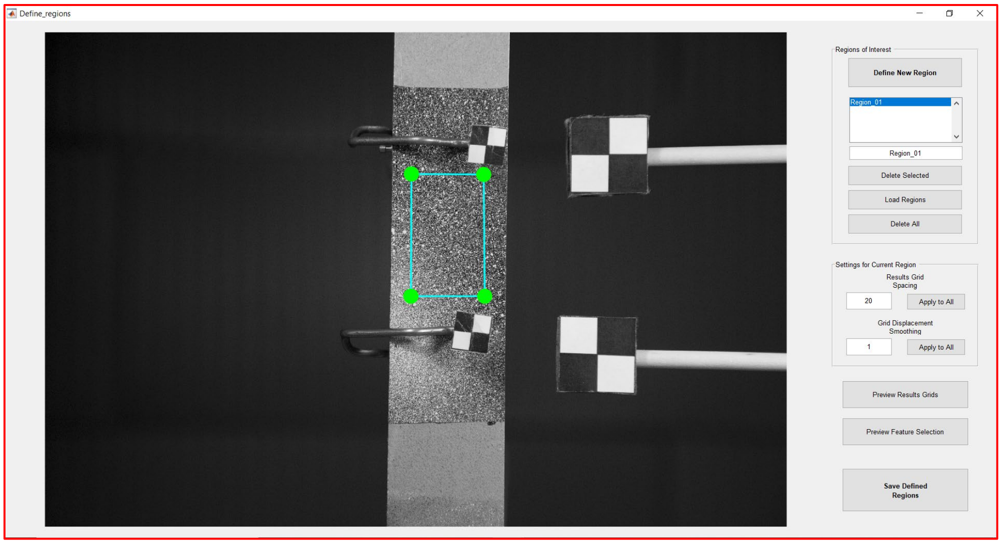
13. Tracking regions are best drawn in rectangles. They are easier to deal with and simple to draw. To draw a tracking region, begin by placing the cursor over the location where the upper right-hand corner of the tracking region should be. Click the mouse to set a node in that position. Next, select the bottom right-hand corner of the tracking region by clicking the mouse in the desired location. Continue selecting corners of the tracking region in a clockwise manner until the upper left-hand corner has been selected again.



Results Grid Spacing: 20

Grid Displacement Smoothing: 1

14. Once all the nodes are placed, press “Enter” on the keyboard. The corners of the tracking region selected will then be outlined in blue with each corner node selected highlighted in green.

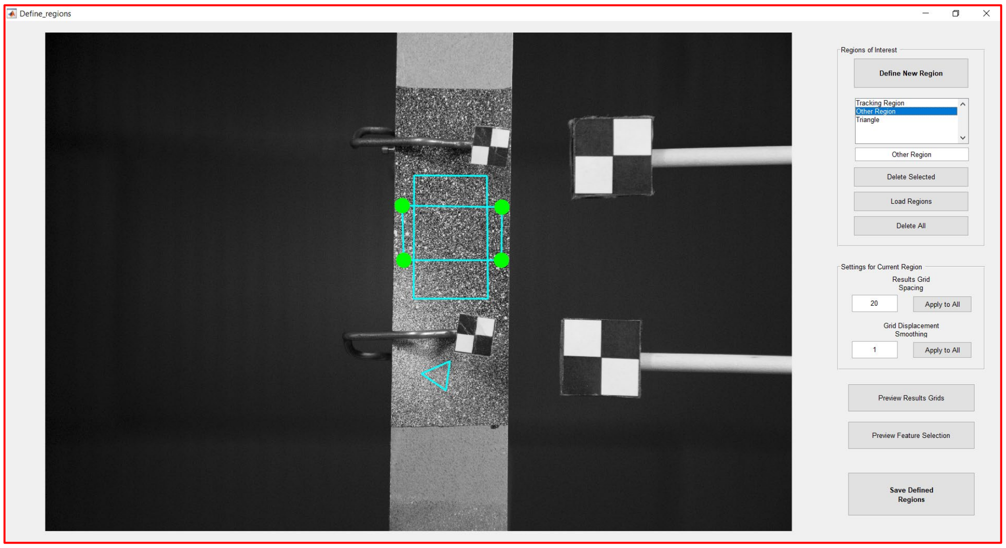


Region_01

Results Grid Spacing: 20

Grid Displacement Smoothing: 1

15. This process can be repeated to analyze multiple regions at once. The regions can overlap and be any shape although rectangular regions are still recommended and preferred.



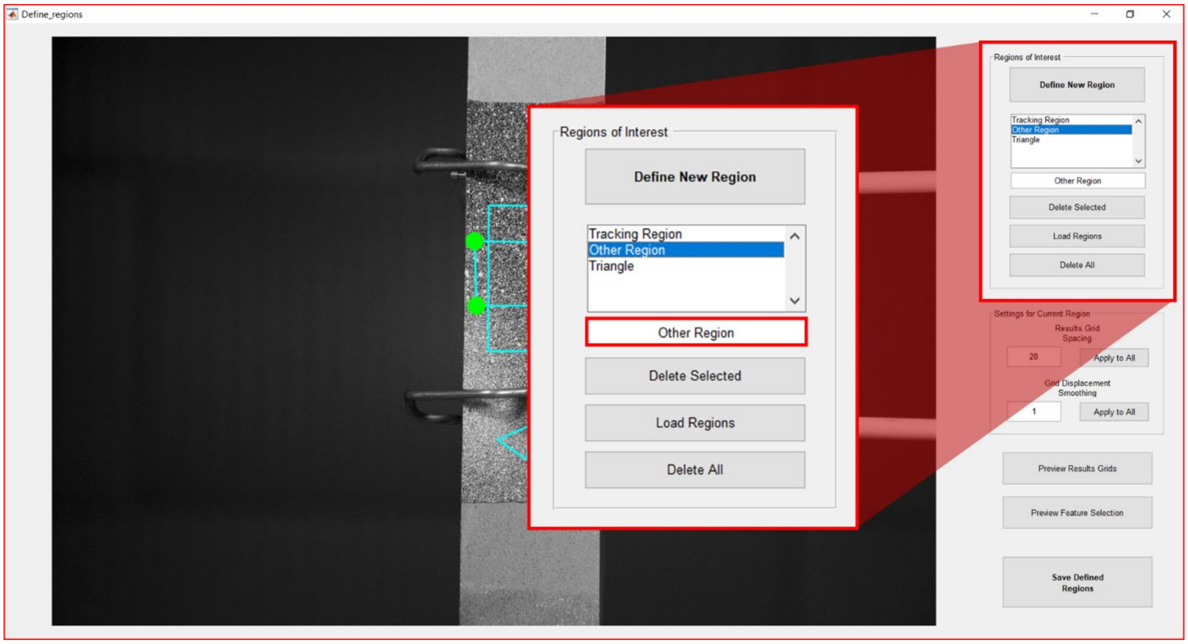
Tracking Region
Other Region
Triangle

Other Region

Results Grid Spacing: 20

Grid Displacement Smoothing: 1

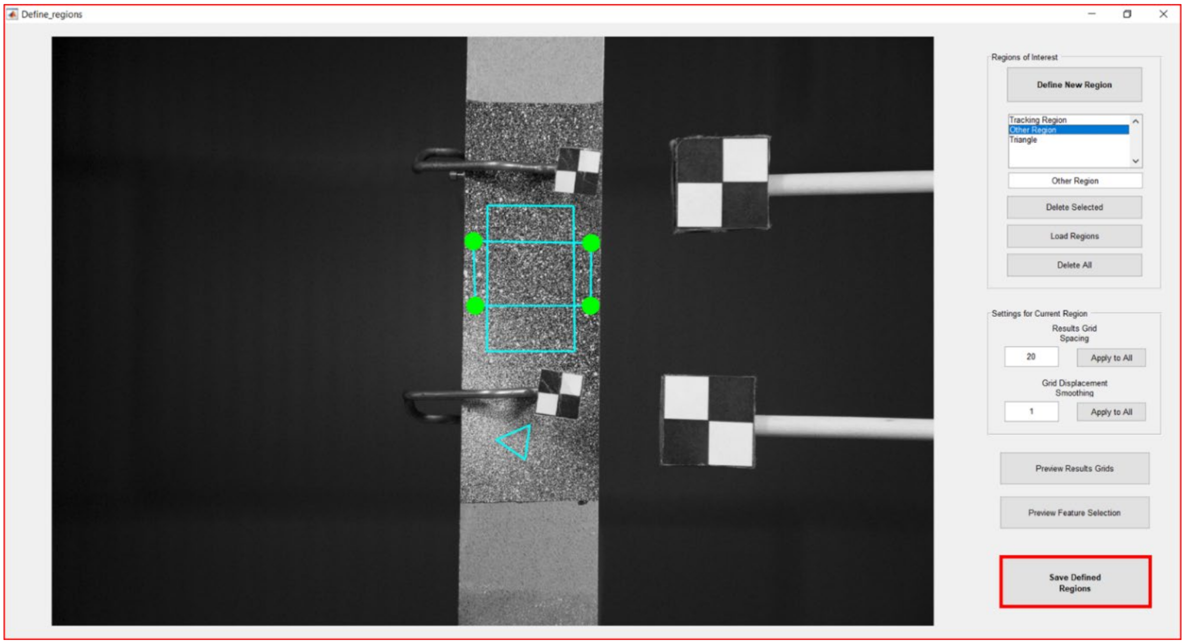
16. The names of the regions can also be changed by selecting the name of the region under the *Region of Interest* section, deleting the existing name in the bar below, and typing in a new name.



Results Grid Spacing 20

Grid Displacement Smoothing 1

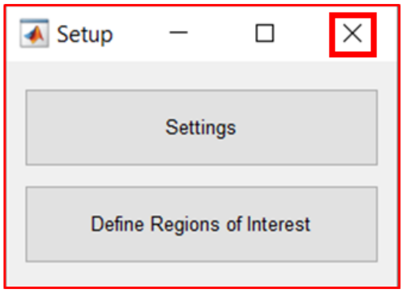
17. After all the tracking regions are drawn, press the “Save Defined Regions” button.



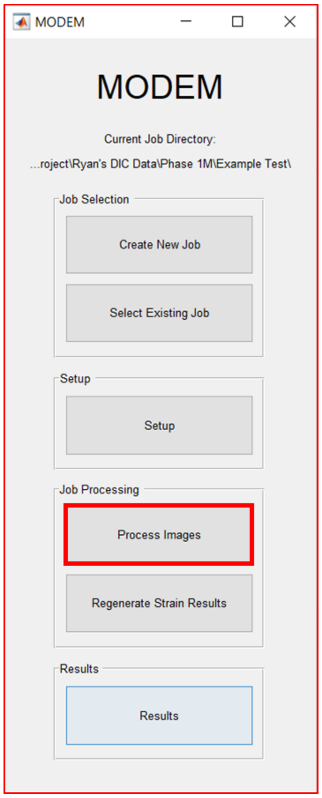
Results Grid Spacing 20

Grid Displacement Smoothing 1

18. The *Setup* window will once again appear. Press the "Close" button to return to the main *MODEM* window.

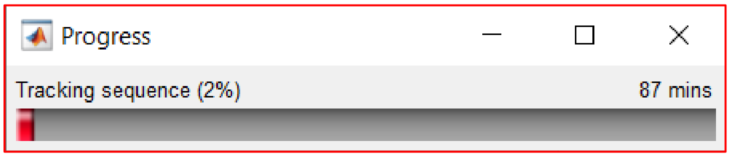


19. Press the "Process Images" to process and analyze the image set of the defined regions selected.



Current Job Directory:
...roject\Ryan's DIC Data\Phase 1M\Example Test\

20. It might take a minute after selecting the "Process Images" button for the progress bar to appear and indicate that MODEM is analyzing the images. This estimated time shown on the progress bar tends to change quickly and is not often accurate.



Tracking sequence (2%) 87 mins

Once the progress bar disappears MODEM is done analyzing the image set and the images are ready to be post processed.

5.4 HOW TO CALIBRATE AN EXPERIMENT

5.4.1 MODEM's Displacement Values

After every 2D analysis, MODEM generates a set of displacements in the x and y directions that measure distance in pixels (explained in Section 5.7.2.3). This matrix of displacement values is not in real world units and does not take into account optical distortion; however, there are multiple methods that can convert this matrix into optically corrected real-world units.

5.4.2 MATLAB Camera Calibration for Distortion Correction

MODEM is capable of correcting for optical distortion by utilizing a calibration file created using *MATLAB's Camera Calibrator App*. This app utilizes a series of photos in which a black and white checkered calibration board is photographed in various positions both in and out-of-plane of the specimen. The objective is to track the intersection of the black and white squares in the checkerboard. Given the dimensions of the calibration board it is possible to determine the location of the specimen in relation to the camera. This data is compiled into a calibration file that can be input into MODEM, which is described in a MathWorks webpage dedicated to explaining how its *Camera Calibrator App* works [18].

This document does not include the evaluation of the accuracy of raw displacements (uncalibrated displacements) compared to calibrated displacements, so users will need to assess this themselves. Also, in its original form, MODEM is not able to upload and utilize calibration files. However, with a few extra lines of code, this can be rectified using the process explained in Section 5.4.2.1. Related to this, instructions for how to create a calibration board can be found in Section 5.4.2.2. Section 5.4.2.3 details how to capture calibration images and Section 5.4.2.4 states how to use these images with the *App* to generate a calibration file. Section 5.4.2.5 specifies how the calibration file can be uploaded to MODEM.

5.4.2.1 CORRECTIONS TO MODEM CODE

MODEM is not, in its original form, able to upload a calibration file created from MATLAB's *Camera Calibrator App* correctly. This section explains the code to allow MODEM to read these calibration files.

The first section of code needs to be placed in line 48 of MODEM's "Run_strain.m" file, located in the *Strain* folder which is nested in the *Core_Code* folder of the main *MODEM* folder (see Figure 5.8).

```
% Loading active matrix
    active = raw_results.active;
```

Figure 5.8: Calibration Code Added to 'Run_strain.m'

The second section of code needs to be placed in line 10 of MODEM's "Correct_coordinates.m" file, located in the same folder as the "Run_strain.m" file (see Figure 5.9).

```
% Load camera parameters from cameraParams
    fc = cameraParams.FocalLength;
    cc = cameraParams.PrincipalPoint;
    kc = [ cameraParams.RadialDistortion(1) , ...
        cameraParams.RadialDistortion(2), ...
        cameraParams.TangentialDistortion(1), ...
        cameraParams.TangentialDistortion(2), ...
        cameraParams.RadialDistortion(3) ];
```

Figure 5.9: Calibration Code Added to 'Correct_coordinates.m'

With these sections of code, MODEM will now be able to read and upload calibration files from *MATLAB's Camera Calibrator App*.

5.4.2.2 CREATE A CALIBRATION BOARD

Calibration photos must be taken by the user before an experiment is conducted to help determine the proper distance between the camera and the specimen as well as the size of the specimen in the camera view. This is done by creating a calibration board that meets the following parameters [18]:

- The checkerboard pattern should be made entirely of squares
- All squares within the board must be the same size in height and width
- All squares should alternate between black and white in both orthogonal directions like a chess board
- The board must be rectangular with one side of the board containing an odd number of squares and the other side containing an even number
- The board should take up approximately 20% of the camera's field of view
- The board must be rigid. If the checkerboard pattern is printed out, then it should be flat with little to no imperfections
- The board pattern should have a matte finish

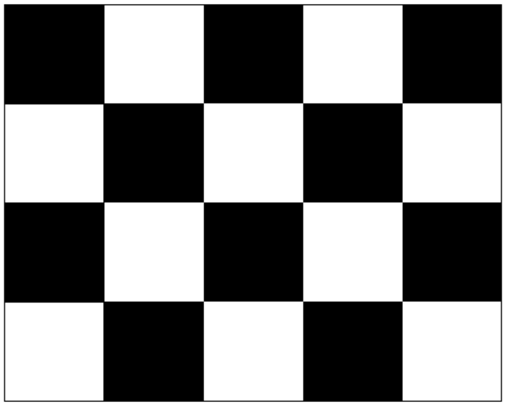


Figure 5.10: Example of a Checkerboard Pattern Used in MATLAB's Camera Calibrator App

5.4.2.3 PROCEDURE FOR TAKING CALIBRATION PHOTOS

When taking calibration photos for MATLAB's *Camera Calibrator App*, around 30 images should be taken. This provides the app with numerous inputs since some images will be rejected. Images should also be an uncompressed or lossless-compression file format like TIFF or PNG to provide the app with more robust images which can also be taken in color.

When taking calibration images, the following recommendations should be used:

1. Complete Steps 1 through 4 from Section 4.2.3.
2. Place the calibration board directly onto or near the specimen being tested. The board, if large enough, can be leaned against or adhered to the specimen with adhesive tape.
3. Set the camera to take TIFF or PNG images (in color or monochrome). Also make sure the white balance is set so the calibration board looks black and white in the images as shown in Figure 5.10. The camera settings (like image size, focal length, and zoom) should remain constant throughout all the calibration images.
4. Measure the distance from the calibration board to the camera. This distance should be parallel to the ground. Also measure the size of the squares that compose the calibration board.

5. Take around 30 calibration images, changing the position of the calibration board in every image. This change should include in-plane (rotation of up to 360°) and out-of-plane rotation (max rotation of 45° out-of-plane). The board should be placed throughout the field of view of the camera. This includes the edges of the camera's field of view to provide the app corrections for optical distortion at the edges of the image.

Some examples of calibration images are shown in Figure 5.11.

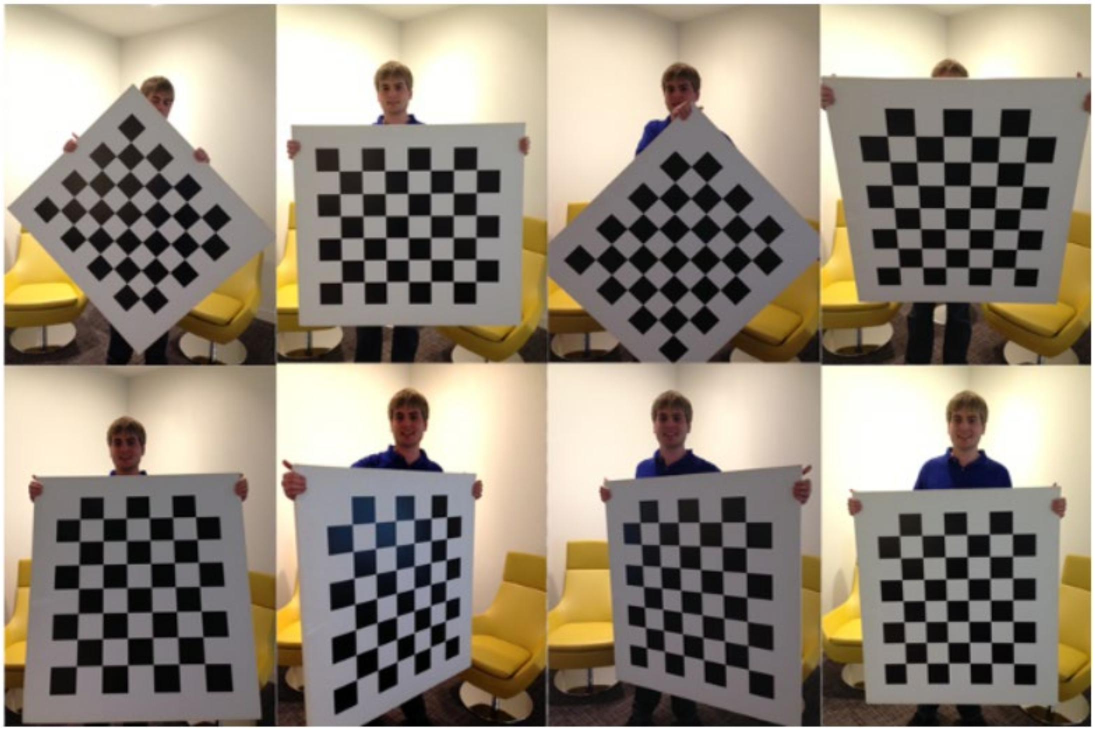


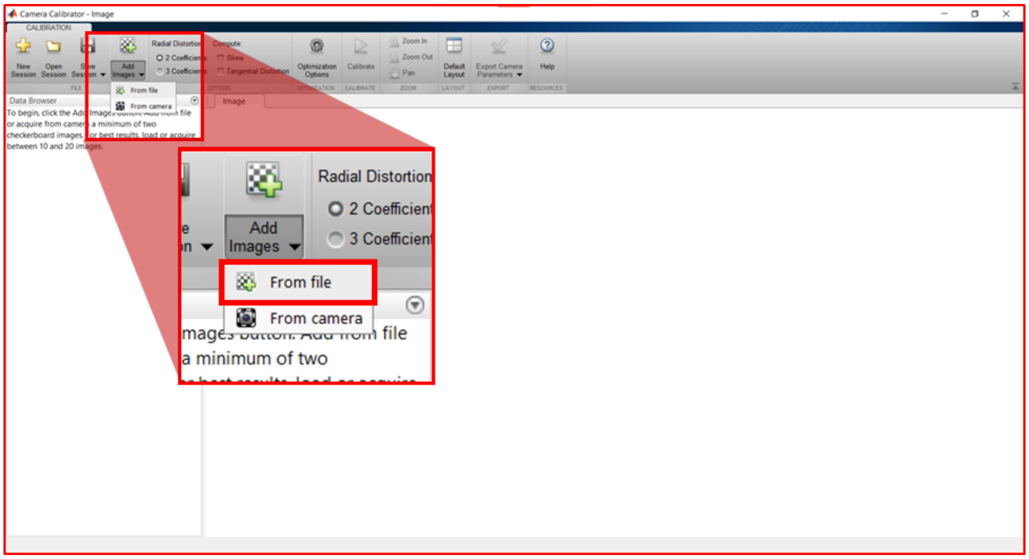
Figure 5.11: Example Calibration Images for MATLAB's Camera Calibrator App [18]

5.4.2.4 PROCEDURE TO USE MATLAB'S CAMERA CALIBRATOR APP

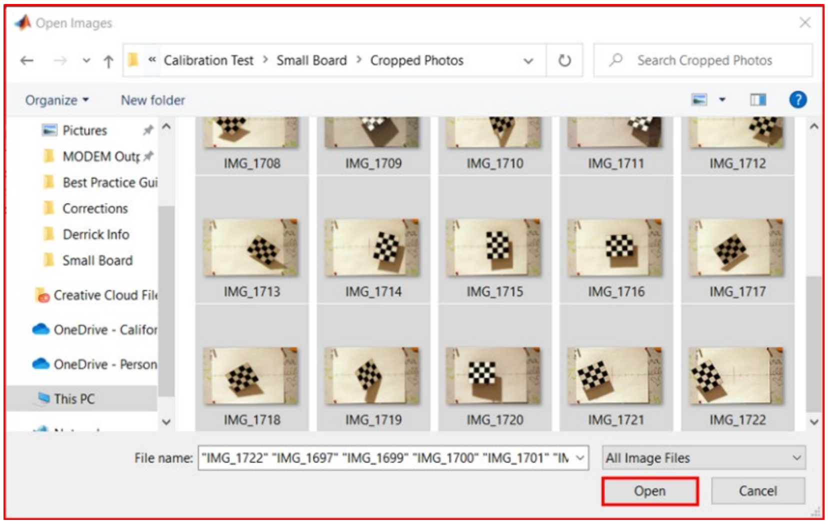
To conduct a calibration using *MATLAB's Camera Calibrator App,* do the following:

1. Open MATLAB's Camera Calibrator App which is located in the *Apps* tab.

2. Open the “Add Images” drop-down menu and select the “From File” button.

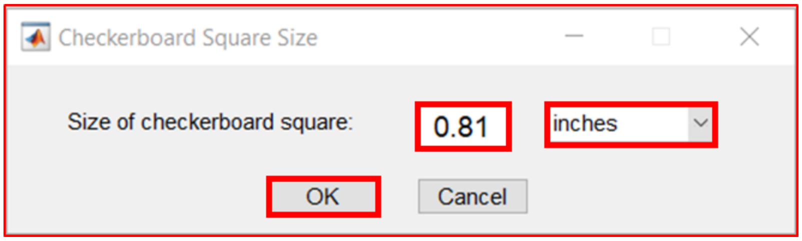


3. Open the file location containing the calibration images taken in Section 5.4.2.3 and select all the calibration images. Then select the “Open” button.



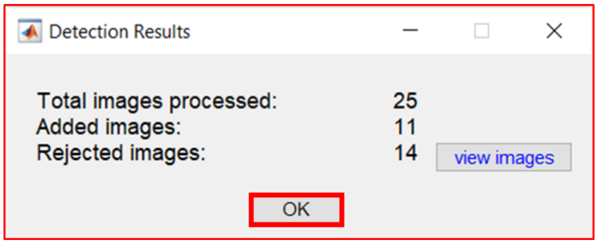
File name: "IMG_1722" "IMG_1697" "IMG_1699" "IMG_1700" "IMG_1701" "IN

4. Once the *Checkerboard Square Size* window opens, insert the size of the checkerboard squares and its units. Once completed, press “OK”.



Size of checkerboard square: 0.81 inches

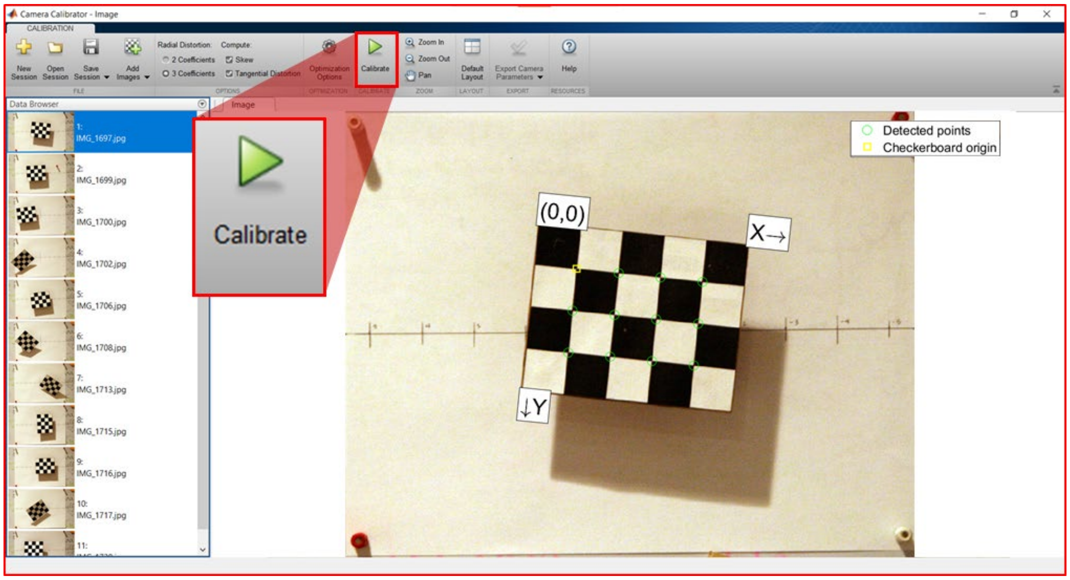
5. Once the *Detection Results* window opens, select the “OK” button.



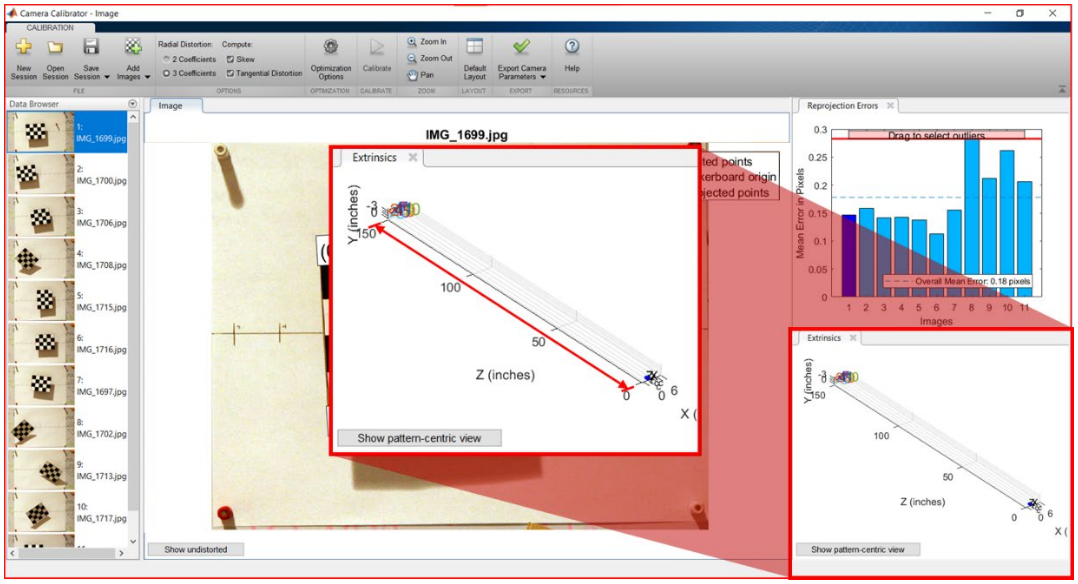
Detection Results

Total images processed:	25
Added images:	11
Rejected images:	14

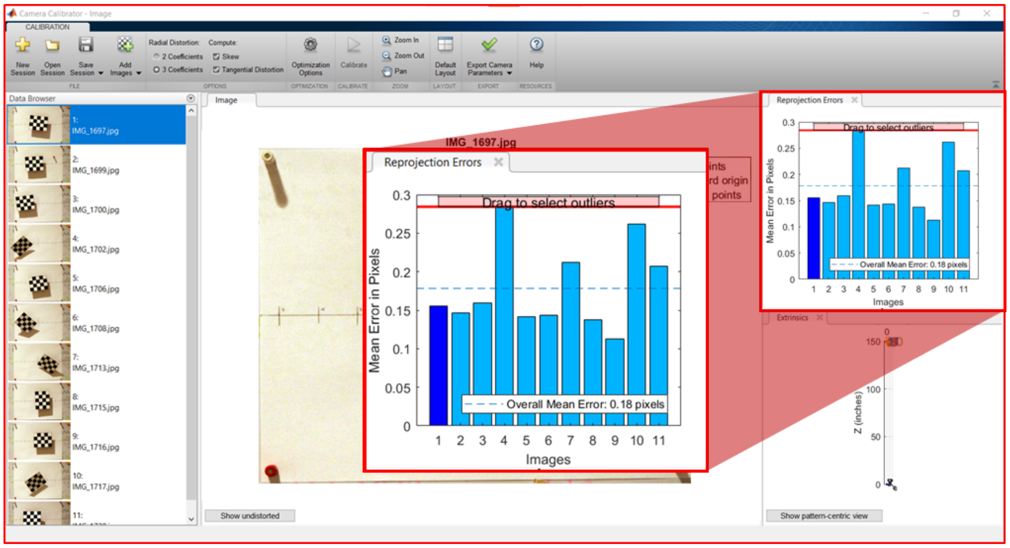
6. Select the “Calibrate” button.



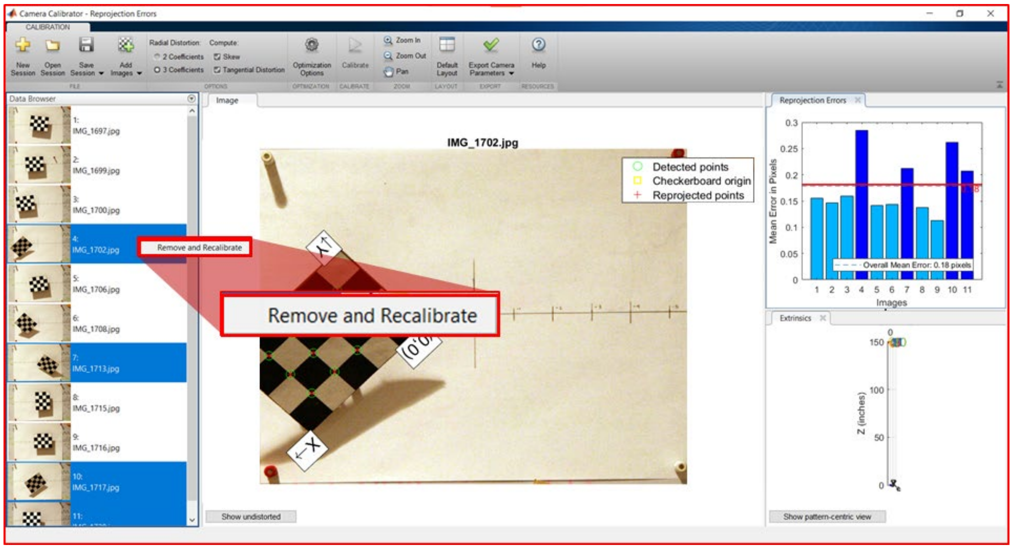
7. Look in the *Extrinsic* window and see, based on the initial calibration conducted, if the distance between the calibration board and the camera is close to what was measured then continue to Step 11. If not, continue to Step 8.



8. In the *Reprojection Errors* window, slide the red bar stating “Drag to select outliers” downward, intersecting the red line with the highest mean reprojection error in the histogram. This bar will become dark blue when highlighted.

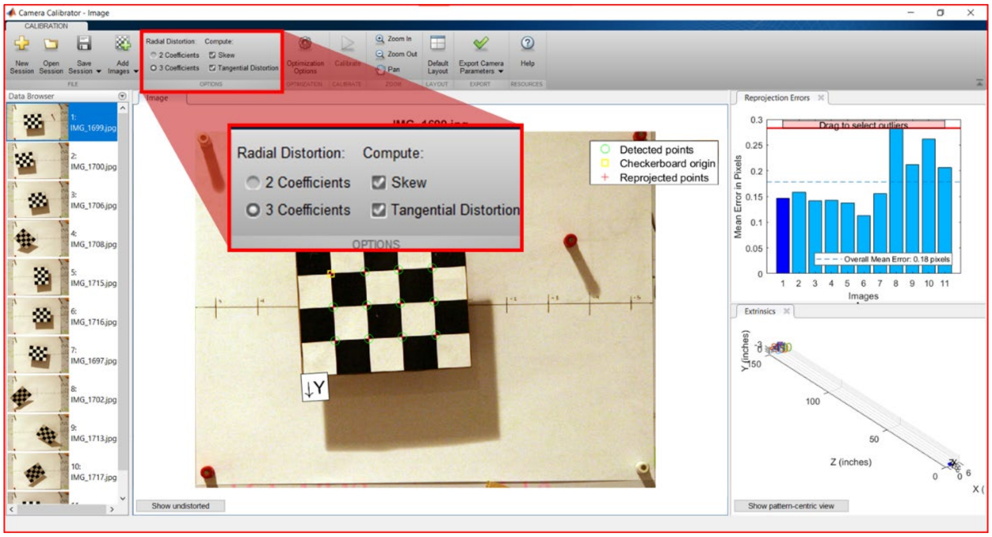


9. The image accompanying the mean error highlighted in the *Reprojection Errors* window will also be highlighted in the *Data Browser* window on the left. Right click on this highlighted image and click the “Remove and Recalibrate” button. It is recommended that around ten images are used in a calibration so be aware of how many images are being removed. The minimum number of images needed is three.



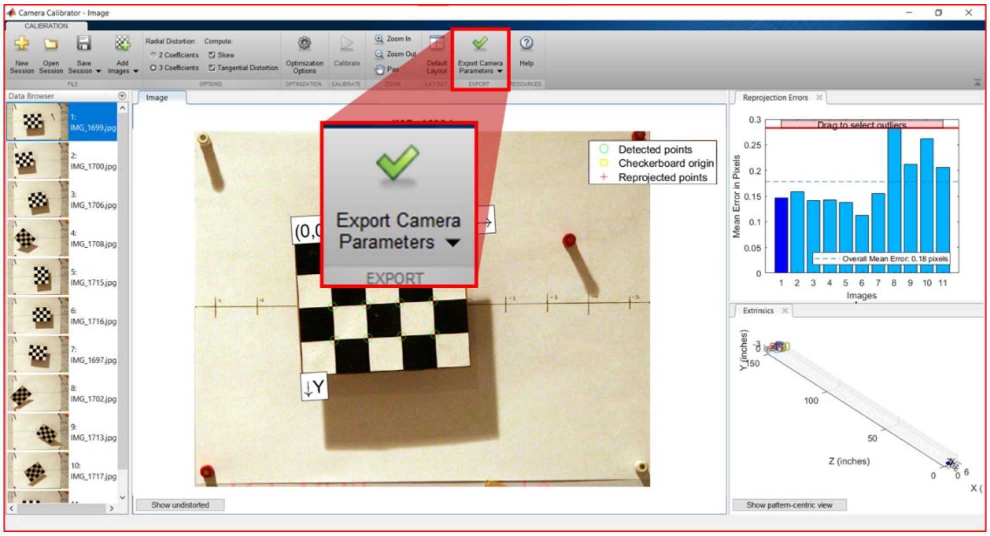
10. Repeat Step 7.

11. In the *Options* section of the *Calibration* tab, select the “3 Coefficients” button under *Radial Distortion* and select the “Skew” and “Tangential Distortion” check boxes under *Compute*.

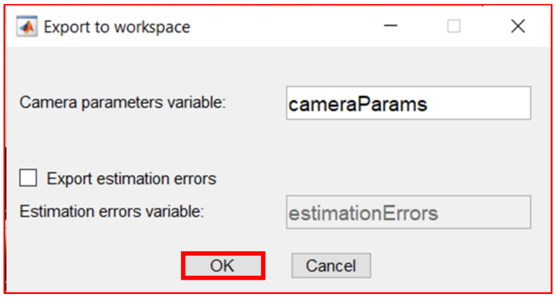


Radial Distortion: Compute:
○ 2 Coefficients ☑ Skew
● 3 Coefficients ☑ Tangential Distortion

12. Select the “Export camera parameters” drop down menu and select the “Export camera parameters” button.



13. Select the “Ok” button in the *Export to workspace* window.



Export to workspace

Camera parameters variable: cameraParams

☐ Export estimation errors

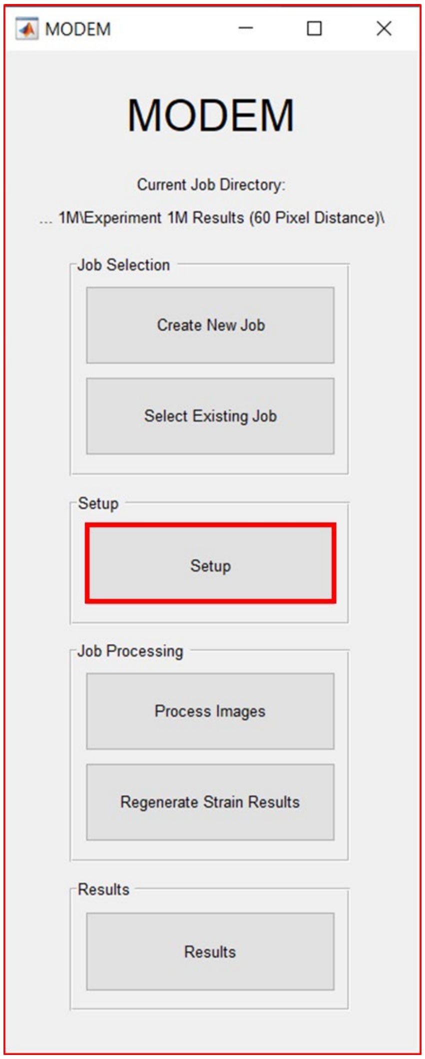
Estimation errors variable: estimationErrors

14. Right click on the “cameraParams” file in the Workspace and select the “Save as” button in the drop-down menu. Save the calibration file wherever the user finds most suitable.

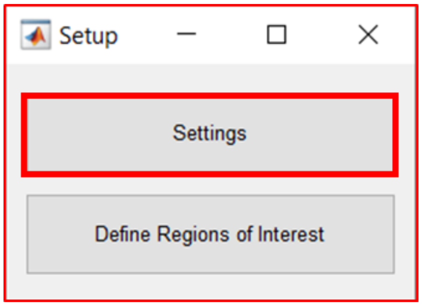
5.4.2.5 HOW TO UPLOAD CAMERA CALIBRATION APP TO MODEM

To upload a calibration file made in MATLAB’s Camera Calibrator App to MODEM, do the following:

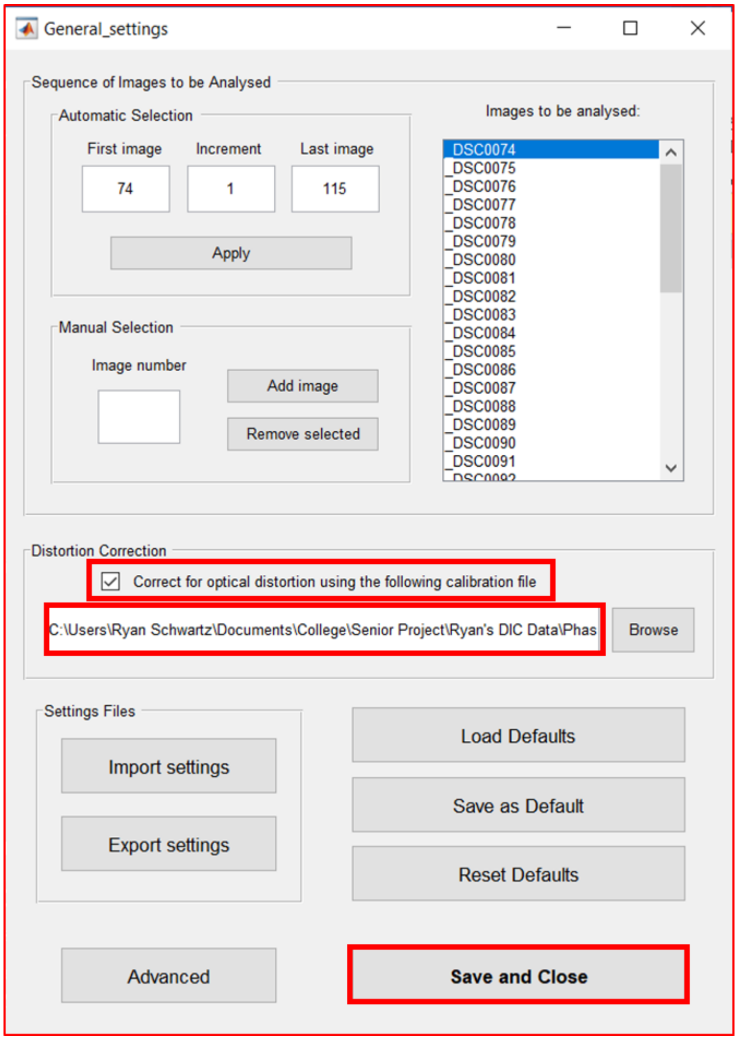
1. Conduct a 2D MODEM analysis as specified in Section 5.3.
2. Select the “Setup” button in *MODEM* window.



3. Select the “Settings” button in the *Setup* window.



4. Under the “Distortion Correction” section of the *General_settings* window, select the checkbox “Correct for optical distortion using the following calibration file.”



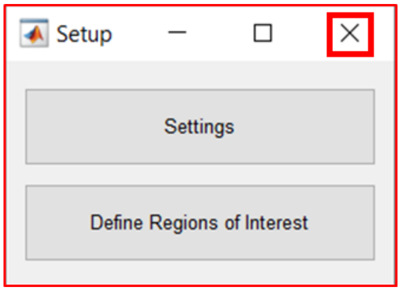
First image: 74
Increment: 1
Last image: 115

DSC0074
DSC0075
DSC0076
DSC0077
DSC0078
DSC0079
DSC0080
DSC0081
DSC0082
DSC0083
DSC0084
DSC0085
DSC0086
DSC0087
DSC0088
DSC0089
DSC0090
DSC0091

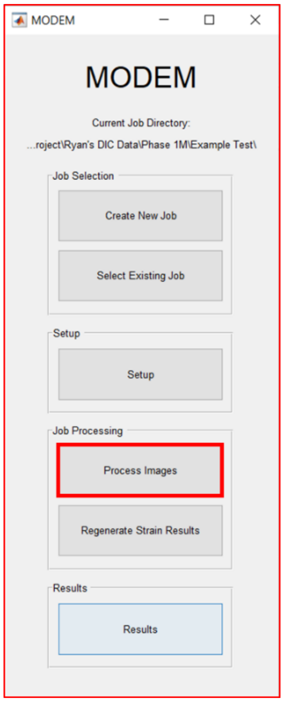
☑ Correct for optical distortion using the following calibration file

C:\Users\Ryan Schwartz\Documents\College\Senior Project\Ryan's DIC Data\Phas

5. Insert the file path for the calibration file generated in Section 5.4.2.4 in the text box under “Distortion Correction.” This file path can be acquired in Windows computers by right-clicking on the calibration file and selecting the “Properties” button. Then, once the *Properties* window opens, copy the file location under the “Location” label. After inputting the file path, select the “Save and Close” button.
6. Select the “Close” button in the Setup window.



7. Select the “Process Images” button in the *MODEM* window.



Current Job Directory:
...roject\Ryan's DIC Data\Phase 1M\Example Test\

MODEM will now have computed calibrated displacement values which can be found in the displacement matrix explained in Section 5.7.2.3. The uncalibrated displacements can be found in the displacements_raw matrix explained in Section A.1.2.3.2.

5.4.3 Manual Calibration to Generate Displacements with Real World Units

The displacement values MODEM computes are not in terms of real world units, but are instead in terms of pixels. So, to acquire MODEM displacement values in real world units (in, mm, etc.), the displacements values within the displacements matrix must be multiplied by a conversion factor. This factor (shown in Equation 2), can be determined by using the following steps:

1. Before the experiment begins, determine the real-world distance between two known points in an experiment (e.g. distance between the center of two static targets).
2. Determine the same distance from Step 1 in pixels by analyzing the reference image in a photo-editing software like Adobe Photoshop.
3. Divide the distance measured in Step 1 by the distance measured in Step 2.
4. Multiply the factor computed in Step 3 by the displacements matrix.

$$displacements\ (real - world\ units) = \frac{distance\ in\ real - world\ units}{distance\ in\ pixels} * displacements\ (pixels)$$

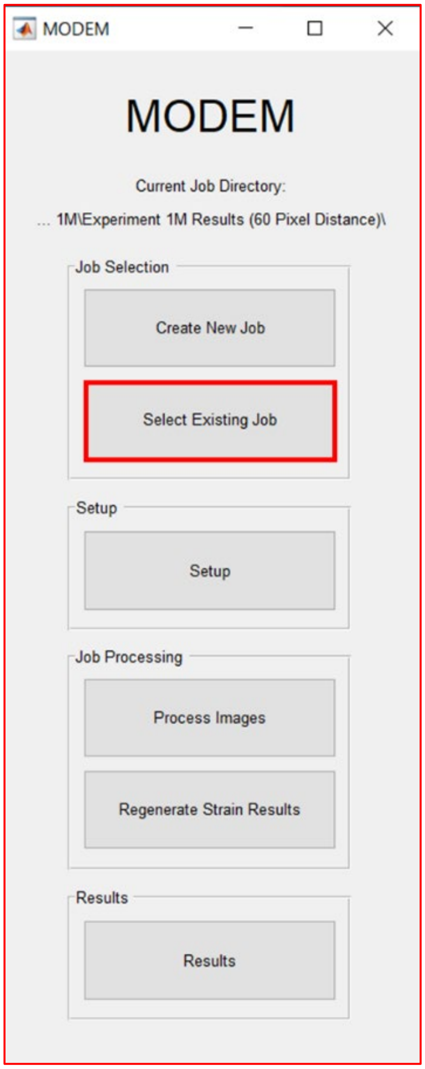
Equation 2: Hand-Calibration Equation

Following these steps provides approximate real-world displacements and is completely dependent on the accuracy of the conversion factor and whether it holds for the entire image. If a specimen is photographed at an angle, the calibration factor may not be valid across the entire specimen. Therefore, in using this calibration method, it is recommended that the specimen is perpendicular to the camera. Also, comparing MODEM displacements to traditional instrumentation was not within the scope of this document so this analysis must be conducted by the user as well as how accurate calibrated MODEM displacements are.

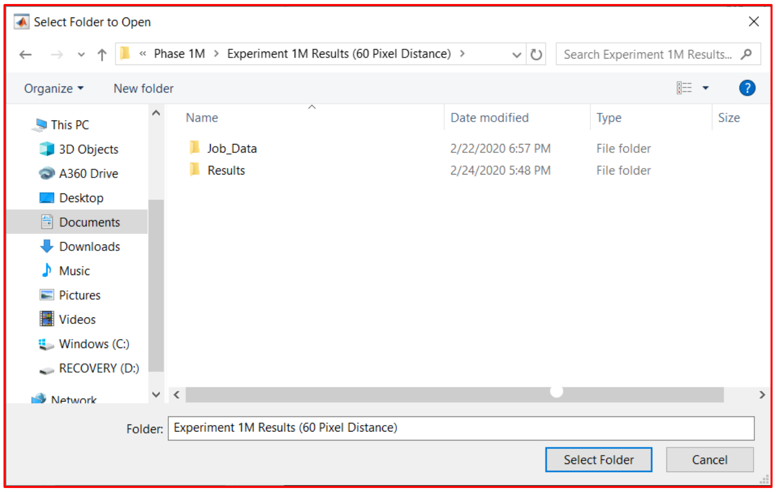
5.5 OPENING AN EXISTING 2D ANALYSIS IN MODEM

To open an existing 2D MODEM analysis, do the following:

1. Open MODEM, as referenced in Step 1 in Section 5.3.
2. Once the *MODEM* window opens, click the “Select Existing Job” button.



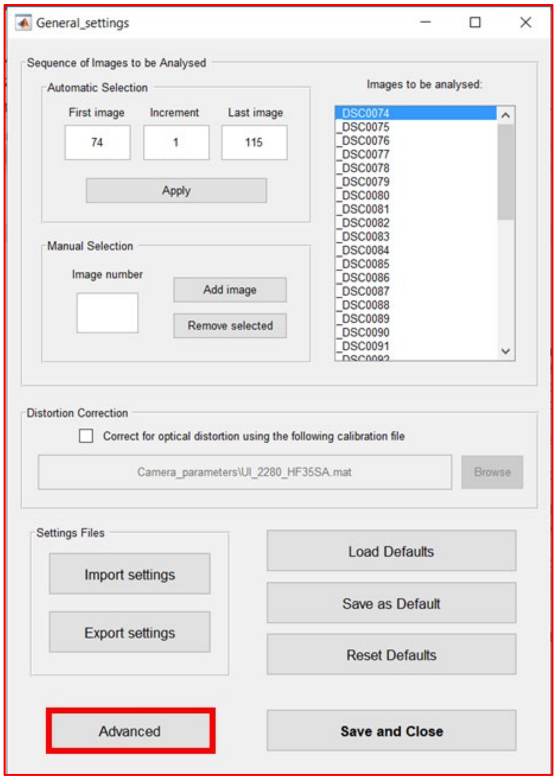
3. Select the file location where the “Job Data” and “Results” folder are for the experimental data that is to be reopened.



Name	Date modified	Type	Size
Job_Data	2/22/2020 6:57 PM	File folder	
Results	2/24/2020 5:48 PM	File folder	

Folder: Experiment 1M Results (60 Pixel Distance)

4. The files needed to analyze the 2D MODEM analysis can now be accessed by MODEM.
    a. A new experiment can be conducted based on the photos selected by choosing new tracking regions (the process for which is shown in Section 5.3 Steps 9 through 14) or by altering the settings used and then conducting a normal 2D MODEM analysis like that explained in Section 5.3. These settings can be found under the “Settings” button in the *Setup* window (as seen in Step 11 of Section 5.3). These settings consist of the general settings in the *General_settings* window and the advanced settings which are accessed by clicking the “Advanced” button in the *General_settings* window. These settings were not explored when creating this document so a researcher will have to determine their uses and values on their own.

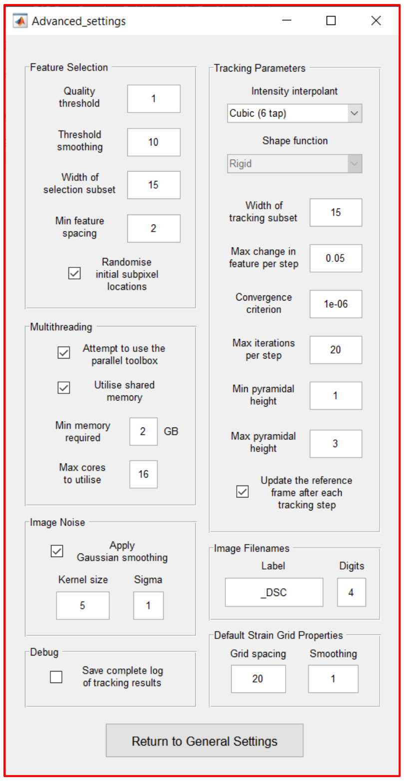


First image: 74
Increment: 1
Last image: 115

Images to be analysed: _DSC0074, _DSC0075, _DSC0076, _DSC0077, _DSC0078, _DSC0079, _DSC0080, _DSC0081, _DSC0082, _DSC0083, _DSC0084, _DSC0085, _DSC0086, _DSC0087, _DSC0088, _DSC0089, _DSC0090, _DSC0091, _DSC0092

☐ Correct for optical distortion using the following calibration file

Camera_parameters\UI_2280_HF35SA.mat



Quality threshold: 1
Threshold smoothing: 10
Width of selection subset: 15
Min feature spacing: 2
☑ Randomise initial subpixel locations
☑ Attempt to use the parallel toolbox
☑ Utilise shared memory
Min memory required: 2 GB
Max cores to utilise: 16
☑ Apply Gaussian smoothing
Kernel size: 5
Sigma: 1
☐ Save complete log of tracking results
Intensity interpolant: Cubic (6 tap)
Shape function: Rigid
Width of tracking subset: 15
Max change in feature per step: 0.05
Convergence criterion: 1e-06
Max iterations per step: 20
Min pyramidal height: 1
Max pyramidal height: 3
☑ Update the reference frame after each tracking step
Label: _DSC
Digits: 4
Grid spacing: 20
Smoothing: 1

b. The experiment accessed can also be reopened by selecting the “Results” button in the *MODEM* window. This will access and open all the data previously computed for this experiment. This same image set can also be reprocessing by selecting either the “Process Images” button or the “Regenerate Strain Results” button (shown in Section 5.3 Step 9). Selecting “Process Images” will reselect the tracking subsets and recompute the strains and displacements. Selecting “Regenerate Strain Results” will only recompute the strains and displacements of the tracking subsets previously identified the last time the image set was processed. From here the results of the tracking can be accessed and contours/videos can be made by selecting the “Results” button and following the process shown in Section 5.8.

5.6 PARAMETRIC STUDY OF NODE SPACING

In every MODEM analysis, as described in Section 5.7.2.2, the spacing for the nodes is user-defined. The variability of this spacing can greatly influence the results of a MODEM analysis. Each node represents the averaged strain and displacement values of the tracking subsets around it. If the node spacing is too small, the resulting strain and displacement fields can experience a phenomenon known as periodicity as shown in Figure 5.12. Periodicity causes the fields to exhibit fluctuating values of a seemingly sinusoidal nature that create inaccuracies at low strains [3].

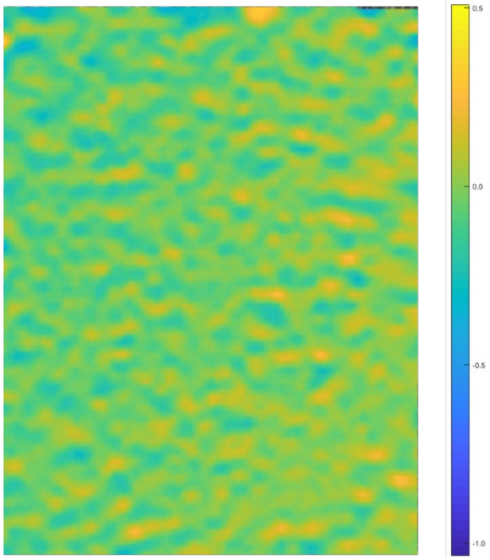


Figure 5.12: Periodicity in a Strain Field in MODEM Due to a 10 Pixel “Results Grid Spacing”

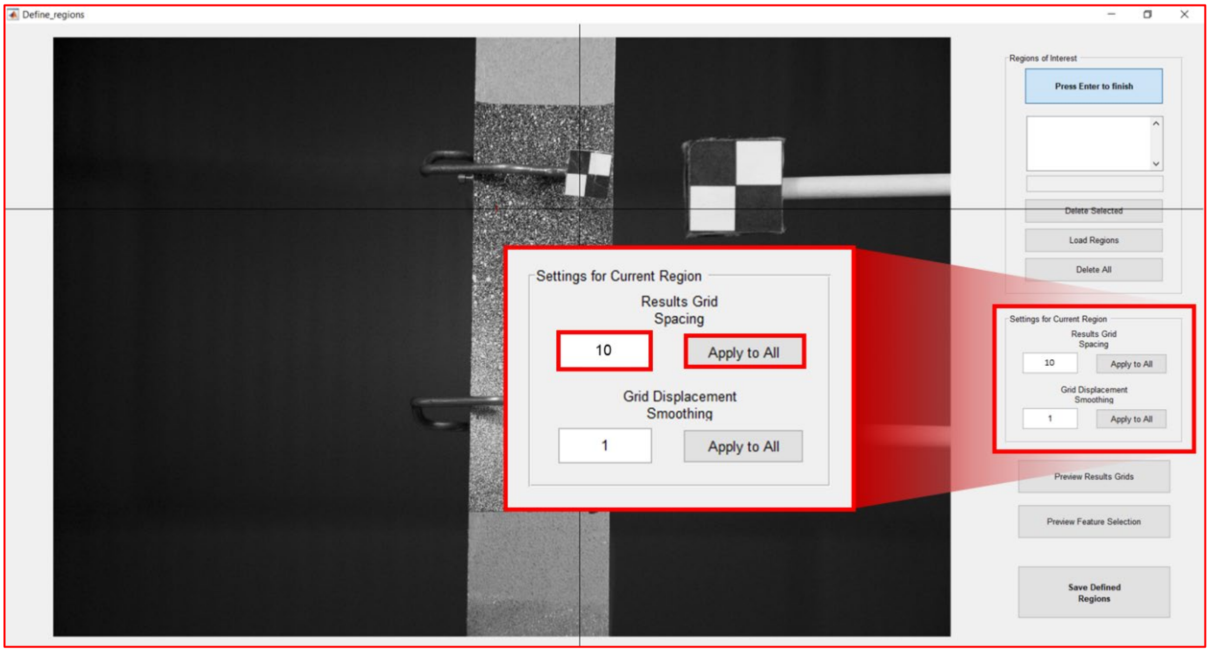
If a node spacing is too far apart then each node will average too many tracking subsets for each node. This can lead to results that are too averaged which can mask and obscure localized behavior, similar to what would happen in an FEM model that is not discretized enough. Thus, the best way to prevent either of these situations is to conduct a parametric study of the node spacing to determine the correct distance. This is done by varying the “Results Grid Spacing” factor in MODEM when selecting tracking regions in the *Define_regions* window (see Section 5.3 Step 13). The procedure for conducting this study is explained in Section 5.6.1.

5.6.1 Procedure for Conducting a Parametric Study of Node Spacing

To conduct a parametric study of nodal spacing, do the following:

1. Follow steps 1-5 found in Section 5.3 to create a 2D MODEM analysis.

2. At step 6 in Section 5.3, after selecting the “Attempt to Infer Pattern and Format,” make the “Last” image the number after the auto-filled “First” image number which makes the image set being analyzed a total of two images.
3. Follow steps 7-12 found in Section 5.3 to finish conducting a 2D MODEM analysis.
4. At step 13 in Section 5.3, select a “Results Grid Spacing” distance, starting at 10 pixels.



Results Grid Spacing: 10
Grid Displacement Smoothing: 1

5. Follow steps 14-16 found in Section 5.3 to finish conducting a 2D MODEM analysis.
6. Repeat steps 1-5 in Section 5.6.1, adding 10 pixels to the “Results Grid Spacing” variable and creating a new 2D analysis. Repeat this step until a new 2D analysis has been generated for every “Results Grid Spacing” from 10 pixels to 100 pixels with a 10-pixel increments.
7. Follow the procedure in Section 5.8.1, generating an x or y normal strain contour plots for each 2D analysis generated in step 6.
8. Once all ten of the 2D analyses are conducted, open all of the contours produced during each separate analysis. These contours can be found by opening the “Results” folder and then the “Contours” folder and then selecting the generated contour image. Compare all of the contours side-by-side as shown in Figure 5.13. After comparing the contours side-by-side, determine which “Results Grid Spacing” produces the least periodicity while also utilizing the smallest spacing.

5.6.2 Analyzing the Results of a Node Spacing Parametric Study

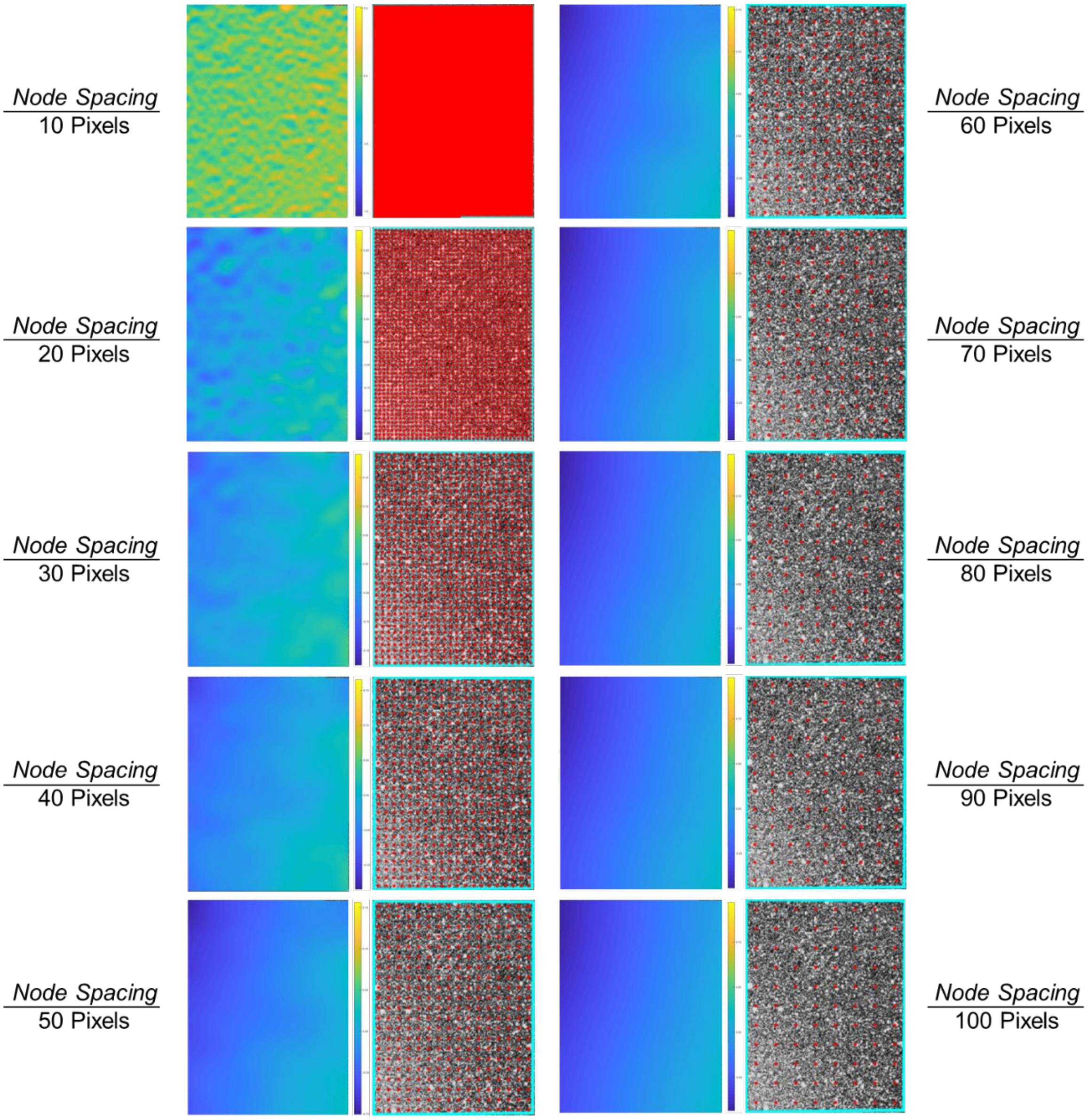


Figure 5.13: Example Pixel Spacing Study (Strain Contour [left] and Node Locations [right])

When analyzing the results of a node spacing parametric study, it is important to locate which spacings create periodicity. In Figure 5.13, this would be the node spacings equaling 10 to 40 pixels. From there, a spacing needs to be selected that displays smooth transitions between the different strain/displacement values. This is exhibited in the contours as a smooth gradient. For the example in Figure 5.13, this would include the nodes spacings equaling 60 to 100 pixels. From these spacings, the smallest node spacing should be taken as the optimal one for the experiment, in this case a node spacing of 60 pixels.

5.7 POST-PROCESSING / EXTRACTING DATA

5.7.1 MODEM's Output Structure

MODEM's output structure is a complex and extensive network of nested MATLAB data structures. It is filled with variables, matrices, and logicals that are helpful for designing and implementing the software but are not as helpful when determining what data necessary for understanding the results of a MODEM analysis of a structural experiment. Therefore, this section will explain how to dissect MODEM's output and determine which pieces of data are essential for understanding any MODEM analysis.

When any MODEM analysis is run, two separate folders are created in the user specified memory location "Target Job Directory" specified when the analysis is being initiated: *Job_Data* and *Results*. The folders and structures that exist within these folders are shown in Figure 5.14 and Figure 5.15.

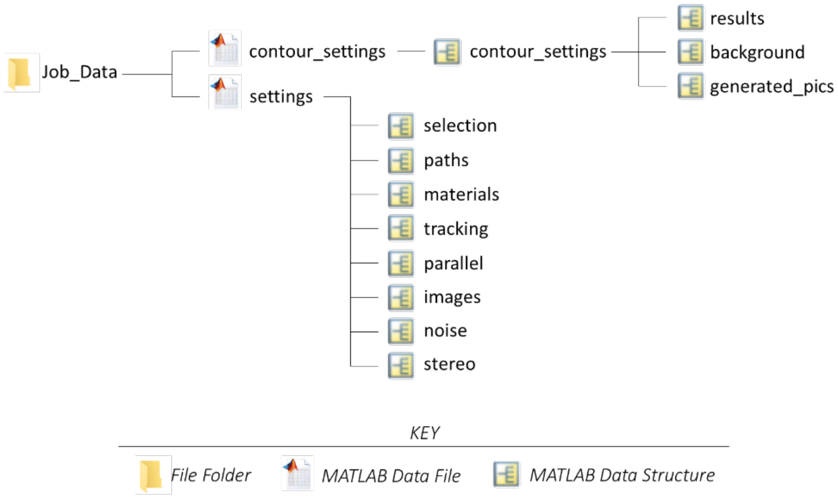


Figure 5.14: Job_Data Folder Output Diagram

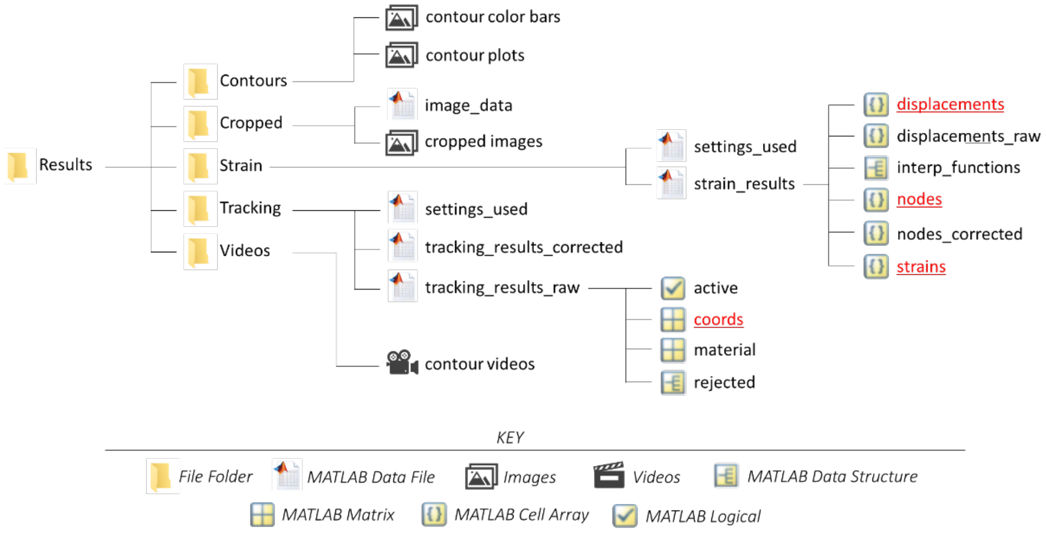


Figure 5.15: Results Folder Output Diagram

5.7.2 Required MODEM Output Files

Only four files are needed from the two folders shown in Figure 5.14 and Figure 5.15 to acquire the data generated from any MODEM analysis. These four files are: coords, nodes, displacements, and strains (as shown in red in Figure 5.15).

5.7.2.1 COORDS MATRIX

The coords matrix is a three-dimensional matrix that contains the locations of the coordinates, or features, that MODEM identifies from the speckle pattern within the user-defined tracking regions. Indexing through the Z dimension of this matrix provides the user with the coordinates for all the features identified by MODEM for a given image from the experiment's image set. For example, the first matrix in the coords file (indexed under *coords(:,:,1)*) contains the locations for all of the features in the first image in the image set MODEM analyzed. The ninth matrix in the coords file (indexed under *coords(:,:,9)*) contains the locations for all of the features in the ninth image of the image set MODEM analyzed.

The first column of the coords matrix (*coords(#,1,#)*) contains the x-coordinates for all the features in a given image while the second column (*coords(#,2,#)*) contains the y-coordinates for all the features of a given image. In order to reference a specific feature's coordinates for a given image, simply index through the first dimension of any given matrix, for tracking subset 23 from image 7 simply type in *coords(23,:,7)* into the command window of MATLAB. To gain a more visual understanding of what is contained within the coords matrix, reference Figure 5.16

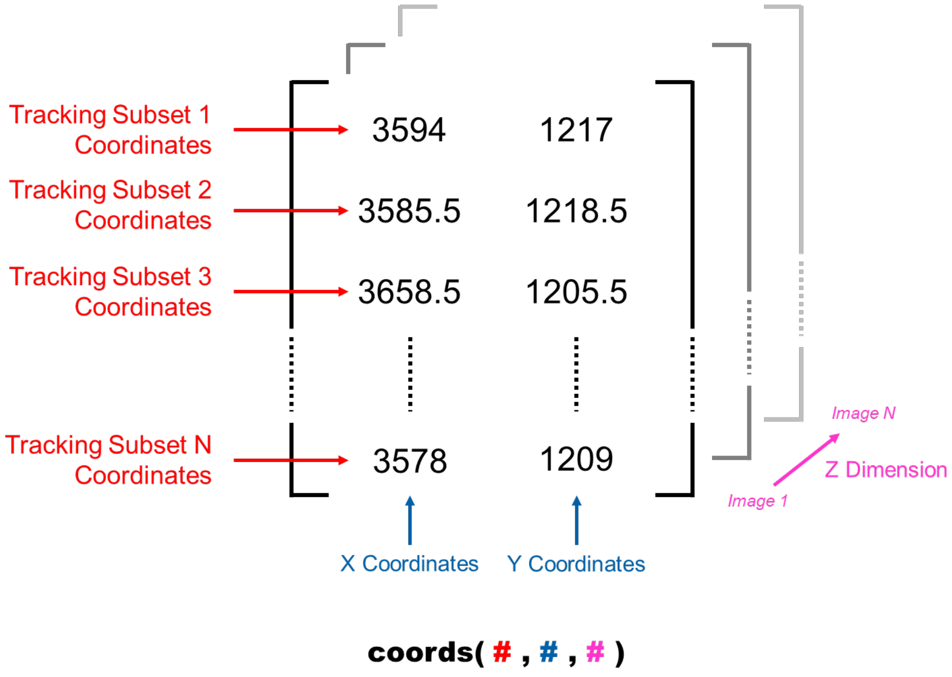


Figure 5.16: Anatomy of the "coords" Matrix

When plotting these coords, care has to be taken since the locations of the coords are made in reference to an image's origin and, in MATLAB and MODEM, the origin of an image is in the upper left corner as illustrated in Figure 5.17. This means that, when plotting the y-coordinates of any tracking subset without overlaying them on an image, the y-coordinates either have to be negated or the y-axis itself has to be flipped in MATLAB. If the y-coordinates are being plotted over a picture, however, the origin will be automatically flipped so there no need to alter the y-coordinates of the tracking subset. The units for the values in the coords matrix are in pixels.

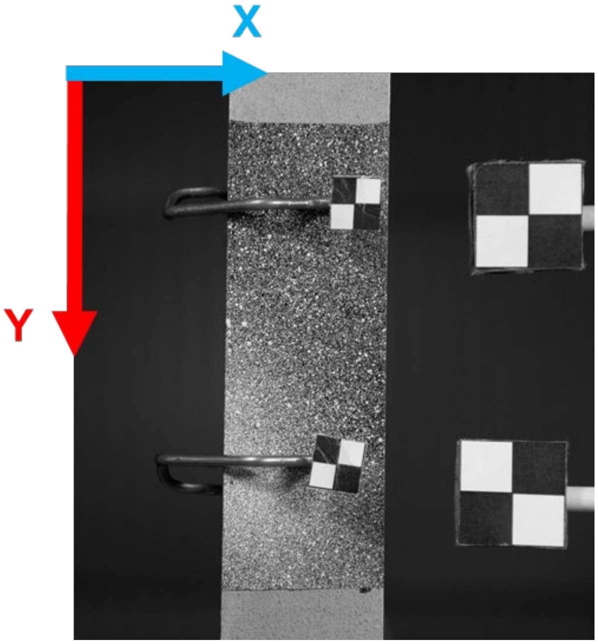


Figure 5.17: MODEM Coordinate Systems for 2D Analysis

5.7.2.2 NODES CELL ARRAY

The nodes cell array contains the coordinates of all of the nodes specified by the user by changing the node spacing (same as the variable "Results Grid Spacing"). This variable dictates how far each node is from all nodes adjacent to it in pixels. These nodes are utilized by the user to minimize the amount of data points that need to be managed for a given tracking area in an experiment. MODEM allows the user to define an overlaying grid of points above the coords where strains and displacements for each feature are averaged to their closest adjacent node. This develops a grid of points that can replace the coords, which are user defined and far fewer in number. Their grid format allows users to compare the results computed in MODEM to results computed using FEM modeling software as well as convert to values that might be seen in traditional instrumentation.

As a comparison, Figure 5.18 visualizes the coords determined for an experiment in comparison to the nodes selected for that same experiment. In Figure 5.18a, the "Results Grid Spacing" was 60 pixels. Figure 5.18a displays the nodes selected while Figure 5.18b displays the coords determined by MODEM for the first image. Since the positions of the nodes are made in reference to an image's origin, the nodes must be plotted in the same way as the coords.

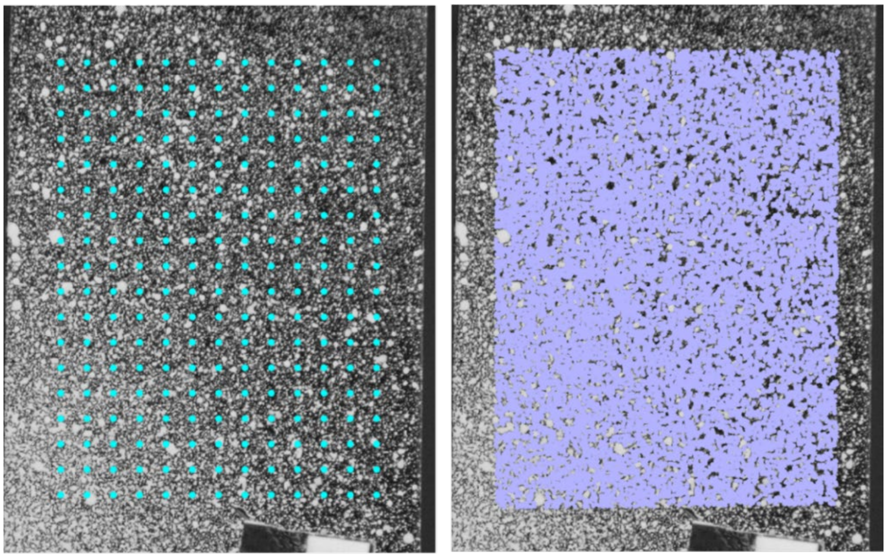

(a) (b)

Figure 5.18: "coords" and "nodes" Plotted on First Image of Experiment 1M

5.7.2.3 DISPLACEMENTS MATRIX

The displacements matrix is 3D matrix that contains the displacement of the nodes made in relation to the reference image (first image) in the image set. These displacements are measured in pixels just like the node location are defined in pixels. The displacements matrix will always be one Z-dimension smaller than the total number of images in the image set since the first image is a reference image. Also, the Z-dimension for the displacement matrix is made in reference to the image number less than one, meaning the displacements for image five can be indexed by using *displacements(-,-,4)*.

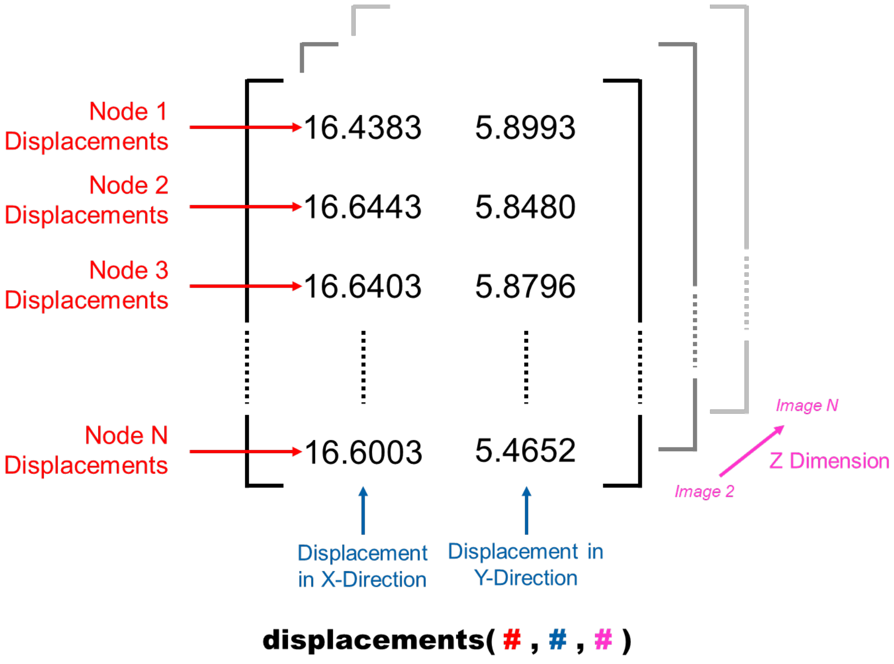


Figure 5.19: Anatomy of the "displacements" Matrix

The indexing for the displacements matrix is shown in Figure 5.19. In order to acquire a node's position for a given image in the image set, add the node's displacement at that given image to the nodes original position in the nodes cell array. Figure 5.20 gives an example of plotting a node's displacement for the fifth image in an image set.

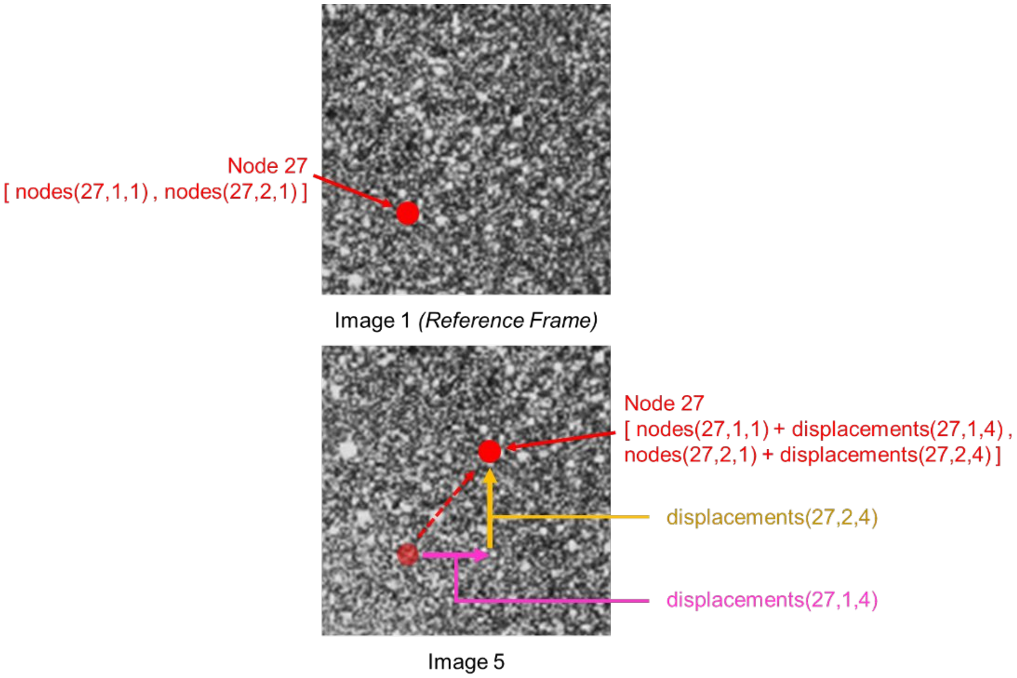


Figure 5.20: Utilizing "displacements" Matrix to Determine Node Locations in Another Image

5.7.2.4 STRAINS MATRIX

The strains matrix is 3D matrix that contains the strains for each node for every image in the image set except the reference frame. Like the displacements matrix, there is one less Z-dimension in the strains matrix than the number of images in the image set. This also means that the Z-dimension for the strains matrix is in reference to the image number minus one. To obtain the twenty-seventh node's strains from the fourth image in the image set, for example, "strains(27,:,4)" should be input into the command window.

The strain values within the strains matrix are computed by deriving the displacements of the nodes which have been computed from the displacements of the adjacent coords. These strains are then averaged to a single value which is assigned to every given node. Thus, strain is no longer a change in distance between two points in MODEM but a change in distance at a point.

For 2D analyses, three different strain values are calculated by MODEM: normal strain in the x-direction, normal strain in the y-direction, and shear strain along the xy plane. The positive sign conventions used by MODEM to generate these strains is shown in Figure 5.21.

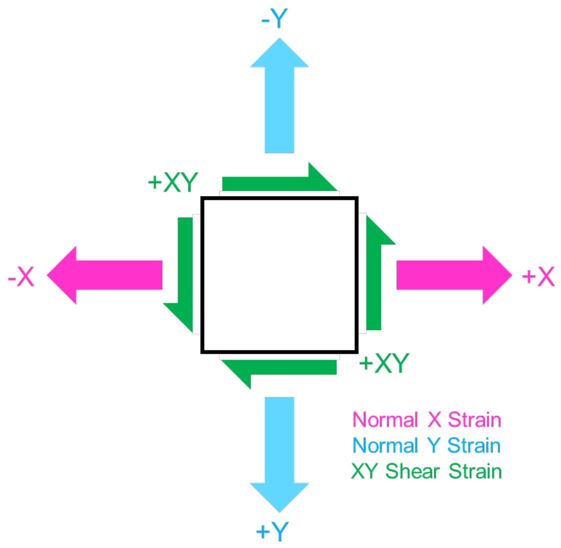


Figure 5.21: Positive Sign Conventions for Strains in MODEM

For the strains matrix, normal strain in the x-direction is located in the leftmost column, the normal strain in the y-direction is located in the central column, and the shear strain in the xy plane is located in the rightmost column. A visual interpretation of the strains matrix is shown in Figure 5.22.

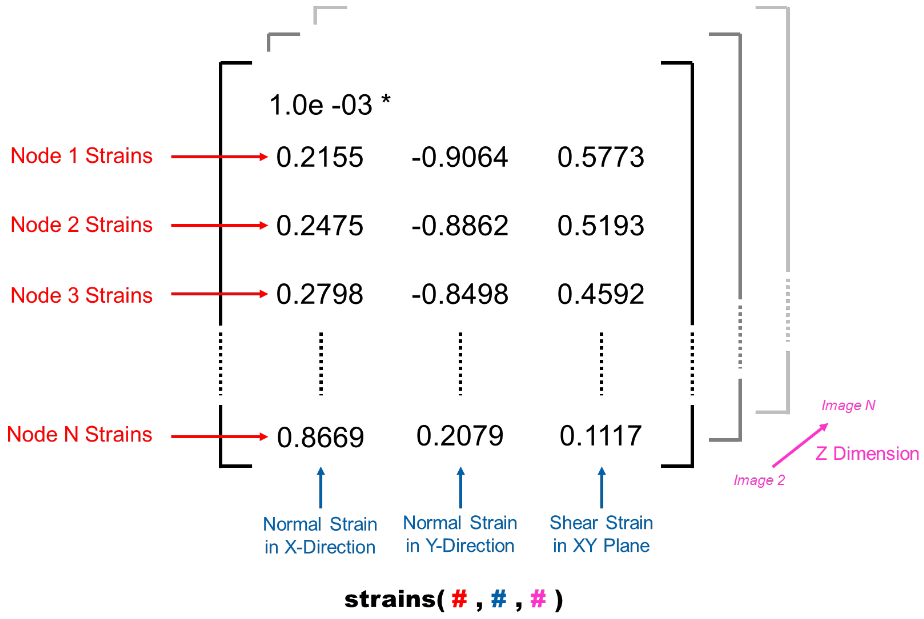


Figure 5.22: Anatomy of "strains" Matrix

5.7.3 Extracting Required Output Files

5.7.3.1 EXTRACTING DATA FROM MODEM OUTPUT TO MATLAB

Based on the format of the Results folder shown in Figure 5.15, a simple MATLAB script (Figure 5.23) was created to acquire all four files referenced in Section 5.7.2 from MODEM's output structure. To run this script, all a user needs to do is to open a given experiment's file location (the location where an experiment’s “Job_Data” and “Results” folders are located) in MATLAB’s Current Folder directory and then execute this script.

```
%-------- Loading 'Results' Folder --------
cd Results
cd Strain
%Loading displacements matrix
load('strain_results.mat','displacements');
displacements = displacements{1,1};
%Loading nodes matrix
load('strain_results.mat','nodes');
nodes = nodes{1,1};
%Loading strains matrix
load('strain_results.mat','strains');
strains = strains{1,1};
cd ..
cd Tracking
%Loading coords matrix
load('tracking_results_raw.mat','coords');
cd ..
cd ..
%------------------------------------------
```

Figure 5.23: MATLAB Script to Extract Relevant MODEM Output Files

5.8 GENERATING CONTOUR PLOTS AND VIDEOS IN MODEM

5.8.1 Generating Contour Plots in MODEM

After an analysis in MODEM has been completed, contour plots and videos showing the strains and displacements calculated by MODEM can be created. These contour plots can show different strain and displacement values as shown in Table 5.1.

Table 5.1: Options for MODEM Strain and Displacement Contours

Contour Abbreviation	Strain/Displacement Plotted
x	Displacement in the X Direction
y	Displacement in the Y Direction
z	Displacement in the Z Direction (Only for 3D Analysis)
xx	Normal Strain in the X Direction
yy	Normal Strain in the Y Direction
xy	Shear Strain in the XY Plane
minP	Minimum Principal Strain
maxP	Maximum Principal Strain
dirP	Principal Direction

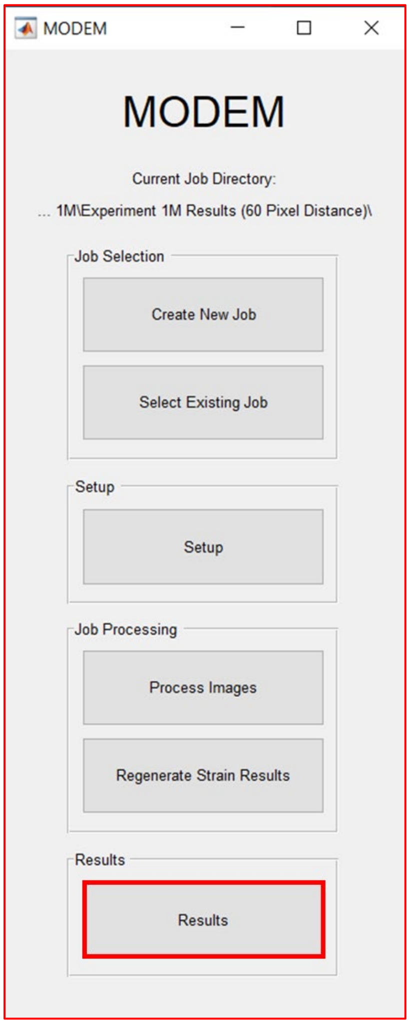
The coordinate axes directions used in MODEM are shown in Figure 5.17 while the positive sign conventions for the strain values are shown in Figure 5.21.

A set of contours can be created for every strain and displacement values described in Table 5.1 with an accompanying color bar-legend. If a new set of contours are generated for a strain or displacement value that has already had contours generated, then the new contours will overwrite the old ones.

Each contour is labeled with the abbreviation of the strain or displacement value it is representing (shown in Table 5.1 “Contour Abbreviation” column) followed by its number given to it by the user. So, for example, if a contour is being generated for an image named “DSC_0015” and will display x displacement values, then the contour generated will be named “x_0015.” These contours will be generated in the “Contours” folder found in the “Results” folder of the given experiment being analyzed.

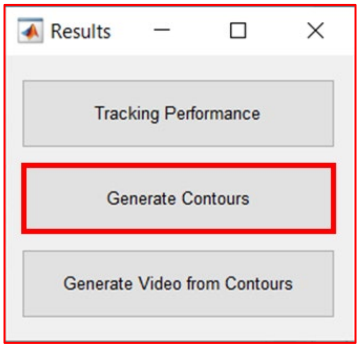
5.8.1.1 PROCEDURE FOR GENERATING CONTOUR PLOTS

1. To generate any given contours shown in the list above, select the “Results” button in the main MODEM directory.

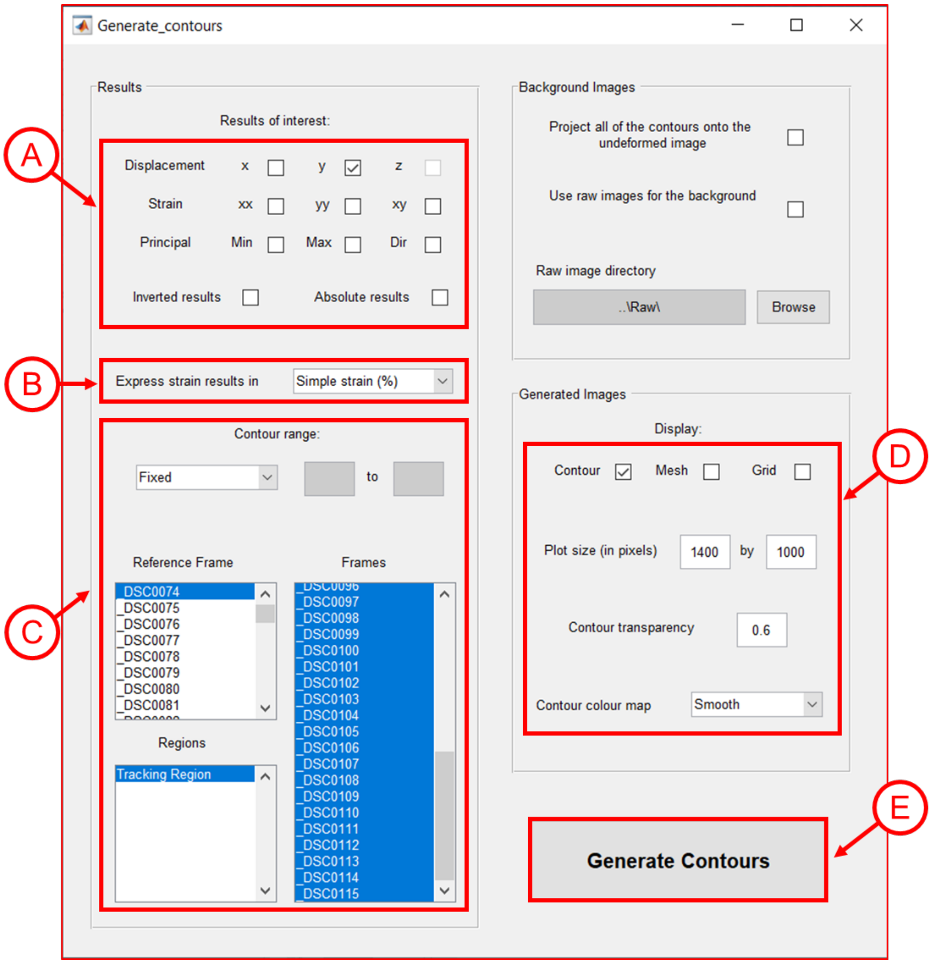


Current Job Directory:
... 1M\Experiment 1M Results (60 Pixel Distance)\

2. In the *Results* window, select the “Generate Contours” button.



3. In the *Generate_contours* window, select any of the displacement or strain values (Section A) to generate contours for the a given range of frames and regions highlighted in the “Frames” and “Regions” lists (Section C). To change the frame range for the contour plots, a reference frame must be selected from the “Reference Frame” list (Section C). This frame is the basis from which all strain and displacement measurements are calculated. The default reference frame is the first image in the image set.



Results

Results of interest:

Displacement x ☐ y ☑ z ☐

Strain xx ☐ yy ☐ xy ☐

Principal Min ☐ Max ☐ Dir ☐

Inverted results ☐ Absolute results ☐

Express strain results in Simple strain (%)

Contour range: Fixed to

Reference Frame: DSC0074, DSC0075, DSC0076, DSC0077, DSC0078, DSC0079, DSC0080, DSC0081

Frames: DSC0096, DSC0097, DSC0098, DSC0099, DSC0100, DSC0101, DSC0102, DSC0103, DSC0104, DSC0105, DSC0106, DSC0107, DSC0108, DSC0109, DSC0110, DSC0111, DSC0112, DSC0113, DSC0114, DSC0115

Regions: Tracking Region

Background Images

Project all of the contours onto the undeformed image ☐

Use raw images for the background ☐

Raw image directory ..\Raw\

Generated Images

Display:

Contour ☑ Mesh ☐ Grid ☐

Plot size (in pixels) 1400 by 1000

Contour transparency 0.6

Contour colour map Smooth

If a strain value is selected, there are four units MODEM can display contours in as shown in Table 5.2 which can selected in Section B.

Table 5.2: Descriptions of Strains Used in MODEM Contour Outputs

MODEM Strain Type	Description of Strain Type
Simple Strain	Change in length over the original member length [Δ/L]
Simple Strain (%)	Simple strain in the form of a percentage
Simple Strain (usn)	Simple strain in terms of microstrain (μn)
True Strain	Instantaneous Elongation per Unit Length of a Specimen [$\ln(L/L_o)$]

4. To define the properties of the contour plots, select from the options in Section D which range from the contour pattern (contour, mesh, or grid) and contour transparency to the smoothness of the contour values. When selecting the type of contour pattern for the contour plots, the user can select more than one option at a time. If more than one option is selected, the contour patterns will be overlaid. Examples of how each contour pattern appears is shown in Figure 5.24.

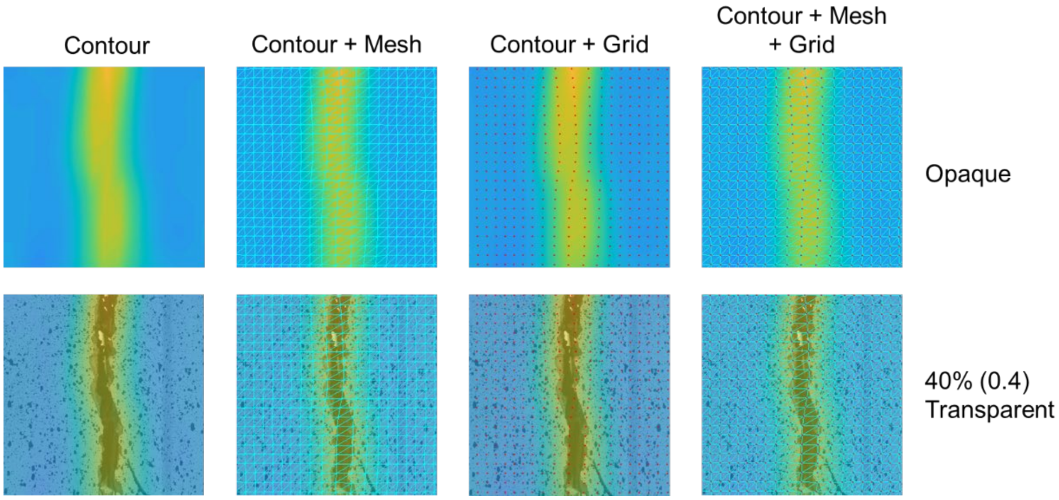


Figure 5.24: Types of Contour Patterns

5. Click the “Generate Contours” button (Section E). Once the contours are created, MODEM will reopen the *Results* window.

5.8.1.2 UTILIZING CONTOUR PLOTS FOR CRACK MAPPING AND PROPAGATION

Contour plots from MODEM can be useful for determining small strain concentrations associated with crack formation in specimens. By minimizing the range of images analyzed in the “Frames” column of Section C in Section 5.8.1.1 for a given experiment, the difference between the minimum and maximum strains in the color bar legend is reduced and causes cracking to become more evident in the generated contours. The images shown in Figure 5.25 show this process of tracking crack propagation in an experiment concrete prism subject to pure tension. As the experiment progresses, it is clear that the cracks appearing in the contours become more defined.

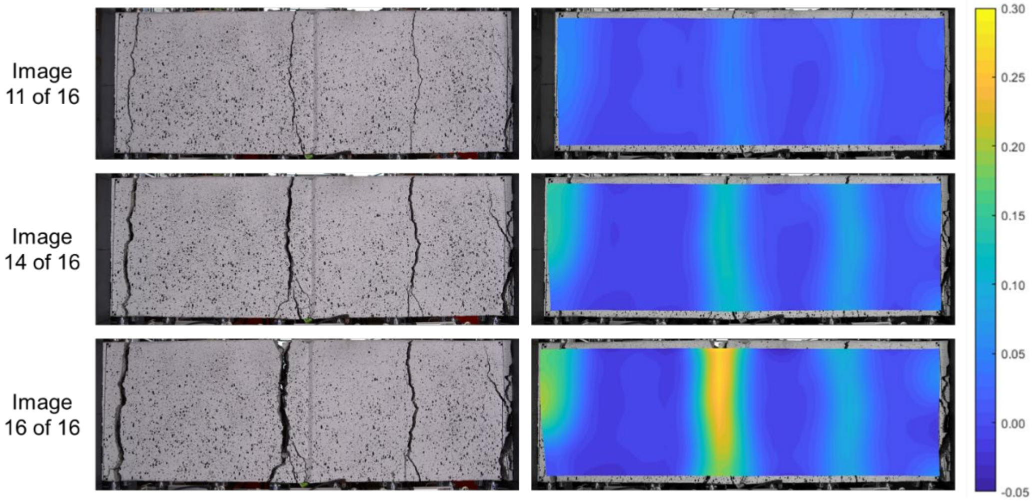
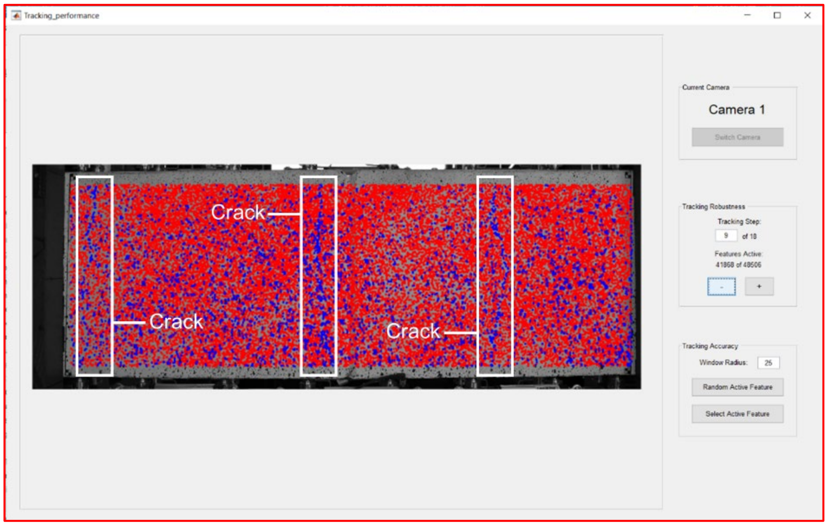


Figure 5.25: Crack Propagation in MODEM Utilizing ε_{xx} *[4] [19]*

Cracking can also be seen through the loss of features in an experiment. When cracks occur, the displacement a feature experiences between two subsequent images may exceed what MODEM can track or may simply disappear altogether, causing them to be lost (identified with blue markers in Figure 5.26). MODEM records which features are lost throughout an experiment and this data can be accessed by selecting the "Tracking Performance" button in the *Results* window as shown in Step 2 in Section 5.8.1.1.



Tracking Step: 9 of 18

Features Active: 41868 of 48506

Window Radius: 25

Figure 5.26: Tracking_performance *Window with Cracking Highlighted*

The number and locations of the lost features may change with every image. When long lines of blue dots form, as those seen in Figure 5.27, this can indicate cracking. If these lines grow throughout the image set, then the cracks are propagating. Determining the difference between cracks and random lost features often requires a user to review the entire image set manually.

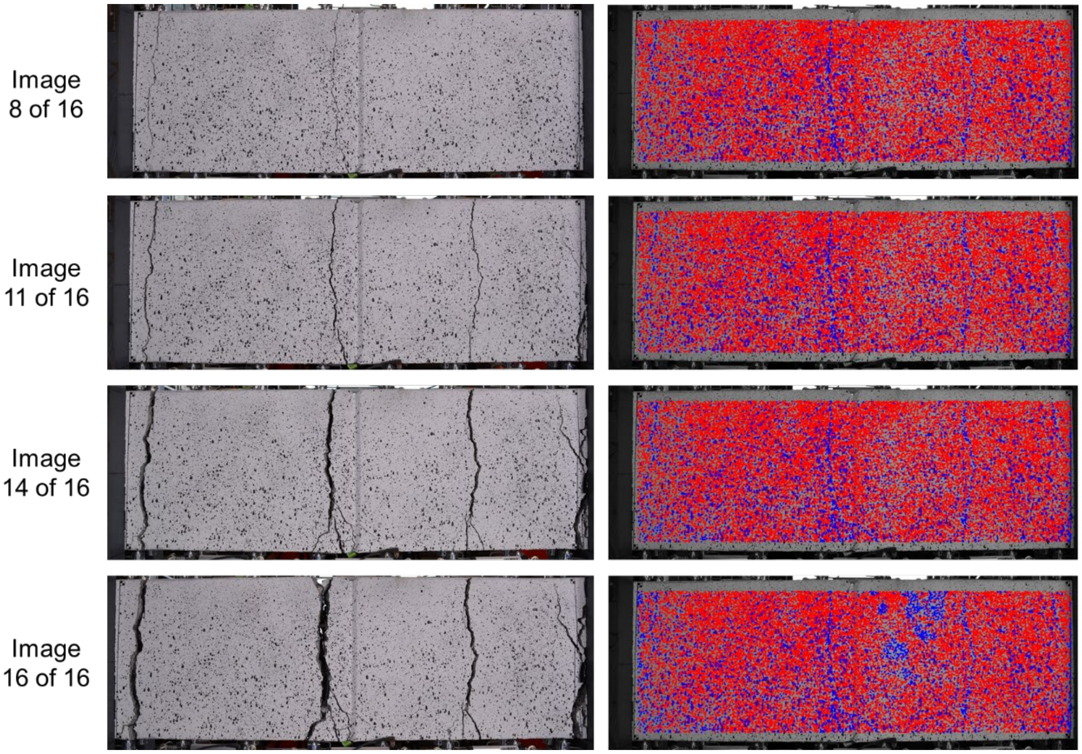


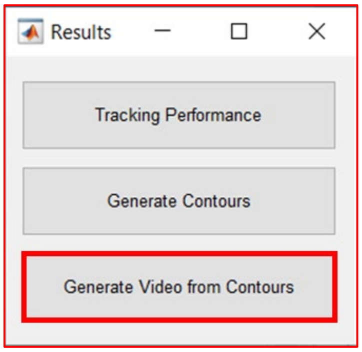
Figure 5.27: Crack Mapping by Analyzing Lost Tracking Subsets

5.8.2 Generating Contour Videos in MODEM

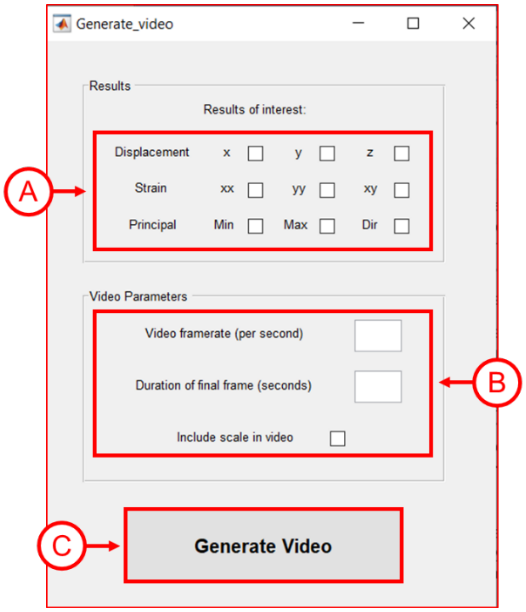
After all of the contours are generated, MODEM can combine these contours into a video, a visual that can be very effective for showing global behavior of the specimens as well as crack formation. MODEM generates MP4 video files from contour plots that have already been generated beforehand, and are saved in the *Videos* folder under MODEM's experimental output as shown in Figure 5.15.

5.8.2.1 PROCEDURE FOR GENERATING VIDEOS IN MODEM

1. To generate videos from contours, select the "Generate Videos from Contours" button in the *Results* window.



2. After the *Generate_video* window opens, select only contours that have already been generated in the *Generate_contours* window to be converted into videos files (Section A). From there, there are three video file parameters that can be customized for any MODEM video file: framerate (frames per second), duration of final frame, and the inclusion of a scale (Section B).



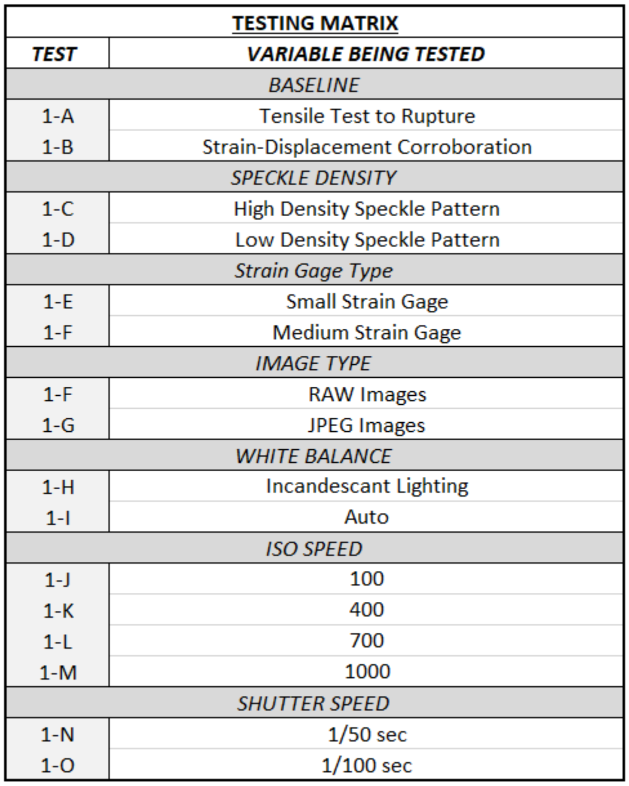
3. Once all video parameters have been selected, click the “Generate Video” button (Section C). From here, a *Progress* bar will appear displaying how far MODEM is in the video generation process. After completion, the *Results* window will appear again.

6 EXAMPLE TESTS AND ANALYSES

6.1 ALUMINUM COUPON – TENSION EXPERIMENTS

6.1.1 Overview

In order to gain a better understanding of MODEM and its intricacies, a series of tensile tests were conducted on aluminum coupons to determine how variables in the setup of an experiment would affect how MODEM would process the images created during that experiment. These variables are shown in the testing matrix in Figure 6.1. The findings of these experiments were used to create the best practice guidelines in Section 4.



TESTING MATRIX

TEST	***VARIABLE BEING TESTED***
BASELINE	
1-A	Tensile Test to Rupture
1-B	Strain-Displacement Corroboration
SPECKLE DENSITY	
1-C	High Density Speckle Pattern
1-D	Low Density Speckle Pattern
Strain Gage Type	
1-E	Small Strain Gage
1-F	Medium Strain Gage
IMAGE TYPE	
1-F	RAW Images
1-G	JPEG Images
WHITE BALANCE	
1-H	Incandescant Lighting
1-I	Auto
ISO SPEED	
1-J	100
1-K	400
1-L	700
1-M	1000
SHUTTER SPEED	
1-N	1/50 sec
1-O	1/100 sec

Figure 6.1: Tension/Pull Experiments Testing Matrix

6.2 EXPERIMENT 1M

Experiment 1M was a tensile/pull test of a simple aluminum specimen conducted to run a 2D analysis of the specimen's behavior in MODEM and verify that the strain values computed by MODEM are consistent with that from traditional instrumentation (i.e. a strain gauge and extensometer). The test setup, procedure, and results of this experiment are elaborated upon in this section as a reference to other researchers for their own DIC experiments.

6.2.1 1M Camera and Experimental Settings

Results from experiments from Figure 6.1, in addition to a literature review, were utilized to develop settings used for Experiment 1M shown in the Table 6.1.

Table 6.1: Experiment 1M Camera and Experimental Settings

Camera Settings	Setting Used in 1M
Camera Used	Nikon D5500 (Nikon AF-S DX NIKKOR 18-55mm Lens)
Image File Format	RAW
Image Coloring	Monochromatic
ISO	1000
F-Stop	F/8
White Balance	Incandescent
Zoom/Camera Distance	140 mm
Shutter Speed	1/100th sec
Image Capture Rate	10 sec
Testing Rate	0.01 in/min

6.2.2 Specimen Geometry and Preparation

The specimen tested (shown in Figure 6.2) was a 1.25” wide by 11” long rectangle made from 21 gauge aluminum. The shape of this specimen was chosen so that it would fit into the jaws of a universal testing machine and provide evenly distributed loading to the entirety of the cross-section. The specimen’s surface was split up into three sections: one central section and two identical clamping sections. These two clamping regions spanned the last 2.25” of either end of the specimen and were bare aluminum. These sections were designated as surfaces for the universal testing machine’s jaws to clamp onto the specimen. The central 6.5” was allocated for the application of the speckle pattern and would contain the region that would be tracked by MODEM.

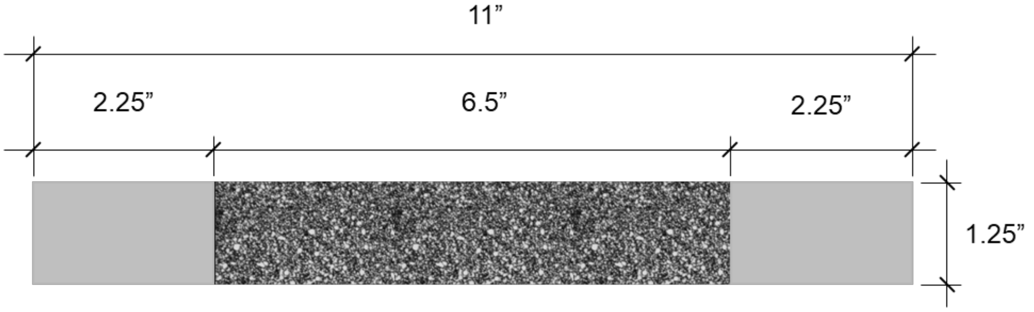


Figure 6.2: Experiment 1M Specimen Geometry

This speckle pattern was a white-on-black pattern made using Rust-Oleum® Universal® Flat Primer + Paint spray paint, a matte spray paint that is designed to be applied to wood, metal, plastic, and masonry. The speckle pattern consisted of a black base layer and a white speckle pattern and was applied utilizing the methods and techniques specified in Section 5.1.

This specimen was instrumented with a strain gage (similar to a CEA-06-250UN-120 from Micro Measurements) measuring 0.3125 inches which was placed in the geometric center of the specimen, 0.625 inches from the long edge of the specimen and 5.5 inches in from the short edge of the specimen. The placement of the strain gage is illustrated in Figure 6.3.

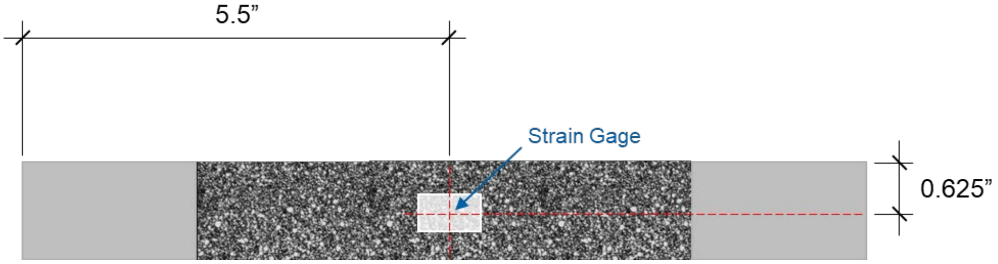


Figure 6.3: Strain Gage Location on Specimen 1M

6.2.3 Experimental Setup

The experiment was conducted using a benchtop universal testing machine (UTM) in Cal Poly’s Berridge Lab. Specimen 1M was placed in the UTM with each of its bare aluminum sections being positioned in one of the UTM’s jaws. It was made sure that the specimen was tightly secured in the testing apparatus. After the specimen was in place and secured, the extensometer (a Tinius Olsen 3542-0200-050-ST axial extensometer with 2 inch gage length) was secured on the specimen. Each of the extensometer’s clips were placed an inch above and below the center of the strain gage as shown in Figure 6.4. This placement allowed the strain gage and extensometer to record the same average strain measurements to verify consistency between traditional instrumentation readings.

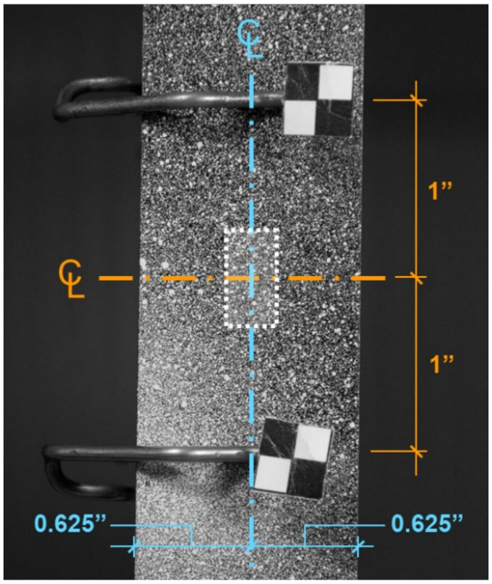


Figure 6.4: Experiment 1M Strain Gage and Extensometer Placement (Image Credit: Nicole Buck)

From here, two incandescent lights with a magnetic base were placed on either side of the specimen, attached to the frame of the UTM. One of these lights was pointing down on the specimen while the other was pointed up, completely illuminating the speckled surface and minimizing the amount of shadows cast on the specimen (shown in Figure 6.5).

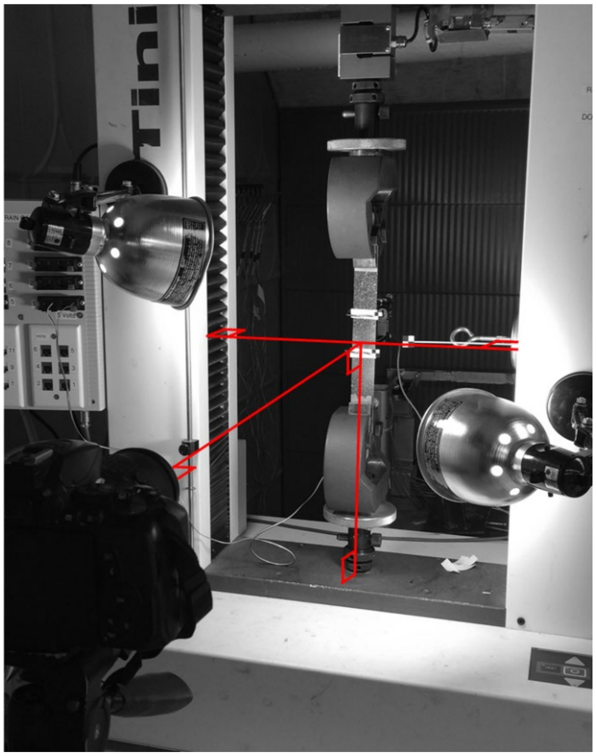


Figure 6.5: Experiment 1M Lighting Setup

After the lighting was in place, the Nikon D5500 camera was positioned orthogonally and about 29 inches from the surface of the specimen (as shown by the red lines in Figure 6.5). This is about as close as the camera could be to the specimen without being placed on the table. The camera was positioned at approximately the same elevation as the centerline of the specimen shown in Figure 6.4. The camera lens was then set to its maximum zoom setting, making sure that the region being tracked was still within the view of the camera. This camera placement and zoom setting were selected so that the camera would be as close to the specimen as possible. This allows for a greater resolution of the speckle pattern and therefore better tracking while also being far enough away so that the entire specimen could be within the frame of the camera.

It was at this point that all the camera settings described in Table 6.1 were set and test photos were taken to verify the image quality - clear resolution and an image brightness - that was satisfactory (an example of which is shown in Figure 6.6). The necessary computer software needed to acquire the strain gage and extensometer data was then prepared and tested to verify that all of the instrumentation was reading properly.

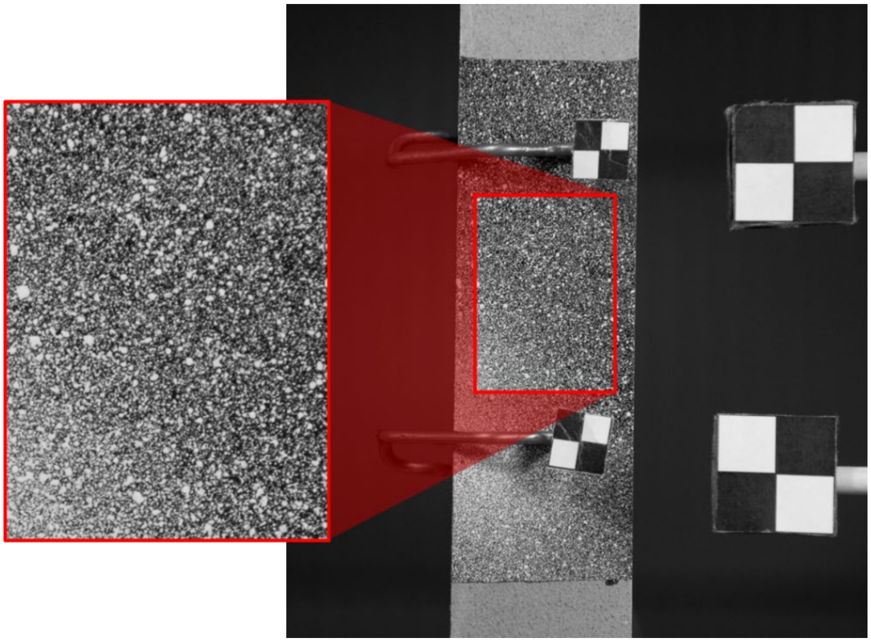


Figure 6.6: Image Quality in Experiment 1M Photos [4]

6.2.4 Calibration

Table 6.2: Experiment 1M Calibration Image and Calibration Board Design

Calibration Parameters	Parameter Used in 1M
Number of Calibration Photos Taken	30
Dimensions of Calibration Board	4 inches by 3.25 inches
Number of Squares on Calibration Board	4 Square Wide and 5 Squares Long
Dimensions of Squares	0.81 inches by 0.81 inches

MATLAB's *Camera Calibrator App* was used to create the calibration file that was then uploaded to MODEM. The dimensions and design of the calibration board used in experiment 1M can be found in Table 6.2. Thirty calibration images using the board described above were taken before the experiment as suggested by MathWorks. All of these photos were varied in their position of the calibration board on and around the specimen to acquire a more accurate reading from the *Camera Calibrator App*.

The calibration board was placed both in-plane (for 25 out of the 30 images) and out-of-plane (for 5 out of the 30 images) for this set of calibration images. This variation allows the MATLAB app to determine depth in a 2D image and the camera's location int relation to the board. All out-of-plane positions shifted at least one corner the calibration board out-of-plane by about 0.75 to 1 inch.

To take these photos, the back of the calibration board was attached to the surface of the specimen with painter's tape to keep the board steady while the images were taken. The tape also allowed for the board to be removed and replaced easily in both in-plane and out-of-plane positions. After every image the calibration board was rotated to provide the app with a variety of board positions to help conduct its calibration. Some examples of the calibration images taken in experiment 1M are shown in Figure 6.7

with the left image depicting the calibration board in-plane and the right image depicting the calibration board out-of-plane.

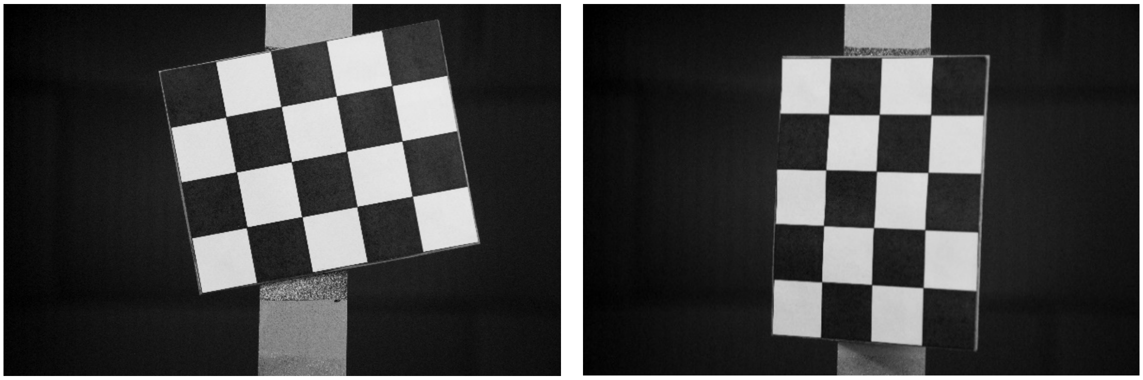
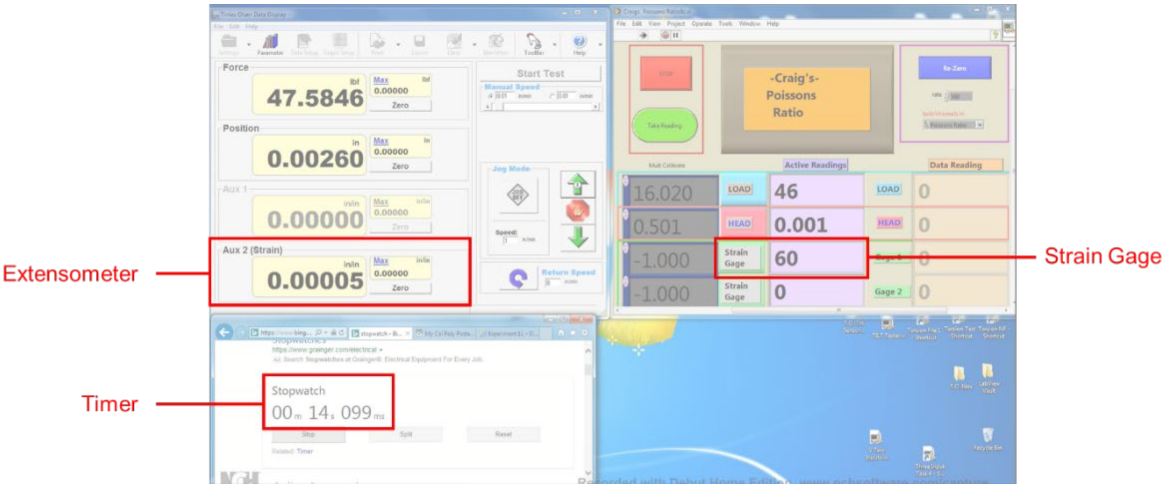


Figure 6.7: Examples of Experiment 1M Calibration Images

6.2.5 Testing Procedure

This testing procedure for experiment 1M, and the other tensile experiments, are described below.

1. The specimen was placed in a tabletop universal testing machine.
2. A strain gage was plugged into a data acquisition system (DAQ) and an extensometer was attached to the specimen and plugged into the tabletop universal testing machine (with which it interfaces).
3. Two incandescent lights with a magnetic base were positioned on either side of the specimen to fully illuminate the tracking region (see Figure 6.5).
4. Both the strain gage and extensometer windows (which continuously display the instruments' readings) were both opened on the same monitor as shown in Figure 6.8. A stopwatch app was also opened on this monitor.



Force 47.5846 lbf
Position 0.00260 in
Aux 1 0.00000 in/in
Aux 2 (Strain) 0.00005 in/in
Stopwatch 00m 14s 099ms
LOAD 16.020 46
HEAD 0.501 0.001
Strain Gage -1.000 60
Strain Gage -1.000 0

Figure 6.8: Snapshot of Screen Record of Experiment 1M Traditional Instrumentation Data Aquisition

5. The camera and tripod were set up according to the specification detailed in Section 6.2.3.
6. The camera settings were set according to Table 6.1.
7. Image(s) of the specimen was taken to ensure lighting properly illuminated the specimen and that the specimen and speckle pattern were in focus.

8. The calibration images were taken of the specimen according to the parameters and procedure described in Section 6.2.4.
9. An image of the specimen was taken to be used as a reference frame for the experiment.
10. The camera’s Time Lapse feature was set to take 40 images at a ten second interval. These images were taken using the Nikon D5500’s Time Lapse feature (where a specific number of images can be taken at a given interval) to make sure there was no movement or changes in camera settings, like white balance or focus.
11. Screen capture software was opened to record the computer monitor and the strain readings, extensometer readings, and stopwatch.
12. The tabletop universal testing machine was set to elongate the specimen at a rate of 0.01 in/min.
13. The stopwatch, screen recorder, tabletop universal testing machine, and camera’s Time Lapse feature were all activated simultaneously.
14. The experiment ran for a predetermined length of 6 minutes and 40 seconds until the camera stopped taking photos.
15. The tabletop universal testing machine, the stopwatch, and the screen recorder were stopped.
16. The screen recording was opened and played. For every ten second interval on the stopwatch the strain gage and extensometer readings were recorded in an Excel file. These values corresponded to the time when the camera was taking an image of the specimen, coinciding the traditional instrumentation’s readings with the images taken.
17. A MODEM analysis was run from the images taken and the results of that analysis were compared to the data collected from the strain gage and extensometer.

6.2.6 Data

The data pulled from the screen recording, showing the extensometer and strain gage data, is shown in Table B.1 in the Appendix.

The images taken during this experiment, the extensometer and strain gage data, and the MODEM data computed from these image are attached to this set of guidelines on Cal Poly Digital Commons under the title “Guidelines for Implementing MODEM: An Open-Source, MATLAB-Based Digital Image Correlation Software.”

6.2.7 MODEM Analysis

Based on the photographs taken during the experiment, the following inputs were given to MODEM in order to conduct the DIC analysis:

Table 6.3: MODEM Input Values for Experiment 1M

Input Parameters	MODEM Input for 1M
Result Grid Spacing	60
All Other Settings	Default Settings

In this analysis most of the input parameters for MODEM were kept as their default settings for simplicity and because MODEM’s default settings provide accurate results to experimental tests, as is described in the Section 6.2.8. The only parameter changed from its default setting was the Results Grid Spacing setting which was changed to 60 pixels. This value was determined by conducting a parametric study (as shown in Section 5.6) which determined that 60 pixels was the smallest distance between the nodes that generated a smooth strain field.

From here a MODEM analysis was run following the procedure indicated in Section 5.3. Based on the Resulting Grid Spacing selected, MODEM produced the following node locations as shown in red in Figure 6.9.

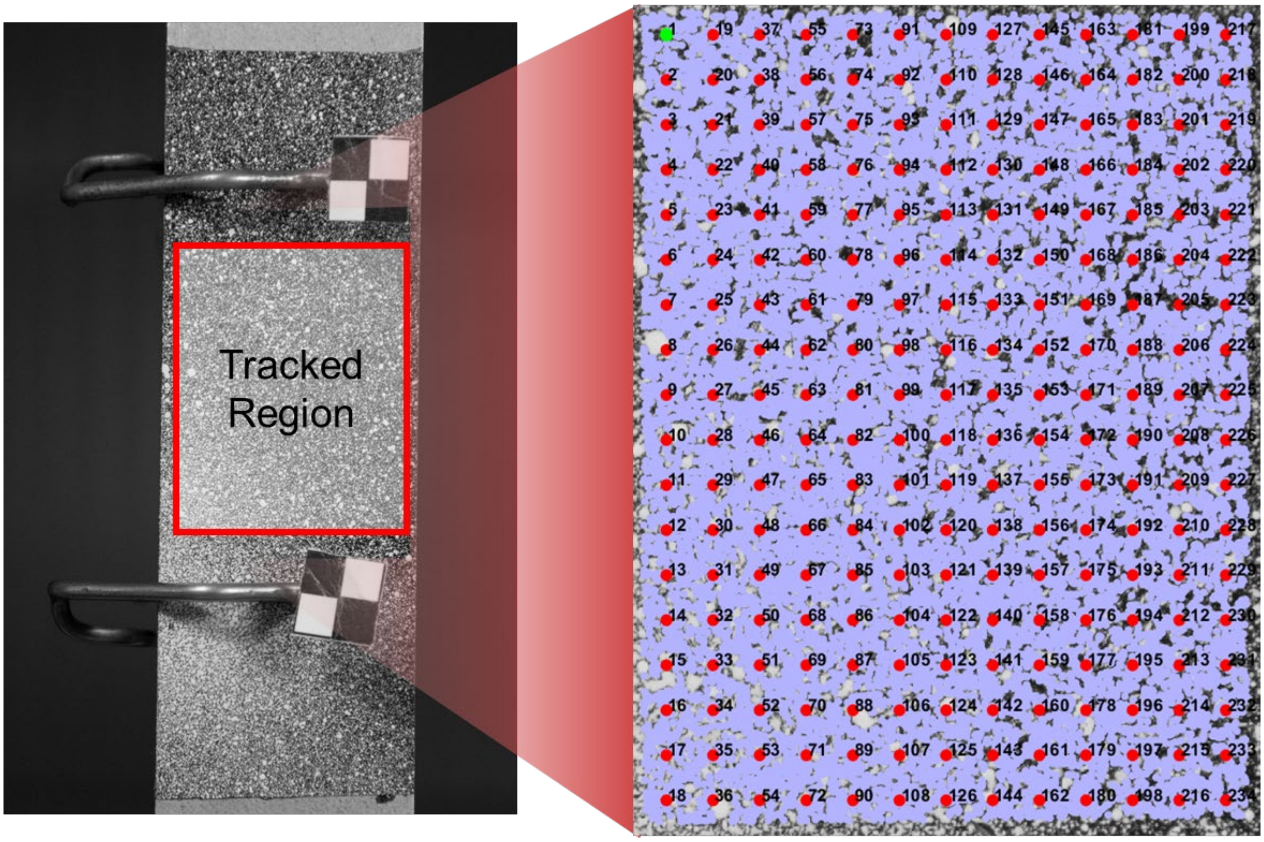


Figure 6.9: Tracked Coords and Nodes from Experiment 1M

A resulting 232 nodes were generated in a 1000 pixel by 1400 pixel tracking region. The violet dots in Figure 6.9 represent the coords tracked during this experiment which were used to compute the strain and displacements at the nodes. This figure was produced by utilizing a MATLAB script developed named “Node_and_Coordinate_Initial_Positions” which can be found attached to this document on Cal Poly Digital Commons.

6.2.8 Results

The 2D MODEM analysis for experiment 1M was strain based and compared to measurements from the strain gage and extensometer. Extensometer strain was computed from measured displacements divided by the by its initial gage length of 2 inches. The data for these sensors are found in Table B.1.

Since the extensometer and strain gage strains were in close agreement, the strain gage values the were used to compare to those acquired from MODEM.

Of the strains at the 232 nodes (shown in Figure 6.9), only the six nodes that overlapped with the strain gage were selected and compared to the strain gage (shown in Figure 6.10). These, nodes were determined based on approximation of the known location of the strain gage.

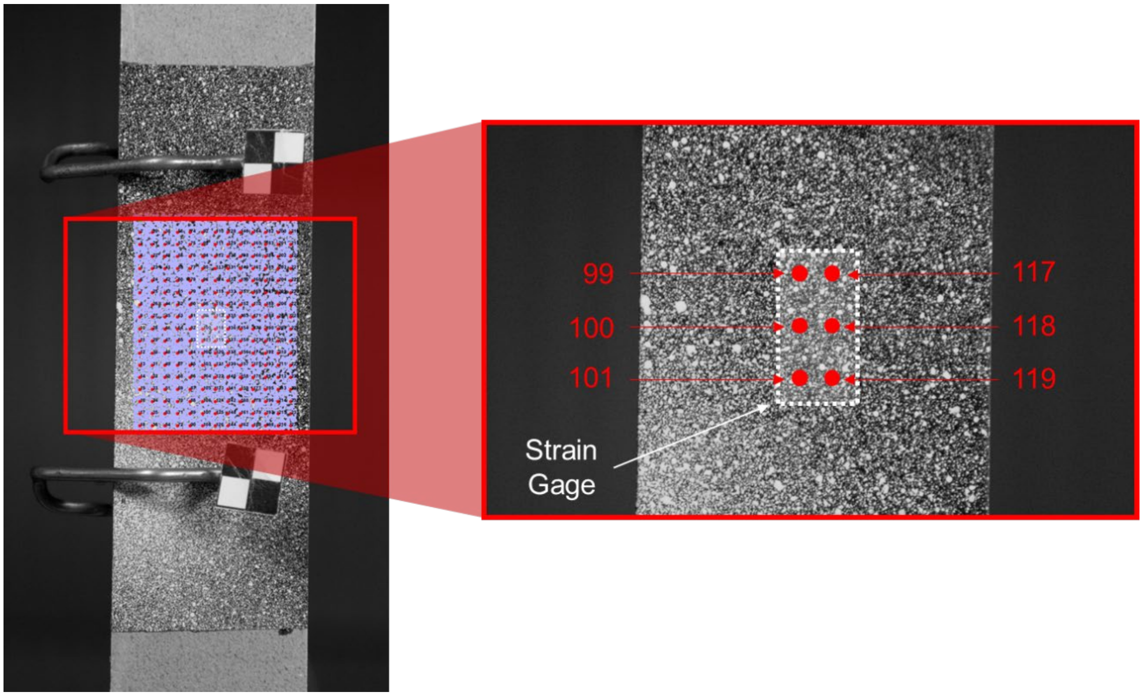


Figure 6.10: MODEM Nodes Utilized in 1M Analysis

Once the nodes were selected, the associated strain values were extracted from the MODEM output (using procedure described in Section 5.7.3.1) and plotted in relation to image number to produce Figure 6.11. The strain values of each node were then averaged at each image number.

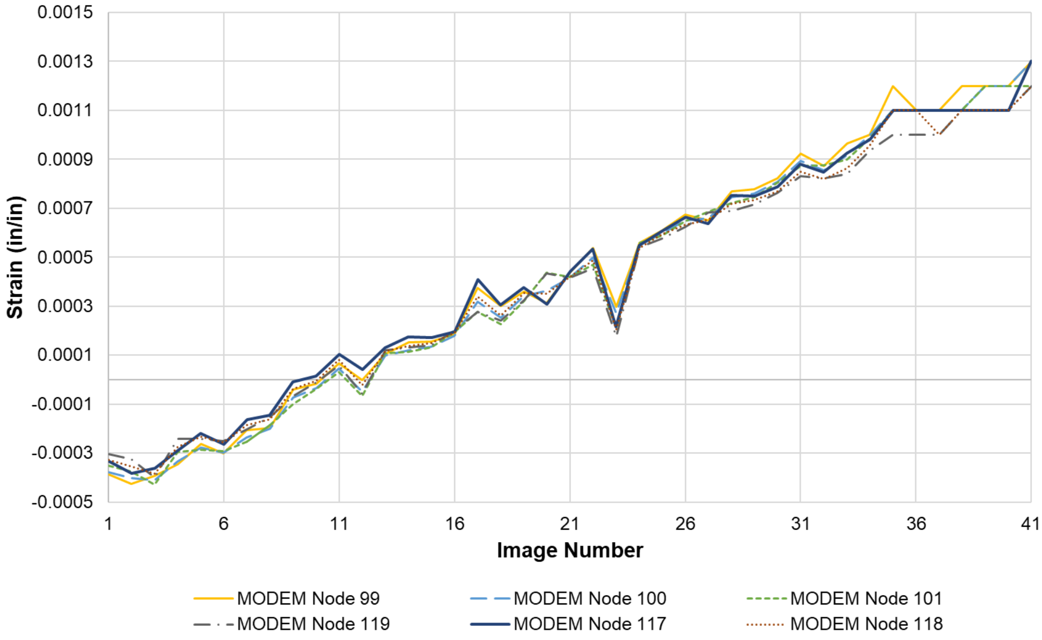


Figure 6.11: Raw MODEM Nodal Strains for Experiment 1M

At this point two aspects of the MODEM strains were incongruent with the experimental data collected: the MODEM strains did not start at zero and the averaged MODEM strain fluctuated as the experiment progressed. Both aspects are common to most DIC systems and did not cause great concern. To rectify the fact that the strains were initially negative, the initial strain value of the averaged nodal strain was subtracted from every strain value in the set. Then, to account for the fluctuation in the MODEM strain values, the values were linearly fit to the data. The reason the data was linear fit was because Experiment 1M was tested within the elastic range, so a linear deformation was assumed. This fitted set of strains was also initiated at zero as well. The resulting strain values are shown in the Figure 6.12.

Figure 6.12: Linearly Fit 1M MODEM Data

Once the MODEM strains were linearized, they were then compared to those captured from the strain gage and found to be, on average, within 5% of the strain gage results. The percent error of these strain readings was higher at the beginning of the experiment and decreased as the experiment progressed. Likely due to the fact that early in the experiment the strain readings were quite small and would lead a very small number in the denominator of the percent error calculation.

A summary of the results of Experiment 1M are shown in the table below:

Table 6.4: Percent Error Results for Linearized Experiment 1M Data

Input Parameters	**MODEM Input for 1M**
Avg. % Error of MODEM Strains	5%
Minimum % Error of MODEM Strains	0.23%
Maximum % Error of MODEM Strains	14%

7 RECOMMENDATIONS AND CONCLUSIONS

7.1 RECOMMENDATIONS

This section describes other recommendations for conducting tensile experiments in MODEM based on the author's experience conducting the suite of aluminum coupon tests (as listed in Figure 6.1). These recommendations can also be generalized for experiments using MODEM beyond tensile experiments.

7.1.1 Approximate Yield Point Prior to Experiment

Determine the material properties of the specimens being tested so that their behavior can be approximated beforehand, specifically when yielding occurs. This can be especially important if the plastic behavior of a specimen is going to be examined. When a specimen begins to yield, the rate at which it deforms can change. If this is not accounted for then, after a specimen yields, the capture rate being utilized for the elastic behavior might be too infrequent to capture the specimen's plastic behavior. This can result in data loss. To account for this change in deformation rate select a capture rate that would capture the specimen's elastic and inelastic deformation. Another solution could be to change the capture rate of the camera when the specimen begins to yield. This could be done automatically or manually based on the strains being acquired from traditional instrumentation.

7.1.2 Implement a DAQ System that Timestamps Instrumentation Data

Utilizing a screen recording to capture instrumentation data was not efficient and likely created error in the results. Thus, it is recommended to implement an automatic form of syncing data and image acquisition. Having all systems linked together, possibly using a MATLAB or LabView program, would reduce the errors seen between the two sets of data. One method for doing this would be to have a DAQ system that timestamps instrumentation data. Then the timestamps found in the image metadata could be used to match that image to the data point from the traditional instrumentation.

7.2 CONCLUSIONS

The suite of aluminum coupon tests explained in Section 6.2 provided great insights into the proper procedures necessary to successfully conduct DIC analyses in MODEM for structural engineering purposes. This suite of experiments, considered Phase 1, led to Phase 2 and 3 work conducted by Buck [4] which focused on small and moderately scaled concrete specimens. That report should be referenced for those interested in using MODEM on concrete specimens since it details the challenges that arise when utilizing MODEM with concrete and techniques that can be used to improve MODEM's results for that specimen type.

Some future work that should be conducted based on the results of these aluminum coupon tests should include determining the accuracy of calibrated and converted MODEM displacement values compared to displacement data collected from traditional instrumentation. Other work that should also be carried out includes utilizing two or more cameras to conduct side-by-side MODEM analyses on a specimen that is too large to be captured by a single camera, like a wall. These side-by-side MODEM analyses could then be combined to create one large, single MODEM analysis. Another future project could include determining how to accurately remove contributions of any out-of-place deformation from 2D MODEM analyses. Work should also be done to better incorporate the Parallel Computing Toolbox into MODEM to speed up computation time since errors during processing arise and require a lengthy re-processing of data. Lastly, future work should be conducted to determine how to utilize the 3D analysis capabilities of MODEM to expand the usability of this new instrument for structural engineering purposes.

REFERENCES

[1] N. McCormick and J. Lord, "Digital image correlation for structural measurements," *Civil Engineering*, vol. 165, no. CE4, pp. 185-190, Oct. 2012, doi: 10.1680/cien.11.00040

[2] A. H. Salmanpour and N. Mojsilovic, "Application of Digital Image Correlation for strain measurments for large masonry walls," *presented at APCOM & ISCM*, Singapore, December 11-14, 2013.

[3] R. M. Stubbing, "Characterisation of Polymeric Foam Cores Using Digital Image Correlation," PhD thesis, Dept. Mechanical Engineering, Univ. of Auckland, Auckland, New Zealand, 2013. [Online]. Available: https://researchspace.auckland.ac.nz/handle/2292/21682?show=full

[4] N. Buck, "Implementation of an Open-Source Digital Image Correlation Software for Structural Testing," Master's thesis, Dept. of Architectural Engineering, California Polytechnic State University, San Luis Obispo, California, USA, 2020.

[5] M. Hikko, N. Mirow, H. Nagel, M. Irqsusi, S. Vogt, and V. Kuokkala, "In-vivo Deformation Measurements of the Human Heart by 3D Digital Image Correlation," *Journal of Biomechanics*, vol. 48, no. 10, pp. 2217-2220, July 2015, doi: 10.1016/j.jbiomech.2015.03.015

[6] B. Pan, K. Qian, H. Nagel, and A. Asundi, "Two-dimensional digital image correlation for in-plane displacement and strain measurements: a review," Measurement Science and Technology, vol. 20, no. 6, pp. 1-17, June 2009, doi: 10.1088/0957-0233/20/6/062001

[7] M. A. Sutton, J. Orteu, and H. W. Schreier, *Image Correlation for Shape, Motion, and Deformation Measurements*, New York, NY, USA, Springer Science+Business Media, 2009.

[8] Y. L. Dong and B. Pan, "A Review of Speckle Pattern Fabrication and Assessment for Digital Image Correlation," *Experimental Mechanics*, vol. 57, no. 8, pp. 1161-1181, Oct. 2017, doi: 10.1007/s11340-017-0283-1

[9] N. McCormick and J. Lord, "Digital Image Correlation," *Materials Today*, vol. 13, no. 12, pp. 52-54, Dec. 2010, doi: 10.1016/S1369-7021(10)70235-2

[10] Nikon Corporation. "Imaging Products." Nikon. https://imaging.nikon.com/lineup/software/control_pro2/ (accessed August 29, 2020).

[11] "digiCamControl." digiCamControl. http://digicamcontrol.com/ (accessed August 29, 2020).

[12] "VideoProc V3.8." Digiarty Software, Inc. https://www.videoproc.com/ (accessed August 29, 2020).

[13] ShareX Team. "ShareX." https://getsharex.com/. (accessed August 29, 2020).

[14] S. Cox. "What Is F-Stop and How Does It Work?" Photography Life. https://photographylife.com/f-stop. (accessed September 1, 2020).

[15] N. Mansurov. "What Is White Balance?" Photography Life. https://photographylife.com/what-is-white-balance. (accessed September 1, 2020).

[16] N. Mansurov. "Understanding Shutter Speed in Photography." Photography Life. https://photographylife.com/what-is-shutter-speed-in-photography. (accessed September 1, 2020).

[17] R. Naryškin, "Understanding Histograms in Photography," Photography Life. https://photographylife.com/understanding-histograms-in-photography (accessed August 18, 2020)

[18] The MathWorks, Inc. “Single Camera Calibrator App.” MathWorks Help Center. https://www.mathworks.com/help/vision/ug/single-camera-calibrator-app.html#:~:text=The%20suite%20of%20calibration%20functions%20used%20by%20the,of%20functions%2C%20see%20Single%20and%20Stereo%20Camera%20Calibration. (accessed September 17, 2020).

[19] Q. Wang, R. Henry, L. Hogan, and A. Scott, "Testing of Reinforced Concrete Prisms with Different Loading Rates," presented at *17th World Conference on Earthquake Engineering*, Sendai, Japan, 2020.

[20] R. Naryškin, “What is Metadata in Photography?,” Photography Life. https://photographylife.com/what-is-metadata-in-photography (accessed August 29, 2020)

APPENDIX A

A.1 MODEM OUTPUT FILE

This section provides a detailed description of the structure and contents MODEM's output files for a given 2D analysis. These output files are automatically generated in the folder location the user defined in the "Target Job Directory" box of the *Prepare_2D_Images* Window (Section 5.3 Step 6).

Two folders are created with each analysis: the "Job_Data" folder and the "Results" folder. Within each folder are multiple MATLAB data files, the contents of which are explained in the following subsections. Within each of these subsections, various MATLAB data types will be referenced and, for simplicity, these data types are condensed into the following acronyms:

FPN-L = Floating-Point Numbers that are treated as logicals [Yes = 1 & No = 0]

FPN-R = Floating-Point Number (Integers that represent selections from a list)

FPN = Floating-Point Number

FPA = Floating-Point Array (1xN or Nx1)

FPM = Floating-Point Matrix (MxN)

CHR = Character Array

LOG = Logical

CELL = Cell Array

DATA = Data structure

A.1.1 "Job_Data" Folder

The "Job_Data" folder contains all of the basic settings used by MODEM in its analysis. This folder includes two MATLAB data files show in Figure 4.2: *contour_settings* and *settings*.

The *contour_settings* file only appears if strain or displacement contours are generated in MODEM. Within this MATLAB data file is a structure by the same name which itself contains three additional structures: *results*, *background*, and *generated_pics*. Each of these structures contain variables, logicals, and values that are defined in Table A.1.

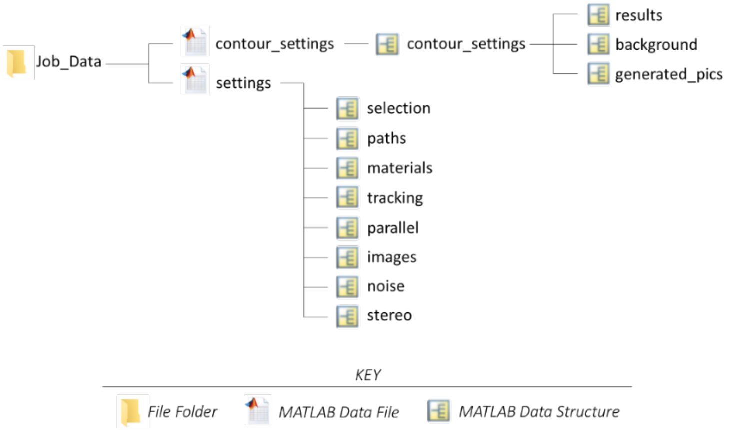


Figure A.1: Job_Data Folder Data Structure

A.1.1.1 CONTOUR_SETTINGS DATA STRUCTURE

The *contour_settings* data structure contains all of the data used to generate strain or displacement contours based on MODEM analyses. This data structure contains three nested data structures: results, background, and generated_pics. This data structure can be found in the within the *contour_settings* data file located in the “Job_Data” folder.

A.1.1.1.1 RESULTS DATA STRUCTURE

The *results* data structure can be found in the *contour_settings* structure which is nested within the “Job_Data” folder. This data structure contains the eighteen entries shown in Table A.1.

Table A.1: Contents of results Data Structure

VARIABLE NAME	VARIABLE TYPE	DESCRIPTION
plot_x	FPN-L	Indicates if x-displacement contours are generated
plot_y	FPN-L	Indicates if y-displacement contours are generated
plot_z	FPN-L	Indicates if z-displacement contours are generated
plot_xx	FPN-L	Indicates if xx-strain contours are generated
plot_yy	FPN-L	Indicates if yy-strain contours are generated
plot_xy	FPN-L	Indicates if xy-strain contours are generated
plot_minP	FPN-L	Indicates if minimum principle stress contours were generated
plot_maxP	FPN-L	Indicates if maximum principle stress contours were generated
plot_dirP	FPN-L	Indicates if principle direction contours were generated
Invert	FPN-L	Indicates if contour results are inverted
absolute	FPN-L	Indicates if contour results are absolute values
strain_out	FPN-R	Indicates type of strain generated in contour plots (simple strain = 1, % strain = 2, % microstrain = 3, true strain = 4)
c_range	FPN-R	Indicates the user defined range of contours being analyzed (dynamic = 1, fixed = 2, custom = 3)
c_min	FPN	first image number in contour range being generated (for custom range only)
c_max	FPN	last image number in contour range being generated (for custom range only)

ref_frame	FPN	position of reference image within photo set (i.e. if it is the first in the sequence, this value is 1)
Frames	FPA	array of images being converted into contours (1xN matrix)
Mats	FPN	value that indicates which region is being utilized in the contours being generated (1 = first region selected, 2 = second region selected, etc.)

A.1.1.1.2 BACKGROUND DATA STRUCTURE

The *background* data structure can be found in the *contour_settings* data structure which is nested within the “Job_Data” folder. This data file contains the entries shown in Table A.2.

Table A.2: Contents of background Data Structure

VARIABLE NAME	VARIABLE TYPE	DESCRIPTION
undeformed	FPN-L	if true, projects all of the contours on the undeformed image
use_raw	FPN-L	if true, uses raw images for the background of the contours
pic_folder	CHR	file path that leads to the raw images that can be used as the background for the contour

A.1.1.1.3 GENERATED_PICS DATA STRUCTURE

The *generated_pics* data structure can be found in the *contour_settings* data structure which is nested within the “Job_Data” folder. This data file contains the seven entries shown in Table A.3.

Table A.3: Contents of generated_pics Data Structure

VARIABLE NAME	VARIABLE TYPE	DESCRIPTION
show_contour	FPN-L	indicates if a heat map was applied to the generated contours
show_mesh	FPN-L	indicates if a mesh was applied to the generated contours
show_grid	FPN-L	indicated if a grid was applied to the generated contours
plot_width	FPN	plot width of the generated contours in pixels
plot_height	FPN	plot height output in generated contours in pixels
transparency	FPN	transparency of generated contours as a fraction of one

colour_map	FPN-R	Indicates the type of heat map that will be generated (Smooth = 1, Stepped = 2, Design Limit = 3)

A.1.1.2 SETTINGS DATA STRUCTURE

The *settings* data structure contains the parameters used by MODEM to determine tracking subsets, store important file locations, determine parallel computing options, store image naming data, record noise properties, track tracking subsets, and any other parameters used to initiate a MODEM analysis. This data structure is located in the *settings* data file located in the “Job_Data” folder.

A.1.1.2.1 SELECTION DATA STRUCTURE

The *selection* structure can be found in the *settings* data structure which is nested within the “Job_Data” folder. This data file contains the five entries shown in Table A.4.

Table A.4: Contents of selection Data Structure

VARIABLE NAME	VARIABLE TYPE	DESCRIPTION
subset	FPN	width of tracking subset (in pixels)
spacing	FPN	minimum feature spacing
threshold	FPN	quality threshold for feature selection
smoothing	FPN	threshold smoothing
randomization	LOG	if true, randomizes the initial subpixel locations of tracking features

A.1.1.2.2 PATHS DATA STRUCTURE

The *paths* structure can be found in the *settings* data structure which is nested within the “Job_Data” folder. This data file contains the seven entries shown in Table A.5.

Table A.5: Contents of paths Data Structure

VARIABLE NAME	VARIABLE TYPE	DESCRIPTION
camera	CHR	file path to the calibration file
raw_images	CHR	file path to raw images in reference to the input photo file path
cropped	CHR	file path to cropped images in reference to the input photo file path

tracking	CHR	file path to tracking data in reference to the input photo file path
strain	CHR	file path to strain data in reference to the input photo file path
contours	CHR	file path to contour data in reference to the input photo file path
videos	CHR	file path to video data in reference to the input photo file path

A.1.1.2.3 MATERIALS DATA STRUCTURE

The *materials* structure can be found in the *settings* data structure which is nested within the "Job_Data" folder. This data file contains the six entries shown in Table A.6.

Table A.6: Contents of materials Data Structure

VARIABLE NAME	VARIABLE TYPE	DESCRIPTION
default_smoothing	FPN	default smoothing values provided by MODEM
default_grid	FPN	default grid spacing provided by MODEM (in pixels)
boundaries	CELL	1x1 cell array containing user defined tracking regions' boundary coordinates (x,y)
grids	CELL	user defined results grid spacing (in pixels)
smoothing	CELL	user defined grid displacement smoothing
names	CELL	1x1 cell array containing the names of the user defined tracking regions

A.1.1.2.4 TRACKING DATA STRUCTURE

The *tracking* structure can be found in the *settings* data structure which is nested within the "Job_Data" folder. This data file contains the thirteen entries shown in Table A.7.

Table A.7: Contents of tracking Data Structure

VARIABLE NAME	VARIABLE TYPE	DESCRIPTION
pyramid_max	FPN	maximum pyramidal height
pyramid_min	FPN	minimum pyramidal height

threshold	FPN	unknown
resolution	FPN	tolerance to which MODEM has to converge during analyses (convergence criterion)
subset	FPN	width of tracking subset (in pixels)
iterations	FPN	maximum amount of iterations per step during tracking
interpolant	CHR	intensity interpolant used during tracking
shape	CHR	shape function used during tracking
fixed_reference	LOG	unknown
normalise_subsets	LOG	unknown
distortion_correction	LOG	unknown
debug	LOG	if true, saves a complete log of tracking results
approximate_increment	LOG	unknown

A.1.1.2.5 PARALLEL DATA STRUCTURE

The *parallel* structure can be found in the *settings* data structure which is nested within the “Job_Data” folder. This data file contains the four entries shown in Table A.8.

Table A.8: Contents of parallel Data Structure

VARIABLE NAME	**VARIABLE TYPE**	**DESCRIPTION**
attempt	LOG	if true, means MODEM will attempt to use the Parallel Computing Toolbox
mem_lim	FPN	minimum amount of computer memory required by MODEM
max_cores	FPN	maximum amount of computer cores MODEM is allowed to utilize
shared_memory	LOG	if true, means MODEM will utilize shared memory

A.1.1.2.6 IMAGES DATA STRUCTURE

The *images* structure can be found in the *settings* data structure which is nested within the “Job_Data” folder. This data file contains the eight entries shown in Table A.9.

Table A.9: Contents of images Data Structure

VARIABLE NAME	VARIABLE TYPE	DESCRIPTION
cropped_format	CHR	file type of images used in MODEM analysis
label	CHR	label for image set (label in front of all images in the image set)
first	FPN	number of first image in image set
increment	FPN	increment between images in the image set
last	FPN	number of the last image in image set
list	FPN	1xN array that includes all of the images MODEM analyzes
digits	FPN	number of digits in image name

A.1.1.2.7 NOISE DATA STRUCTURE

The *noise* structure can be found in the *settings* data structure which is nested within the “Job_Data” folder. This data file contains the three entries shown in Table A.10.

Table A.10: Contents of noise Data Structure

VARIABLE NAME	VARIABLE TYPE	DESCRIPTION
gaussian_apply	LOG	if true, means MODEM will utilize Gaussian smoothing in its analysis
gaussian_kernel	FPN	kernel size utilized in Gaussian smoothing
gaussian_sigma	FPN	sigma used in Gaussian smoothing

A.1.1.2.8 STEREO DATA STRUCTURE

The *stereo* structure can be found in the *settings* data structure which is nested within the “Job_Data” folder. This data file contains the two entries shown in Table A.11.

Table A.11: Contents of stereo Data Structure

VARIABLE NAME	VARIABLE TYPE	DESCRIPTION
available	LOG	indicates if a 3D analysis was conducted
tracking	STR	contains data similar to the first 6 entries of Table A.7 for stereo calcs

A.1.2 “Results” Folder

The “Results” folder contains all of the results generated by MODEM during an analysis. This folder includes five folders containing various MATLAB files shown in Figure A.2: “Contours”, “Cropped”, “Strain”, “Tracking”, and “Videos” folders.

The “Contours” folder contains all of the contours and color bars generated for every series of contour plots generated. The “Cropped” folder contains a series of blank images equal to the number of images in the data set being analyzed and a MATLAB data file named *image_data* that contains some basic information for the images uploaded to MODEM for the experiment being analyzed. The “Strain” folder contains the *settings_used* MATLAB data file, which is the same file as the *settings* data file described and explained in Section A.1.1.2, and the *strain_results* data file which contains the node locations, their displacements, and their accompanying strain values. The “Tracking” folder contains three MATLAB data files: the *settings_used* file, the *tracking_results_corrected* file, and the *tracking_results_raw* file. The *tracking_results_raw* file contains the feature locations and documentation as to whether or not the features were tracked throughout the experiment or rejected. The *tracking_results_corrected* file contains the feature locations that have been calibrated for the actual experiment if a calibration file was uploaded into MODEM. If not, the file is simply the same coords file as that found in the *tracking_results_raw* file. Lastly, the “Videos” folder contains all of the videos generated for the given experiment. Each of these folders contain structures, variables, logicals, and values that are defined in Sections A.1.2.1 to A.1.2.5.

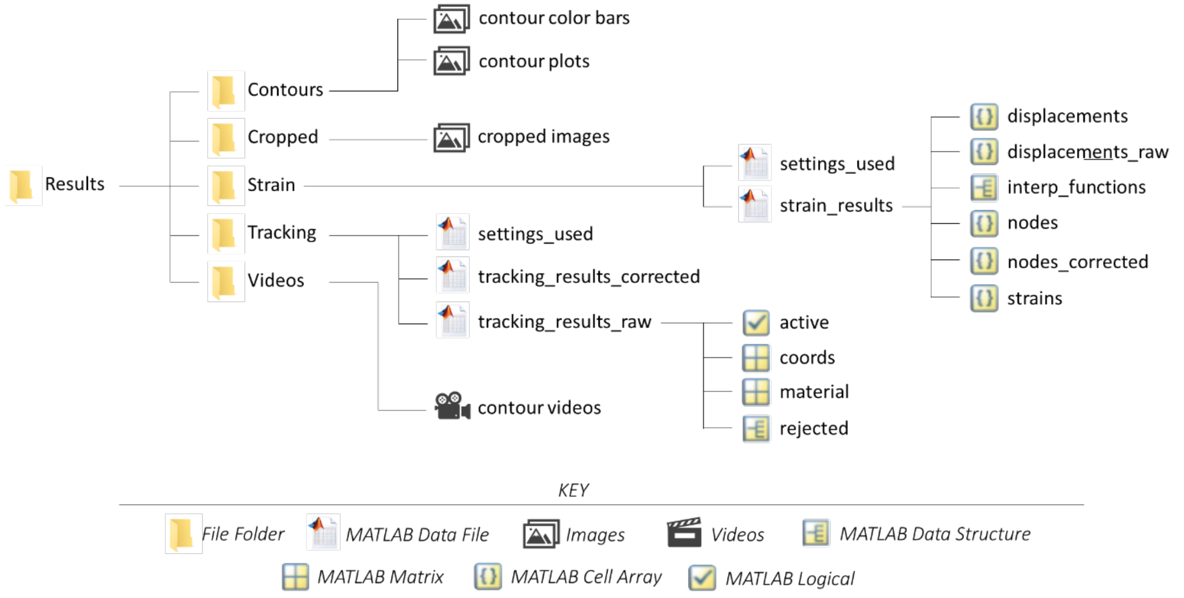


Figure A.2: Results Folder Data Structure

A.1.2.1 “CONTOURS” FOLDER

The “Contours” folder contains all of the contour plots and color bars generated for any given MODEM analysis being conducted. If no contours are created, the folder is not created. This folder can house one set of contours for every given displacement and strain value MODEM can analyze. So, at once, the folder can contain eight different sets of contours for the same image set pertaining to all of the displacements and strains that can be analyzed in a 2D MODEM analysis. If another set of contour plots are being generated for the same given value, then the previous contour plots and color bars created will be overwritten by the newer version. So, as an example, if a set of contour plots for strain in the y direction are created but then another set of contour plots in created that have a more opaque contour, then the previous contours generated will be overwritten. If only a few images in an image set are being created, then only those images will be overwritten.

A.1.2.2 “CROPPED” FOLDER

The “Cropped” folder contains a black image for every image in the image set corresponding to the size of the photos being analyzed by MODEM for a given experiment. This will contain the cropped characteristics of the images if the images input into MODEM that are cropped within. This folder also contains a file known as i*mage_data*. the contents of which are explained in Table A.12.

Table A.12: Contents of image_data Data File

VARIABLE NAME	VARIABLE TYPE	DESCRIPTION
BitsPerPixel	FPN	The bit depth of the images used in MODEM analysis
crop_mark	FPA	unknown
image_size	FPA	1x2 array containing the size of the images used in the MODEM analysis. The number of pixels in the X direction is in cell 1x1 while the number of pixels in the Y direction is in cell 1x2.

A.1.2.3 “STRAIN” FOLDER

The “Strain” folder contains all of the information users input in MODEM (a copy of the *settings* data structure described in Section A.1.1.2) and the MODEM generated from those inputs to compute the strain and displacement values of a given experiment. This strain and displacement data occurs in both its original form (when it was tracked) and its condensed form in cell arrays and matrices. The condensed forms specified in Section 5.7.2 are the ones researchers should use when post processing MODEM’s output.

A.1.2.3.1 SETTINGS_USED DATA FILE

The *settings_used* data file contains a copy of the settings data structure described in Section A.1.1.2. All information for this structure can be found there.

A.1.2.3.2 STRAIN_RESULTS DATA FILE

The *strain_results* structure can be found in the “Strain” folder which is nested within the “Results” folder. This data file contains the six entries shown in Table A.13.

Table A.13: Contents of strain_results Data File

VARIABLE NAME	VARIABLE TYPE	DESCRIPTION
displacements	CELL	Calibrated x & y displacements of nodes per image except for first image
displacements_raw	CELL	Raw x & y displacements of nodes per image except for first image
interp_functions	DATA	Receptacle for data used in computing strain calcs
nodes	CELL	Calibrated initial nodal locations in relation to image origin
nodes_corrected	CELL	Raw initial nodal locations in relation to image origin
strains	CELL	x, y, & xy Strains for every node in every image of image set except for the first image

A.1.2.3.2.1 INTERP_FUNCTIONS DATA STRUCTURE

The *interp_functions* data structure can be found within the *strain_results* data file. This structure contains the four entries shown in Table A.14.

Table A.14: Contents of interp_functions Data Structure

VARIABLE NAME	VARIABLE TYPE	DESCRIPTION
displacements	CELL	Calibrated x & y displacements of nodes per image except for first image
displacements_raw	CELL	Raw x & y displacements of nodes per image except for first image
strains	CELL	a nested series of cell arrays containing strain data for each tracking region throughout the image set
undeform_coords	CELL	a nested series of cell arrays containing the location of features throughout the image set

A.1.2.3.2.1.1 STRAINS CELL ARRAY

The *strains* cell array is a nested series of cell arrays found in the *interp_functions* data structure which is stored in the “Strain” folder. This cell array is created when MODEM computes the strains at each node based on the positioning of the features or “coords” it tracks.

The first level of this array, which is shown in Figure A.3, is a 1xN array where N equates to the number of tracking regions generated in a given experiment. If only one tracking region is generated, then only one cell will exist at this first level. This level is indexed by typing the desired cell (N) behind the name of the cell array in the following fashion: “interp_functions.strains{1,N}”. Within each cell of the array is a 1x3 cell array. This is considered the second level of the array.

The cell arrays that make up the second level of *interp_functions.strains* represent the strains calculated for the nodes in the selected tracking region (selected in level one). The xx strains, yy strains, and xy strains are contained within this level and are in the {1,1}, {1,2}, and {1,3} positions of the array respectively. Each array in the second level is 1xM in size. The M represents the number of images in the image set excluding the reference image. These strains are indexed by placing the location of the desired strain within the second layer (K) behind the indexing for the first layer: “interp_functions.strains{1,N}{1,K}”.

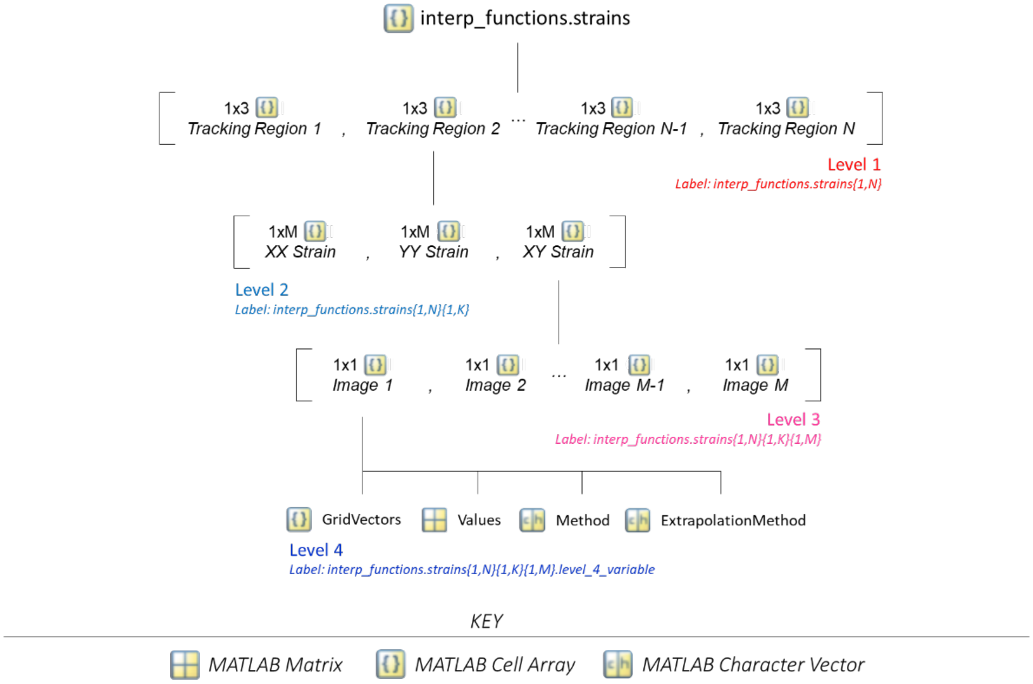


Figure A.3: interp_functions.strains Data Structure

The third layer of this cell array contains a 1x1 gridded interpolant for each image in the image set excluding the reference image. Each image can be indexed by placing the desired image within the image set (M) behind the indexing for the second layer: “interp_functions.strains{1,N}{1,K}{1,M}.

Within each of these gridded interpolants if the fourth layer of the cell array. This layer contains four variables: *GridVectors*, *Values*, *Method*, and *ExtrapolationMethod*. The contents of each variable is explained in

Table A.15. The variables within the fourth layer of the array can be indexed by adding the variable name of the variable being selected to the end of the indexing for the third layer in the following manner: “interp_functions.strains{1,N}{1,K}{1,M}.level_4_variable”.

Table A.15: Contents of interp_functions.strains Cell Array

VARIABLE NAME	**VARIABLE TYPE**	**DESCRIPTION**
GridVectors	CELL	1x2 array containing vectors of the node locations. The column one entry contains the X coordinates of all of the nodes while the column two entry contains the Y coordinates of all of the nodes.
Values	FPM	strain values for the given strain in question (based on the selected second layer entry) and the given image in question (based on the selected third layer entry). These strain values are oriented in a grid to correspond with the node they represent. The node locations, however, are inverted about the y = -x line.
Method	CHR	character vector that indicates what shape function was used to determine the strain value
ExtrapolationMethod	CHR	character vector that indicates which extrapolation method was used to determine the strain

A.1.2.3.2.1.2 UNDEFORM_COORDS CELL ARRAY

The *undeform_coords* cell array is a nested series of cell arrays found in the *interp_functions* data structure which is stored in the “Strain” folder. This cell array is created when MODEM computes the strains at each node.

The first level of the *undeform_coords* cell array is a 1xN array where N equated to the number of tracking regions generated as shown in Figure A.4. This layer is indexed by typing the desired cell (N) behind the name of the cell array in the following fashion: “interp_functions.undeform_coords{1,N}”.

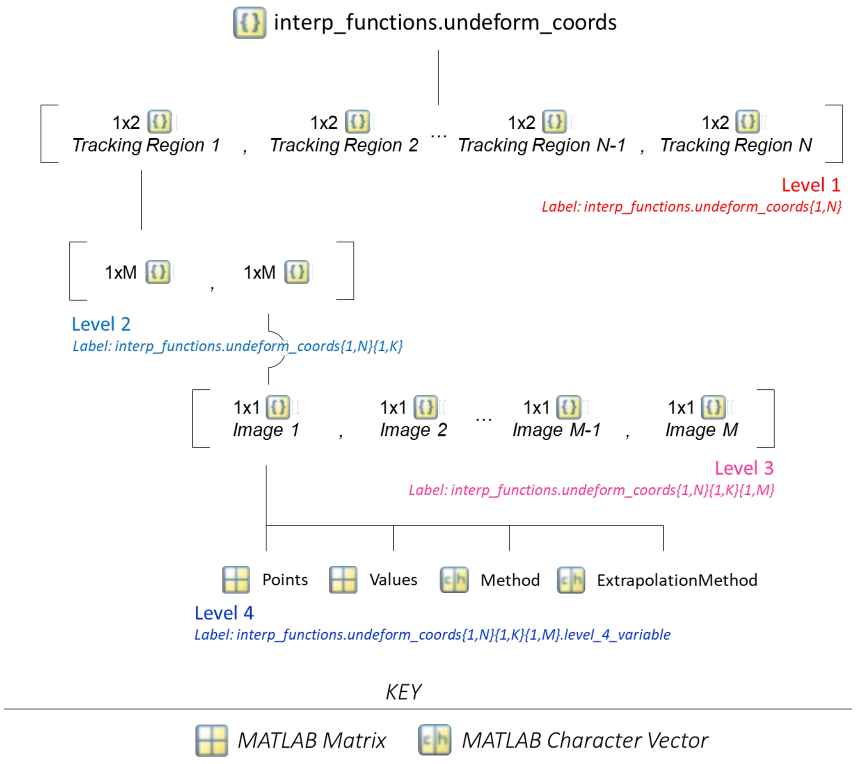


Figure A.4: interp_functions.undeform_coords Cell Array

The contents of each entry in the first level of the array constitute the second level of the array. Each entry consists of a 1x2 array and can be indexed by adding the desired entry of the second layer array (K) to the end of the first layer indexing in the following manner: “interp_functions.undeform_coords{1,N}{1,K}”.

Each entry of the second layer contains a 1xM array where M is the number of images in the image set excluding the reference image. These arrays make up the third layer of the interp_functions.undeform_coords cell array. The entries within the third layer of this array can be indexed by adding the desired entry in the third layer (M) to the end of the indexing of the second layer in the following manner: “interp_functions.undeform_coords{1,N}{1,K}{1,M}”.

Each entry in the third level contains a 1x1 scattered interpolant which constitutes the fourth level of the interp_functions.undeform_coords cell array. Each scattered interpolant contains four variables: *Points*, *Values*, *Method*, and *ExtrapolationMethod*. The contents of these four variables are explained in Table A.16. The contents of the fourth layer of the array can be indexed by adding the variable name of the variable being selected to the end of the indexing for the third layer in the following manner: “interp_functions.undeform_coords{1,N}{1,K}{1,M}.level_4_variable”.

The uses of this cell array as a whole are unknown.

Table A.16: Contents of interp_functions.undeform_coords Cell Array

VARIABLE NAME	VARIABLE TYPE	DESCRIPTION
Points	FPM	Mx2 matrix where M is the number of nodes in the tracking region selected in the first level of the array. The importance of the values included in the matrix are unknown
Vaules	FPA	Mx1 array where M is the same values as that in the *Points* matrix. The importance of the values included in the array are unknown.
Method	CHR	unknown
ExtrapolationMethod	CHR	unknown

A.1.2.4 “TRACKING” FOLDER

The “Tracking” folder contains all the information MODEM generated while tracking the subsets it identified throughout an image set. This folder contains three data structures: *settings_used*, *tracking_results_corrected*, and *tracking_results_raw*. These data structures are explained in Section A.1.2.4.1 through Section A.1.2.4.3.

A.1.2.4.1 SETTINGS_USED DATA FILE

The *settings_used* data file contains a copy of the settings data structure described in Section A.1.2.3.1. All information for this structure can be found there.

A.1.2.4.2 TRACKING_RESULTS_CORRECTED DATA FILE

The tracking_results_corrected data file is contained within the “Tracking” folder nested within the “Results” folder. This data file contains one entry shown in Table A.17.

Table A.17: Contents of tracking_results_corrected Data File

VARIABLE NAME	VARIABLE TYPE	DESCRIPTION
coords	FPM	Locations of the features in relation to the image origin calibrated based on the input calibration file. If not, this matrix is the same as the coords file in *tracking_results_raw*

A.1.2.4.3 TRACKING_RESLUTS_RAW DATA FILE

The tracking_results_raw data file is contained within the “Tracking” folder nested within the “Results” folder. This data file contains four entries shown in Table A.18.

Table A.18: Contents of tracking_results_raw Data File

VARIABLE NAME	VARIABLE TYPE	DESCRIPTION
active	LOG	logical matrix detailing, in a grid, which cords remain tracked throughout the image set
coords	FPM	locations of the tracking subsets in relation to the image origin
material	FPA	unknown
rejected	DATA	data structure containing information on rejected tracking subsets

A.1.2.4.3.1 REJECTED DATA STRUCTURE

The rejected data structure is found in tracking_results_raw data file located in the "Tracking" folder. This data structure contains two entries shown in Table A.19.

Table A.19: Contents of rejected Data Structure

VARIABLE NAME	VARIABLE TYPE	DESCRIPTION
quality	FPM	unknown
spacing	CELL	This is an empty cell array

A.1.2.5 VIDEOS FOLDER

The "Videos" folder contains all of the videos MODEM generated for a given image set based on user input. If no videos are generated, the folder is not created. This folder can only contain a maximum of eight videos, one for each type of contour that can be generated in MODEM for a 2D analysis. Just as with contours, if a new video is to be created using contours that have already been converted into a video, the video being generated will overwrite the previous video made using the same contours.

A.1.3 Quickly Referencing MODEM Data

The four files that are needed to post-process MODEM data are the *coords*, *nodes*, *displacements*, and *strains* files described in Sections 5.7.2.1 through 5.7.2.4. Information about what each file contains, how it can be used, and how it can be indexed through can be found in these sections. Their locations within the larger MODEM output structure can be found in Section A.1.2.3.2 for the *nodes*, *displacements*, and *strains* files while the *coords* file can be found in Section A.1.2.4.3.

Section 5.7.3.1 contains a MATLAB script that extracts the *coords*, *nodes*, *displacements*, and *strains* files "Job_Data" and "Results" folders. This script requires that the "Job_Data" and "Results" folders be in MATLAB's Current Folder and deposits the four files mentioned into MATLAB's Workspace.

APPENDIX B

B.1 EXPERIMENT 1M RAW DATA

The data collected from the traditional instrumentation during experiment 1M (the strain gage and extensometer) are shown in Table B.1.

Table B.1: Strain Gage and Extensometer Values from Experiment 1M

Image Number	Strain Gage Reading (in/in)	Extensometer Reading (in/in)	Strain Gage Reading / Extensometer Reading (%)
1	0.00000	0.00000	100.00%
2	0.00005	0.00005	102.04%
3	0.00008	0.00007	85.37%
4	0.00012	0.00012	103.45%
5	0.00015	0.00015	97.40%
6	0.00019	0.00018	93.26%
7	0.00023	0.00022	94.42%
8	0.00028	0.00027	97.47%
9	0.00032	0.00030	93.17%
10	0.00037	0.00035	93.83%
11	0.00042	0.00040	95.24%
12	0.00047	0.00043	92.27%
13	0.00051	0.00048	94.12%
14	0.00055	0.00052	94.89%
15	0.00059	0.00055	94.02%
16	0.00062	0.00060	96.31%
17	0.00066	0.00063	95.74%
18	0.00070	0.00067	96.40%
19	0.00074	0.00070	95.24%
20	0.00078	0.00073	94.19%
21	0.00082	0.00078	95.59%
22	0.00086	0.00082	95.57%
23	0.00090	0.00087	96.35%
24	0.00095	0.00092	96.84%

25	0.00100	0.00097	96.81%
26	0.00105	0.00102	96.77%
27	0.00111	0.00107	96.22%
28	0.00117	0.00113	96.33%
29	0.00124	0.00118	95.47%
30	0.00130	0.00125	96.30%
31	0.00136	0.00132	97.35%
32	0.00141	0.00137	97.03%
33	0.00146	0.00142	97.06%
34	0.00150	0.00145	96.41%
35	0.00155	0.00148	95.55%
36	0.00159	0.00152	95.60%
37	0.00163	0.00155	95.21%
38	0.00167	0.00158	94.72%
39	0.00171	0.00162	94.85%
40	0.00175	0.00165	94.50%
41	0.00179	0.00170	95.13%

This data shown in Table B.1 (which is also provided in an Excel spreadsheet with this document on Digital Commons) can be compared with the strains from MODEM utilizing the experimental photos. Readers can implement the MODEM analysis methods described in this document to test their ability to accurately run an experiment through the DIC program. An in-depth comparison between the results of the traditional instrumentation and MODEM results for Experiment 1M can be found in Section 6.2.8.

B.2 LIST OF PUCHASED ITEMS

Table B.2 is an example of a list of purchased items bought to create a low-cost DIC system for a university level structural engineering lab. This list is based off of the required materials detailed in Section 2 and is mean to be used by other researchers as an example for their own systems. This list uses the US Dollar as its currency and utilizes suppliers and websites that are found in the US.

Table B.2: Example List of Items to Purchase for a DIC System

Item Description	**Quantity**	**Unit Cost**	**Total Cost**	**Purchase Form**	**Link to Product**
Nikon DSLR 5500 Camera with 18-140mm lens	2	1,200.00 USD	2,400.00 USD	Nikon	https://www.nikonusa.com/en/nikon-products/product/dslr-cameras/1548/d5500.html
Camera Bag	2	20.00 USD	40.00 USD	Amazon	https://www.amazon.com/CADeN-Crossbody-Compatible-Mirrorless-Waterproof/dp/B07H9TBT5H/ref=sr_1_10?keywords=Nikon+camera+case&qid=
Camera Tripod	2	30.00 USD	60.00 USD	Best Buy	https://www.bestbuy.com/site/sunpak-platinumplus-5858d-58-tripod-black/1802042.p?skuId=1802042&ref=212&loc=1&extStoreId=396&ref=212&loc
Portrait Studio 60W Lighting Kit	1	50.00 USD	50.00 USD	Efavormart	https://www.efavormart.com/products/photography-photo-portrait-studio-600w-day-light-white-umbrella-continuous-lighting-
USB 3.0 USB Extender (2 Pack)	1	12.50 USD	12.50 USD	Amazon	https://www.amazon.com/Extension-Besgoods-Extender-Gold-Plated-Connectors/dp/B06XCB5234/ref=sr_1_7?crid=1T9ZV1RFMX6B2&keywords=us
Lens Filter for 52mm Dia. Lens (1 Pack)	4	6.00 USD	24.00 USD	Amazon	https://www.amazon.com/AmazonBasics-UV-Protection-Lens-Filter/dp/B00XNMWCF8/ref=sr_1_3?keywords=nikon%2Bdslr%2Blens%2Bfilter
SD Card [125GB] (2 Pack)	2	23.00 USD	46.00 USD	Amazon	https://www.amazon.com/SanDisk-Memory-Standard-Packaging-SDSDUNC-128G-GN6IN/dp/B0143IISD0/ref=sr_1_5?crid
External Hard Drive [1 TB]	1	47.38 USD	47.38 USD	Amazon	https://www.amazon.com/Toshiba-HDTB410XK3AA-Canvio-Portable-External/dp/B079D359S6/ref=sr_1_5?crid=8JA48EUO7DAX&keywords=externa
		TOTAL	**2679.88 USD**		

B.3 CHECKLISTS

The checklists found in this section are designed to provide experimentalists with some basic steps and parameters to consider when a DIC experiment is being run. All experiments and lab facilities are different some of the steps or parameters stated in these checklists might be unnecessary. Section B.3.1 is an example of a checklist for a tensile experiment using a thin aluminum specimen. Section B.3.2 is an example of a checklist for a compressive experiment using a concrete cube as its specimen [4].

B.3.1 Example Aluminum Tensile Test Checklists

Order for Powering Up the Tinius Olsen and its Accessories

1) Connect extensometer to Tinius Olsen
2) Turn on strain gauge input box
3) Turn on displacement box
4) Turn on DAQ
5) Wait 30 seconds and then turn on the Tinius Olsen
6) Wait 30 seconds and then turn on the computer

Calibration Images

☐ Use rectangular checkerboard as target.
☐ Make sure the static targets and displacement targets are not in the calibration images so that they do not get the calibration software confused between the targets and the checkerboard.
☐ Vary checkerboard placement in three dimensions in calibration photos.
☐ Make sure the calibration board in completely within the image.
☐ Make images JPGs to minimize memory storage used.
☐ Check all calibration images after taking series of images to verify they are in focus and meet the parameters above.

Test Setup

☐ Place static targets on Tinius Olsen to track machine vibration.
☐ Make sure the tripod is stable. Add weight if necessary.
☐ Mark tripod location on the floor for future tests so that camera position is a constant variable.
☐ Make sure the jaws are in the correct position for the tension test and have the correct plates (flat plates) within the jaws to properly engage with the specimen.
☐ Place a piece of electrical tape or painter's tape on the back of the specimen next to the strain gage to indicate the location of the strain gage without blocking out the speckle pattern on the other side of the specimen.

Prior To Experiment

☐ Measure camera placement in relation to specimen to corroborate calibration results:
    Distance Between Camera Lens and Specimen: ____________________
    Distance Between Camera Sensor and Specimen: _________________
☐ Take photo to test lighting before experiment and to have an image at t = 0.
☐ Change camera parameters from defaults settings to the following ranges:
    White Balance: [Incandescent] __________________
    F-Stop: [~F8] __________________
    Shutter Speed: [1/30 to 1/100]: __________________
    ISO: [500 to 1000] __________________
    Picture Control: [Monochromatic] __________________
    Zoom: [140mm or Max] __________________
  Take a test picture to verify that these are the camera parameters.

☐ Record all relevant camera settings in a log.
☐ After changing and recording the camera settings, then connect the camera to the computer.
☐ Measure the dimensions of specimen before conducting the experiment (if applicable).
☐ Record the distances between any given reference points on the specimen (red dots).
☐ Check the quality of the camera image to verify the specimen is in focus.
☐ Do not let computer go to sleep. Connection between Tinius Olsen and computer may be lost.
☐ Take out the extensometer pin before running experiment but after attaching it to the specimen.
☐ Conduct a brief elastic test before actual test to verify instrumentation is working properly.
☐ Make sure to be careful with the strain gages. They can easily be ripped off. Before running the test make sure the strain gage(s) are completely intact.

Experimental Procedure

7) Screen record.
8) Start Tinius Olsen machine.
9) Start delayed image capture and stopwatch on screen simultaneously.

Post-Processing

☐ Verify there is no external movement between the images.
☐ Plot strain and extensometer data before conducting MODEM analysis to see if it is viable data.

B.3.2 Example of a Concrete Compressive Test Checklist

Test Setup

☐ Measure the dimensions of the specimen
☐ Place specimen in testing machine
☐ Connect instrumentation to data acquisition (DAQ) box
☐ Place plywood screen around testing machine
☐ Set up tables, laptops, extension cords and DIC Camera
☐ Measure camera placement in relation to specimen (x, y, z) to corroborate calibration results and record readings in excel file
☐ Close lab doors
☐ Set up lighting and adjust camera settings: white balance, F stop, ISO, auto features off, and record settings in excel file
☐ Take series of photos to test tether software and verify there is no external movement between images, delete photos, reset tether photo counter
☐ Set up Windows computer webcam
☐ Set up Windows DAQ program and initialize bias (take out extensometer pin before bias is run)
☐ Run image calibration

Calibration Images

☐ Vary checkerboard placement in three dimensions
☐ Make sure static targets are not in the calibration images
☐ Make sure the calibration board is completely within the image

Prior To Experiment

☐ Place Static Targets on testing machine
☐ Check lighting on webcam
☐ Record max load/increment and image capture time interval
☐ Take photo of test set up with doors closed
☐ Ensure webcam can see speed dial

Experimental Procedure

- ☐ Set compression test head (red light turns off)
- ☐ **Start** phone stopwatch and Laptop 1 webcam
- ☐ **Lap** stopwatch and **start** Laptop 1 (DAQ)
- ☐ **Stop** stopwatch and **start** Camera interval shooting
- ☐ After first image has been taken, **start** test machine
- ☐ After interval shooting has finished, stop test machine, video recording, DAQ readings

Post-Experiment

- ☐ Use PhotoScapeX to batch convert images to JPEG and save
- ☐ Upload images from Mac computer to flash drive
- ☐ Save webcam recording and .txt file from Windows
- ☐ Upload images from flash drive to Windows computer
- ☐ Upload images to MODEM on Windows computer