MEASURING FULL-FIELD DEFORMATION OF HYPERELASTIC MATERIAL USING DIGITAL IMAGE CORRELATION

Sujeeka Nadarajah

198035H

Degree of Master of Science

Department of Civil Engineering

University of Moratuwa
Sri Lanka

Aug 2021

MEASURING FULL-FIELD DEFORMATION OF HYPERELASTIC MATERIAL USING DIGITAL IMAGE CORRELATION

Sujeeka Nadarajah

198035H

Thesis submitted in partial fulfillment of the requirements for the degree Master of Science in Civil Engineering

Department of Civil Engineering

University of Moratuwa

Sri Lanka

Aug 2021

DECLARATION

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

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

Signature: UOM Verified Signature Date: 22/03/2022

The above candidate has carried out research for the Masters thesis under my supervision.

Name of the supervisor: Dr. H. M. Y. C. Mallikarachchi

Signature of the supervisor: UOM Verified Signature Date: 22/03/2022

ABSTRACT

Digital image correlation, an optics-based strain measuring technique, has gained popularity during the last decade. The practice of employing digital image correlation-based measuring techniques in experiments has progressed dramatically owing to many incorporated sophistications: ease of use, greater precision, extensive measurement range, and stability of the results. While cutting-edge DIC measurement systems are commercially available, their high capital costs make them unaffordable for widespread usage. Several studies on establishing DIC-based measurement techniques with experimental validation have been undertaken over the years. The majority of prior DIC research concentrated on the testing of concrete, masonry, and metal alloy specimens, and little effort was made in assessing materials with substantial elongations.

This dissertation presents an innovative DIC-based measuring tool for capturing surface deformation information with better precision. The proposed system is composed of two commonly accessible digital cameras as well as MATLAB-based algorithms. Uniaxial tensile tests of latex specimens are used to demonstrate the aptness of the proposed approach in capturing substantial deformations, and displacement estimates are validated against Vernier Caliper readings. A secondary high-precision measuring tool, the Ncorr program, was used to validate the integrity of the results produced by this technique.

According to the comparison of results, it is proven that the system is capable of producing precise full-field displacement and strain maps with an apparent accuracy greater than 96 %. Furthermore, the similarity of the contour plots obtained using the proposed approach and outcomes of the Ncorr program verified the reliability of the proposed technique.

Keywords: Digital Image Correlation, 3D reconstruction, surface deformation information, latex

ACKNOWLEDGMENT

Accomplishment of the objectives of this research is not possible without the guidance and comments given during the progress. Therefore, I would like to express my deepest gratitude to all the individuals who helped to complete the research successfully.

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

I obligate to thank Professor W.P.S. Dias, Dr. R. Rodrigo, Dr. C. S. Lewangamage and Professor Indika Perera for the constructive comments, valuable advice, guidance, and support in the success of this research and report. My sincere thanks to Dr. J. C. P. H. Gamage and Dr. Nimal Wijeratne, Research Coordinators, Department of Civil Engineering, University of Moratuwa, for their supportive guidance, providing necessary guidelines regarding the research project evaluation.

Furthermore, sincere thanks to the staff of the Department of Civil Engineering, University of Moratuwa for the support and guidance provided towards this research. I extend my gratitude to all previous researchers, whose contributions I have made use of in carrying out this research project.

Last but not least, I am immensely grateful to many individuals including my friends, who provided me with the digital cameras and mobile phones and supported in experimental procedures, non-academic staff in the laboratory for helping me in setting up the experimental setup, and my colleagues for assisting and providing feedback on the output in various ways throughout the research. I would like to make this an opportunity to express my sincere gratitude towards them all. Thank you.

N. Sujeeka

Department of Civil Engineering

University of Moratuwa

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

DIC – Digital Image Correlation

FIB - Focused Ion Beam

WTP - Water Transfer Printing

SSD - Sum of Squared Difference

FAST - Features from Accelerated Segment Test

SIFT - Scale Invariant Feature Transform

SURF - Speeded-Up Robust Features

BRISK - Binary Robust Invariant Scalable Keypoints

PNG – Portable Network Graphics

JPEG-LS - Joint Photographic Experts Group - Lossless

CCD – Charge – Coupled Device

CMOS - Complementary Metal–Oxide–Semiconductor

SGM - Semi Global Matching

RoI – Region of Interest

NaN – Not a Number

LAR - Least Absolute Residuals

SSE - Sum of Squares Error

RMSE - Root Mean Squared Error

SSR - Sum of Squares of Regression

SST - Sum of Squares Total

RANSAC - Random Sample Consensus

LIST OF APPENDICES

1. INTRODUCTION

1.1. Overview

The accurate measurement of strain is an important aspect of material testing. The mechanical properties of materials are often represented as a stress-strain curve, and understanding the stress-strain curve helps engineers to compare various materials and anticipate how a part or structure manufactured from a specific material will respond during processing and service. It's also important in Low-cycle fatigue testing, which determines the resilience of materials subjected to cyclic loading during operation.

There is a wide range of strain measurement techniques that may be classed as contact systems and non-contact systems, depending on the interaction of equipment with the object being tested. However, local practice is limited to contact-based approaches mostly involving measuring probes, strain gauges, and extensometers. Strain measurement with contact-based methods is challenging for many reasons. Depending on the material and the test, the equipment must be able to measure a wide range of deformations, ranging from a few millimeters when testing a stiff material such as metal or composite to over a meter when testing an elastomer. It's also crucial that the equipment does not affect the material being tested. Physical contact with the specimen can lead to premature failure in some cases. Equipment must be able to withstand the shocks of frequent specimen breaks, in addition to providing extremely precise measurements.

Various studies have been conducted on addressing the aforementioned challenges, and numerous techniques have been devised in recent years. These methods can be further classified as sensor-based and optics-based techniques. Non-contact-based deformation measuring techniques such as GPS, laser scanning, close-range photogrammetry and videometry, and radar interferometry system are commonly employed for short- and long-range measurements [1], [2]. Among these, the techniques suitable for the short-range measurements as in laboratory experiments are the laser scanning and optical-based measuring system.

Laser scanning is a form of digital modeling technique that uses laser light and basic mathematics to build realistic 3D reconstructions. A laser point or a laser line is aimed towards the body to be scanned, depending on the 3D laser scanner. Then, using either triangulation or time of flight, specialized software calculates and records data points where the laser makes contact with the object's surface. Cartesian coordinates are used to represent these positions (X, Y, and Z planes). This technology may be used for reverse engineering, quality control and assurance, dimensional analysis, and surveying. Although this procedure is precise, rapid, and safe, there are some drawbacks to it that make it unsuitable for experimental studies. Ambient light can interfere with the accuracy of 3D laser scanners since they acquire data by reading the light of a laser. Depending on the level of interference, the scan may be noisy or even unusable. The cost of laser scanning equipment is high. A typical industrial 3D laser scanner costs between $50,000 and $150,000. Optics-based measuring techniques on the other hand have progressed greatly owing to the numerous advantages it possesses such as ease of testing, higher precision over the entire specimen, wide measurement range, and strong stability of measurements [2]– [19]. The use of optical measurements facilitates capturing the complete surface deformation information, unique to each surface point.

Digital Image Correlation (DIC) is an optics-based measurement approach that has the potential to be an excellent solution. It entails recording images of the object under consideration, storing them in digital format, image processing, and full-field displacement computation. A set of calibration images is collected in this approach to determine the camera locations and certain other parameters needed for subsequent processing, which will be utilized for displacement computation [20].

1.2. The state-of-the-art facilities and recent developments

There are numerous commercial DIC systems available, consisting of software packages and dedicated hardware. Correlated Solutions, one of the market's top providers of DIC solutions, offers software like VIC 2D and VIC 3D for 2D and 3D analysis respectively. These devices can generate in-plane and out-of-plane full-field displacement and stresses in specimens ranging in size from microns to meters. These systems have integrated image acquisition software, outstanding speed and resolution for short image sequences, substantial internal image storage for higher frame rates, and a sophisticated interface for ultra-fast image downloading and cost-effective integration. These systems are widely utilized in academia and various industries in vibration, infrared, microscopic, and high-speed measurements.

Josts Engineering, another renowned DIC solution provider, offers full-field and three-dimensional dynamic measurement of shape, displacements, and strains with the DIC Q 400, DIC Q 450, and DIC Q 480 systems. The system's versatility allows it to be used for a wide range of tasks, from microscopic studies of microelectronic or biomedical materials to large-scale assessments of aeronautical, automotive, and marine elements. However, the costs of these commercial packages have become incredibly expensive (ranging between 35 000 and 60 000 USD) and unaffordable in the local context.

In recent years, numerous studies have been conducted to develop DIC-based algorithms employing various programming and numeric computing platforms. The majority of the studies used experimental validation to determine the accuracy of the technique proposed. Most of the investigations looked into concrete cubes and masonry stones, as well as examining metal and metallic alloy specimens. However, the applicability of the proposed DIC techniques in studying rubber-like materials, which have a wide range of uses is underexplored.

1.3. Scope and aims

The use of commercial DIC systems for deformation assessment in the local community is very unrealistic, as such systems incur substantial capital costs. The research works that have been conducted in the past years in efforts of establishing an open source DIC system, including the development of software in various programming languages and experimentation with broad range of materials to ensure its reliability, have been presented in the extensive literature review in previous chapters. However, the study on usage of DIC in measuring substantial deformation like in hyperelastic materials need to be investigated further.

Therefore, primary objective of this study is set to formulate an algorithm that facilitates 3D full-field deformation measurement of specimens with greater accuracy in a cost-effective manner. And verify its applicability and reliability in measuring substantial deformations as in hyperplastic materials. Initiative of this preliminary design tool will lead to a powerful, low cost, user-friendly 3D DIC system that will be available for a wide range of applications.

To attain this status, the following sub-objectives were formulated:

- Image acquisition using readily available digital cameras – One of the reasons for the high capital cost is the advanced cameras that are included in commercial DIC kits. Commonly accessible digital cameras, such as mobile phone cameras or low-cost CCTV cameras, were suggested to be included into the design to minimize the cost of the proposed system, which should be sufficient for short-range DIC experiments.
- Develop an algorithm using a multi-paradigm programming platform: MATLAB - MATLAB is a high-performance computing language. It combines computation, visualization, and programming in a user-friendly interface. The built-in MATLAB computer vision toolbox applications make constructing codes for the image analysis step a lot easier.
- Experimental validation of the results - The verification of the source codes by experimental validation is the foundation of confidence in the proposed system. A validation experiment is required in order to make more quantified claims about the amount of agreement between computational and experimental results. The specifics of the experiments included in the study are presented in chapter 3.1.
- Comparison of the results with well-established open-source DIC algorithm: Ncorr - The study also includes a comparison of results with a widely used DIC system as a last step in ensuring the system's accuracy. Most of the 3D DIC systems are expensive. Consequently, image analysis was performed using Ncorr, a well-established 2D DIC system, and the results were compared to the proposed system's 2D analysis results.

1.4. Chapter organization

This thesis is organized into five chapters. Following the current introductory chapter, Chapter 2 presents extensive literature on the progress of DIC techniques. The first section of the chapter discusses the history, early studies conducted, and evolution in this field, while the second section describes studies related to the basic components and functions of DIC: speckle pattern and image matching algorithms that have been developed over time. This chapter also includes applications of DIC in various experiments. Chapter 3 is divided into two sections. The first section presents the details of the experiments conducted. The proposed technique is sequentially demonstrated in the second section. The major components of the proposed technique, related theories, and important aspects to be considered during the process are explained in depth in 9 subsections.

The deformation measurements obtained using the proposed method and the experimental data are compared in tabular form in Chapter 4. The full-field deformation maps produced for each loading and unloading step are also presented in this chapter. Furthermore, the authenticity of the proposed method is proved by comparing the colormaps produced by this method to those obtained using a well-known open-source DIC technique. Chapter 5 presents a summary of the proposed technique followed by conclusions. Finally, some recommendations for further development of the proposed method are discussed.

2. LITERATURE REVIEW

This chapter presents an overview of the literature on DIC techniques. The chapter begins with the history and important studies conducted in early times. Next, a review of important aspects of DIC and related studies is presented. The last section describes the current state of the art and development in recent years.

2.1. History and early studies

The notion of measurement through images was initially established in the field of photogrammetry. However, evidence for the usage of image and viewpoint concepts in the 1480s can be found in the literature [20]. In 1961, Gilbert Hobrough published the first study on DIC. He demonstrated an analog image correlation concept, which includes comparing the gray levels of an image using hardware. Later the same year, in 1961, he worked on combining high-quality reconnaissance and survey images to facilitate a precise measurement of the changing terrain [21].

Digital images became widely accessible in the 1970s, motivating computer science and robotics researchers to develop stereo-vision concepts and vision-based algorithms, along with photogrammetry applications. There is a wealth of literature on the advancement and applications of DIC techniques over the years. Most studies that involve experiments in the field of solid mechanics were focused on using different speckle application techniques. Yamaguchi published a paper in 1981 on deformation and reduced autocorrelation of speckles in image fields for small object deformation. He evaluated the cross-correlation functions in the deformed and undeformed states to explore the displacement of the laser speckle pattern accompanied by structure change. The observed correlations were physically interpreted by adding a small change in the diffraction rays [22].

In 1982, an intriguing early study was carried out by Peters and Ranson on the development and the use of the idea of digital image methods for stress analysis. The work included a computer-connected image scanner that records a laser speckle pattern of an image in the reference and current deformed states. In calculating displacements, small segments of the deformed image termed as subsets were numerically correlated with the reference images. The authors recommended using the fundamental concepts of continuum mechanics that control smaller regions in the "matching process" as a part of this strategy [23]

In 1983, Sutton et al. employed this methodology to formulate a more sophisticated DIC method to acquire complete in-plane surface deformation information of the target. In this approach, the deformations are estimated by mathematically correlating the intensity of the subsets in the undeformed state image and the results were proven to accord with theoretical calculations. The well-known 2D DIC is the result of this method [24].

The inability of 2D DIC approaches to measure out-of-plane motion or deformation was well understood, thus the development of 3D DIC techniques was prioritized. Frederic and Olivier devised a method for obtaining the characteristics of 3D forms without using 3D models in 1994. Estimation of binocular stereo disparity is a fundamental idea introduced in that paper. They presented a novel approach for extending the conventional correlation method to properly estimate the disparity and derivative directly from image data, which was then utilized to extract surface characteristics like curvatures and orientations [25].

2.2. Digital Image Correlation

Computer vision is a branch of artificial intelligence and computer science that attempts to offer machines a better visual knowledge of the environment by analyzing a set of digital images. It deals with the automated acquisition of images, processing, and the subsequent interpretation of usable information from digital images. Image matching, which includes DIC, is a field of computer vision that is important in a wide range of applications.

DIC is an optical-numerical measuring technique that uses a mathematical correlation analysis to investigate a set of digital image data. The DIC method compares and matches the grayscale intensity of images taken at different perspectives and times, and various levels of deformation. The DIC approach can calculate the deformation, geometry, and motion vector fields, as well as associated gradient maps, in 2D and 3D spaces by tracking every few pixels in the region of interest in the reference and current state images, then performing data interpolation [26].

2.2.1. Speckle pattern

This technique utilizes the randomized intensity fluctuation including an image's basic features, such as lines, grids, points, and random textures. Surfaces without apparent distinct texture should be coated with a randomly spaced dot pattern termed as a speckle pattern. A good speckle pattern should be random in nature to guarantee that different spots exhibit distinct patterns. Speckles of 3-7 pixels in the image are recommended. The application of the pattern of the speckle should not influence the object's characteristics and should stick to the surface and deform with the body [27], [28].

There are several ways for creating random patterns, such as spin-coating, lithography, Focused Ion Beam (FIB), etc, but the most popular approach is the use of an airbrush or spray can [29]. In general, black paint is sprayed over a white or light-colored surface to provide a high contrast design [30], [31]. In scenarios with large deformation, speckles may detach from the surface of the specimens. In such situations, ink markings would be the simplest and finest option.

It is not always viable to rely on air brushing or spraying since these techniques are time-consuming and may not be appropriate in some contexts. Using modern technology, several ways for applying random speckle have been developed. Zenning et al. presented a simple, repeatable Water Transfer Printing (WTP) approach in 2015 to eliminate artificial measuring errors. Water, brush, and a piece of transfer paper composed of prefabricated decal paper, a protected sheet, and print of speckle patterns are used in this process. Computer simulations were used to develop and optimize the speckle patterns, which were subsequently printed on decal paper. It was also demonstrated to be capable of retaining coherence and adhesion to the specimen surface, as well as producing precise displacements [32]. Recent developments and applications in laser speckle and electronic speckle patterns are well documented in the literature [33]–[37].

Over the years, several kinds of research have been conducted to determine the optimal speckle pattern. In 2007, Lecompte et al. investigated the efficiency of a random speckle pattern and its impact on observed in-plane displacements as a function of subset size. To investigate the impact, four distinct speckle patterns with different dot sizes were deformed in a controlled manner. It was concluded that the size of the speckles, as well as their spectral content, when paired with the size of the subset, has a significant impact on the accuracy of the measured displacements [38]. Many additional studies on the impact of speckle characteristics on computing precise displacements have been published in the literature [39]– [43].

2.2.2. Image analysis

The correspondence problem is the first step in calculating deformation. It is a subject of identifying which sections of the first image correspond to which sections of the second image, where the differences are caused by the deformation or motion of the object captured. In general, it is not possible to determine the correlation between individual pixels in consequent images since the intensity value of a single pixel in the reference image is often found in hundreds of other pixels in the current state image. It's achieved by dividing the reference image's RoI into numerous subsections called subsets and locating their positions in the current image. Several algorithms for solving the correspondence problem have been developed over the years.

The template matching process, pattern recognition, feature matching, Hough transformations, the RANSAC algorithm, the optical flow-based approach, up sampled cross correlation, and orientation code matching, etc., are some of the fundamentals to the image analysis process [44]– [48]. Template-based matching and feature-based matching are the most often utilized categories among them.

Template-based matching is a powerful tool that detects image segments that match a specified template. These approaches are versatile and simple to implement, making them one of the most widely used ways of object localization for displacement measurement. When dealing with complicated templates, the aptness of template matching is restricted by the capacity of the computer used, as it is time-consuming.

The fundamental approach of template-based matching involves comparing the current image to a subset of the reference image (template). Numerical measures of similarity of the template and its overlap, cross-correlation, and sum of squared difference (SSD), are calculated when the template is moved pixel by pixel over the current state image.

This method is simpler to implement compared to feature-based matching. However, scale changes and rotations have an impact on the algorithm's efficiency [49]. In recent years, numerous algorithms have been created to address the shortcomings of the original template matching algorithms, as well as to improve the speed and accuracy, and to tailor it to specific applications [50]– [54].

A feature is a component of an image that is useful for addressing computational problems. The presence of a large number of such local features in an image offers crucial information. They're image patches that differ in texture, color, or intensity from the surrounding region. Edges, corners, blobs, and ridges are examples of different structures found in an image. Image registration, object identification and classification, tracking, motion prediction, and content-based image retrieval are some of the applications.

Feature-based matching algorithms use these local features to better handle scale and, rotation changes in images. These algorithms include three main steps: feature detection, extraction, and matching. Feature detection identifies areas of an image with distinct content. The basic idea of feature-based detection is discovering features that are locally invariant, meaning they can be identified even when rotated or scaled. Computation of a descriptor, which is generally done on areas centred around identified features, is the first step in feature extraction. To convert a local neighbourhood of a pixel into a compact vector representation, descriptors rely on image processing. This new representation makes it possible to compare neighbourhoods despite differences in scale or orientation [55].

Various detection algorithms for specific features have been developed over time. A corner is a typical low-level feature that is defined by the intersection of two lines and displays a substantial intensity variation in an image. With perspective changes, corner detection usually acquires more stable characteristics than patching technique across changes of viewpoint [56]. The Harris and Fast algorithms are the most frequently utilized corner detection techniques. Harris technique, developed by Harris and Stephens in 1988, is a combination of corner and edge detection algorithms that extracts stability and low self-similarity points [57]. Rosten and Drummond proposed the Features from Accelerated Segment Test (FAST) method in 2006. This technique compares pixels only on a circle of fixed radius around a point as illustrated in Figure 1. A point is detected as a corner only if there exist a huge number of pixels on the abovementioned circle around the point are all notably darker or brighter than the centre [58].

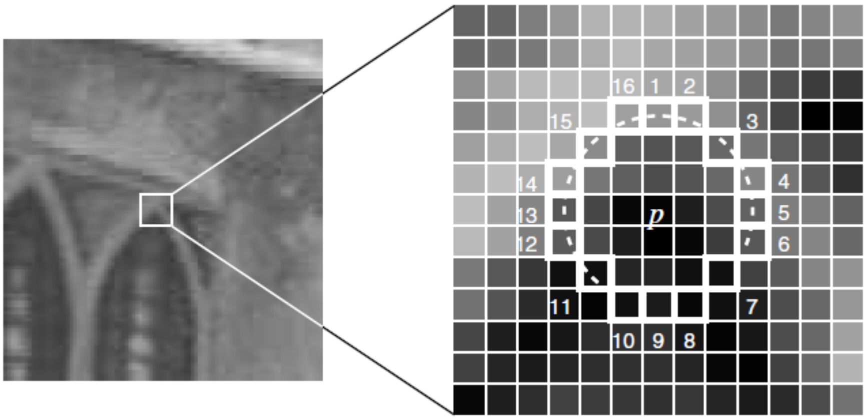


Figure 1: Corner detection using FAST algorithm [58]

Lowe suggested the Scale Invariant Feature Transform (SIFT) method in 2004 for extracting unique invariant characteristics from images. SIFT is widely utilized in applications such as object identification, robot localization, and mapping because it is simple to extract, distinct, and resistant to changes in light, noise, and small perspective modifications [59].

Herbert Bay et al. presented the Speeded-Up Robust Features (SURF) technique in 2006. It is a fast and reliable technique for representing and comparing images in a local, similarity invariant manner. The highlight of the SURF method is its ability to compute operators quickly using box filters, allowing for real-time applications like tracking and object recognition. The Hessian Matrix is generated on every pixel to extract SURF features, which is then followed by a non-maximal suppression process [60], [61].

Leutenegger et al. presented a recently developed well-established feature point detection and description method namely Binary Robust Invariant Scalable Keypoints (BRISK) with scale invariance and rotation invariance in 2011. The findings showed that the proposed method performs equivalent to the well-known SURF and SIFT methods, but at a lower computational cost [62].

Pablo et al. introduced KAZE in 2012 as a new multiscale 2D feature identification method in nonlinear scale spaces. To find and describe features at multiple scale levels, previous techniques approximated an image's Gaussian scale space. Gaussian blurring, on the other hand, ignores natural object boundaries and smoothes both features and noise to the same extent, diminishing localization accuracy and uniqueness. In this technique, nonlinear diffusion filtering was used to discover and characterize 2D features. The results indicated that while this approach is more computationally expensive than SURF, it outperforms the existing state-of-the-art methods [63].

The feature detection and description algorithm are chosen solely on the basis of the data's intended application and nature. Authors have studied the performance of these algorithms in comparison to one another for various applications in the past few years. Oliveira et al. have published a detailed comparison of the SURF, ORB, BRISK, and SURF-Harris algorithms when applied to acoustic data gathered under water, based on the performance of respective feature detection and description processes. Several metrics were used to conduct the study, including the number of features discovered, the total number of associations conducted, and the number of erroneous connections discovered [40].

Anjana et al.'s study in 2017 compared two robust techniques, SIFT and SURF in image mosaicking. It also provides a method for extracting invariant characteristics from images that may be used to match various perspectives of the same scene or object [41]. Several comparative studies assessing the performance of such algorithms have been reported in recent years [64], [65].

2.3. Applications of Digital Image Correlation

DIC has evolved in numerous ways throughout the years as a result of enormous advancements in computer science and optical technology. Several studies have been conducted to create DIC-based methods, and accuracy has been determined by experimental validation. The DIC method has been applied to both laboratory and field research.

In a study carried out by Lecompte et al. in 2006 two optical measuring techniques were developed for the detection of cracks on the surface of a concrete beam that was deformed under flexible circumstances of stress. The first approach was based on the comparison of several digital images obtained with a speckle pattern during the loading of the beam. The second method was the positioning of several diodes (LEDs) that emit lights, attached to the beam's surface. Displacements and strains were calculated using both techniques in a discrete number of points on the surface. It was found that, even before cracks became visually apparent, the presence and development of cracks may be detected using both approaches [66]. Similar research has been undertaken utilizing digital image cross-correlation to measure reinforced concrete beam displacement to study the in-situ behavior of cracks [67] [68], [69], [70], [71]

In 2007, Suat et al. employed image processing analyses to evaluate the compressive strength of the stones in the ancient construction of masonry. An artificial neural network based on image processing was used to assess 100 stones with varying characteristics. The results obtained through the proposed method were validated experimentally [72]. In 2015, Tiago et al. measured the in-plane and out-of-plane displacements and strains of masonry walls using the commercial software package Vic-3D. In the study, two separate tests were conducted. The first test was an in-plane shear test to replicate seismic activity effects and promote crack development, while the second was a quasi-static out-of-plane cycle test to trigger wall collapse. In both 2D and 3D studies, Vic-3D was used to get the in-plane and out-of-plane deformation fields [73].

Several pieces of research addressing the evaluation of metal specimens using image-based measuring techniques have been conducted. In 2013, Yu proposed an algorithm for DIC and utilized it to inspect medium and heavy steel plates. The findings showed that the method presented outperforms traditional measurement techniques [74]. In 2016, Andrew et al. published a fascinating study on the testing of thin sheet metal specimens used in the manufacture of beverage cans. The material was subjected to a process known as die necking, which involves changing the body such that it connects to both the top and bottom. The compressive force applied in all directions might induce wrinkling. Therefore, the sheet metal's characteristics were of interest. Conventional extensometers didn’t work since their weight caused significant deformations in the thin metal sheet. Vic-3D was used to measure the displacements, and the results were consistent for each batch [75]. Similarly, numerous research have been conducted on the use of DIC-based measuring techniques in the testing of metal specimens [76], [77], [78], [79].

Polymer composite materials have grown in prominence as a result of their superior performance and capacity to tailor mechanical properties. This advantage allows them to be used in a wide range of industrial applications, including aircraft, automotive, architecture, and marine. Deformation measurement of soft materials using conventional contact-based measuring instruments is challenging. In these cases, DIC-based measurement methods have been frequently employed as they give complete field deformation information and reliable results. A large number of authors have explored the use of the DIC technique to investigate behavioral aspects of polymer composite materials in the literature [80]– [85].

Rubber is a type of hyperelastic material that is utilized in a variety of applications in the industry. Rubber and rubber products are used in almost every major industry. These are used in situations when longer-term flexibility is required under heavy loads. Rubber materials have been employed in anti-vibration mounts, bearings, conveyor belts, drive couplings, and other engineering applications. Therefore, it is critical to investigate the behavior of hyperelastic materials under various loading situations, as the serviceability of rubber components must be monitored. DIC has almost solely been utilized to analyze the compressive and shear behavior of rubbers in previous researches [86]– [88]. Only a few papers present how DIC may be employed to capture large deformations in hyperelastic materials [89]. Therefore, the objective of our research is intended to develop and validate a 3D full-field deformation measurement technique for capturing large deformations in hyperelastic materials.

3. FULL-FIELD DEFORMATION MEASUREMENT

3.1. Experiment

This study is focused on developing an optics-based 3D full-field deformation measuring technique based on stereo-reconstruction and confirming its applicability in testing hyperelastic material. Five independent uniaxial tensile tests of three different latex specimens (60 mm x 20 mm x 0.21 mm, 60 mm x 60 mm x 0.21 mm and 20 mm x 60 mm x 0.21 mm) were carried out. The outcomes of the proposed technique were validated against experimental results and results of the 2D analysis were compared to the maps generated by the Ncorr program.

Speckle patterns of optimal size should be chosen in order to ensure that the maximum information content of the RoI is captured in the images. Concurrently, the speckle dots should remain adhere to the specimen when subjected to extreme deformations. In consideration of these facts, permanent marker pens were used to apply a medium-sized speckle pattern on the specimens. The specimens were loaded uniaxially in the vertical direction from one end while the other end was attached to the stationary frame of the apparatus. Image pairs of the RoI were captured at each step of the loading process when the specimen was in a stable state. For validation purposes, the length of the speckled region was measured using a Vernier caliper with a minimum count of 0.05 mm attached to a stable fixture in the apparatus for each deformed configuration.

In contrast with other materials, hyperelastic materials will deform substantially, which may cause difficulty in matching corresponding subsets in deformed and undeformed state images. Therefore, the image pairs were captured at small incremental loading steps and successive image pairs were considered for the correspondence problem. Figure 2 shows the deformation of a single speckle during consecutive loading steps.

Figure 2: Deformation of a speckle dot in subsequent loading steps

3.2. The proposed method

The overall design concept of the proposed system includes image acquisition of specimen and calibration grid, camera calibration, rectification, 3D reconstruction, refinement of the resulting point clouds, feature detection, estimation of geometric transformation, and generation of full-field displacement and strain maps. The framework of the proposed method is given in the form of a flow chart in Figure 3.

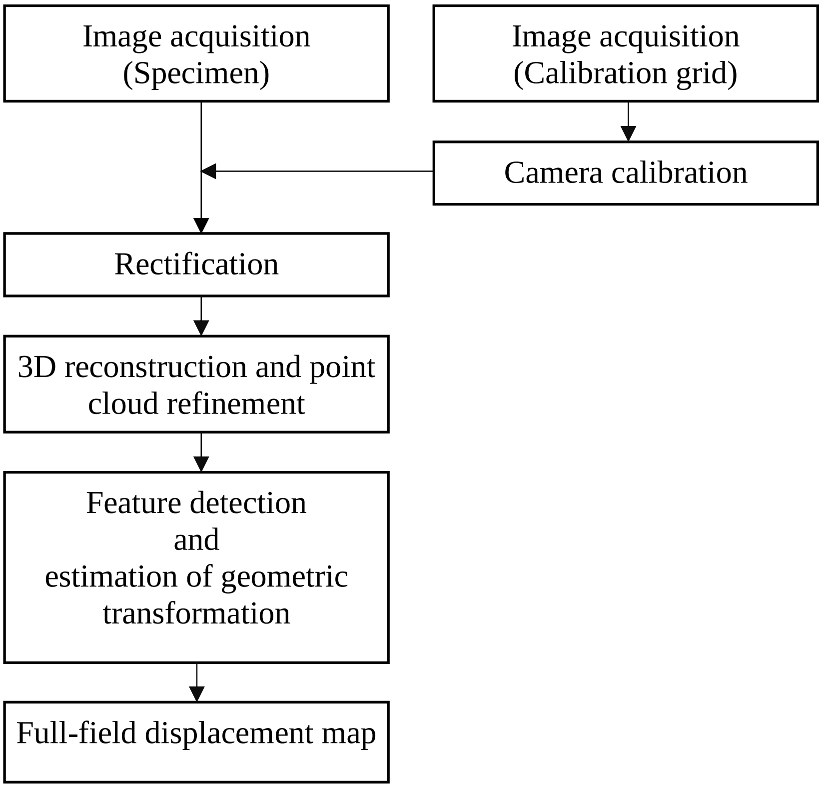


Figure 3: Framework of the proposed system

3.2.1. Image acquisition

The scene depth information should be accurately captured since the proposed system's goal is to capture the displacement in 3D space. The system consists of two digital cameras with equal resolutions and pixel densities installed using a rigid frame. The cameras are positioned so that the maximum overlap of the left and right image frame is achieved, enabling the camera's resolution to be effectively utilized for improved accuracy. When the position is established, cameras should not be disturbed, so external triggers are included in the system, to facilitate the image acquisition process. The autofocus function was avoided, and artificial lighting was used during the image acquisition process. The major components of the system and experimental setup are illustrated in Figure 4.

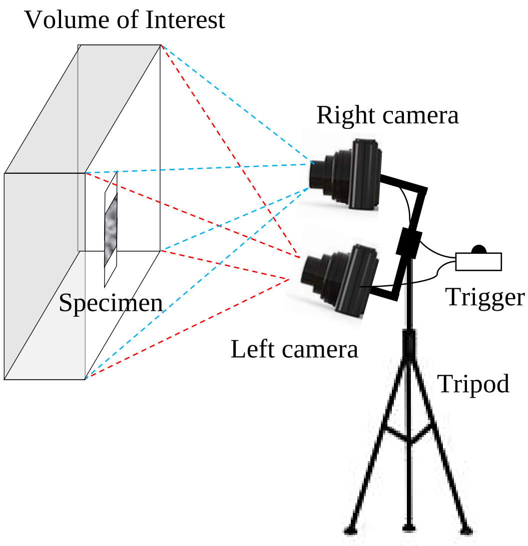


Figure 4: Experimental setup

Two sets of image pairs need to be acquired: calibration images and images of specimen for calibration process and computation of deformation respectively. The images of the specimen captured by the left and right camera at a particular loading step are shown in Figure 5.

(a) (b)

Figure 5: Images of specimen captured by (a) left camera and (b) right camera

3.2.2. Camera calibration

The subject of camera calibration has piqued the interest of the computer vision community, and various calibration methods have been developed over the years. It is the process of assessing a set of calibration images taken in various orientations to obtain the camera intrinsic parameters: internal geometrical and optical characteristics, as well as the extrinsic parameters: relative positions and rotation of cameras in the real space [28]. Stereo Camera Calibrator, one of MATLAB's built-in programs, was used to accomplish the calibration process.

This process requires a set of image pairs of a flat checkerboard pattern spanning the entire VoI in different locations and orientations as shown in Figure 6. It is preferred to store images in lossless formats such as PNG and JPEG-LS or any other uncompressed formats. The design of the checkerboard comprises even and odd numbers of squares on perpendicular axes, allowing the application to identify the plate's orientation. The calibrator program usually takes the longer side as X direction. To prevent unwanted warping effects that may be taken into consideration erroneously in the estimation of the stereo parameters, the surface of the pattern was pasted on a flat glass piece. The length of a single checkerboard square was measured (14.5 mm) that is needed for the calibration.

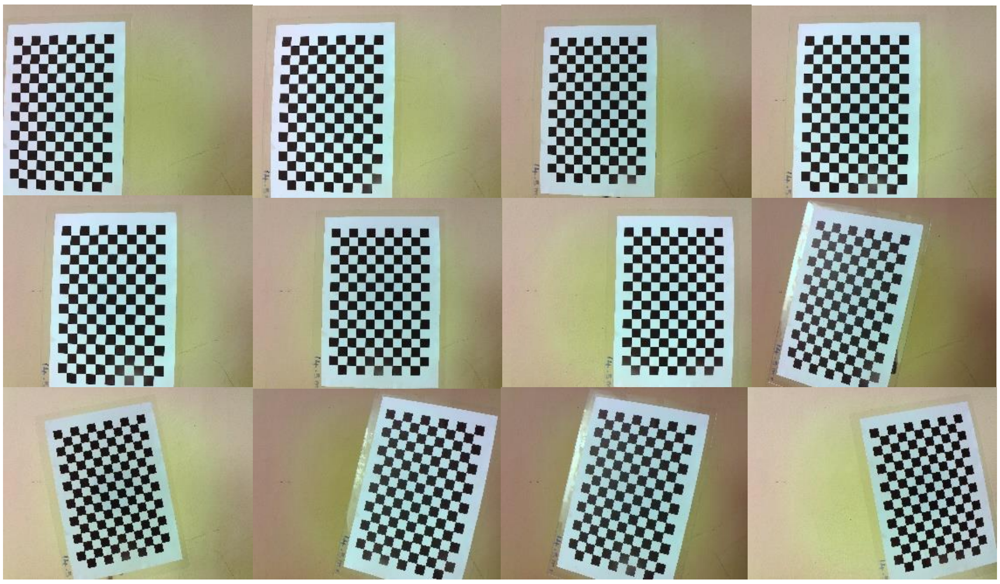


Figure 6: Checkerboard pattern in different locations and orientations

The pattern was focused very well and the autofocus function was disabled when the photos are captured. Furthermore, the zoom settings between photos were not adjusted as that will change the focal length. The distortion of the lens radially increases from the image center. Therefore, the images were also shot around frame boundaries to capture the distortion properly. In every location and orientation, the checkerboard pattern was held in a manner that it is completely visible in the image frame. Although three numbers of image pairs are enough for calibration, having more than 20 image pairs is preferred as the duplicated, distorted, and blurred images will be eliminated. More than 20 images were captured and used for the calibration process in all five experiments. And in order to obtain higher precision, the calibration grid orientations were maintained to have angles lower than 45° with the camera plane as depicted in Figure 7.

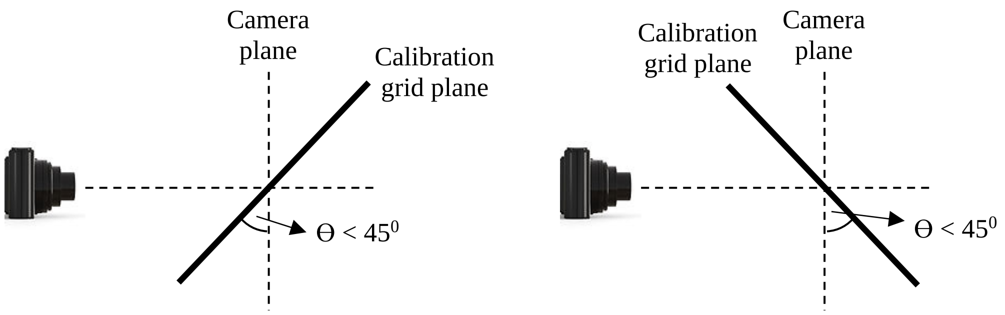


Figure 7: The orientation of the calibration grid with relation to the camera plane

The checkerboard square size should be inserted when the images are loaded into the application. The application tries to recognize the pattern of the checkerboard in every pair of images and rejects the low-quality images and the duplicates. The precision of the calibration process is evaluated by the resulting mean reprojection error, examining the camera extrinsic or viewing the undistorted image pairs. For greater accuracy, it is recommended to use all three methods [55].

The Stereo Camera Calibrator application detects the origin (yellow square) and corner points of the grid (green circles), as shown in Figure 8. The grid pattern is then projected with the estimated parameters and the projected corner points (red crosses) positions are recorded. Finally, the average distance value between the corner points detected and projected was computed and used to measure the precision of the calibration process. The image pairs leading to a higher reprojection error were deleted and recalibrated during the process until the final mean reprojection error value is less than 0.5 pixel which is the recommended value. The resulting plot of reprojection error for the first calibration process is given in Figure 9.

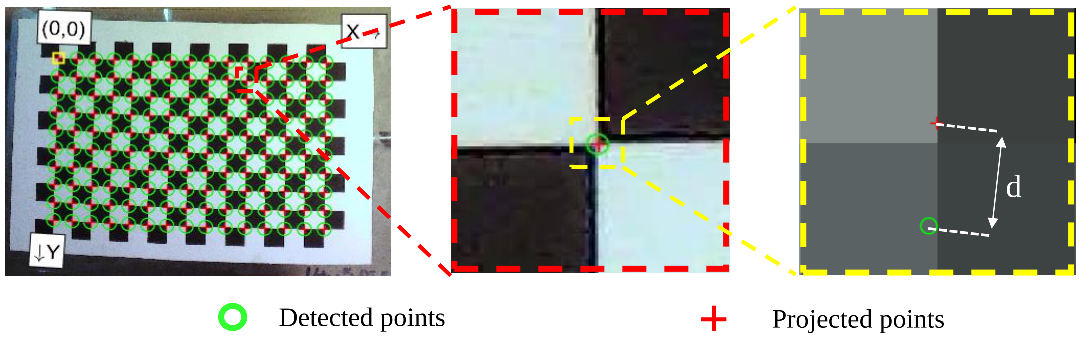


Figure 8: Reprojection error, d

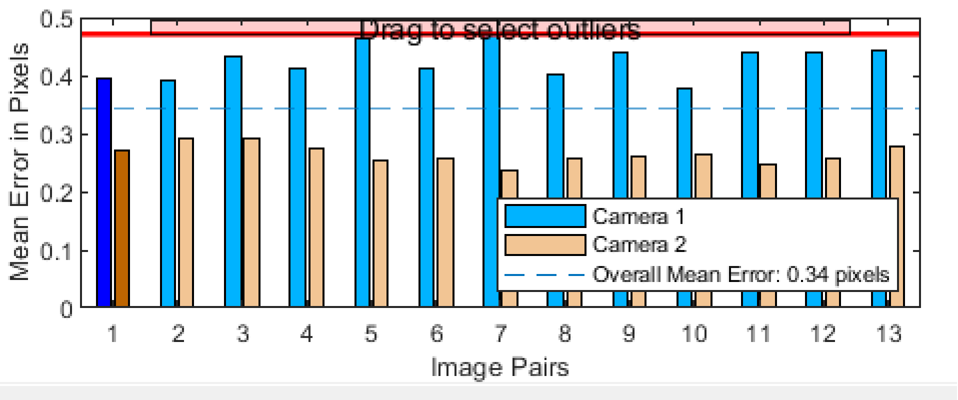


Figure 9: Plot of reprojection error

Also, the Stereo Camera Calibrator application facilitates the 3-D extrinsic parameters plot in the forms of camera-centric views of the patterns and pattern-centric views of the cameras [55]. When the cameras are held stationary, the camera-centric view of the patterns becomes useful. This allows checking if the relative locations of cameras and calibration grids match the real configuration. Figure 10 displays the camera-centric view of the calibration grids in the first calibration process.

The view of undistorted row-aligned is another way of assessing the accuracy of the calibration. A CCD or CMOS sensor is used to produce a digital image that turns the light from the scene into electrical impulses. The amount of light that comes from different regions of the scene which fall into the sensor determines the intensities of the electrical signals. These signals are amplified and transformed into digital signals and are captured in the image of the scene as pixels [90]. An ideal camera should capture an image that is identical to the scene. However, the optical design of the lens, as well as the camera's position relative to the scene captured, will cause optical (or lens or radial) and perspective distortion in a digital image.

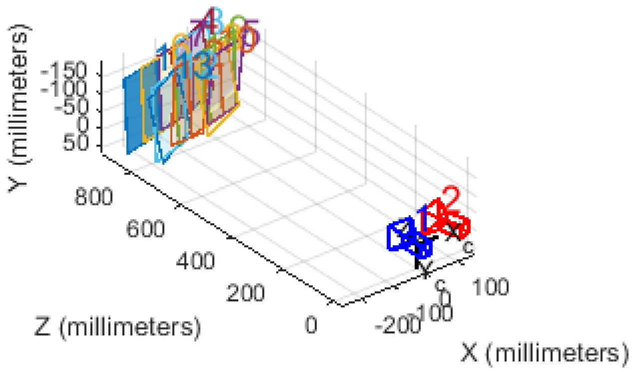
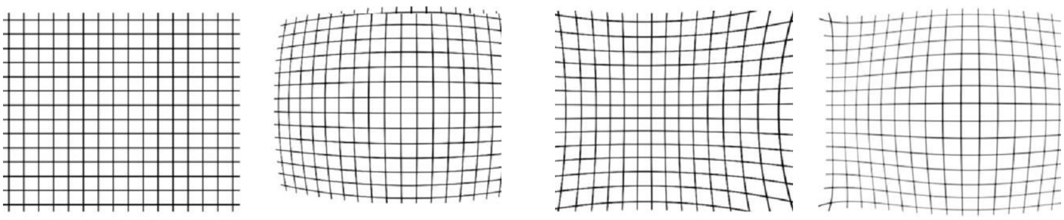


Figure 10: Camera-centric view of the calibration grids

Lens distortion is defined as a form of optical irregularity that deforms straight lines, making them look curved in images [91]. This phenomenon occurs when light rays bend more at the periphery of a lens than in its optical center. Higher distortions are resulted in smaller lenses and vice versa [92]. Three types of lens distortions are mainly considered that is illustrated in Figure 11. The number of radial coefficients can be specified in the Stereo Camera Calibrator application. In most cases, two coefficients are enough for calibration. Three coefficients can be used for extreme distortion, such as in wide-angle lenses [92].



(a) (b) (c) (d)

Figure 11: Types of distortions: (a) Zero distortion. (b) Barrel distortion. (c) Pincushion distortion. (d) Mustache distortion [96]

3.2.3. Rectification

3D reconstruction from stereo vision is the key aspect of the proposed technique. A stereo image is a pair of images taken by coplanar cameras separated by a distance. When using two separate cameras, it is not always possible to hold them in exact parallel orientations. Thus, the left and right images captured need to be rectified in order to produce the stereo image of that scene captured.

The acquired left and right images are projected onto a common plane that's parallel to the baseline: the line joining the camera centres. The outcome of this process will be the rectified images, equivalent to the image pair of the scene captured by a stereo rig. The epipolar geometry: the geometry of stereo vision is illustrated in Figure 12 to discuss the rectification process in detail.

The epipolar plane is the plane that contains the target point as well as the optical centers of both the left and right cameras. Epipolar lines are defined as the lines formed by the intersection of image planes with the epipolar plane. The resulting epipolar lines will be inclined if the cameras are not aligned in parallel. Rectification is the estimation of the transformation of both the left and right image planes so that the epipolar lines become collinear and parallel to one of the image axes. The advantage of rectification is that it reduces the complexity of the correspondence problem which is a 2D problem to a 1D search problem, which is generally along the horizontal raster lines of rectified images.

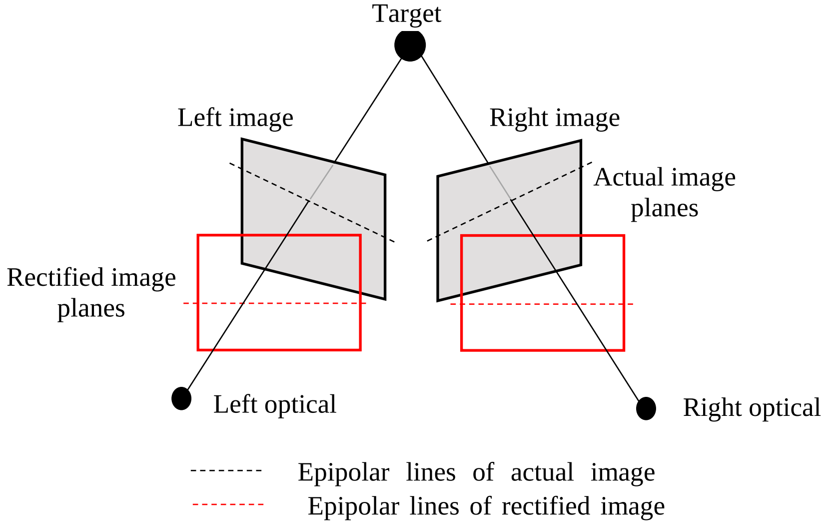
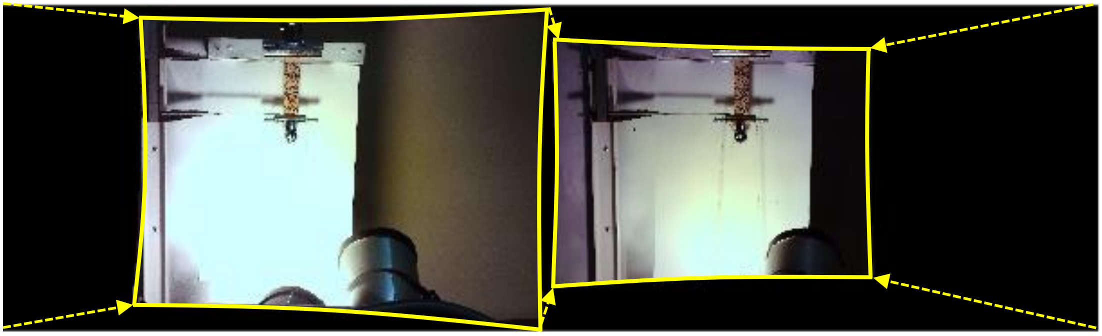


Figure 12: Epipolar geometry

Following the rectification process, the image pairs will be warped to align the matching points horizontally, as shown in Figure 13. The extent of the raw image frame is shown by the black backdrops, and the boundaries of the rectified/warped images are displayed by yellow lines.



(a) (b)

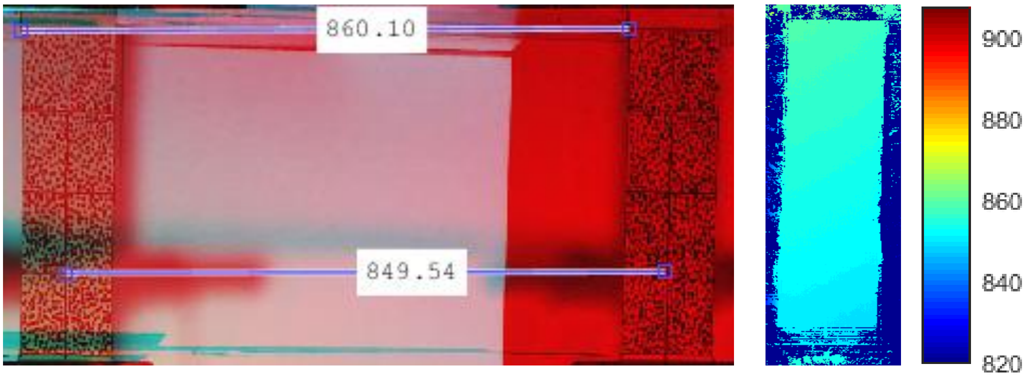
Figure 13: Warped image pairs after rectification (a) left image and (b) right image

3.2.4. Disparity map generation

Following the rectification process, the resulting stereo images will have corresponding points in both images aligned horizontally with a distance. In other words, locations of a specific point in the left and right image will have a horizontal shift that's inversely proportional to the distance between the camera and the object. This is known as disparity, and it may be calculated for each location in order to create a disparity map. [93]

Several methods for disparity computation and mapping have been explored and developed over the years. They may be divided into two categories: local methods and global methods [94]. The Semi Global Matching technique (SGM) is a commonly used local approach based on the idea of matching mutual information at each pixel and approximating a global, 2D smoothness constraint by integrating multiple 1D constraints [95].

The range of disparity in the SGM must be set to encompass the minimum and maximum amount of horizontal shift between the matching pixels in the rectified stereo image pair. The actual horizontal shift range may be measured in the stereo anaglyph of the stereo image pair, and the disparity map can be generated using a range that is close to the actual range divisible by 8. The stereo anaglyph and disparity map generated are shown in Figure 14.



(a) (b)

Figure 14: (a) Measurement of disparity range in stereo anaglyph (b) disparity map

3.2.5. 3D reconstruction

The x, y, and z coordinates at each pixel point of the image, that describes the position in the real 3D space can be calculated from the disparity map. The resulting 3D point cloud in this reconstruction process will be in the form of a three-dimensional array, with the first two dimensions being the pixel coordinate system. The x, y, and z point cloud coordinates are saved in the first, second, and third stack of the three-dimensional array respectively. The resulting point cloud is shown in Figure 15.

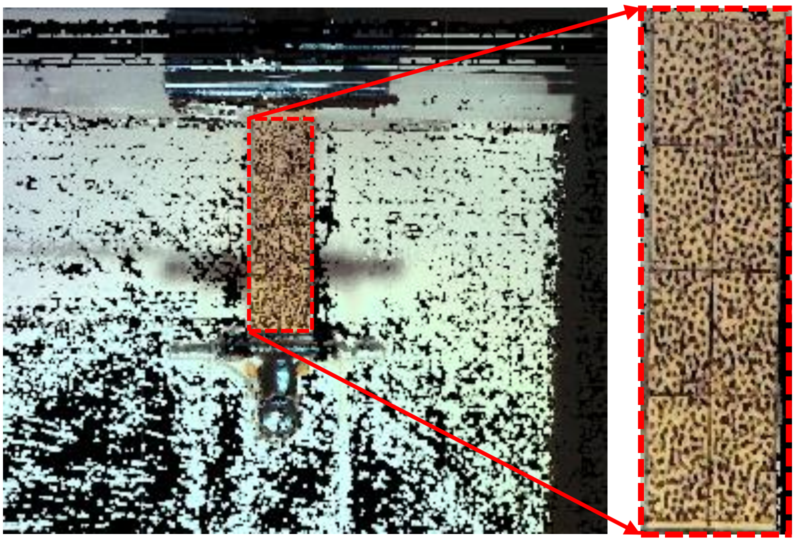


Figure 15: Point cloud of the RoI

The optical centre of the left camera is the origin of the coordinate system on which all the pixels are projected to generate the point cloud. As given in Figure 16, the XY plane and Z-axis are parallel to the image plane and optical axis of the left camera, respectively. The 3D coordinates of a certain image point can be determined by identifying its pixel coordinate value and then indexing that value to find x, y, and z values in the point cloud.

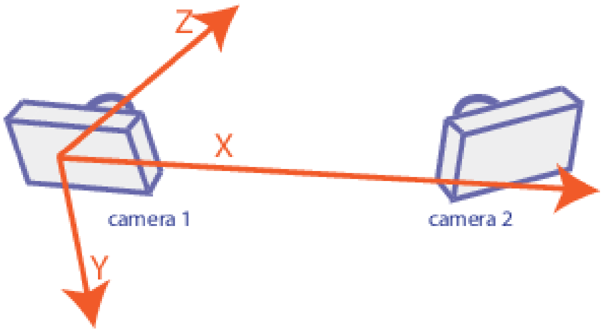


Figure 16: 3D coordinate system of the point cloud [55]

Although the resulting 3D point cloud seems smooth, there are numerous null spots where appropriate disparity values have not been properly recorded. And small deviations in the disparity value will affect its depth value significantly. Therefore, the outliers along with the null points should be eliminated by the application of smoothing filters.

3.2.6. Point cloud refinement

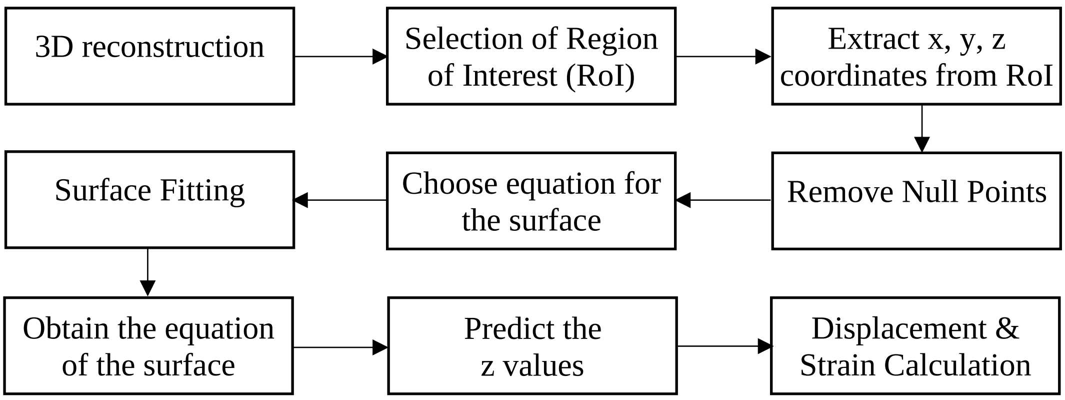
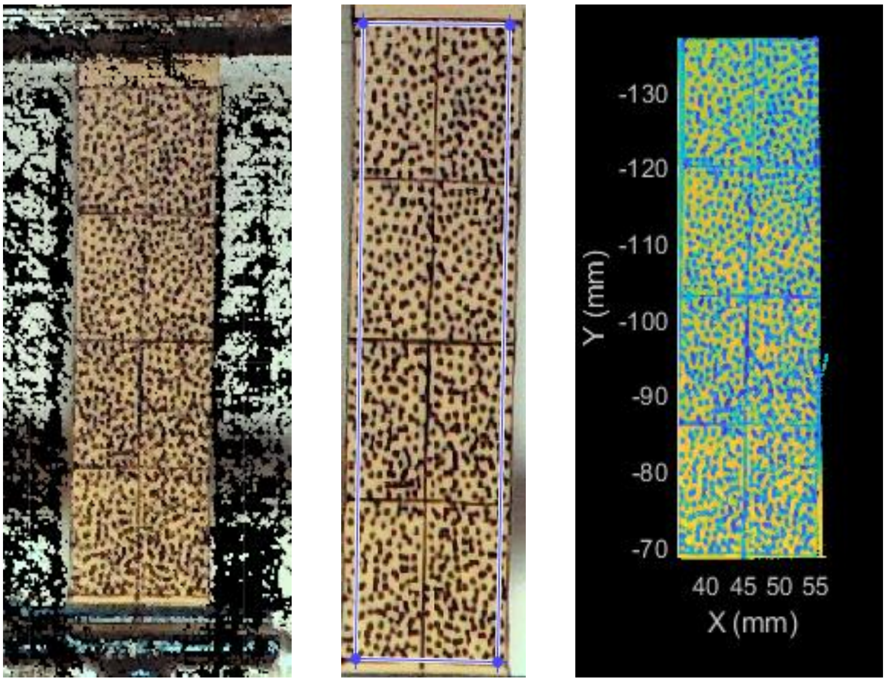


Figure 17: Point cloud refinement algorithm

The steps followed in refining the resulting point cloud are described in Figure 17 as a flowchart. The Region of Interest (RoI) is a small segment of the image frame, and the resulting point cloud will be the same size as the entire image frame. Smoothing filters and eliminating NaN points is a time-consuming and computationally expensive procedure. Therefore, the RoI should be specified, and the smoothing process should be limited to the RoI.

The proposed technique allows the user to manually specify the RoI by picking the vertices of the rectangular region that encompasses the RoI. The x, y, and z coordinate arrays are extracted from the raw point cloud after the vertices' coordinates have been saved. In iterations, a 20 × 20 pixel sliding window is shifted with an overlap of 5 pixels in each stack of the extracted point cloud, and existing NaN points are replaced with the window's mean value at the current location. The abovementioned steps of the refinement process and the outcome are illustrated in Figure 18.



(a) (b) (c)

Figure 18: Steps in point cloud refinement process (a) Raw point cloud (b) RoI (c) extracted point cloud without any NaN points

At the present moment, there will be no NaN points in the extracted point cloud, but there may be noise or outliers in the x, y, and z coordinates. Before moving on to the displacement computation, the extracted region should be smoothed down for accurate results. Outliers in the x and y coordinates are small in comparison to the z coordinates, however, cleaning out all the stacks is recommended to minimize undesired errors.

The approach used to eliminate outliers in this technique is surface fitting of the coordinates. The MATLAB Curve Fitting Toolbox allows the user to fit a data set interactively. Custom equation, interpolant, lowess, and polynomial are the four fit models provided by the application. In comparison to the first two models, the latter two are significantly better at fitting the coordinates. In the lowess (locally weighted scatter plot smooth) model, linear and quadratic polynomials may be utilized for regression, whereas in the polynomial model, we can go up to the 5th degree. Polynomial models are preferred over lowess models because pointwise regression calculations in the lowess approach are computationally intensive and time-consuming.

We can utilize the robust mode for better fit, in which, two options are available for each polynomial equation: LAR (Least Absolute Residuals) and bisquare. The LAR approach looks for a curve that results in the least value for the absolute difference of the residuals. As a result, peak values will have less impact on the fit. The bisquare technique tries to minimize the weighted sum of squares, in which the weight assigned to each data point is proportional to its distance from the fitted line. Points that are close to the line are given full weight. The weight of points that are further away from the line is lowered. Points that are distant from the line than predicted by chance are given no weight. In most situations, the bisquare weight technique is favored over LAR because it uses the standard least-squares approach to create a curve that fits the bulk of the data while also minimizing the influence of outliers.

A polynomial degree that results in a decent fit should be selected. The first step in determining the fit's quality is a visual assessment. Additionally, the goodness of fit statistics can be utilized. In the Curve Fitting Toolbox, four parameters are calculated:

Sum of Squares due to Error (SSE), R-square, adjusted R-square, and Root Mean Squared Error (RMSE), with the first two being relevant and adequate [55].

The overall deviation of the data from the fit to the data is measured by SSE. A number around zero implies that the model's random error component is less, and therefore the fit will be more helpful for prediction. R-square, which is defined as the ratio of the sum of squares of the regression (SSR) to the total sum of squares (SST), is a metric for how well a model fits the data. It is always in the range of 0 and 1. The greater the R-squared, the better the model fits the data in general. The statistical equations and variation of SSE and R-square with a polynomial degree for a sample surface fitting process are presented in the equations and Table 1 below. The best-fit equation should be chosen based on the evaluation of statical parameters and visual inspection, and it may be utilized to generate point clouds for displacement computation. The raw x, y, and z coordinates and the outcomes of the surface fitting process are resented in Figure 19.

$$\text{SSE} = \sum_{i=1}^{n} w_i (z_i - \hat{z_i})^2$$

$$\text{SSR} = \sum_{i=1}^{n} w_i (\hat{z_i} - \bar{z})^2$$

$$\text{SST} = \sum_{i=1}^{n} w_i (z_i - \bar{z})^2$$

$$R - square = \frac{SSR}{SST}$$

Table 1: Variation of statistical parameters with the degree of the polynomial

Degree of the polynomial	SSE	R-square
1	1.5957×10^{-3}	0.9927
2	1.4180×10^{-3}	0.9935
3	1.4181×10^{-3}	0.9935
4	1.3945×10^{-3}	0.9936
5	1.3566×10^{-3}	0.9938

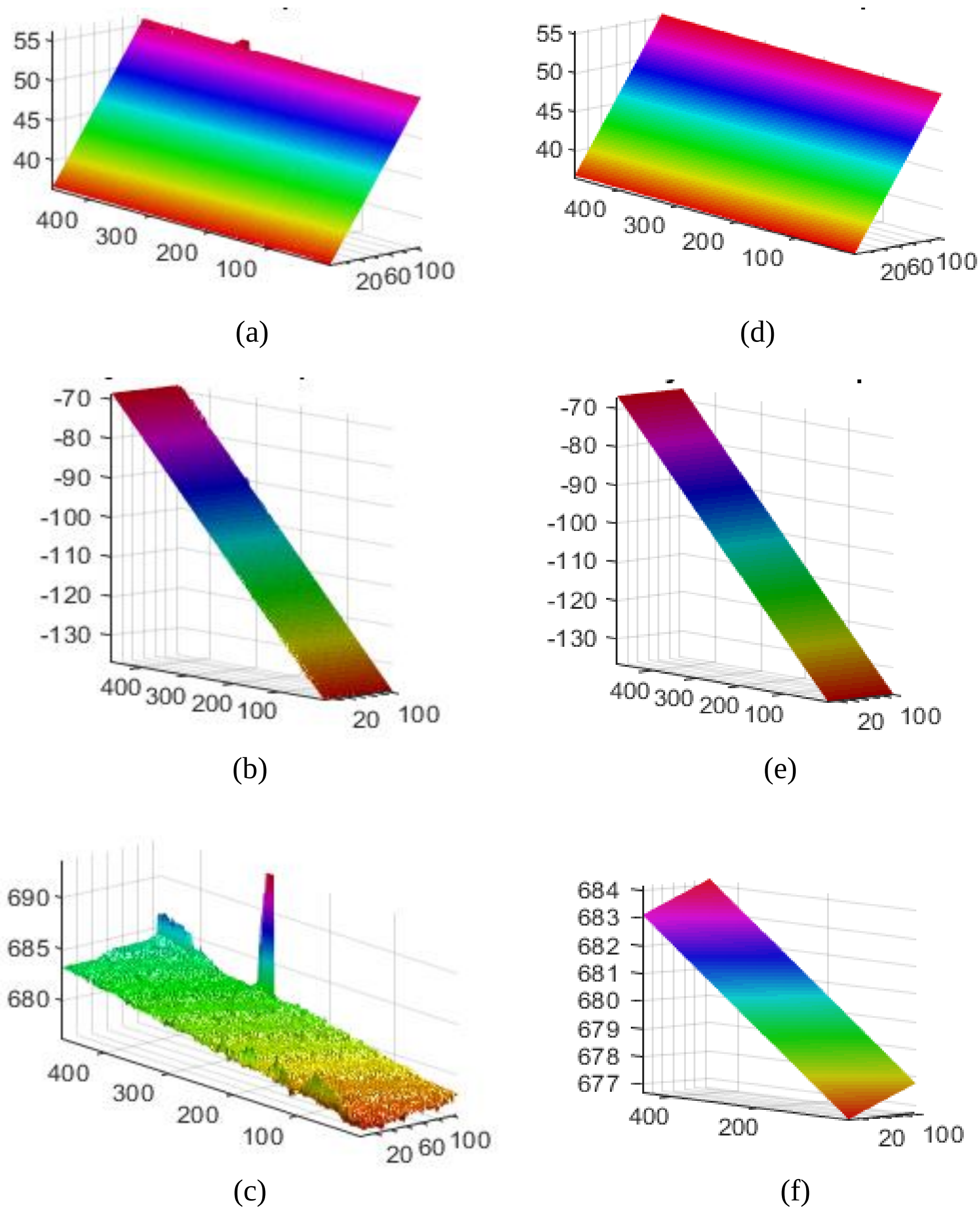


Figure 19: Raw (a) x coordinates (b) y coordinates (c) z coordinates and fitted (d) x coordinates (e) y coordinates (f) z coordinates

3.2.7. Feature detection and estimation of geometric transformation

Finding the location of corresponding regions in successive images is the first step in deformation computation. Figure 20 depicts the entire process of estimating the transformation that occurred between the subset locations of undeformed and deformed state images in a simplified form.

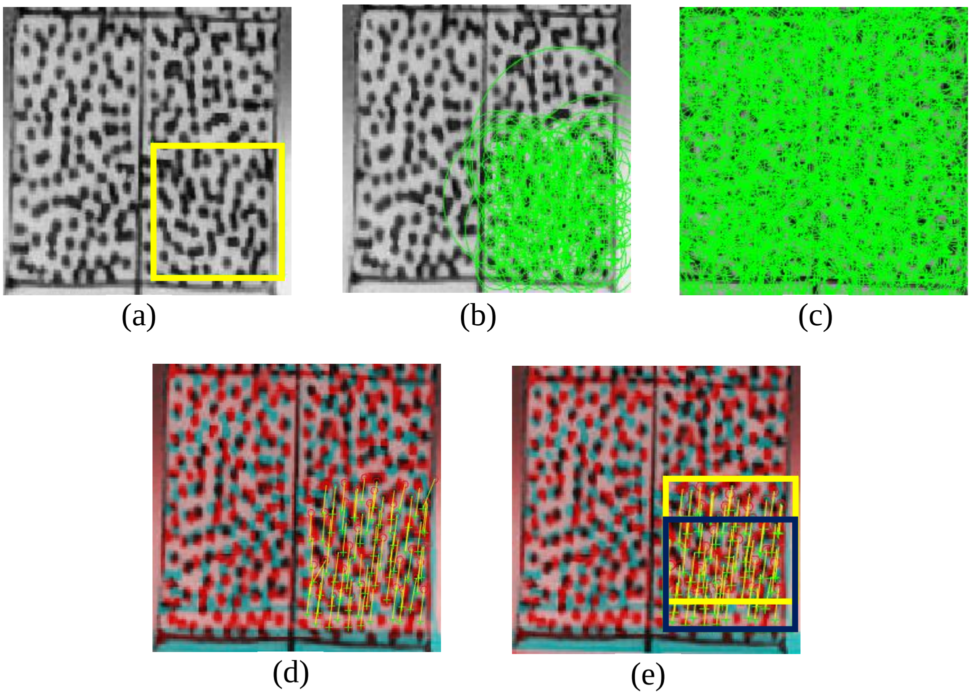


Figure 20: Steps followed in estimating the geometric transformation (a) the subset defined in the reference image (b) features detected in the subset of the reference image (c) features detected in the current state image (d) matched features including outliers (e) matched features without any outliers and subset detected in the current state image

A subset of optimal size is defined in the reference or undeformed state image as shown in the yellow square in Figure 20 (a). The following step is to find the corresponding subset position in the deformed state image by detecting the local distinct features included in the subset and matching them with features in the deformed state image, as shown in Figures 20 (b), Figure 20 (c), and Figure 20 (d). Using the information from these matched features, the geometric transformation that occurred between the subset positions in the considered consecutive images will be calculated. Once the transformation information has been saved, it will be utilized to track down the position of the subset denoted by the blue square in Figure 20 (e). The 2D geometrical transformation of the regions depends on how much the region has deformed. In order to capture that, the above procedure is repeated in iterations by establishing various subset locations in an organized manner in the reference state image.

The Computer Vision Toolbox of MATLAB provides several feature detection and description algorithms that can be mixed and matched. Suitable feature detection and extraction algorithms need to be chosen depending on the criteria of the application and the nature of the image. The available feature detection and description algorithms and respective feature types, characteristics, and applications are presented in Table 2 and Table 3.

Table 2: Feature detection algorithms [55]

Detector	**Feature Type**	**Scale Independent**
FAST	Corner	No
Minimum eigenvalue algorithm	Corner	No
Corner detector	Corner	No
SURF	Blob	Yes
KAZE	Blob	Yes
BRISK	Corner	Yes
MSER	Region with uniform intensity	Yes
ORB	Corner	No

Table 3: Feature descriptor algorithms [55]

Descriptor	**Invariance**		**Typical Use**	
	Scale	**Rotation**	**Finding Point Correspondences**	**Classification**
HOG	No	No	No	Yes
LBP	No	Yes	No	Yes
SURF	Yes	Yes	Yes	Yes
KAZE	Yes	Yes	Yes	Yes
FREAK	Yes	Yes	Yes	No
BRISK	Yes	Yes	Yes	No
ORB	No	Yes	Yes	No
• Block • Simple pixel neighborhood around a keypoint	No	No	Yes	Yes

The algorithms SURF, KAZE, and BRISK are suited for images with scale and rotation variations. The number of matching features in a subset (150 x 150 - pixel window) of

RoI in the reference and current state images at a given loading step obtained from the above algorithms are compared in order to determine the best-suited algorithm that is given in Table 4. The KAZE algorithm resulted in the maximum of matching features, according to the comparative study. As a result, the KAZE method is used to determine the transformation matrix of subsets in image pairs of subsequent loading steps.

Table 4: Comparison of feature detection algorithms

Algorithm	No of features detected in the undeformed image	No of features detected in the deformed image	No of matched features
SURF (Blob)	256	497	35
KAZE (Blob)	1156	2151	234
BRISK (Corner)	1123	2178	7

Following the identification of matching features, the geometric transformation between the defined and detected subsets needs to be estimated. This is completed using a built-in function of MATLAB: *estimateGeometricTransform* and the transformation details are stored in a matrix form namely tform. The initial matching points will include some outliers and tose will be eliminated using the resulting tform matrix. The M-estimator Sample Consensus algorithm, a variation of the Random Sample Consensus method, is used to eliminate the outliers (RANSAC). Because of the randomized nature of the method, the results generated in subsequent runs may change. The transform type should be one of the three methods available: similarity, projective, or affine. Table 5 presents the minimum number of matched pairs of points necessary for each transform type.

Table 5: Required minimum number of matched points

Transform type	Minimum number of matched pair of points
Similarity	2
Affine	3
Projective	4

The similarity transform technique is a shape-preserving algorithm that takes uniform scaling in x and y directions, translation, and rotation into consideration. It has four degrees of freedom that may be calculated using two matched pairs of points. Affine transformation is a type of linear mapping that preserves collinearity and the ratios of distances between points on a line. It accounts for six degrees of freedom, including translation, scale, shear, and rotation. The rotation, translation, shear, scaling, and perspective projection are all accounted for by the projective transformation, which has 8 degrees of freedom and is computed from four pairs of matched points (MATLAB Documentation). The projective transformation approach is chosen over the others because it is capable of completely describing the transformation due to its higher number of degrees of freedom.

3.2.8. Displacement calculation

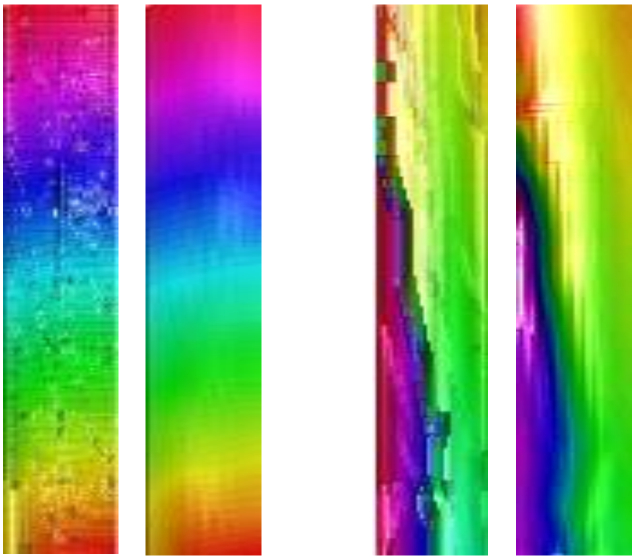
A 3 x 3-pixel window is moved in each subset (described in the previous section), and x, y, and z coordinates are retrieved from the reference state's point cloud. The position of the tiny window in the deformed state image may be determined using the estimated geometric transformation of the bigger subset to which the small window belongs. As a result, the coordinates of the corresponding window in the deformed state may be retrieved and utilized in the equations below to compute displacements. In each step of the iteration, the displacement values are stored in the centroid of the moving window in a matrix the same size as the reference state RoI.

$$dx = \sqrt{(x_1 - x_2)^2 + (z_1 - z_2)^2}$$

$$dy = \sqrt{(y_1 - y_2)^2 + (z_1 - z_2)^2}$$

There will be NaN values in all places except the window centroids at this stage. Therefore, a 20 × 20-pixel window is iterated over the displacement matrix, and all NaN values are replaced by the window's mean value, identical to the method used to eliminate NaN points in the point cloud. Appropriate local regression filters should be used to further smooth out the resulting displacement matrix.

There are several digital filters such as moving average, Savitzky-Golay filters, and local regression with and without weights and robustness (lowess, loess, rlowess, and rloess). The lowess, loess, rlowess, and rloess techniques, as stated in section 3.2.6, are not recommended since they are computationally intensive and time-consuming. The moving average filter smooths away a large amount of the data's peak information, and it can only keep the lower moments of a peak such as a centroid. Savitzky-Golay filters, which use least-squares fitting to fit a polynomial in a window of points around each sample point, are recommended. The polynomial's degree can be specified. Its computation time is proportional to the width of the window however, it has the benefit of retaining the area, location, and width of peaks, which is important for obtaining accurate results.



(a) (b)

Figure 21: Raw and smoothened full-field maps of (a) vertical displacement (b) horizontal displacement

The raw vertical and horizontal displacement fields, as well as the smoothed-out fields, are presented in Figure 21. The brighter uneven spots in the maps denote the outliers. It can be observed that the noise and outliers present in the raw displacement fields were substantially reduced by the Savitsky-Golay smoothing filter. The displacement fields produced indicate the deformation that happened in successive loading steps. To obtain the overall full-field displacement at a given loading step, all of the displacement matrices obtained at prior loading steps were scaled and added.

3.2.9. Strain calculation

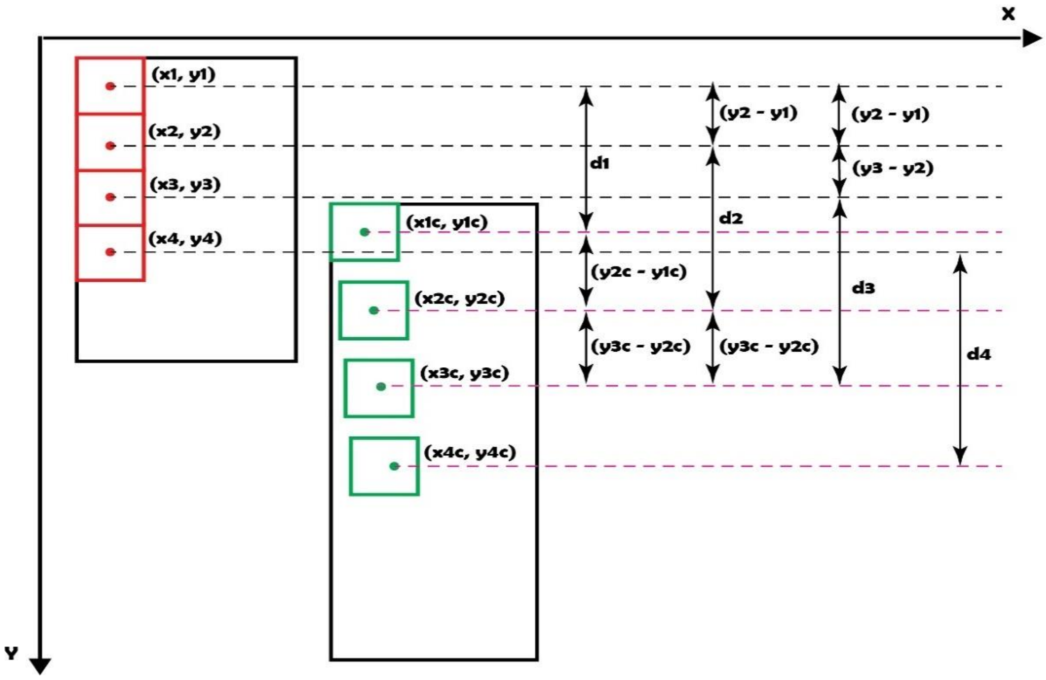


Figure 22: Strain calculation

Figure 22 depicts the method of strain calculation briefly. The red squares represent the small 3x3-pixel windows defined in the undeformed state image with displacement values stored at its centroid denoted by the red dots and green squares are the corresponding transformed windows of them in the deformed state image. The vertical strain can be computed using the equations given below.

$$strain = \frac{deformed\ length - initial\ length}{initial\ length}$$

$$\varepsilon_{yy1} = \frac{(y_{2c} - y_{1c}) - (y_2 - y_1)}{(y_2 - y_1)} = \frac{(d_2 - d_1)}{(y_2 - y_1)}$$

$$\varepsilon_{yy2} = \frac{(y_{3c} - y_{2c}) - (y_3 - y_2)}{(y_3 - y_2)} = \frac{(d_3 - d_2)}{(y_3 - y_2)}$$

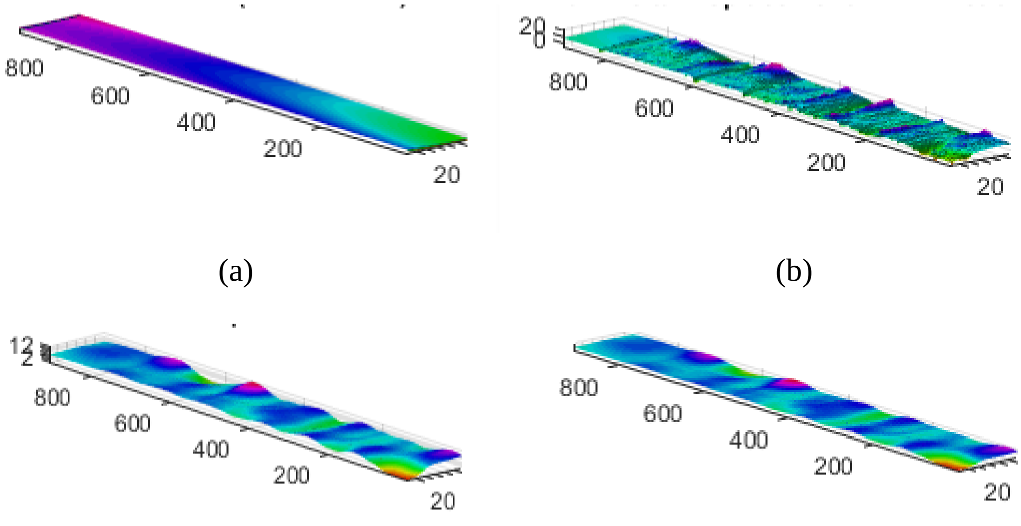
$$\varepsilon_{yym} = \frac{\Delta DY_y}{\Delta Y}$$

As expressed in the above equations, the vertical strain at each pixel is calculated by dividing ΔDY by ΔY. The symbol ΔDY is the change in vertical displacement values and the subscript denotes the direction in which the change is calculated. And ΔY is the amount of change (in the vertical direction) in y coordinates. These matrices are produced by subtracting each row of the matrix from the row above it, excluding the first one. Using a similar approach, the horizontal strain and shear strain matrices can be generated using the following equations.

$$\varepsilon_{xxm} = \frac{\Delta DX_X}{\Delta X}$$

$$\varepsilon_{xym} = \frac{\Delta DY_X}{\Delta X} + \frac{\Delta DX_Y}{\Delta Y}$$

Despite the fact that the displacement matrices were initially smoothed down, anomalies will exist in ΔDY_y, ΔDX_X, ΔDY_X and ΔDX_Y matrices. These matrices are smoothed down using Savitsky-Golay smoothing filters to eliminate errors caused by these outlier points. Figure 23 depicts the outcomes of the stated approach.



(a) (b) (c) (d)

Figure 23: Colormaps of (a) ΔY matrix (b) ΔDY_y matrix (c) raw vertical strain matrix and (d) smoothened vertical strain matrix

4. RESULTS AND DISCUSSION

4.1. Results

The length of the RoI was measured using a Vernier Caliper at each loading step for verification purposes. Tables 6 - 10 present the comparison of displacement and strain values obtained through the Vernier Caliper measurements and the proposed method. The tables also include the percentage error for the difference between computed and measured displacement and strain values. The maximum errors recorded are 3.79 % and 3.66 % for the displacement and strain measurement respectively.

Based on the methodology described in the previous chapter, colormaps of displacements and strains of specimens at each step of the loading and unloading phase can be obtained for the RoI. Figures 24 - 63 provide the graphical representations of displacement and strain variation in both horizontal and vertical directions as well as shear strains variation at each loading step in the form of contour maps. The results for the unloading phase of the specimen in experiment 5 are not included in the study as the specimen tore at the last loading step.

It is intended to compare the outcomes of the proposed method with any well-established, widely used DIC methods in order to verify the integrity of the full-field deformation maps produced. However, the number of open-source algorithms for 3D DIC measurements is limited and most of the studies are still under development. Thus, it was decided to carry out the 2D analysis using the proposed method and validate the results against the widely used open source, 2D DIC program, Ncorr.

The undistorted raw images captured by the left camera were used in the 2D analysis, and the steps included in the proposed method for obtaining 3D reconstructions were skipped during the process. By disabling the impact of the z coordinates, the displacement and strain fields were produced using the same algorithms. The pixel indices were utilized for the displacement and strain computations in the same way as the x, y, and z coordinates extracted from point clouds were used in the 3D analysis, and the resultant displacement measurements were expressed in pixels.

Ncorr was unable to analyze pictures with higher deformation due to the difficulties in capturing the higher correlation. As a result, the displacements that occurred in successive loading steps were compared. Consecutive loading steps with higher step deformation were excluded from the comparison as well. Figures 64 – 73 present the comparison of vertical and horizontal displacement maps obtained using the proposed method and the Ncorr program.

Table 6: Comparison of results obtained for the specimen of dimension 60 mm x 20 mm x 0.21 mm – experiment 1

Load (g)		Vernier Calliper			DIC			Displacement		Strain	
		Length (mm)	**Displacement (mm)**	**Strain**	**Length (mm)**	**Displacement (mm)**	**Strain**	**Error**	**Error (%)**	**Error**	**Error (%)**
Loading	0	59.95	0.00	0.0000	60.05	0.00	0.0000	0.00	0.00	0.0000	0.00
	68.53	64.40	4.45	0.0742	64.41	4.36	0.0726	-0.09	-2.02	-0.0016	-2.19
	118.53	68.10	8.15	0.1359	68.24	8.19	0.1364	0.04	0.49	0.0004	0.32
	168.53	72.65	12.70	0.2118	72.90	12.85	0.2140	0.15	1.18	0.0021	1.01
	218.53	79.20	19.25	0.3211	79.23	19.18	0.3194	-0.07	-0.36	-0.0017	-0.53
	268.53	86.45	26.50	0.4420	86.38	26.33	0.4385	-0.17	-0.64	-0.0036	-0.81
	318.53	91.55	31.60	0.5271	91.52	31.47	0.5241	-0.13	-0.41	-0.0030	-0.58
	368.53	96.30	36.35	0.6063	96.22	36.17	0.6023	-0.18	-0.50	-0.0040	-0.66
	418.53	104.50	44.55	0.7431	104.70	44.65	0.7435	0.10	0.22	0.0004	0.06
	468.53	112.60	52.65	0.8782	112.62	52.57	0.8754	-0.08	-0.15	-0.0028	-0.32
	518.53	125.25	65.30	1.0892	125.44	65.39	1.0889	0.09	0.14	-0.0003	-0.03
	568.53	131.50	71.55	1.1935	131.66	71.61	1.1925	0.06	0.08	-0.0010	-0.08
Unloading	518.53	131.50	71.55	1.1935	131.64	71.59	1.1922	0.04	0.06	-0.0013	-0.11
	468.53	131.50	71.55	1.1935	131.61	71.56	1.1917	0.01	0.01	-0.0018	-0.15
	418.53	130.50	70.55	1.1768	130.64	70.59	1.1755	0.04	0.06	-0.0013	-0.11
	368.53	130.50	70.55	1.1768	130.63	70.58	1.1754	0.03	0.04	-0.0015	-0.12
	318.53	129.50	69.55	1.1601	129.57	69.52	1.1577	-0.03	-0.04	-0.0024	-0.21
	268.53	123.10	63.15	1.0534	123.35	63.30	1.0541	0.15	0.24	0.0007	0.07
	218.53	110.00	50.05	0.8349	110.07	50.02	0.8330	-0.03	-0.06	-0.0019	-0.23
	168.53	88.80	28.85	0.4812	88.68	28.63	0.4768	-0.22	-0.76	-0.0045	-0.93
	118.53	83.30	23.35	0.3895	83.55	23.50	0.3913	0.15	0.64	0.0018	0.47
	68.53	71.00	11.05	0.1843	71.35	11.30	0.1882	0.25	2.26	0.0039	2.09

Table 7: Comparison of results obtained for the specimen of dimension 60 mm x 20 mm x 0.21 mm – experiment 2

Load (g)		Vernier Calliper			DIC			Displacement		Strain	
		Length (mm)	**Displacement (mm)**	**Strain**	**Length (mm)**	**Displacement (mm)**	**Strain**	**Error**	**Error (%)**	**Error**	**Error (%)**
Loading	0	60.05	0.00	0.0000	60.00	0.00	0.0000	0.00	0.00	0.0000	0.00
	68.53	66.45	6.40	0.1066	66.21	6.21	0.1035	-0.19	-2.99	-0.0031	-2.91
	118.53	69.05	9.00	0.1499	69.23	9.23	0.1538	0.23	2.52	0.0039	2.60
	168.53	73.05	13.00	0.2165	73.24	13.24	0.2207	0.24	1.87	0.0042	1.95
	218.53	77.25	17.20	0.2864	77.28	17.28	0.2880	0.08	0.47	0.0016	0.55
	268.53	83.75	23.70	0.3947	83.58	23.58	0.3930	-0.12	-0.50	-0.0016	-0.42
	318.53	90.50	30.45	0.5071	90.36	30.36	0.5060	-0.09	-0.30	-0.0011	-0.22
	368.53	96.70	36.65	0.6103	96.67	36.67	0.6112	0.02	0.06	0.0008	0.14
	418.53	101.55	41.50	0.6911	101.49	41.49	0.6914	-0.01	-0.03	0.0004	0.05
	468.53	109.05	49.00	0.8160	109.29	49.29	0.8214	0.29	0.58	0.0054	0.66
	518.53	117.00	56.95	0.9484	116.94	56.94	0.9489	-0.01	-0.02	0.0006	0.06
	568.53	137.45	77.40	1.2889	137.29	77.29	1.2881	-0.11	-0.14	-0.0008	-0.06
Unloading	518.53	136.45	76.40	1.2723	136.52	76.52	1.2753	0.12	0.16	0.0030	0.24
	468.53	136.45	76.40	1.2723	136.50	76.50	1.2749	0.10	0.13	0.0026	0.21
	418.53	136.25	76.20	1.2689	136.23	76.23	1.2704	0.03	0.03	0.0014	0.11
	368.53	136.25	76.20	1.2689	136.21	76.21	1.2701	0.01	0.01	0.0012	0.09
	318.53	136.00	75.95	1.2648	136.09	76.08	1.2680	0.13	0.18	0.0033	0.26
	268.53	135.50	75.45	1.2565	135.34	75.34	1.2557	-0.11	-0.14	-0.0008	-0.06
	218.53	129.55	69.50	1.1574	129.26	69.26	1.1543	-0.24	-0.34	-0.0030	-0.26
	168.53	93.50	33.45	0.5570	93.49	33.48	0.5580	0.03	0.10	0.0010	0.18
	118.53	88.00	27.95	0.4654	88.27	28.27	0.4712	0.32	1.16	0.0058	1.24
	68.53	75.50	15.45	0.2573	75.29	15.29	0.2548	-0.16	-1.05	-0.0025	-0.97

Table 8: Comparison of results obtained for the specimen of dimension 60 mm x 20 mm x 0.21 mm – experiment 3

Load (g)		**Vernier Calliper**			**DIC**			**Displacement**		**Strain**	
		Length (mm)	**Displacement (mm)**	**Strain**	**Length (mm)**	**Displacement (mm)**	**Strain**	**Error**	**Error (%)**	**Error**	**Error (%)**
Loading	0	60.00	0.00	0.0000	60.27	0.00	0.0000	0.00	0.00	0.0000	0.00
	68.53	65.55	5.55	0.0925	65.69	5.41	0.0898	-0.14	-2.48	-0.0027	-2.93
	118.53	69.05	9.05	0.1508	69.26	8.98	0.1490	-0.07	-0.77	-0.0018	-1.22
	168.53	72.40	12.40	0.2067	72.42	12.15	0.2015	-0.25	-2.03	-0.0051	-2.48
	218.53	76.85	16.85	0.2808	76.83	16.56	0.2747	-0.29	-1.72	-0.0061	-2.17
	268.53	82.05	22.05	0.3675	81.86	21.59	0.3582	-0.46	-2.09	-0.0093	-2.54
	318.53	89.55	29.55	0.4925	89.69	29.41	0.4880	-0.14	-0.46	-0.0045	-0.92
	368.53	95.10	35.10	0.5850	94.90	34.62	0.5744	-0.48	-1.37	-0.0106	-1.82
	418.53	101.85	41.85	0.6975	101.60	41.32	0.6856	-0.53	-1.26	-0.0119	-1.71
	468.53	111.50	51.50	0.8583	111.49	51.21	0.8497	-0.29	-0.55	-0.0087	-1.01
	518.53	117.95	57.95	0.9658	118.18	57.91	0.9607	-0.04	-0.07	-0.0051	-0.53
	568.53	122.85	62.85	1.0475	123.06	62.78	1.0416	-0.07	-0.11	-0.0059	-0.57
Unloading	518.53	122.85	62.85	1.0475	122.85	62.58	1.0382	-0.27	-0.43	-0.0093	-0.89
	468.53	122.85	62.85	1.0475	122.83	62.56	1.0379	-0.29	-0.47	-0.0096	-0.92
	418.53	122.85	62.85	1.0475	122.80	62.52	1.0373	-0.33	-0.52	-0.0102	-0.97
	368.53	122.80	62.80	1.0467	122.77	62.50	1.0369	-0.30	-0.48	-0.0098	-0.93
	318.53	122.45	62.45	1.0408	122.66	62.39	1.0351	-0.06	-0.10	-0.0058	-0.55
	268.53	119.20	59.20	0.9867	119.48	59.21	0.9823	0.01	0.01	-0.0044	-0.45
	218.53	106.45	46.45	0.7742	106.17	45.90	0.7614	-0.55	-1.19	-0.0127	-1.64
	168.53	104.00	44.00	0.7333	103.82	43.54	0.7224	-0.46	-1.04	-0.0109	-1.49
	118.53	88.00	28.00	0.4667	87.77	27.50	0.4562	-0.50	-1.80	-0.0105	-2.25
	68.53	74.55	14.55	0.2425	74.48	14.20	0.2357	-0.35	-2.38	-0.0068	-2.82

Table 9: Comparison of results obtained for the specimen of dimension 60 mm x 60 mm x 0.21 mm – experiment 4

Load (g)		**Vernier Caliper**			**DIC**			**Displacement**		**Strain**	
		Length (mm)	**Displacement (mm)**	**Strain**	**Length (mm)**	**Displacement (mm)**	**Strain**	**Error**	**Error (%)**	**Error**	**Error (%)**
Loading	0	60.35	0.00	0.0000	60.42	0.00	0.0000	0.00	0.00	0.0000	0.00
	68.53	63.20	2.85	0.0472	63.23	2.81	0.0465	-0.04	-1.38	-0.0007	-1.49
	168.53	66.80	6.45	0.1069	67.03	6.62	0.1095	0.17	2.58	0.0026	2.46
	268.53	67.65	7.30	0.1210	67.89	7.47	0.1236	0.17	2.31	0.0027	2.19
	368.53	71.90	11.55	0.1914	71.85	11.43	0.1892	-0.12	-1.00	-0.0021	-1.12
	468.53	76.10	15.75	0.2610	75.91	15.49	0.2564	-0.26	-1.63	-0.0046	-1.74
	568.53	81.00	20.65	0.3422	81.21	20.79	0.3441	0.14	0.67	0.0019	0.56
	668.53	83.00	22.65	0.3753	82.86	22.44	0.3714	-0.21	-0.92	-0.0039	-1.03
Unloading	568.53	80.00	19.65	0.3256	80.12	19.70	0.3261	0.05	0.26	0.0005	0.15
	468.53	75.50	15.15	0.2510	75.51	15.09	0.2497	-0.06	-0.41	-0.0013	-0.52
	368.53	71.00	10.65	0.1765	70.79	10.37	0.1717	-0.28	-2.60	-0.0048	-2.71
	268.53	66.50	6.15	0.1019	66.49	6.07	0.1005	-0.08	-1.25	-0.0014	-1.36

Table 10: Comparison of results obtained for the specimen of dimension 20 mm x 60 mm x 0.21 mm – experiment 5

Load (g)		**Vernier Caliper**			**DIC**			**Displacement**		**Strain**	
		Length (mm)	**Displacement (mm)**	**Strain**	**Length (mm)**	**Displacement (mm)**	**Strain**	**Error**	**Error (%)**	**Error**	**Error (%)**
Loading	0	19.35	0.00	0.0000	19.37	0.00	0.0000	0.00	0.00	0.0000	0.00
	68.53	19.65	0.30	0.0155	19.68	0.30	0.0155	0.00	0.42	0.0000	0.29
	268.53	20.00	0.65	0.0336	20.05	0.67	0.0348	0.02	3.71	0.0012	3.58
	468.53	23.00	3.65	0.1886	23.11	3.74	0.1929	0.09	2.39	0.0043	2.26
	668.53	24.30	4.95	0.2558	24.51	5.14	0.2652	0.19	3.79	0.0094	3.66
	868.53	28.80	9.45	0.4884	28.60	9.23	0.4762	-0.22	-2.37	-0.0121	-2.49

Specimen (60 mm x 20 mm) - 1

Vertical Displacement

Loading

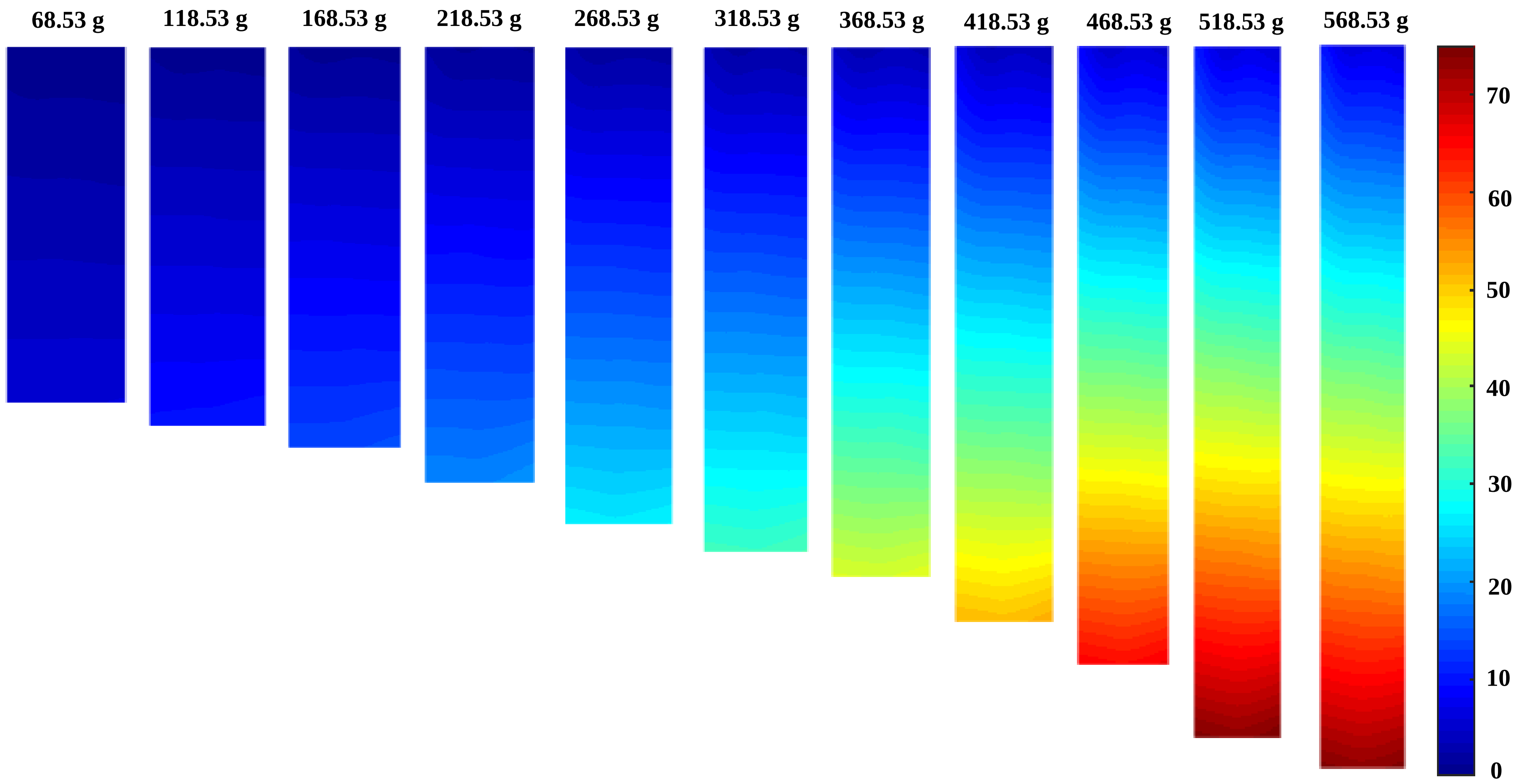


Figure 24: Graphical representation of total vertical displacement during loading process – experiment 1

Unloading

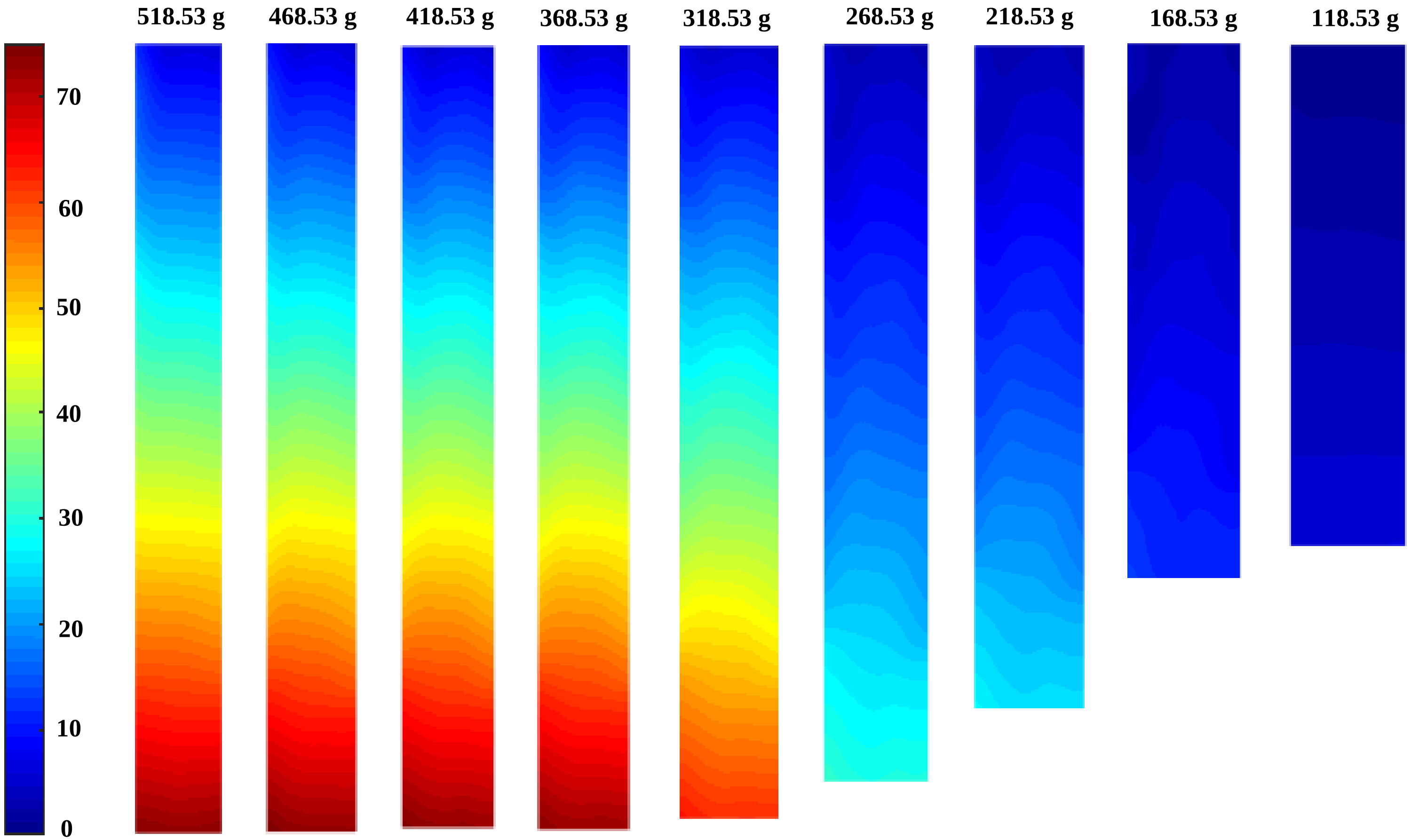


Figure 25: Graphical representation of total vertical displacement during unloading process – experiment 1

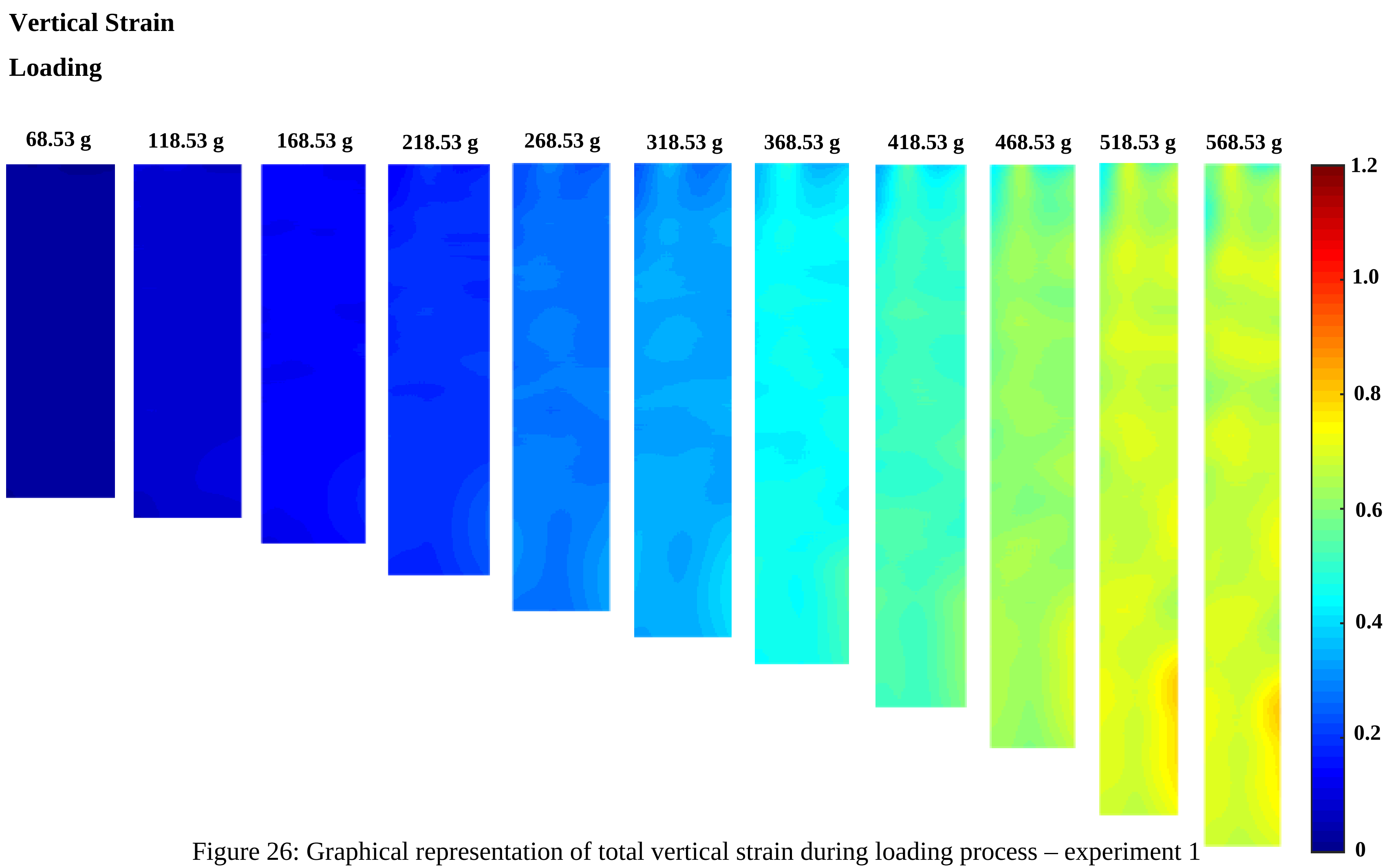


Figure 26: Graphical representation of total vertical strain during loading process – experiment 1

Unloading

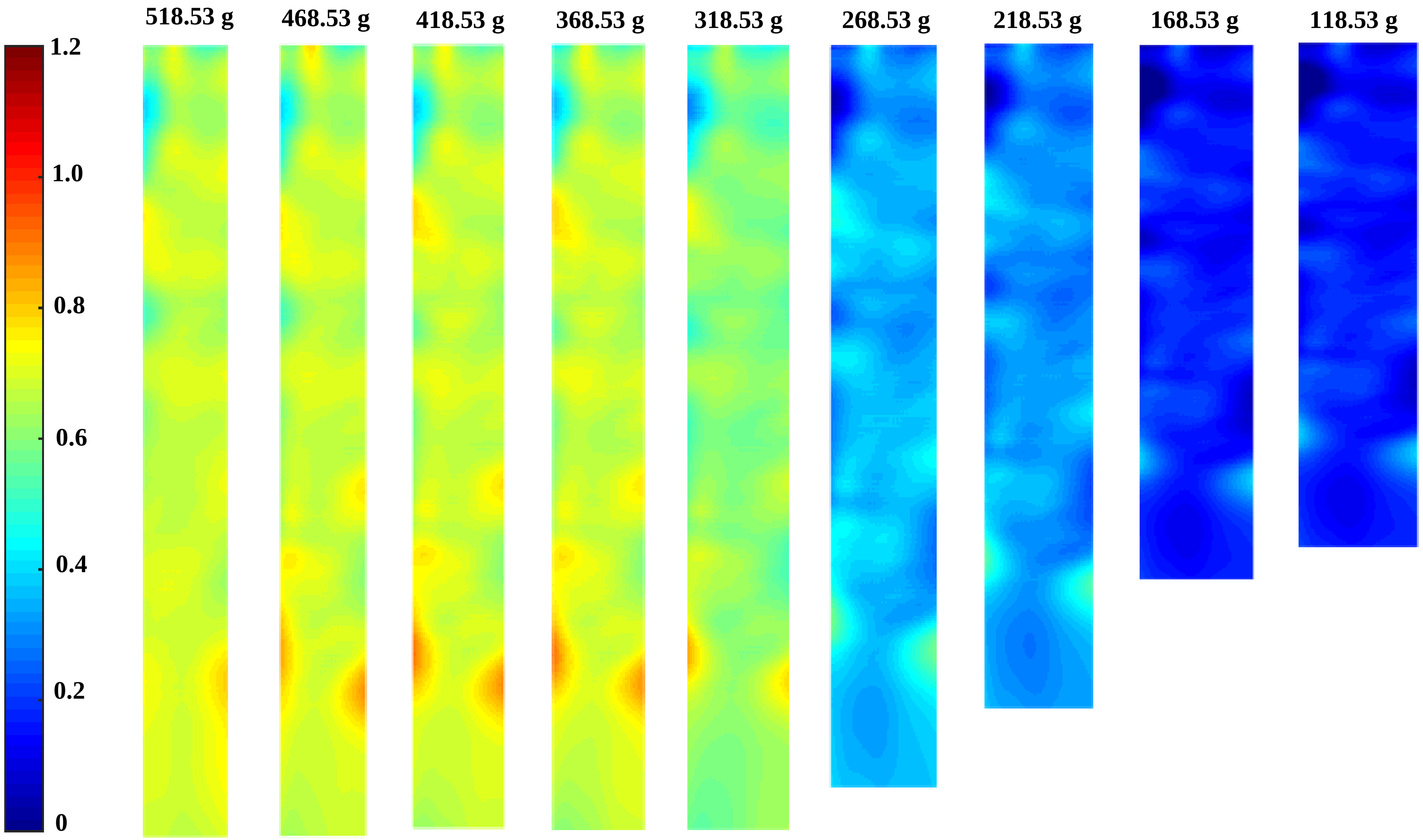


Figure 27: Graphical representation of total vertical strain during unloading process – experiment 1

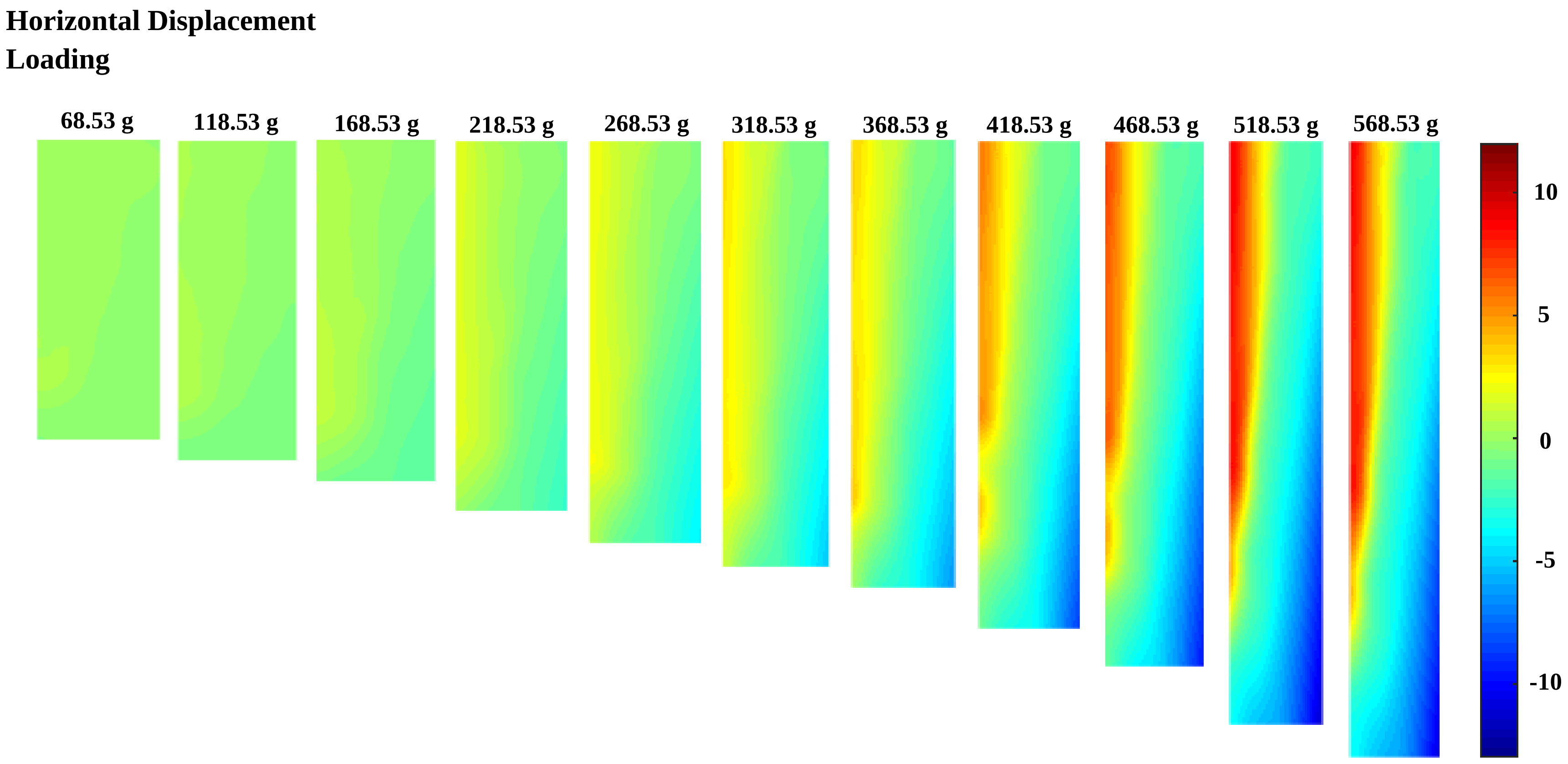


Figure 28: Graphical representation of total horizontal displacement during loading process – experiment 1

Unloading

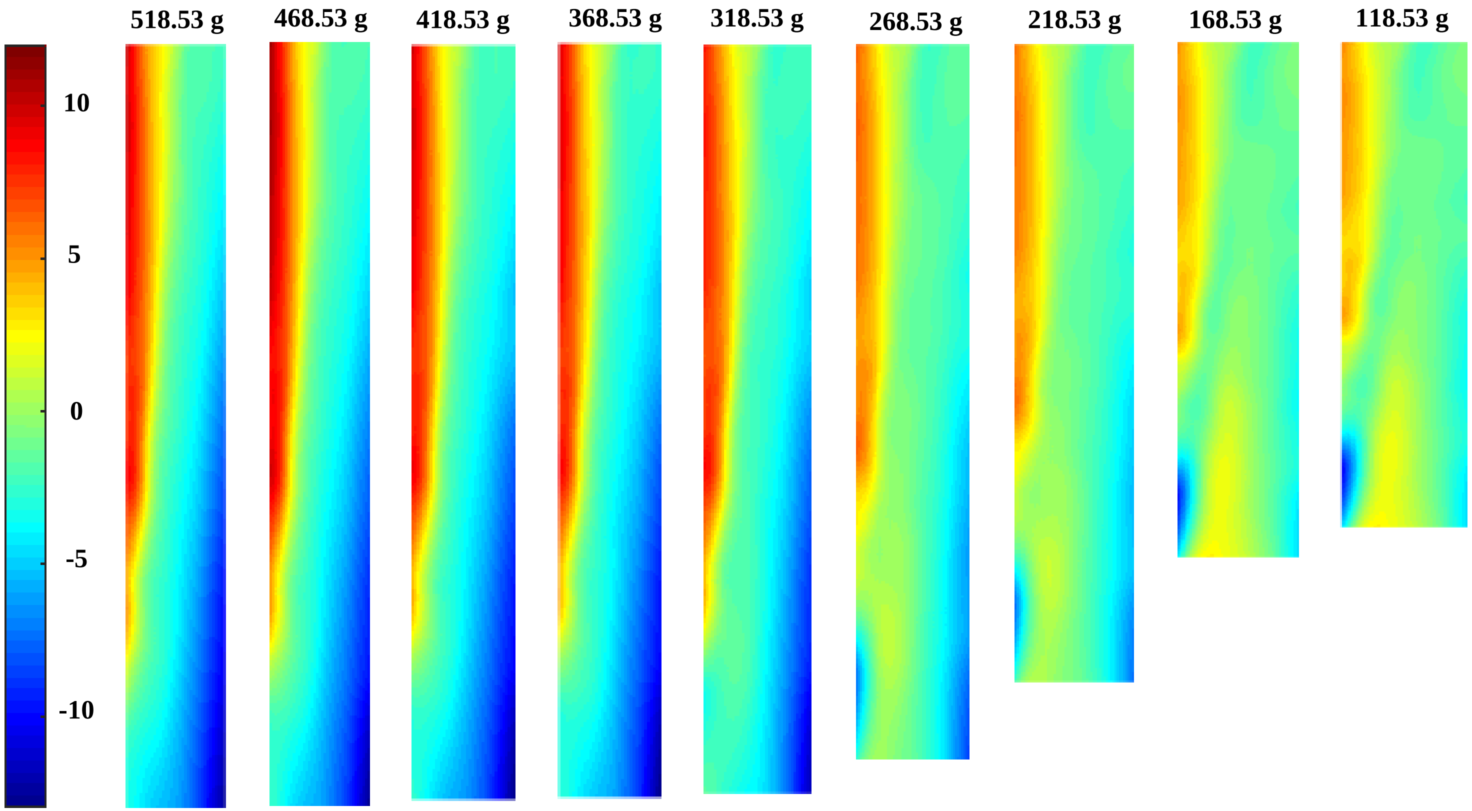


Figure 29: Graphical representation of total horizontal displacement during unloading process – experiment 1

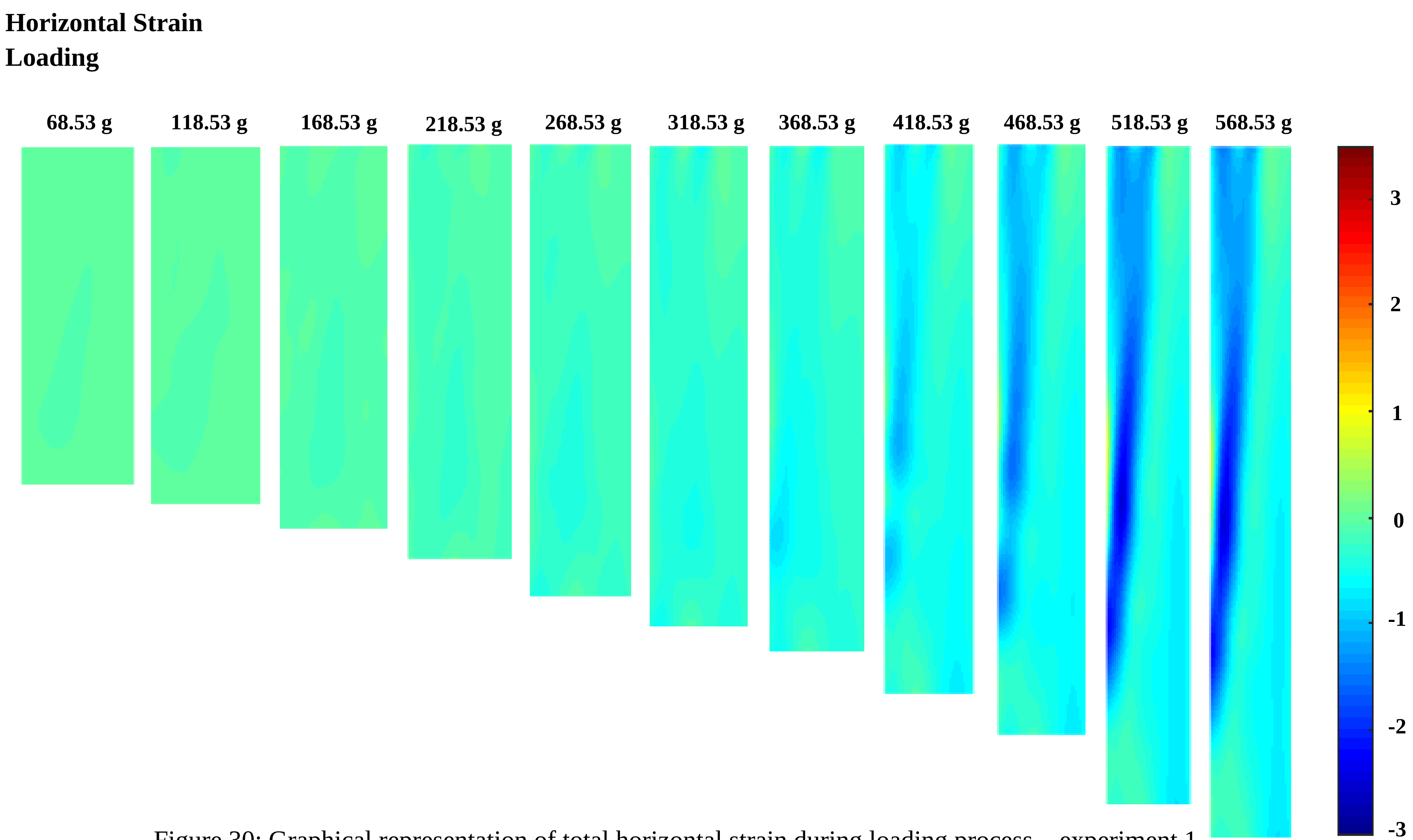


Figure 30: Graphical representation of total horizontal strain during loading process – experiment 1

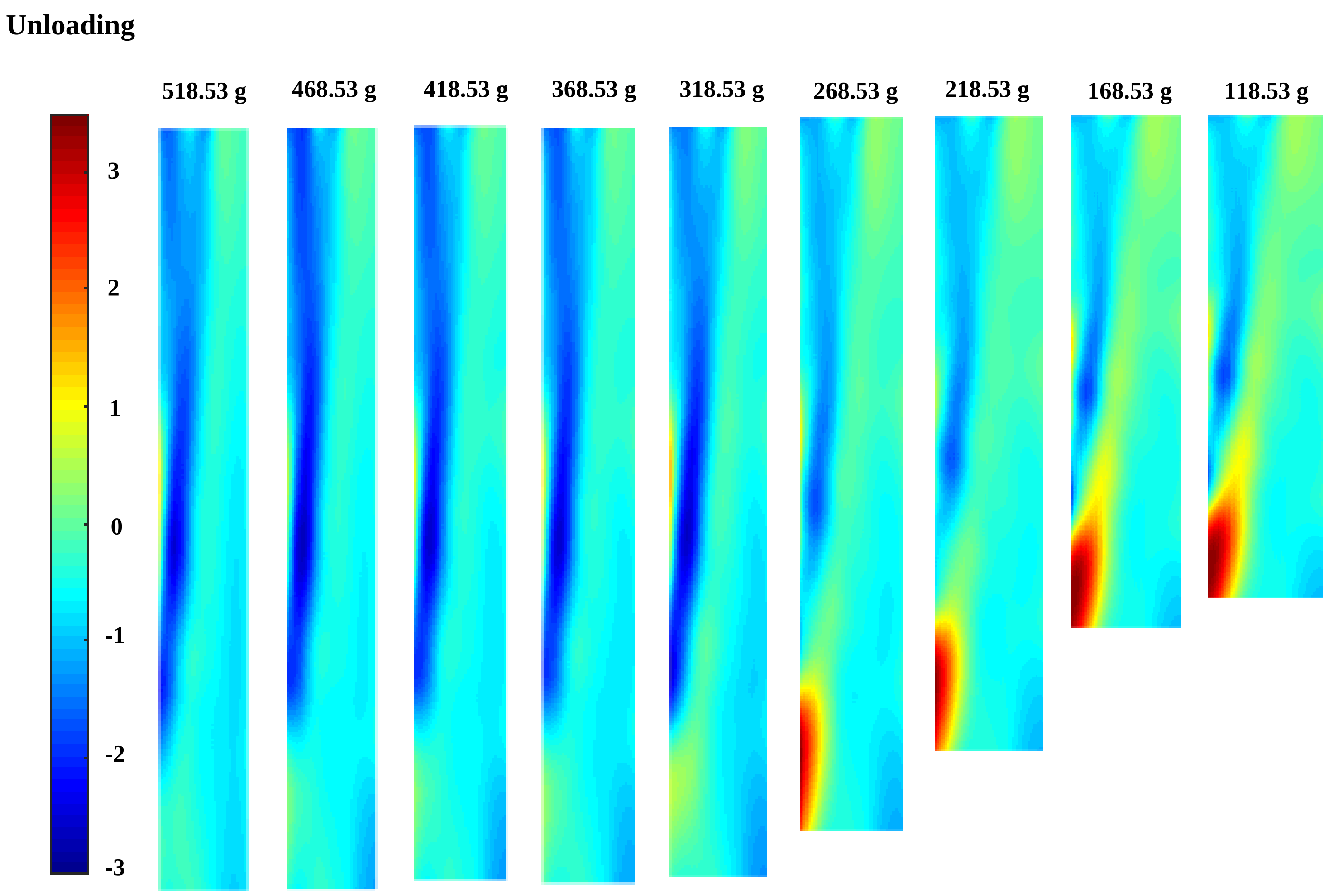


Figure 31: Graphical representation of total horizontal strain during unloading process – experiment 1

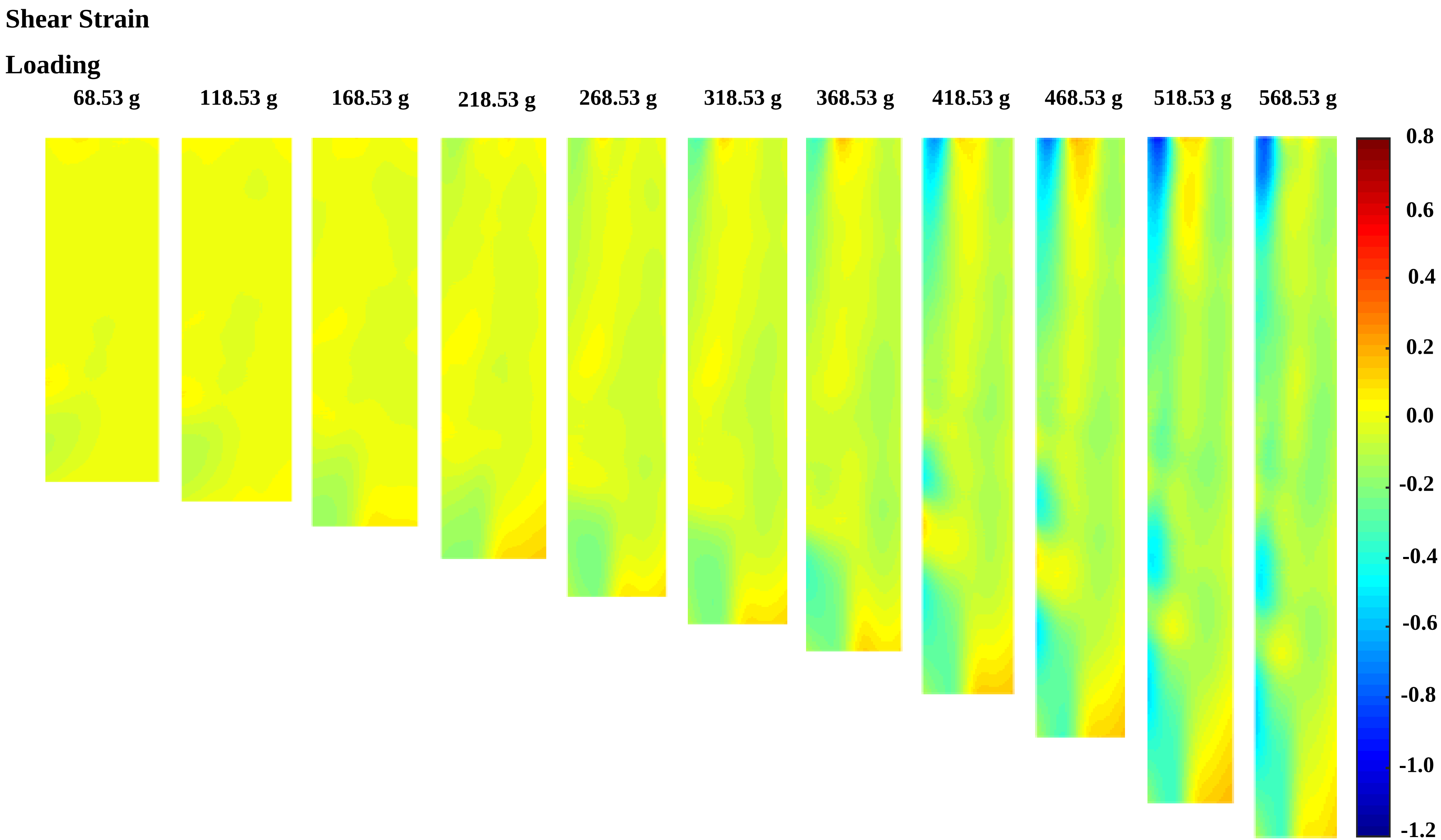


Figure 32: Graphical representation of total shear strain during loading process – experiment 1

Unloading

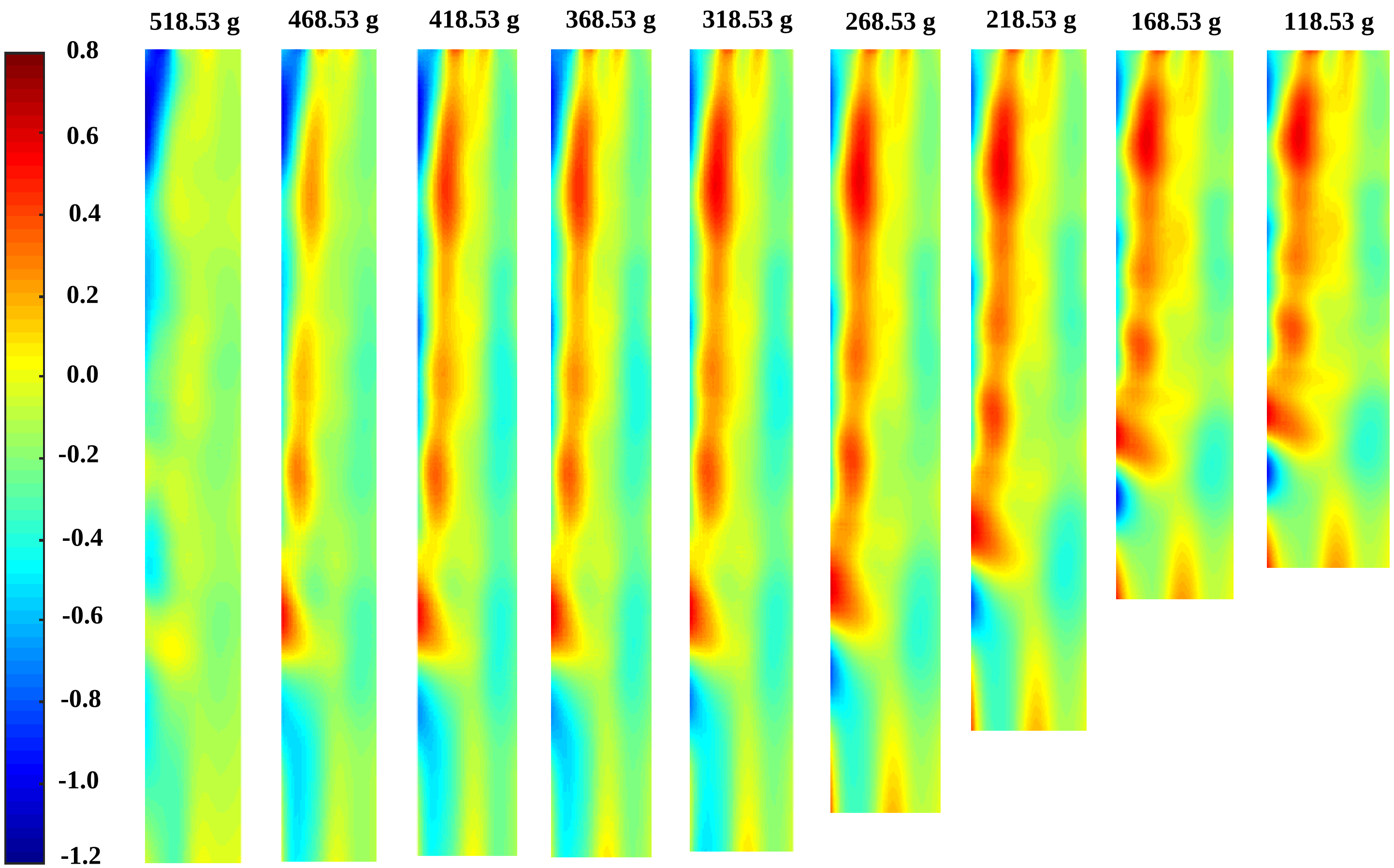


Figure 33: Graphical representation of total shear strain during unloading process – experiment 1

Specimen (60 mm x 20 mm) - 2

Vertical Displacement

Loading

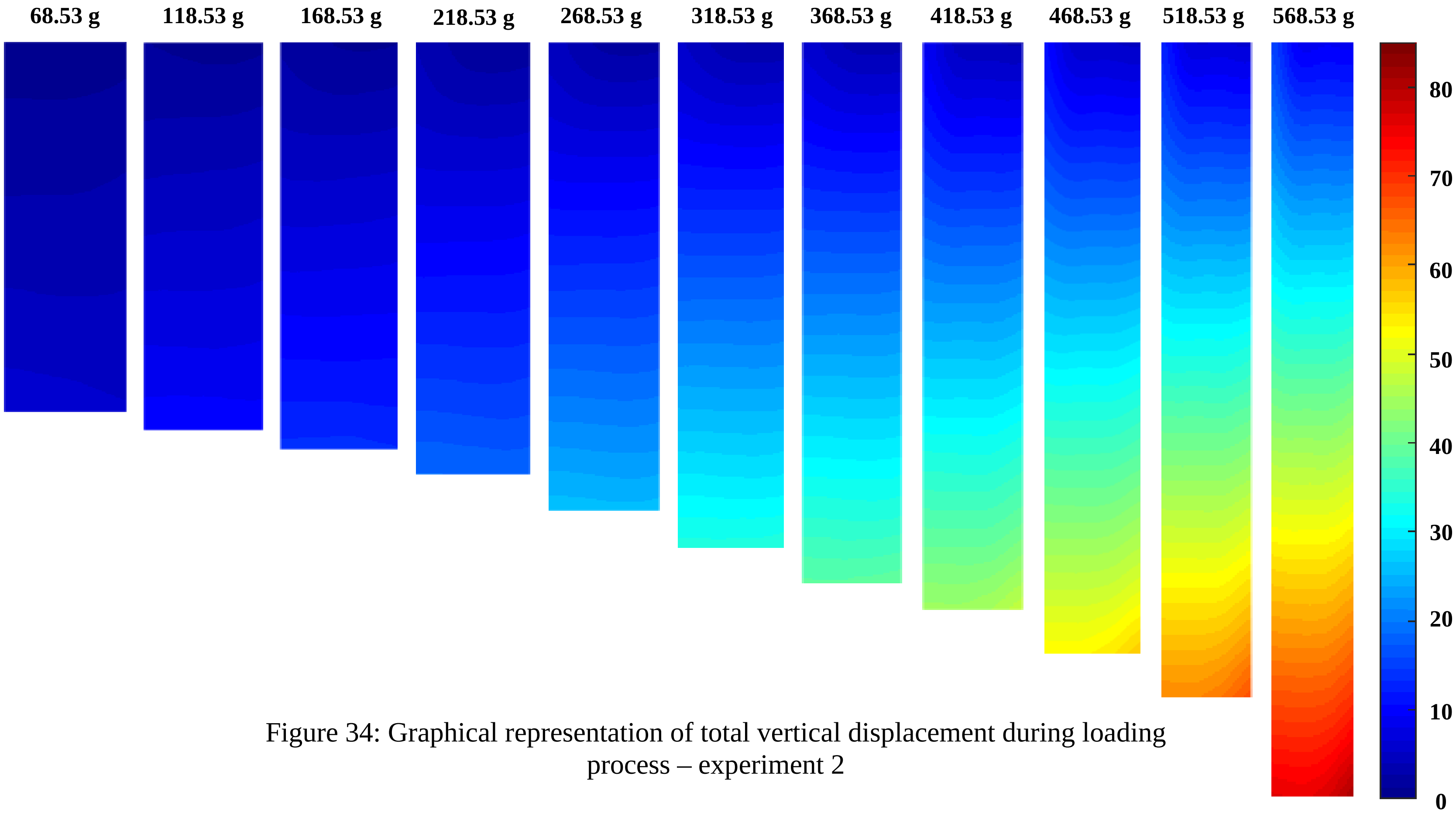


Figure 34: Graphical representation of total vertical displacement during loading process – experiment 2

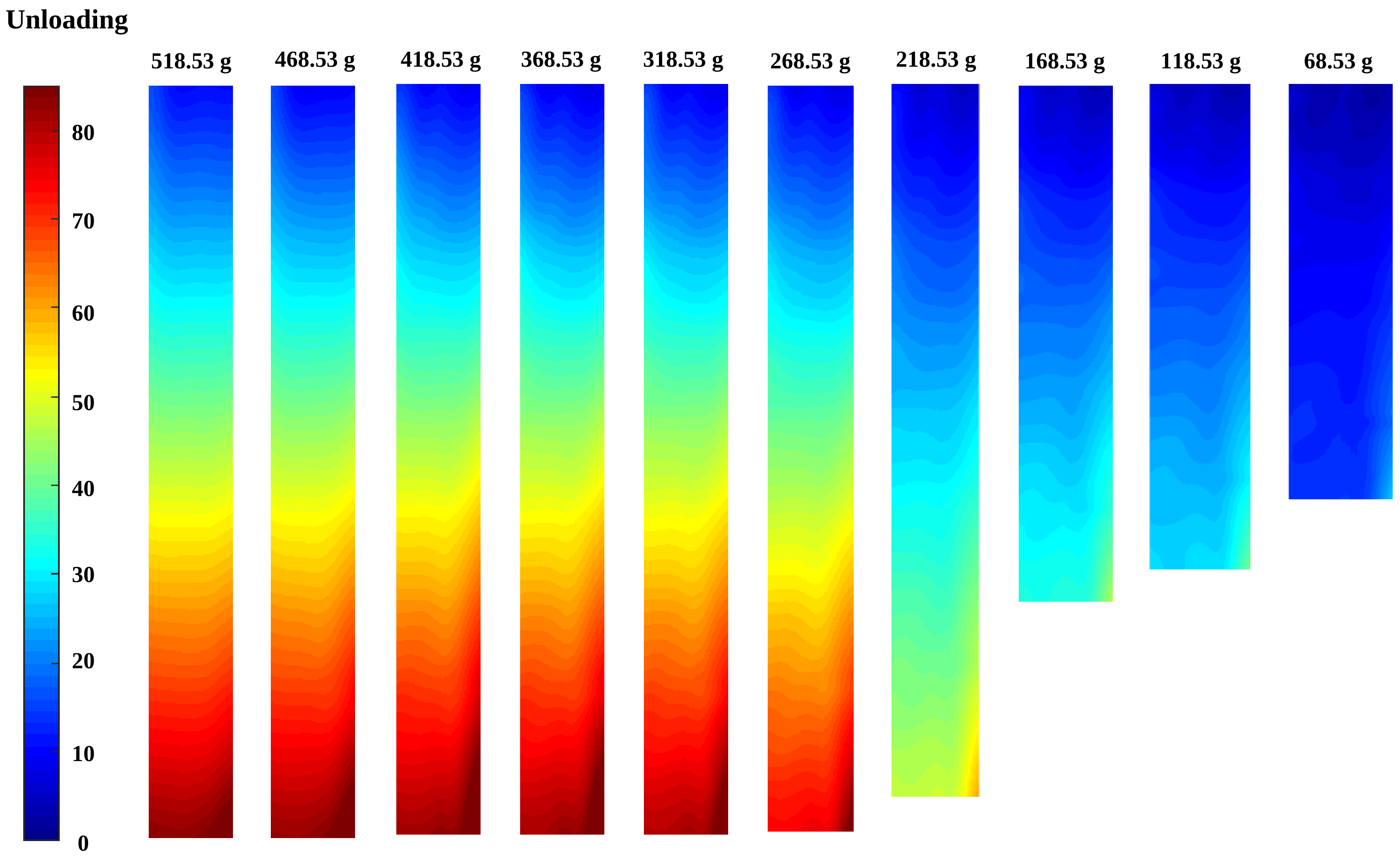


Figure 35: Graphical representation of total vertical displacement during unloading process – experiment 2

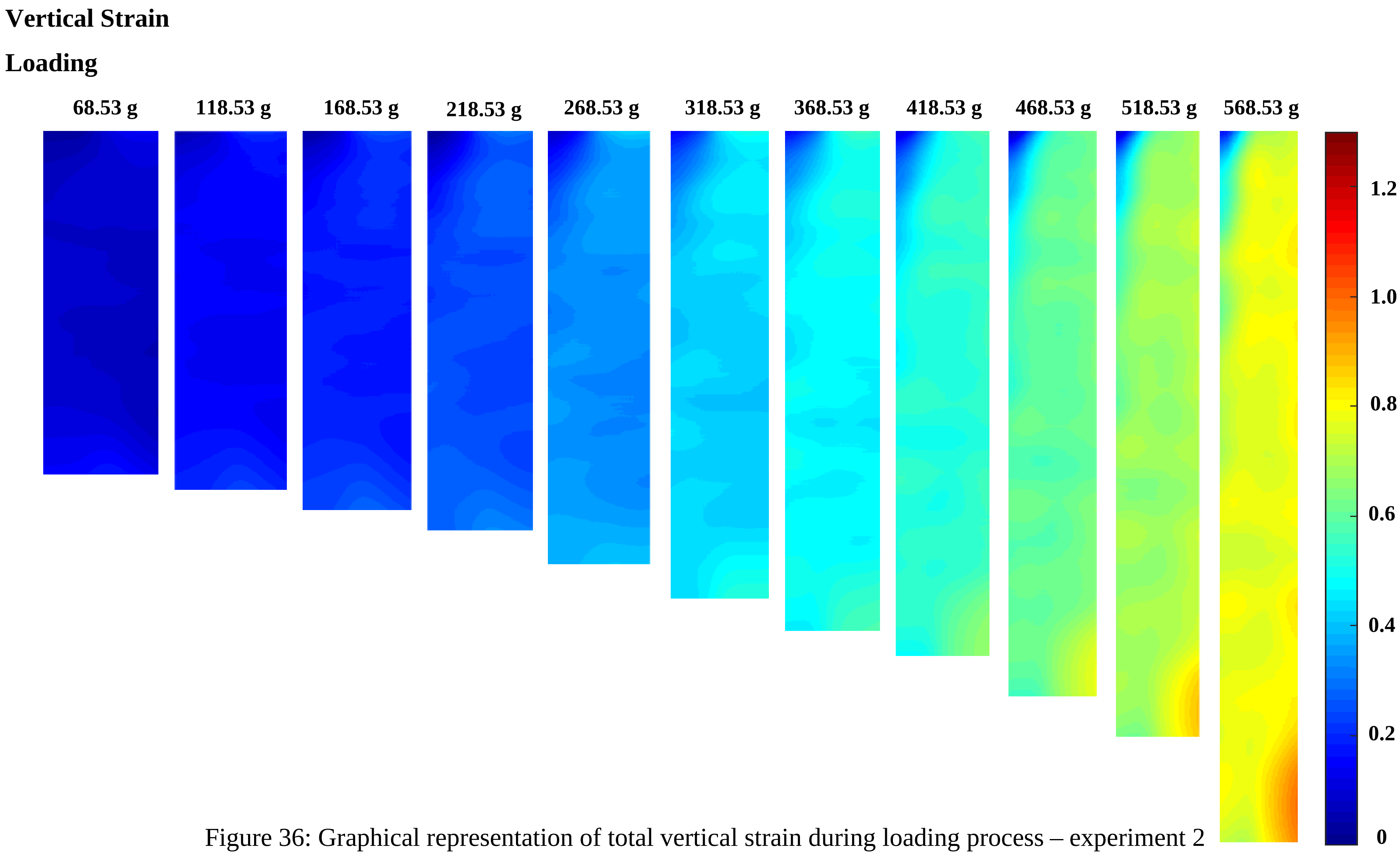


Figure 36: Graphical representation of total vertical strain during loading process – experiment 2

Unloading

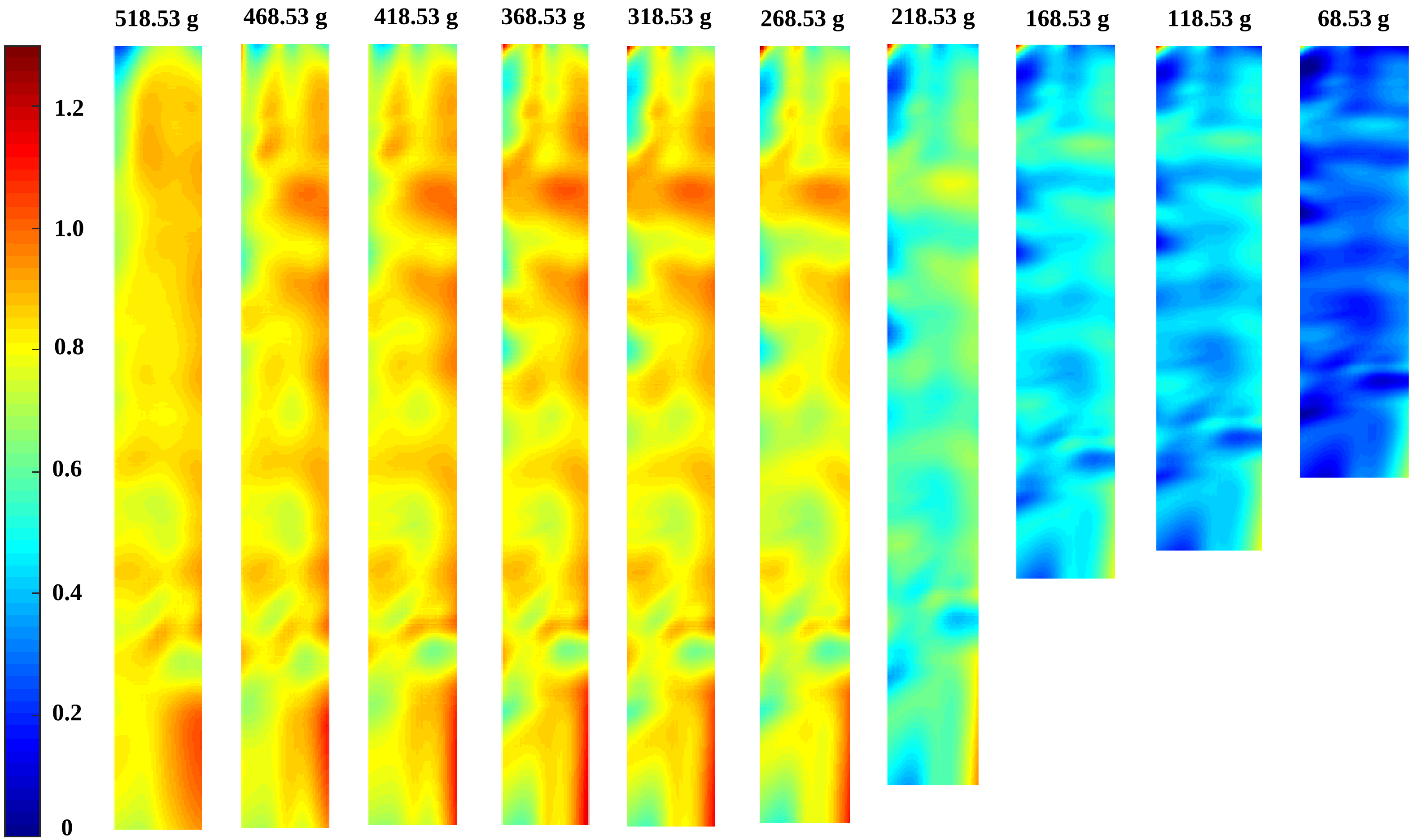


Figure 37: Graphical representation of total vertical strain during unloading process – experiment 2

Horizontal Displacement

Loading

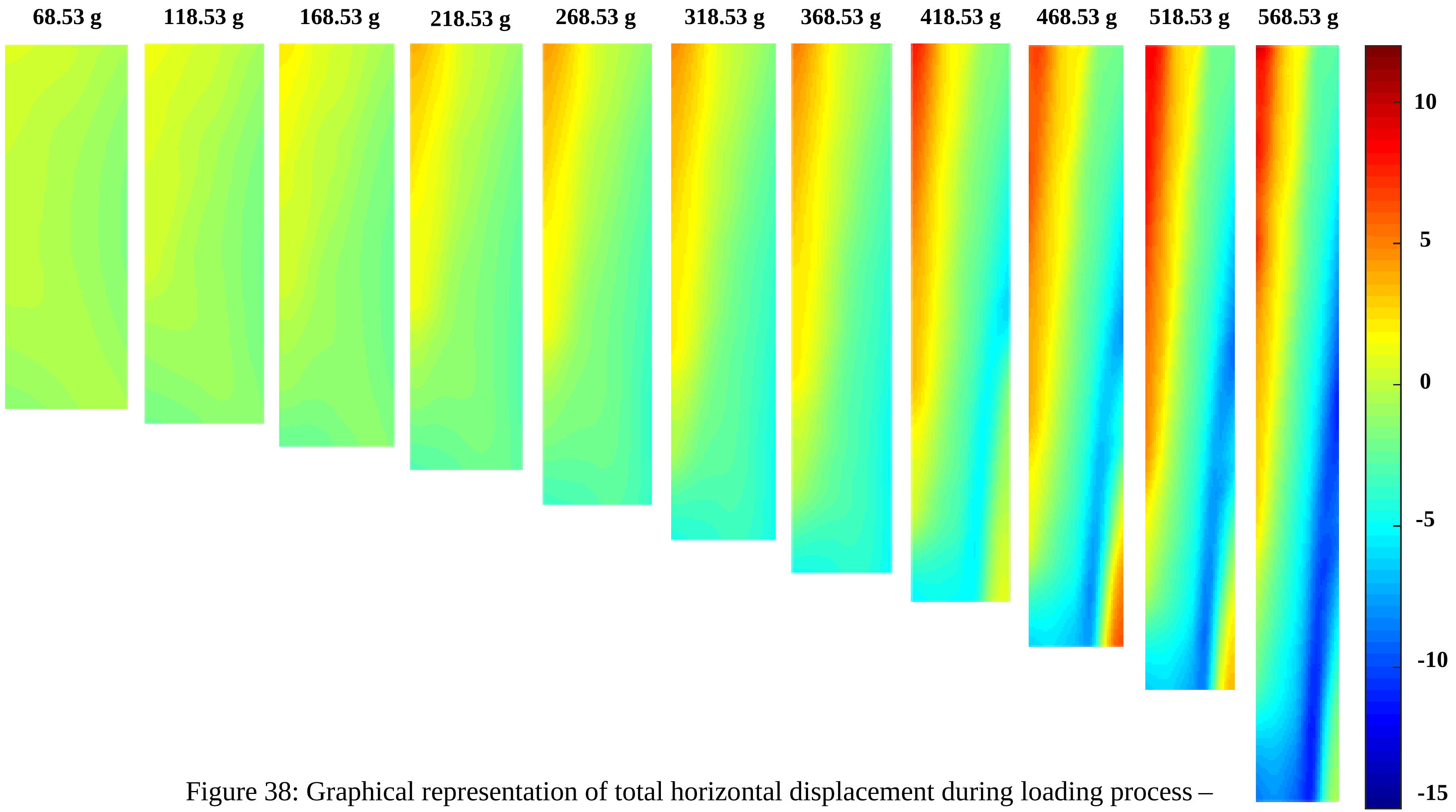


Figure 38: Graphical representation of total horizontal displacement during loading process – experiment 2

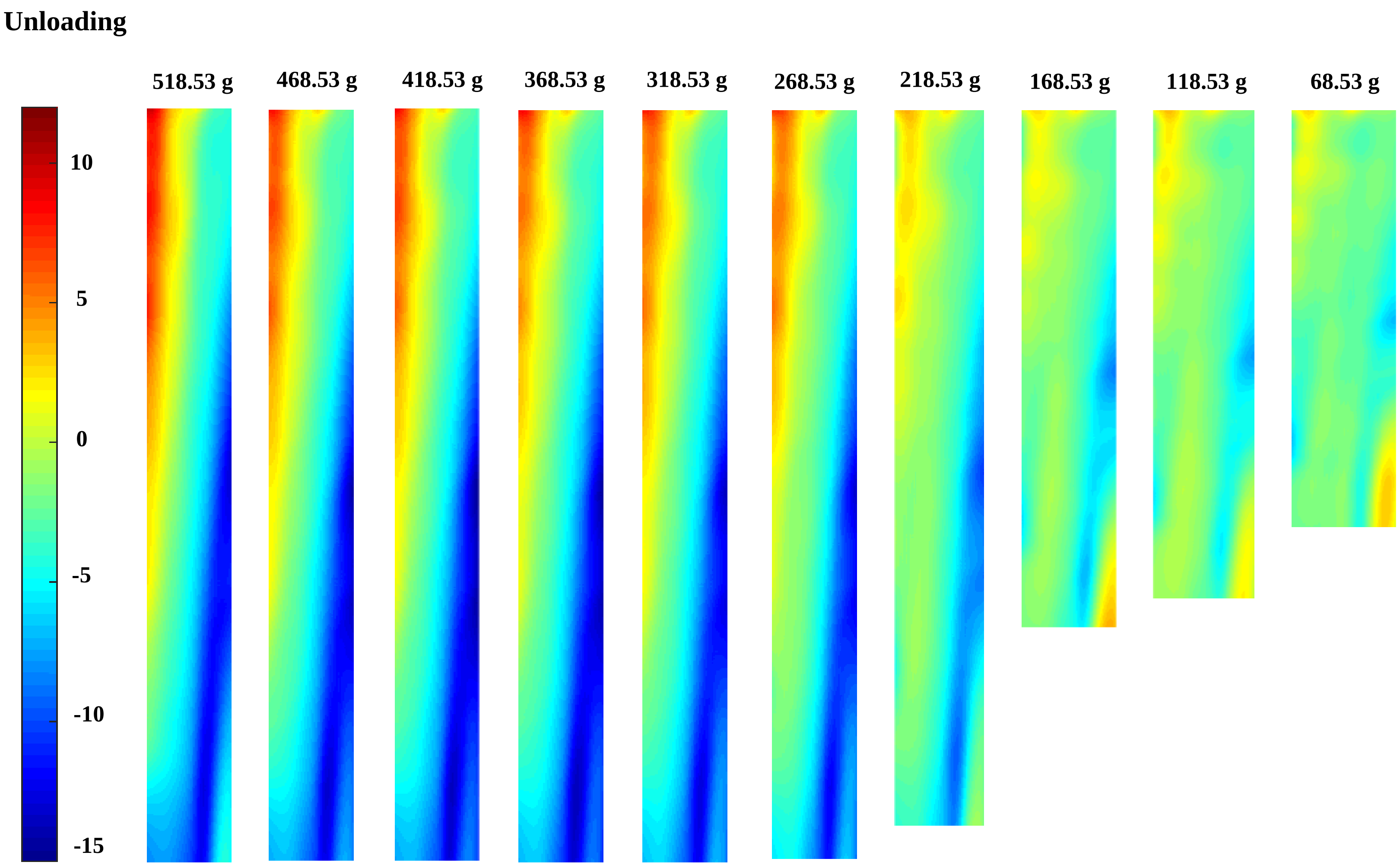


Figure 39: Graphical representation of total horizontal displacement during unloading process – experiment 2

Horizontal Strain

Loading

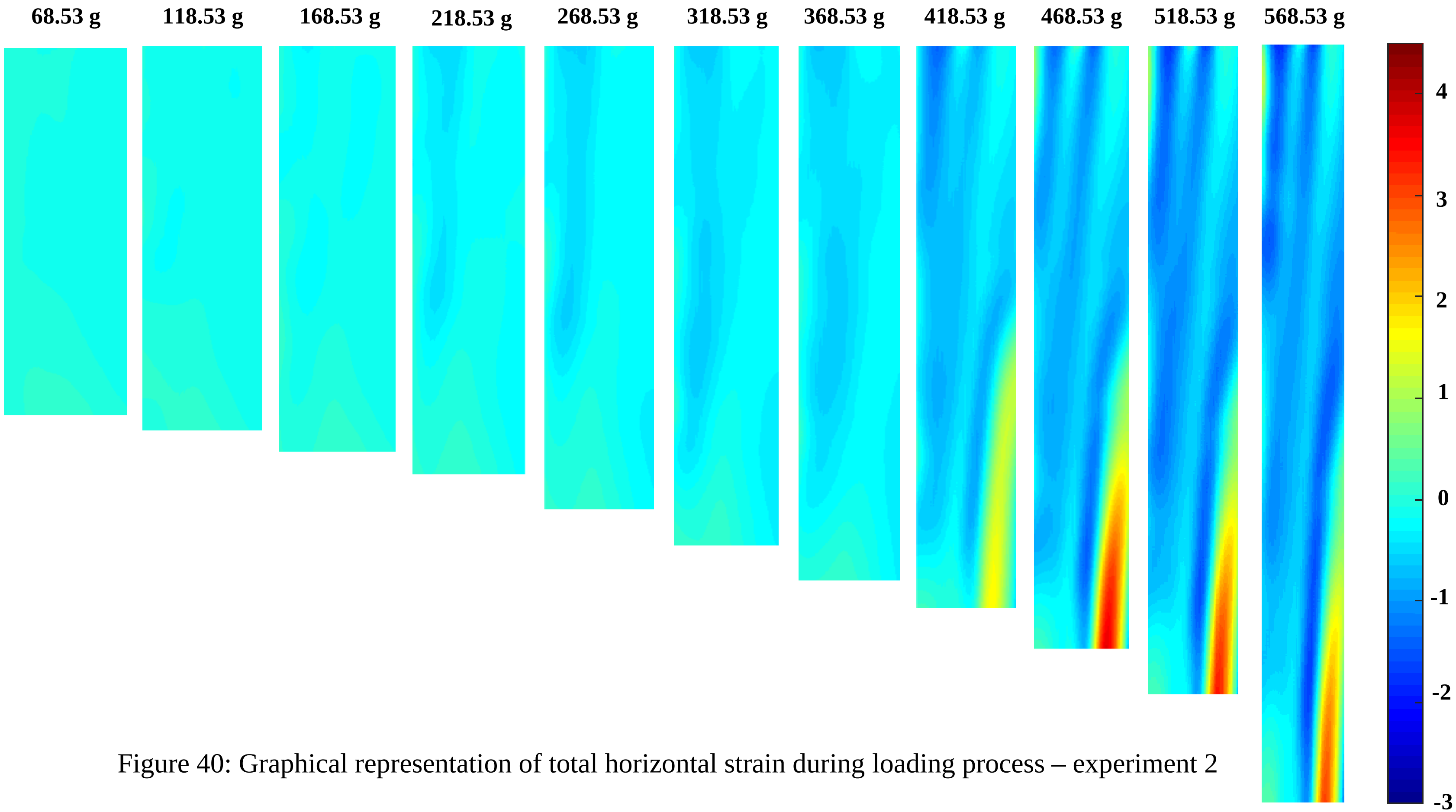


Figure 40: Graphical representation of total horizontal strain during loading process – experiment 2

Unloading

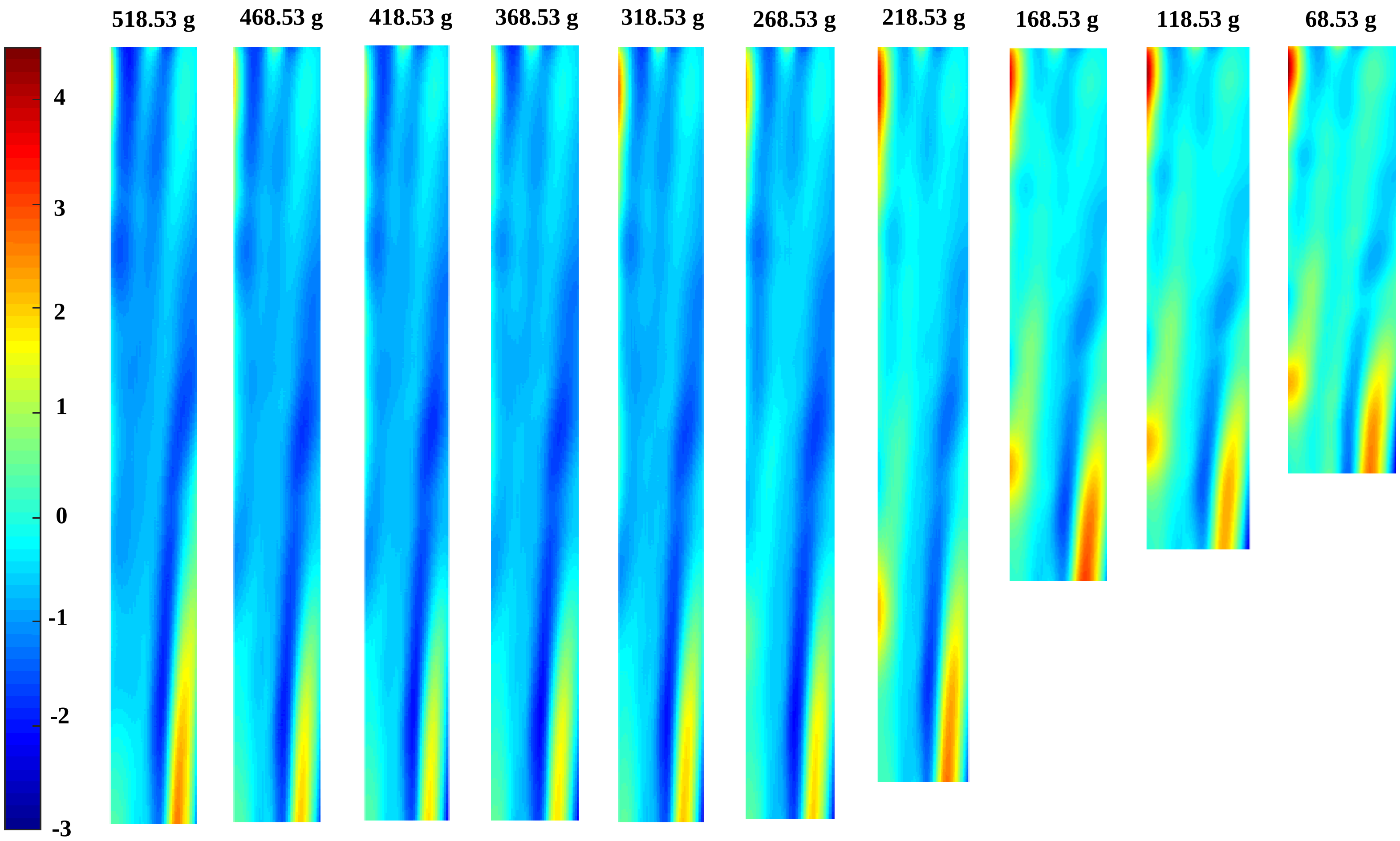


Figure 41: Graphical representation of total horizontal strain during unloading process – experiment 2

Shear Strain

Loading

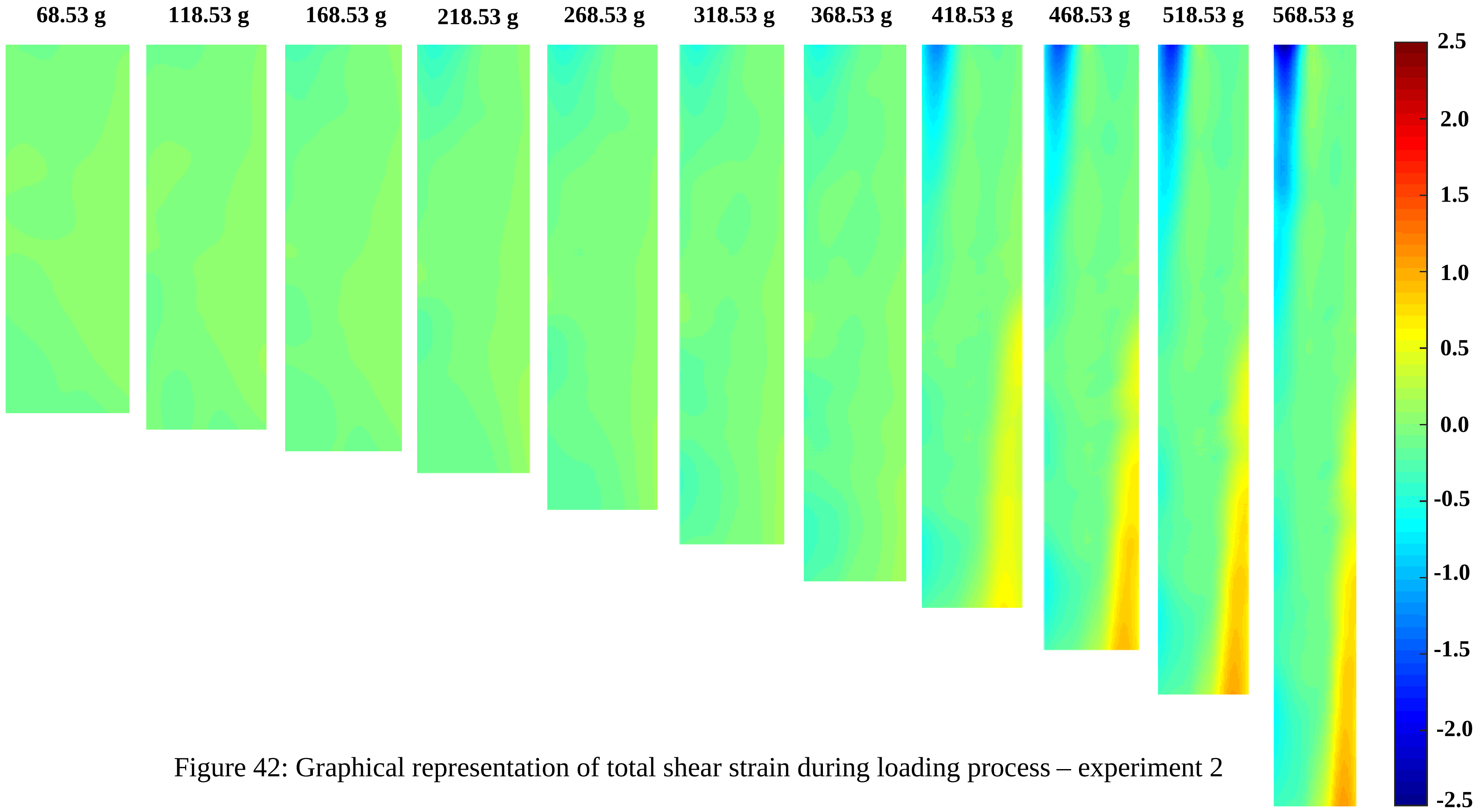


Figure 42: Graphical representation of total shear strain during loading process – experiment 2

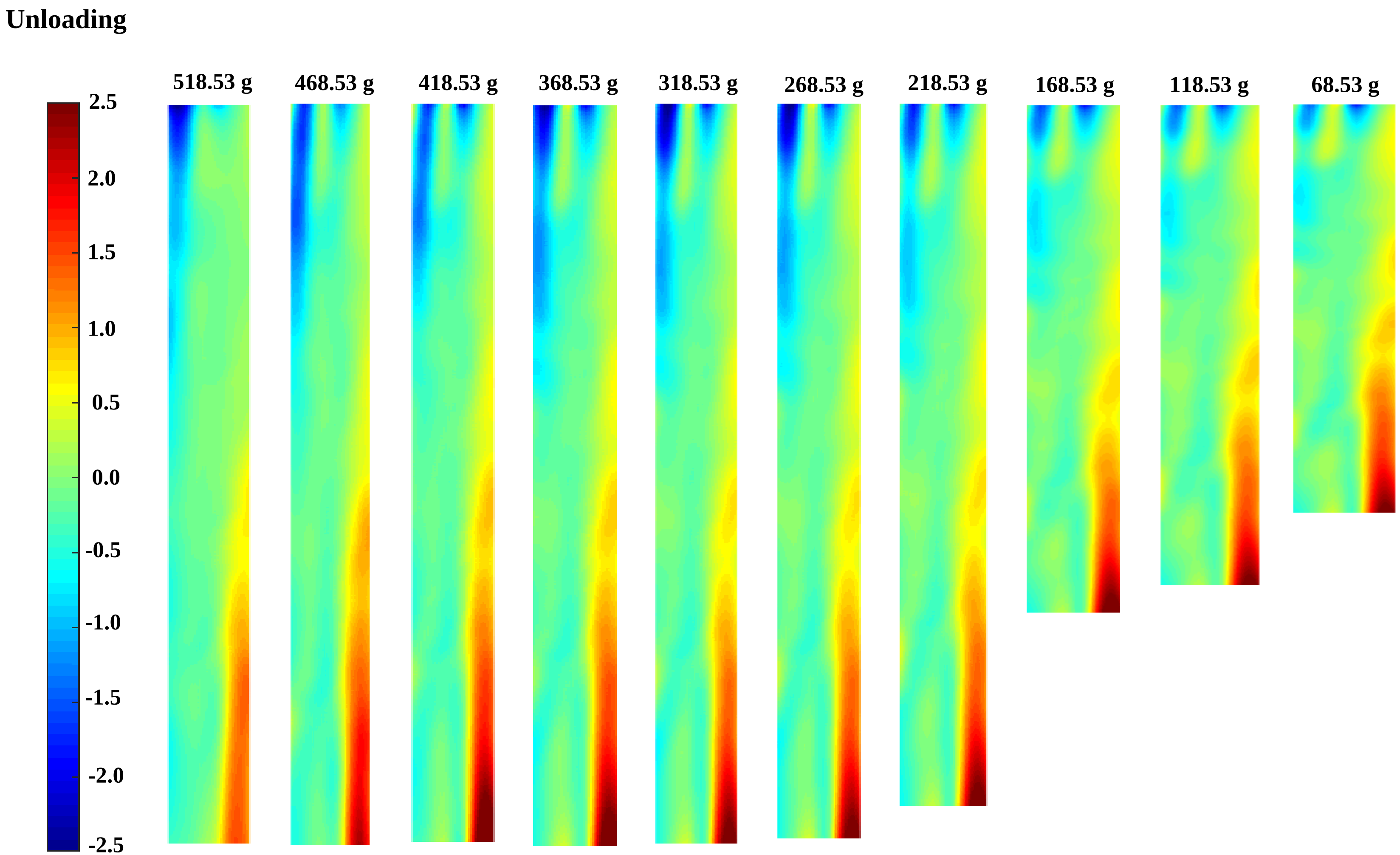


Figure 43: Graphical representation of total shear strain during unloading process – experiment 2

Specimen (60 mm x 20 mm) - 3

Vertical Displacement

Loading

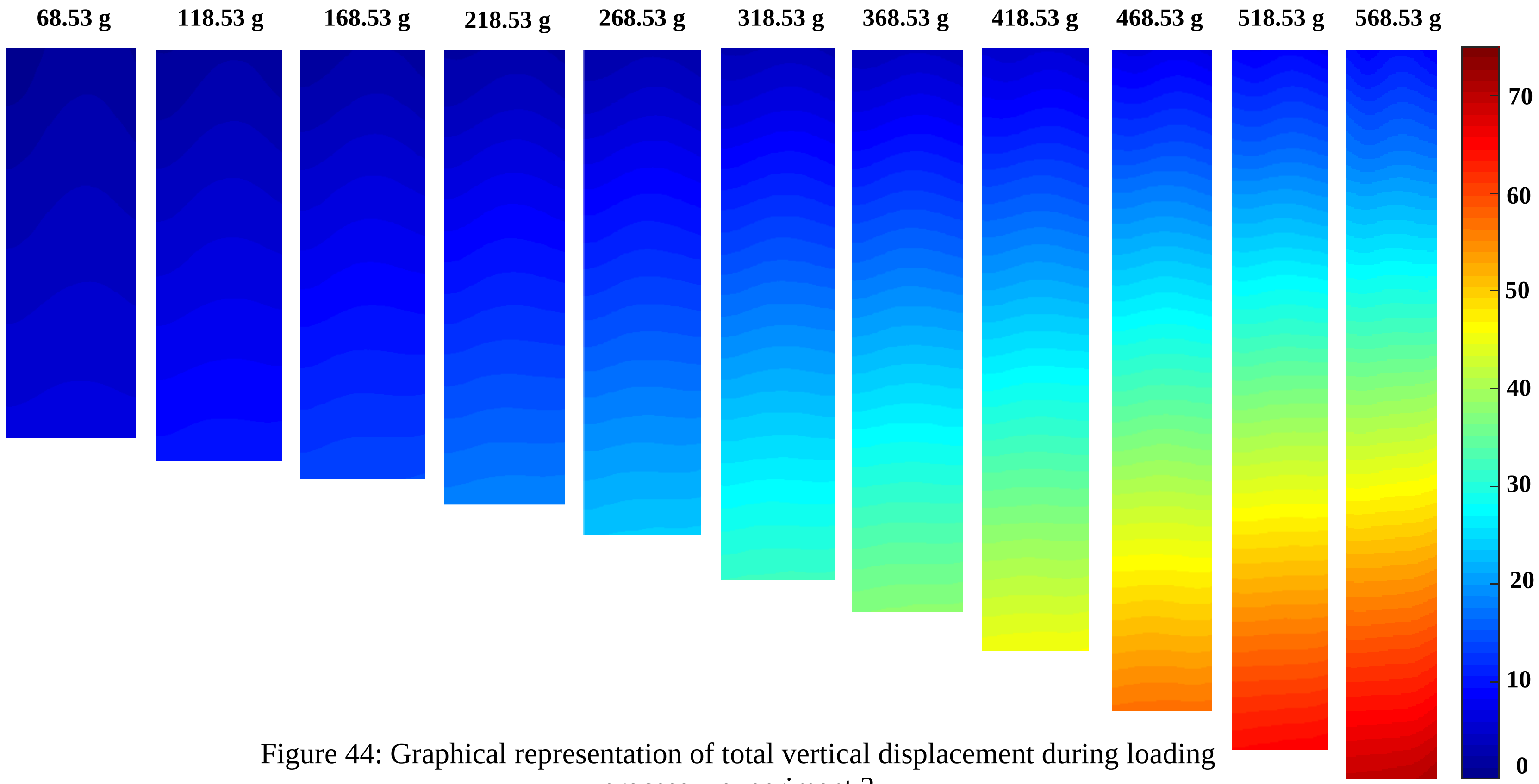


Figure 44: Graphical representation of total vertical displacement during loading process – experiment 3

Unloading

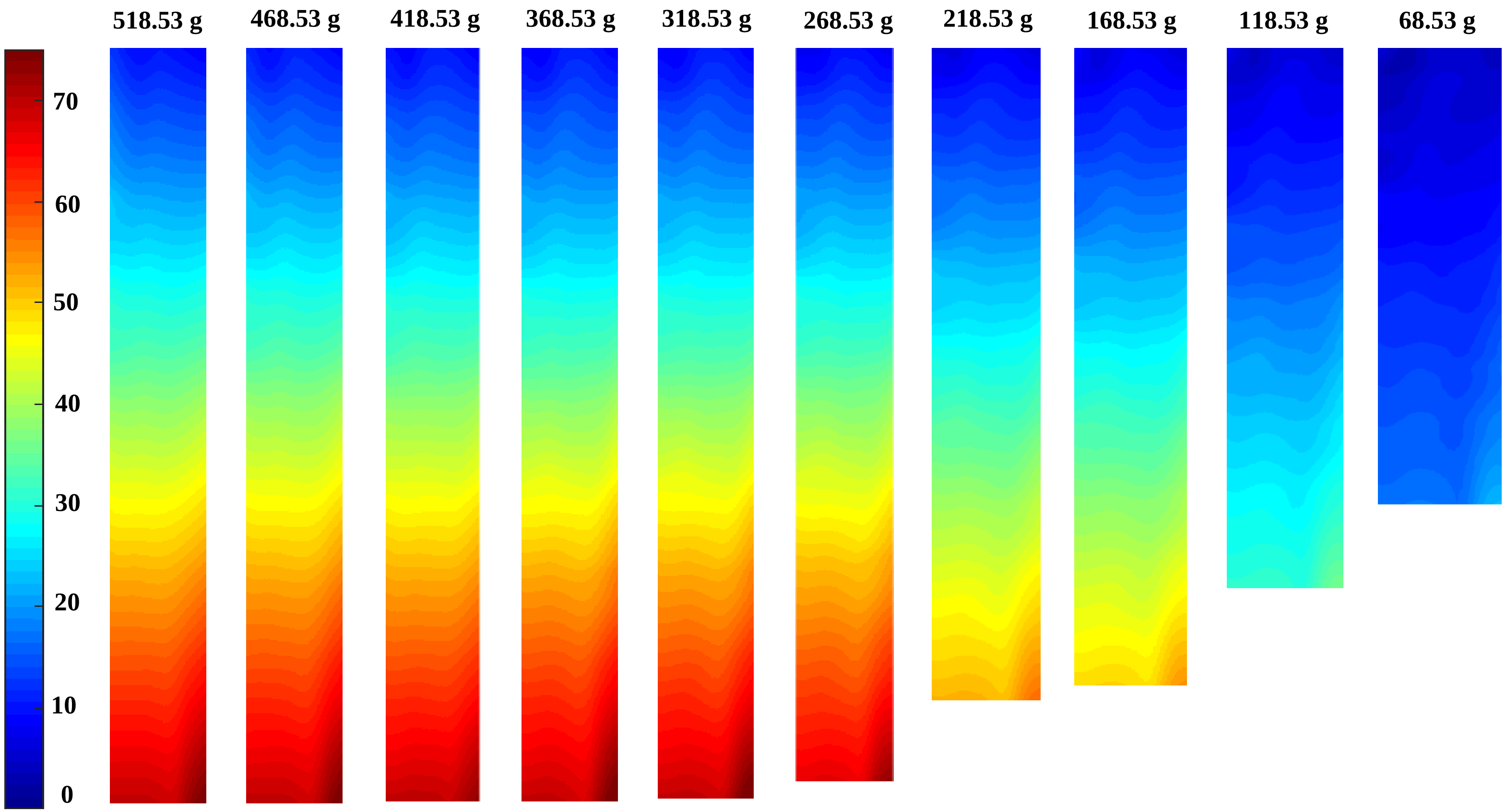


Figure 45: Graphical representation of total vertical displacement during unloading process – experiment 3

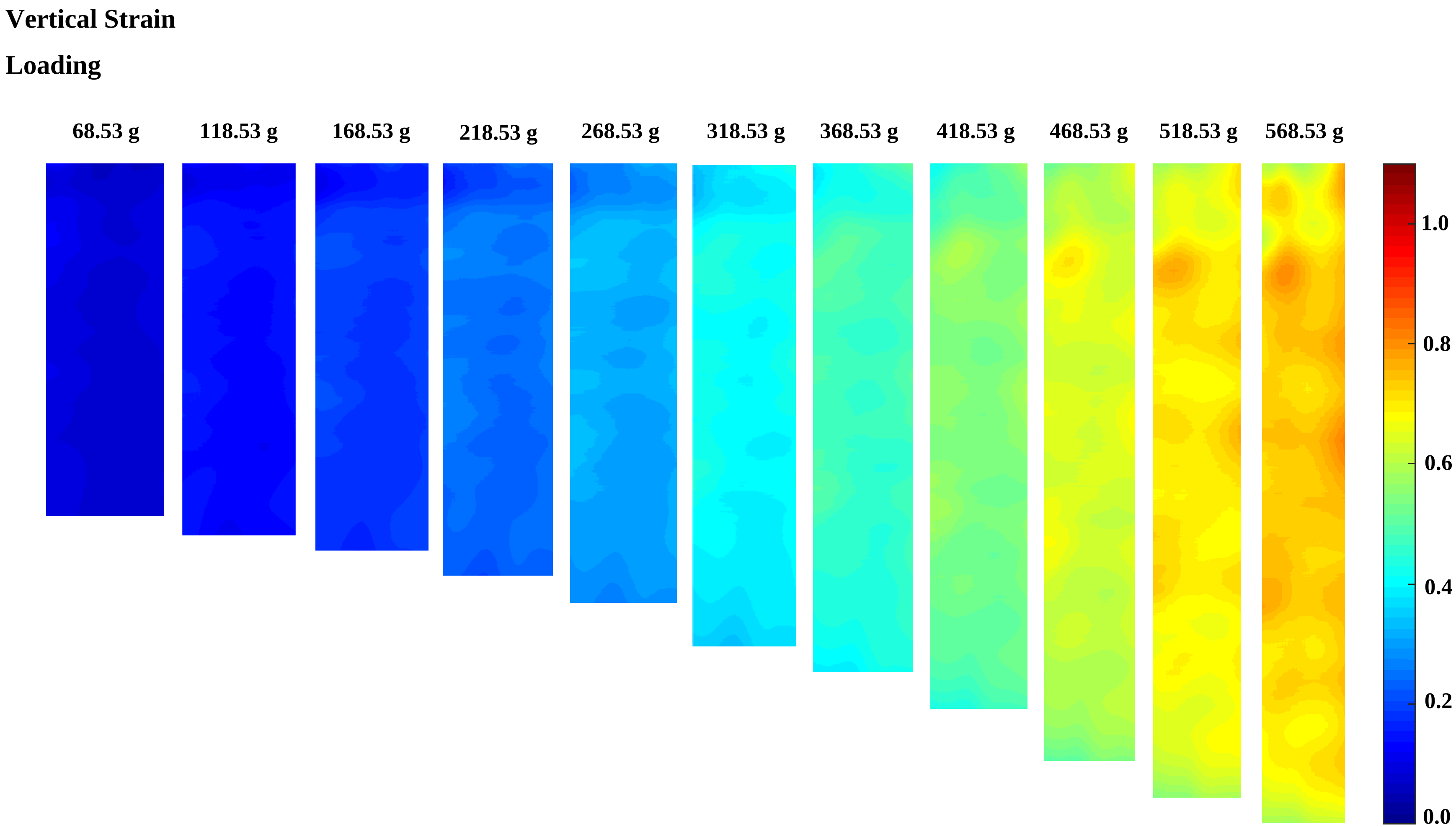


Figure 46: Graphical representation of total vertical strain during loading process – experiment 3

Unloading

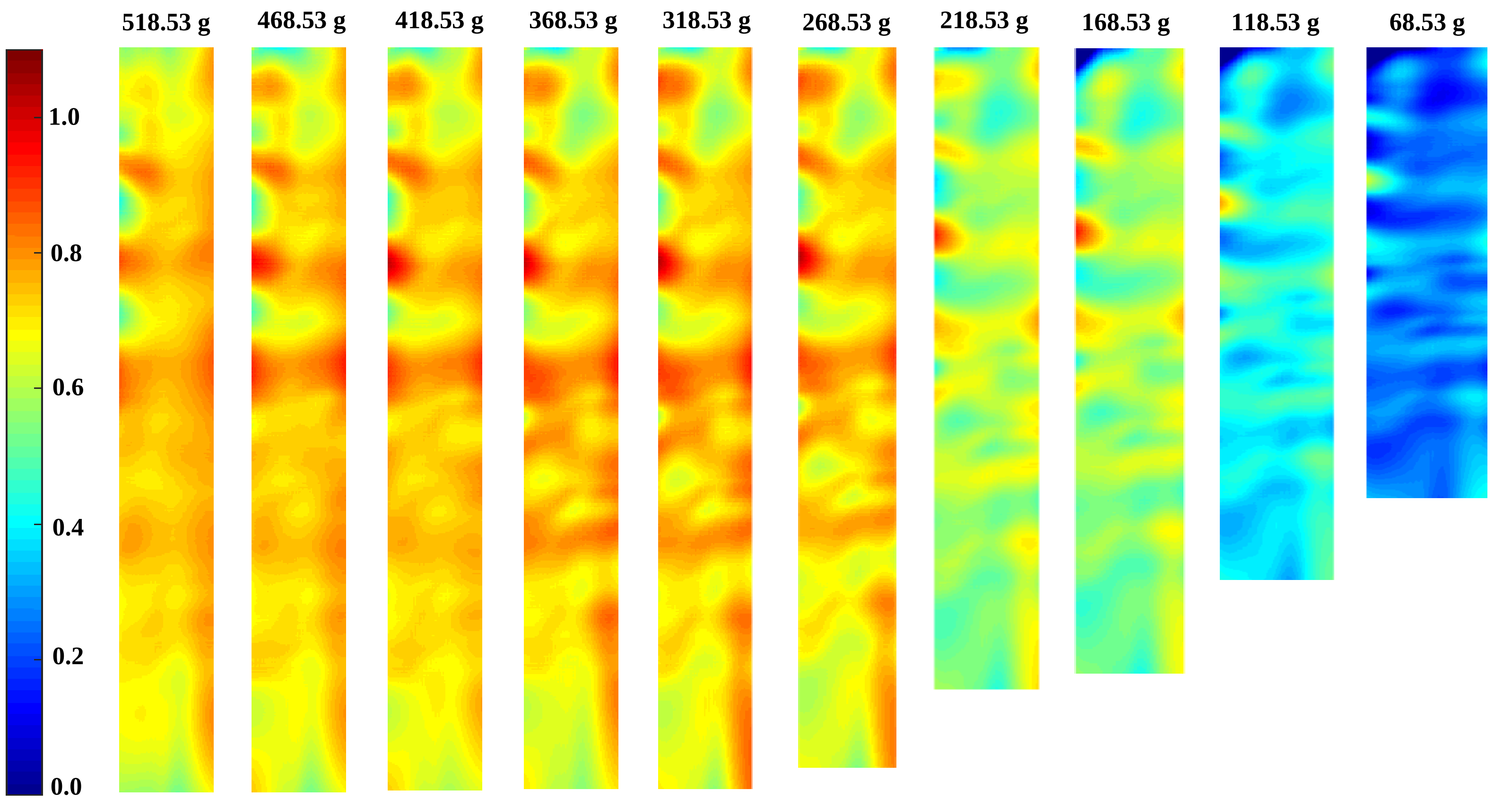


Figure 47: Graphical representation of total vertical strain during unloading process – experiment 3

Horizontal Displacement

Loading

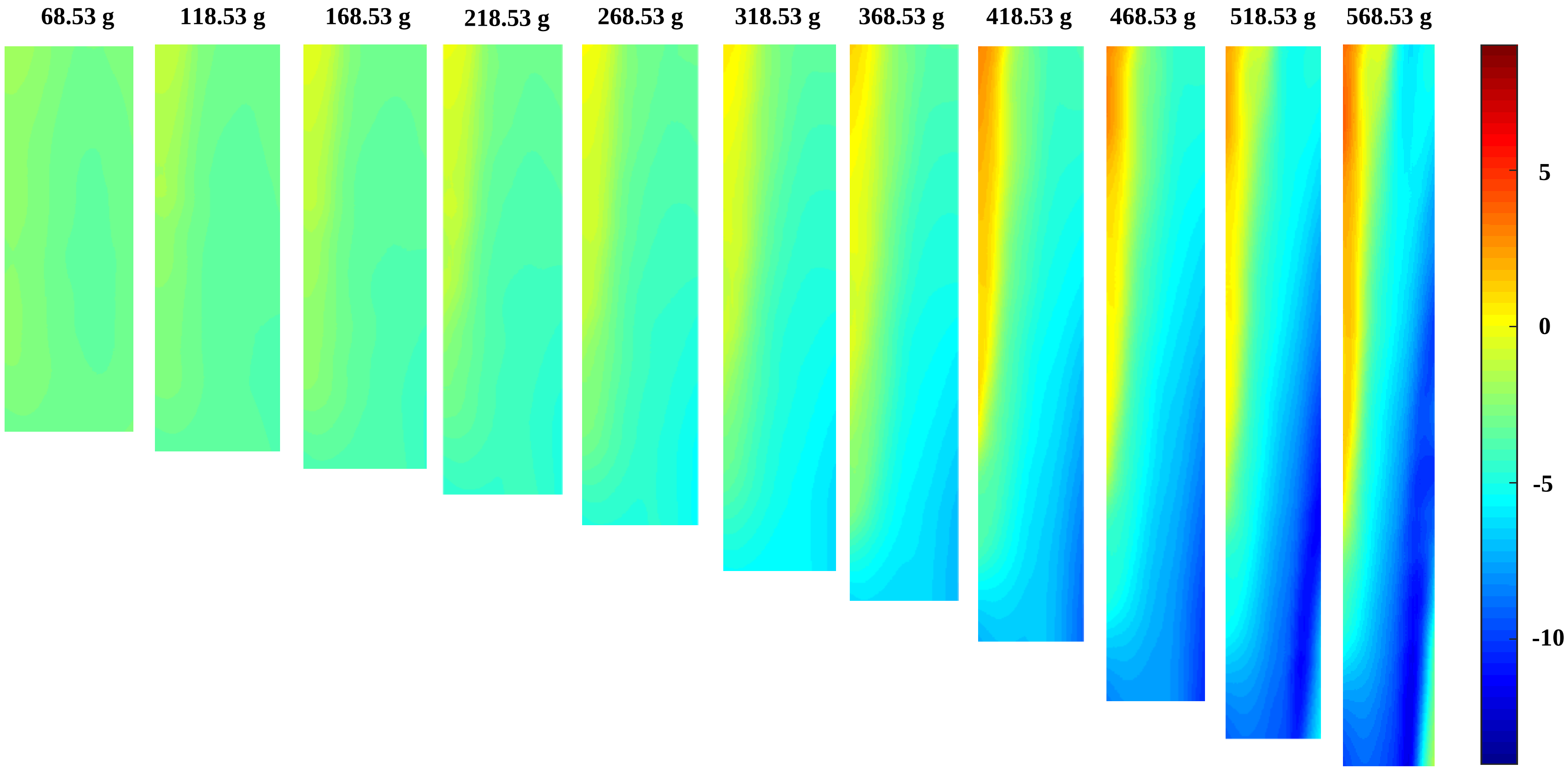


Figure 48: Graphical representation of total horizontal displacement during loading process – experiment 3

Unloading

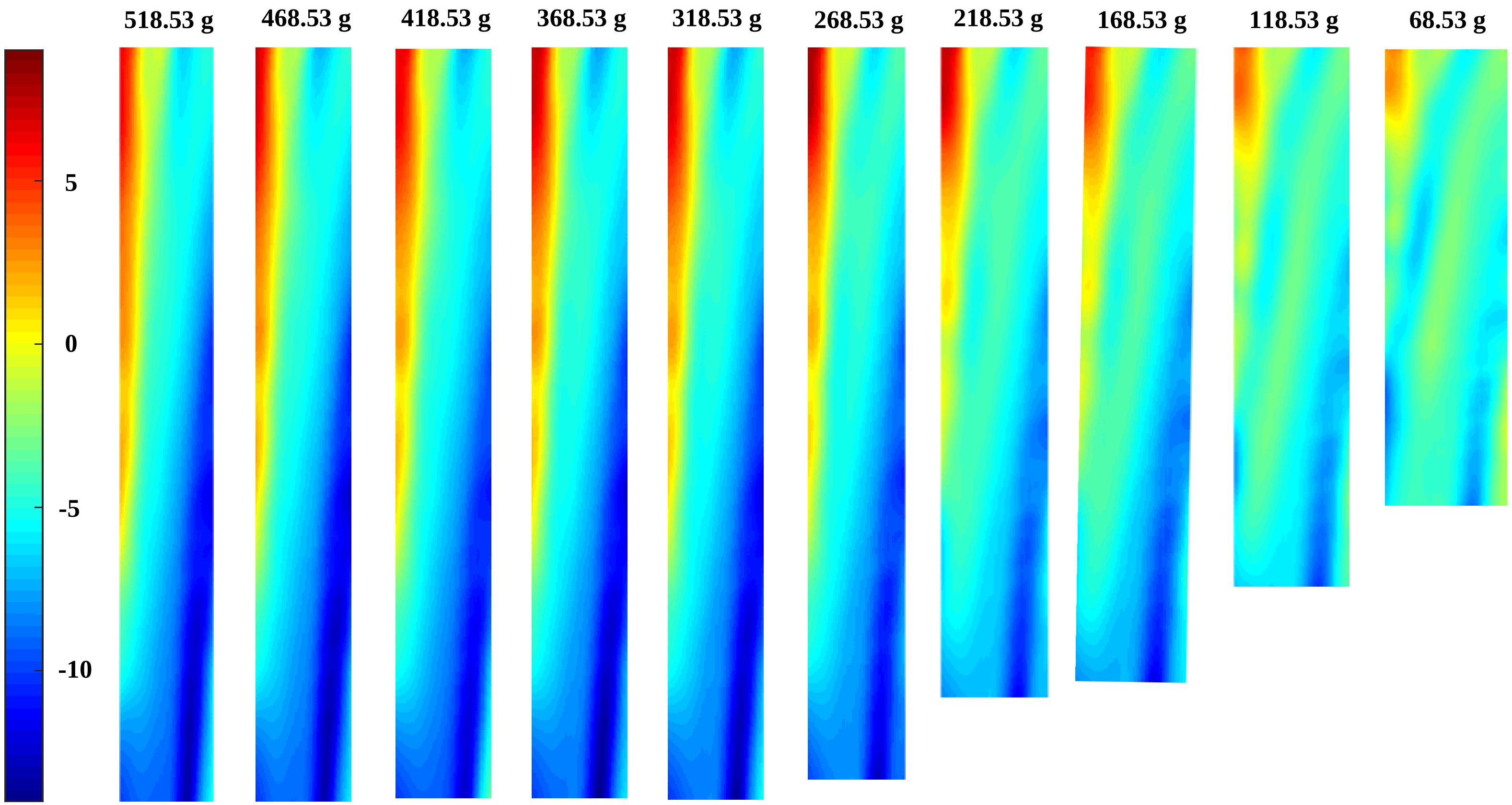


Figure 49: Graphical representation of total horizontal displacement during unloading process – experiment 3

Horizontal Strain

Loading

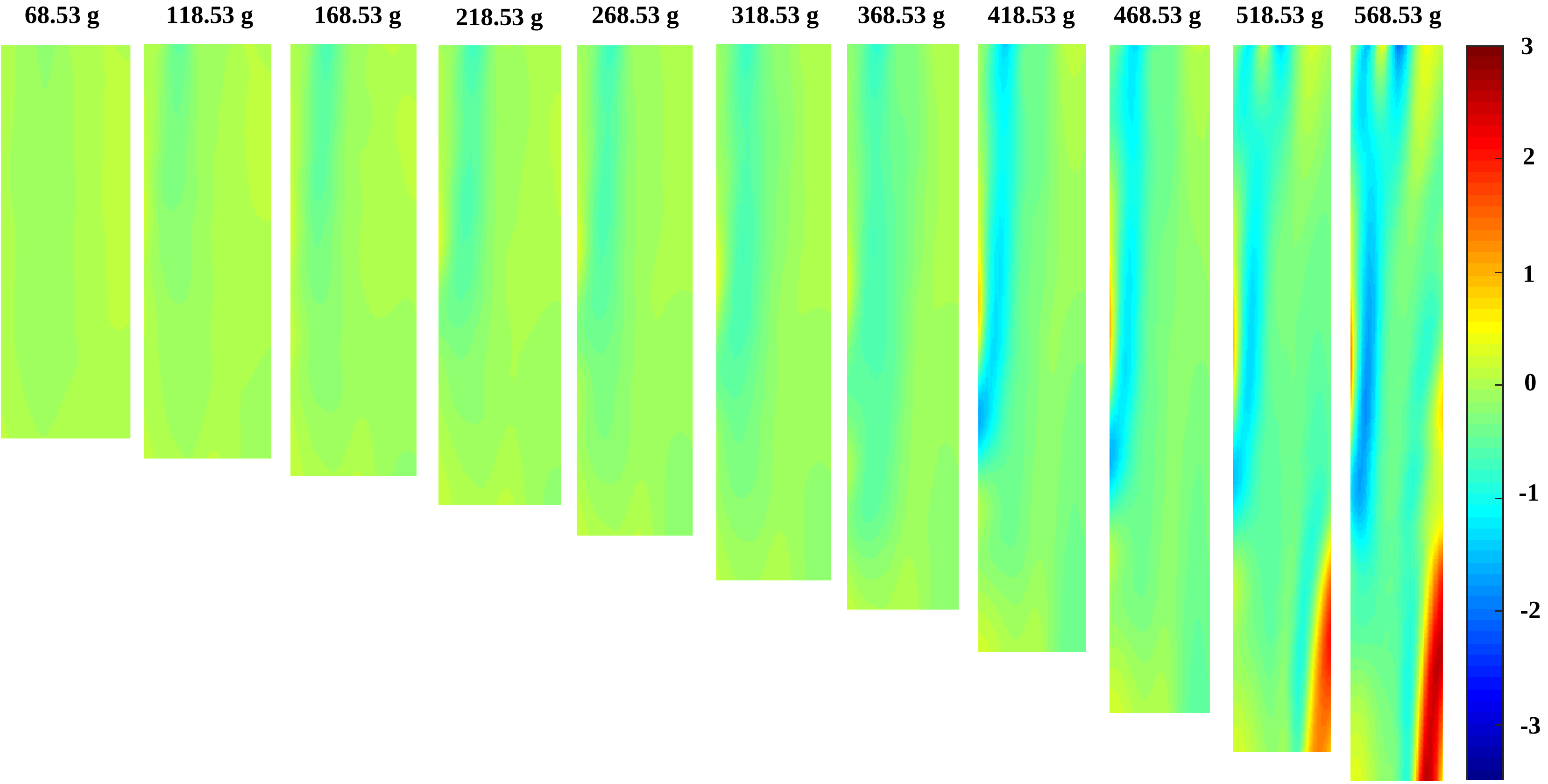


Figure 50: Graphical representation of total horizontal strain during loading process – experiment 3

Unloading

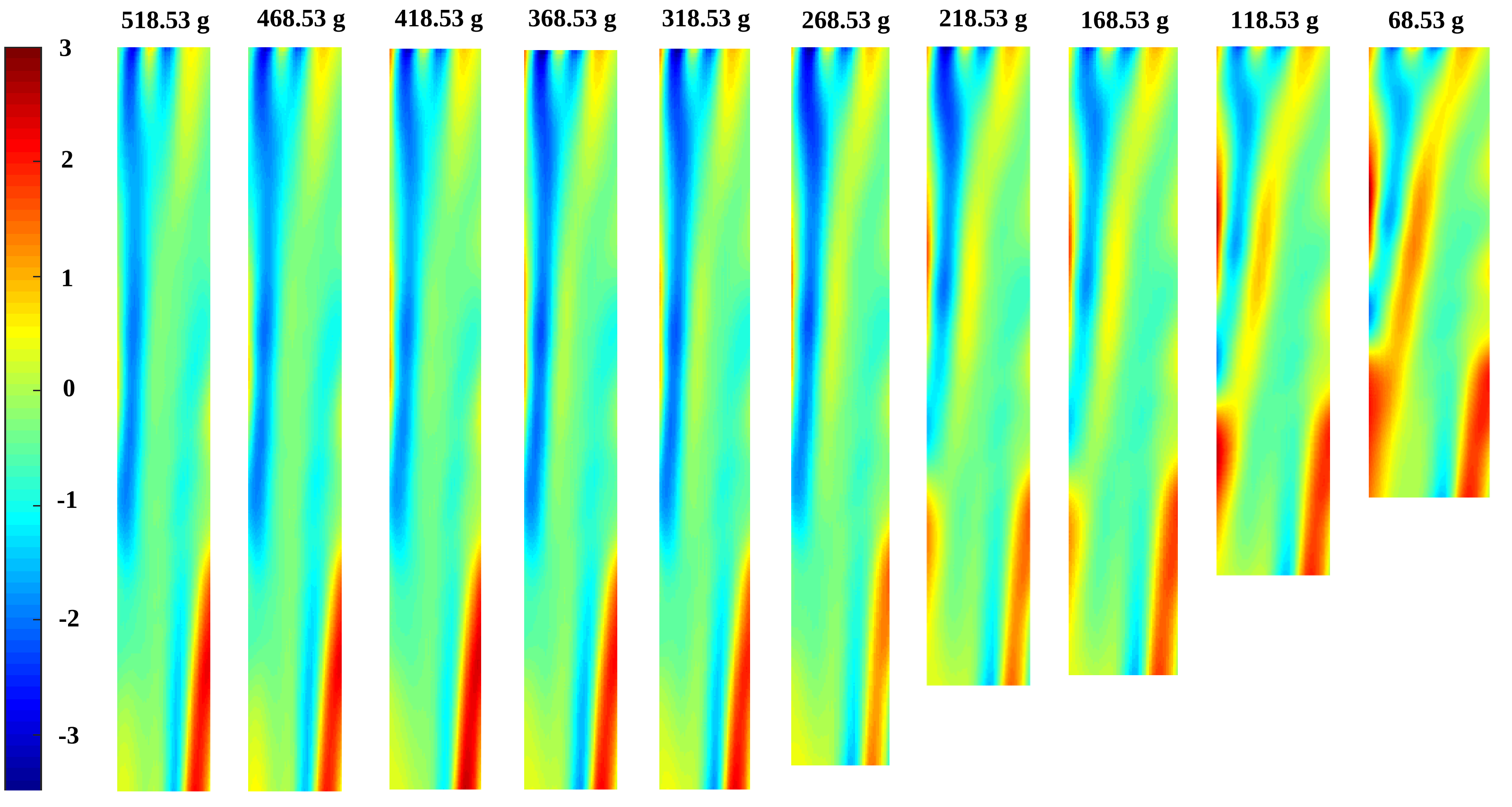


Figure 51: Graphical representation of total horizontal strain during unloading process – experiment 3

Shear Strain

Loading

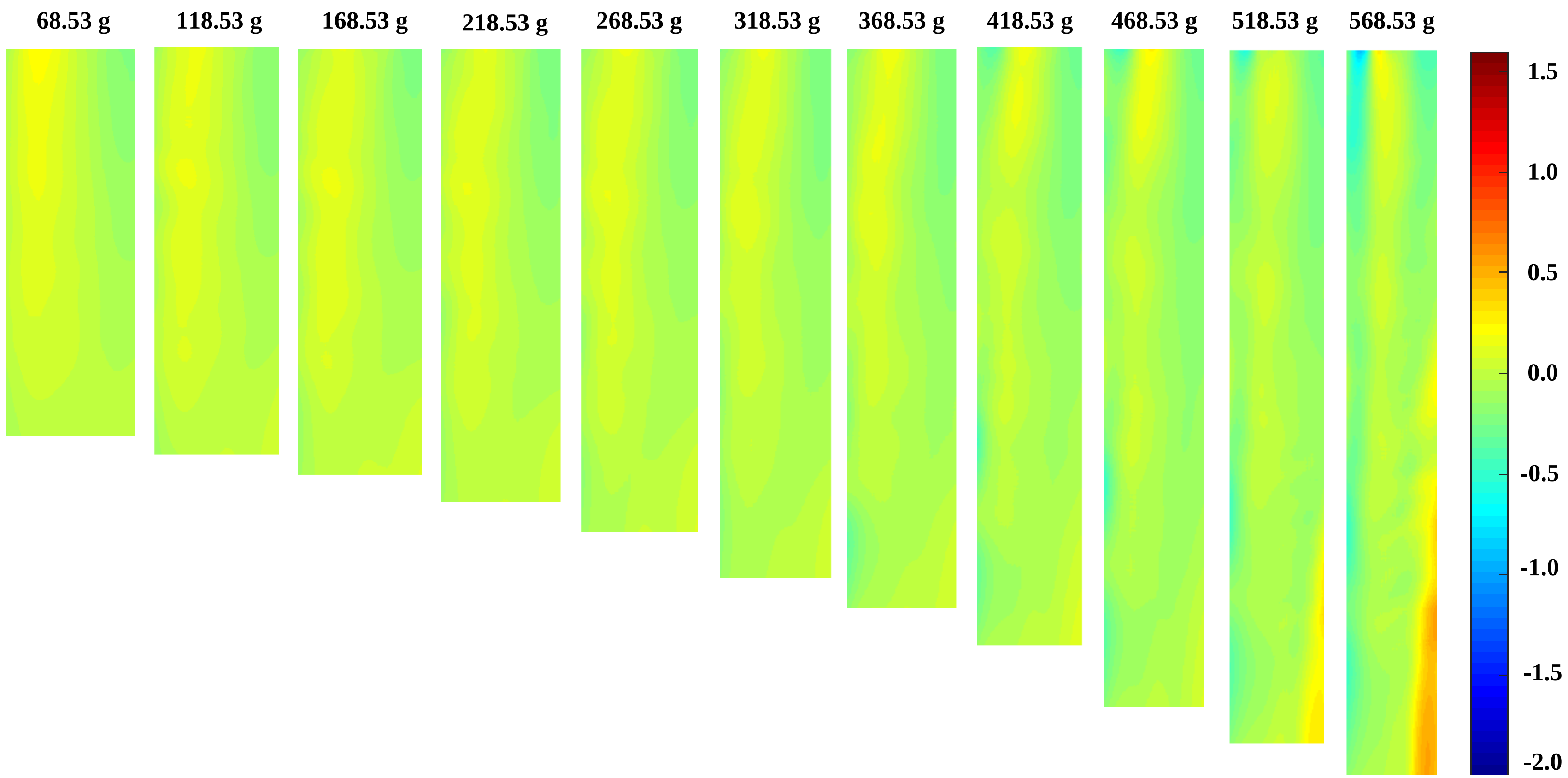


Figure 52: Graphical representation of total shear strain during loading process – experiment 3

Unloading

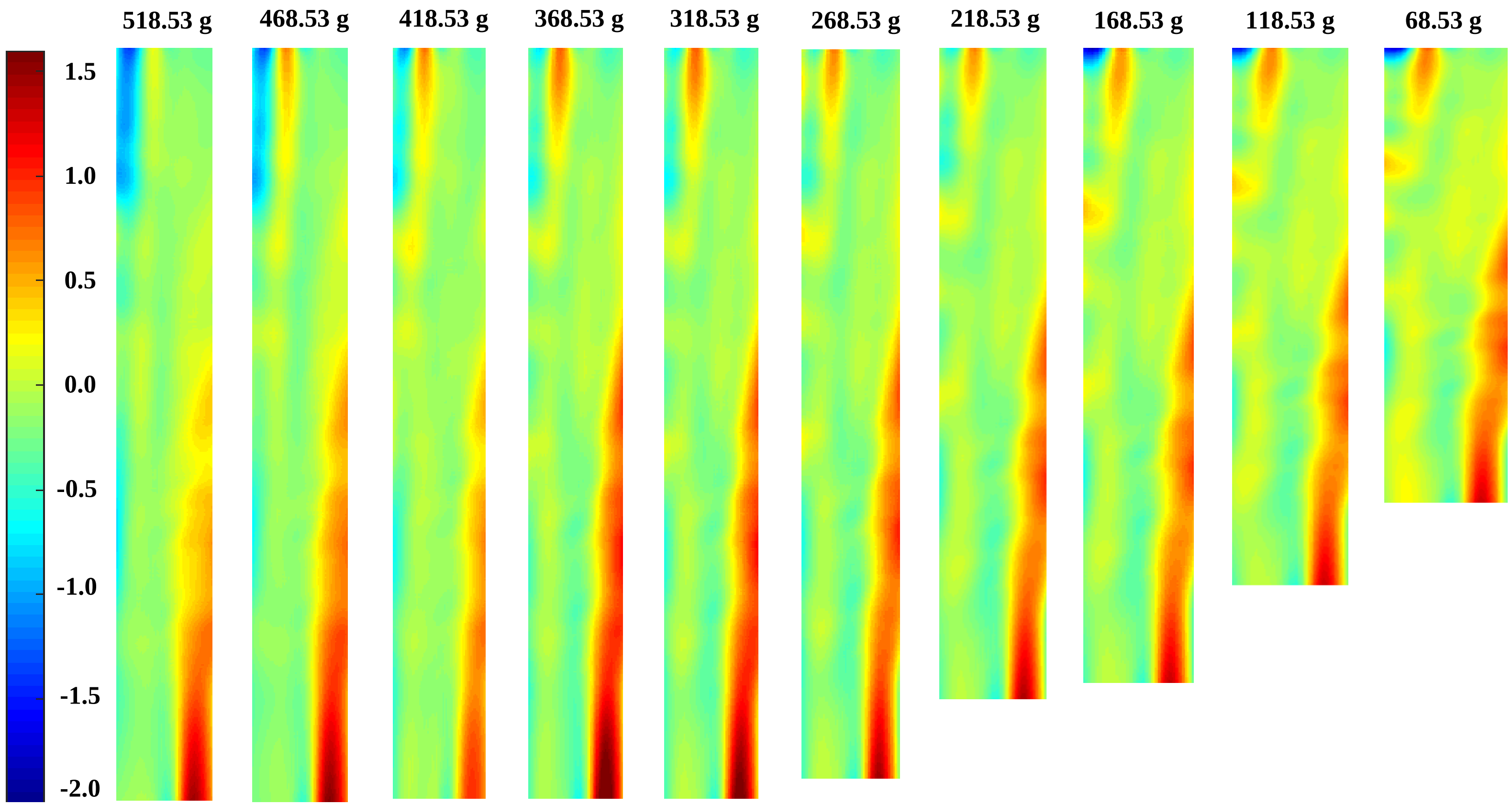


Figure 53: Graphical representation of total shear strain during unloading process – experiment 3

Specimen (60 mm x 60 mm) – 4

Vertical Displacement

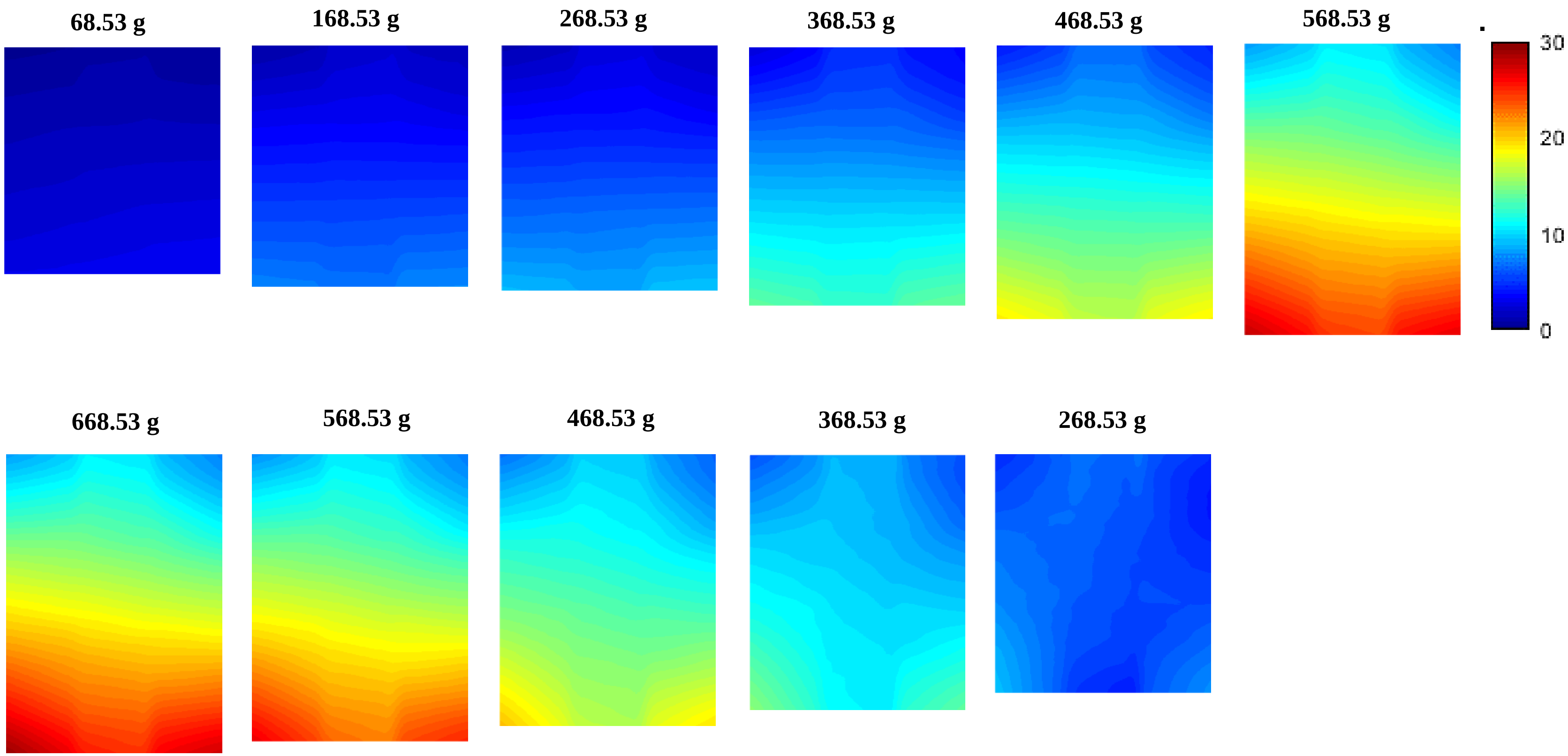


Figure 54: Graphical representation of total vertical displacement – experiment 4

Vertical Strain

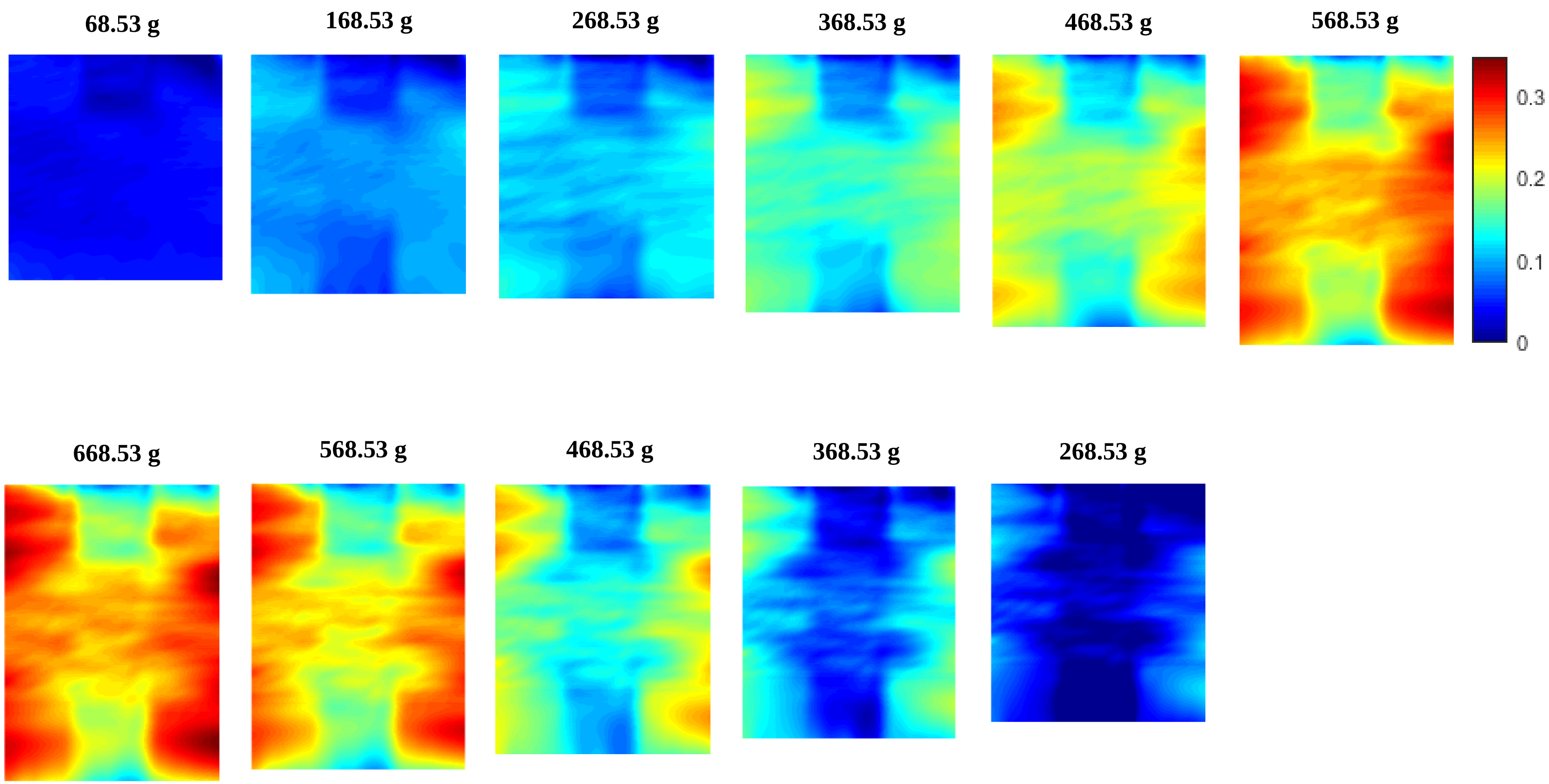


Figure 55: Graphical representation of total vertical strain – experiment 4

Horizontal Displacement

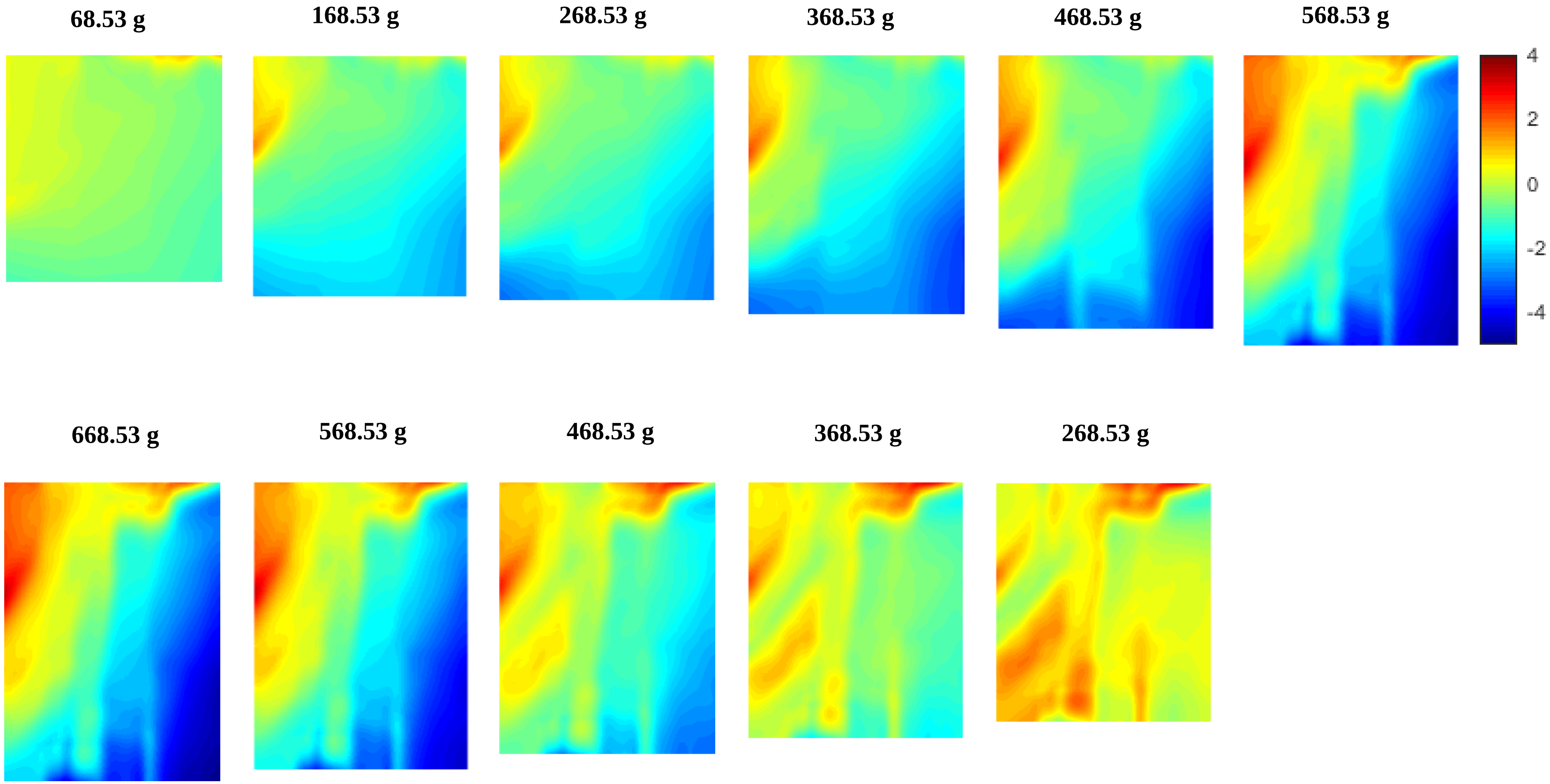


Figure 56: Graphical representation of total horizontal displacement – experiment 4

Horizontal Strain

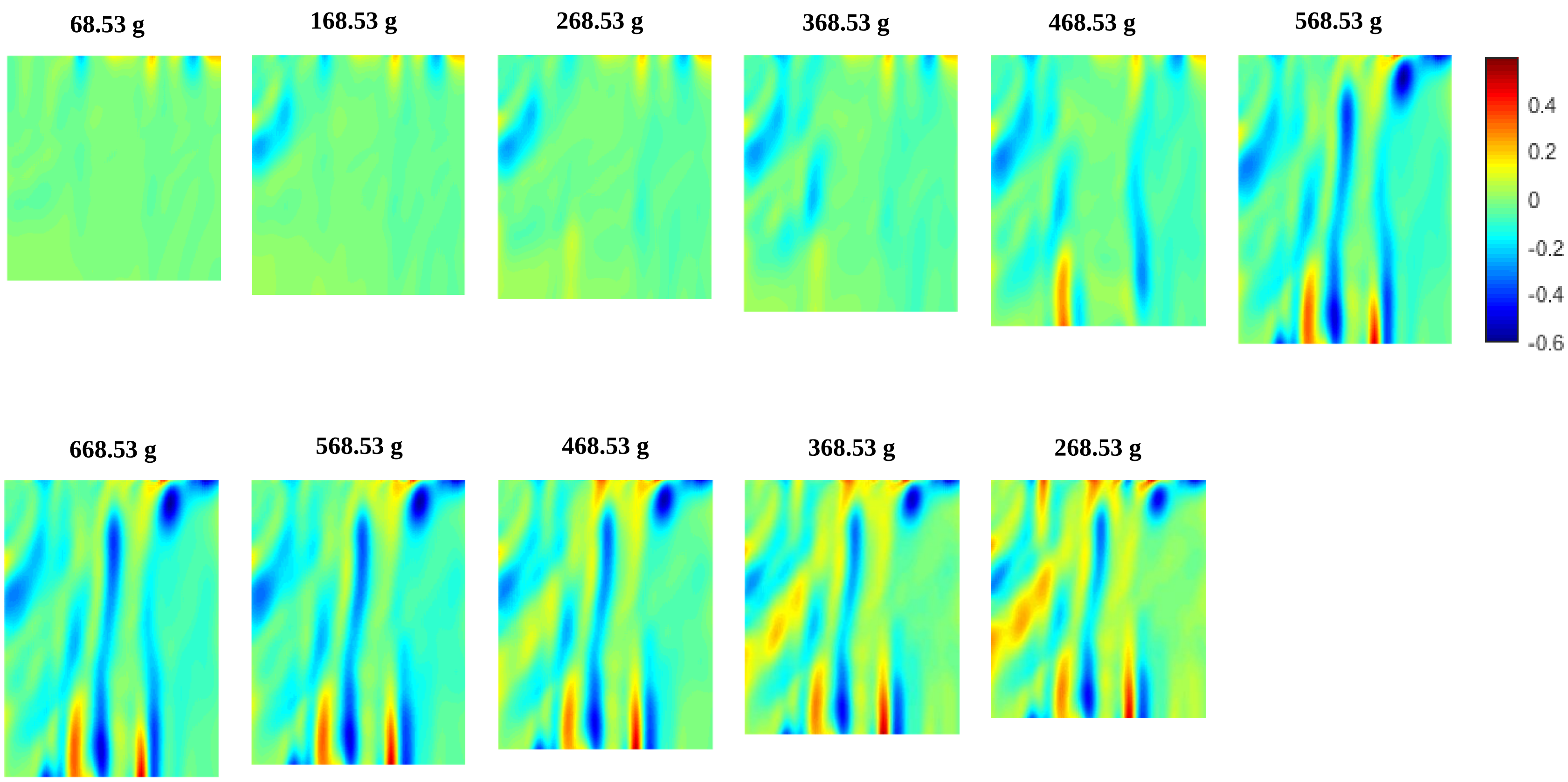


Figure 57: Graphical representation of total horizontal strain – experiment 4

Shear Strain

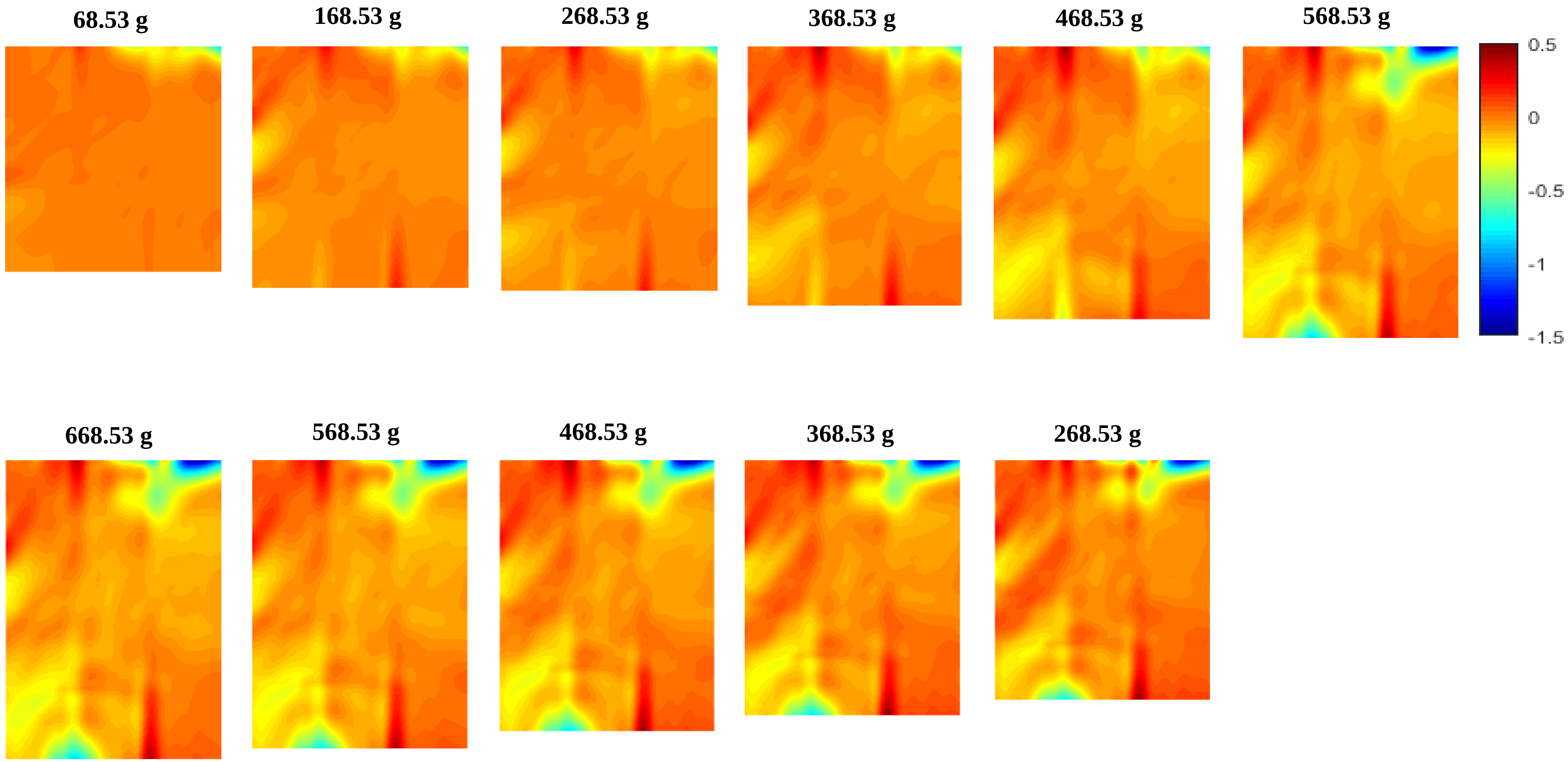


Figure 58: Graphical representation of total shear strain – experiment 4

Specimen (20 mm x 60 mm) - 5

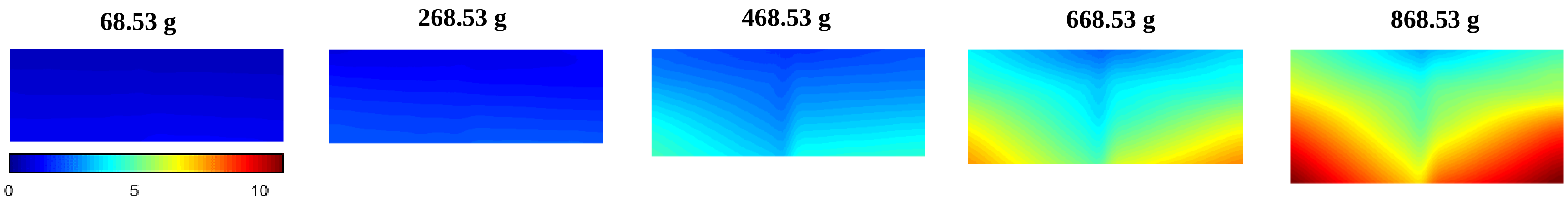


Figure 59: Graphical representation of total vertical displacement – experiment 5

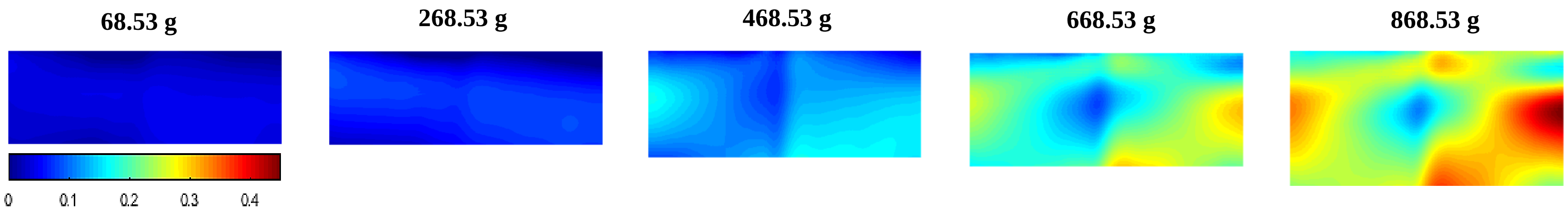


Figure 60: Graphical representation of total vertical strain – experiment 5

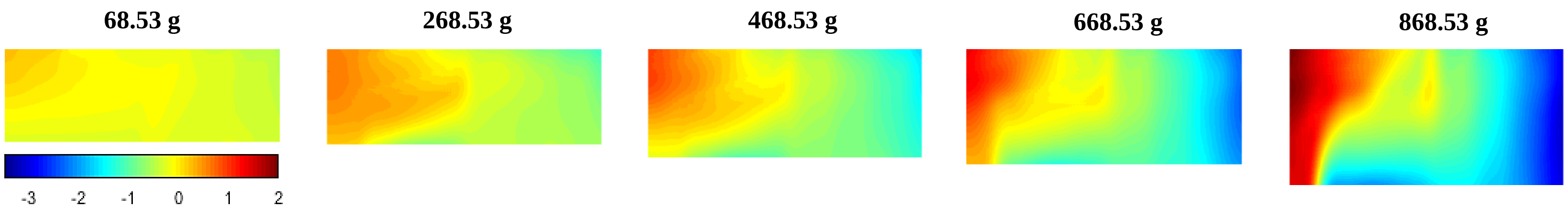


Figure 61: Graphical representation of total horizontal displacement – experiment 5

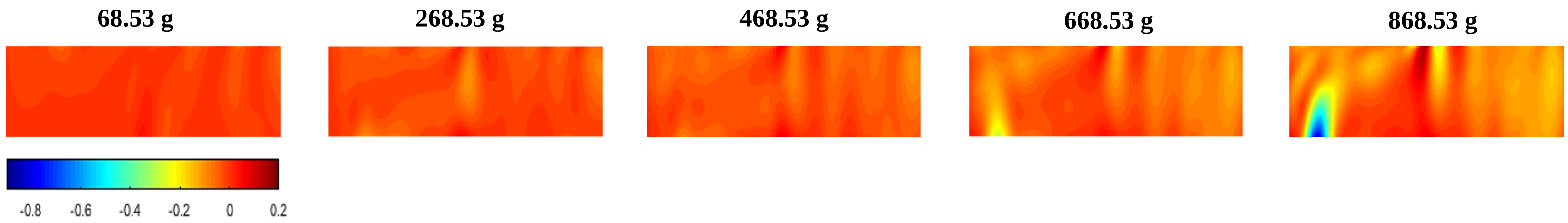


Figure 62: Graphical representation of total horizontal strain – experiment 5

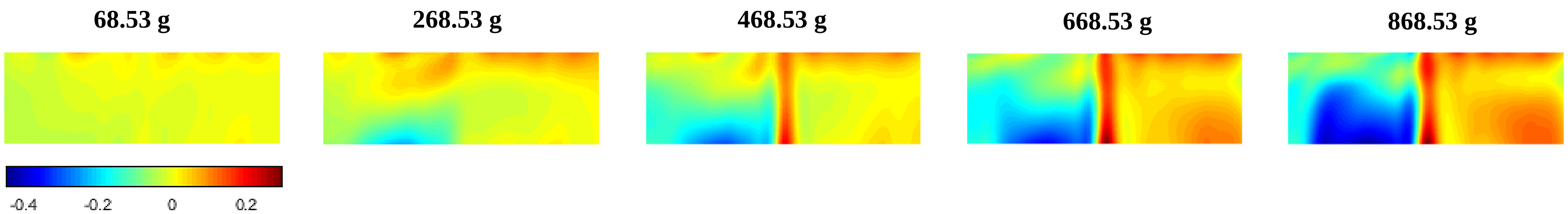
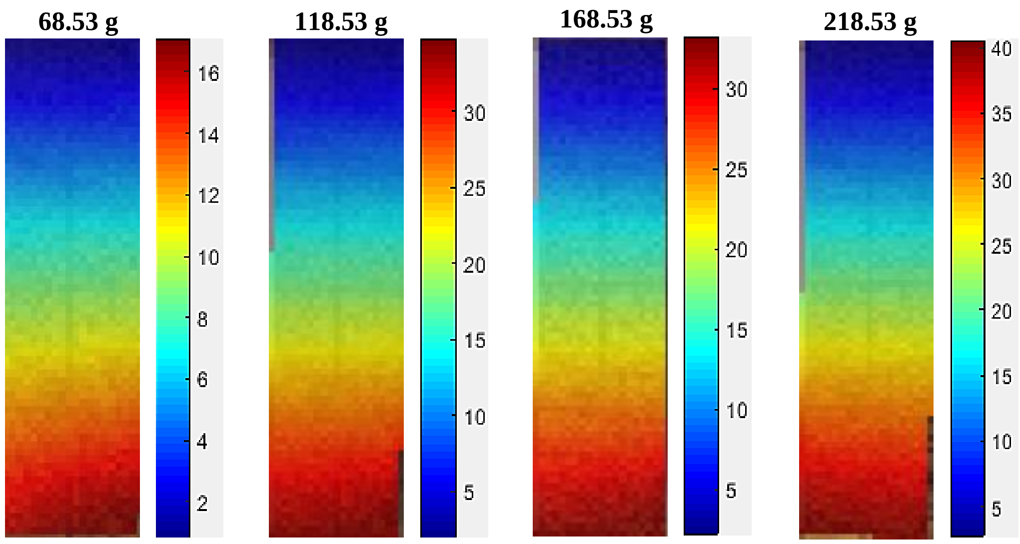
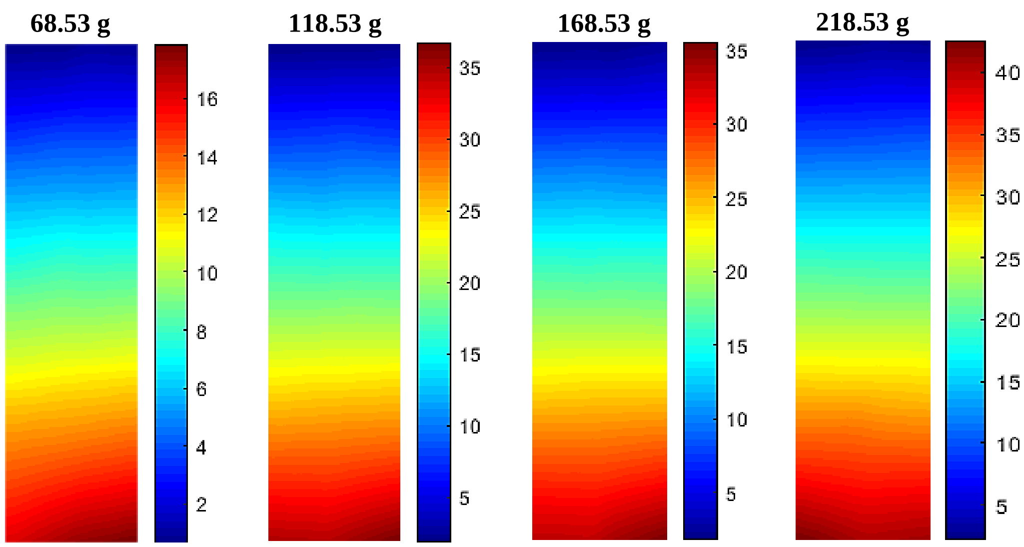


Figure 63: Graphical representation of total shear strain – experiment 5

Comparison of the results



(a)



(b)

Figure 64: Graphical representation of vertical displacement during loading process obtained using
(a) Ncorr and (b) proposed technique – experiment 1

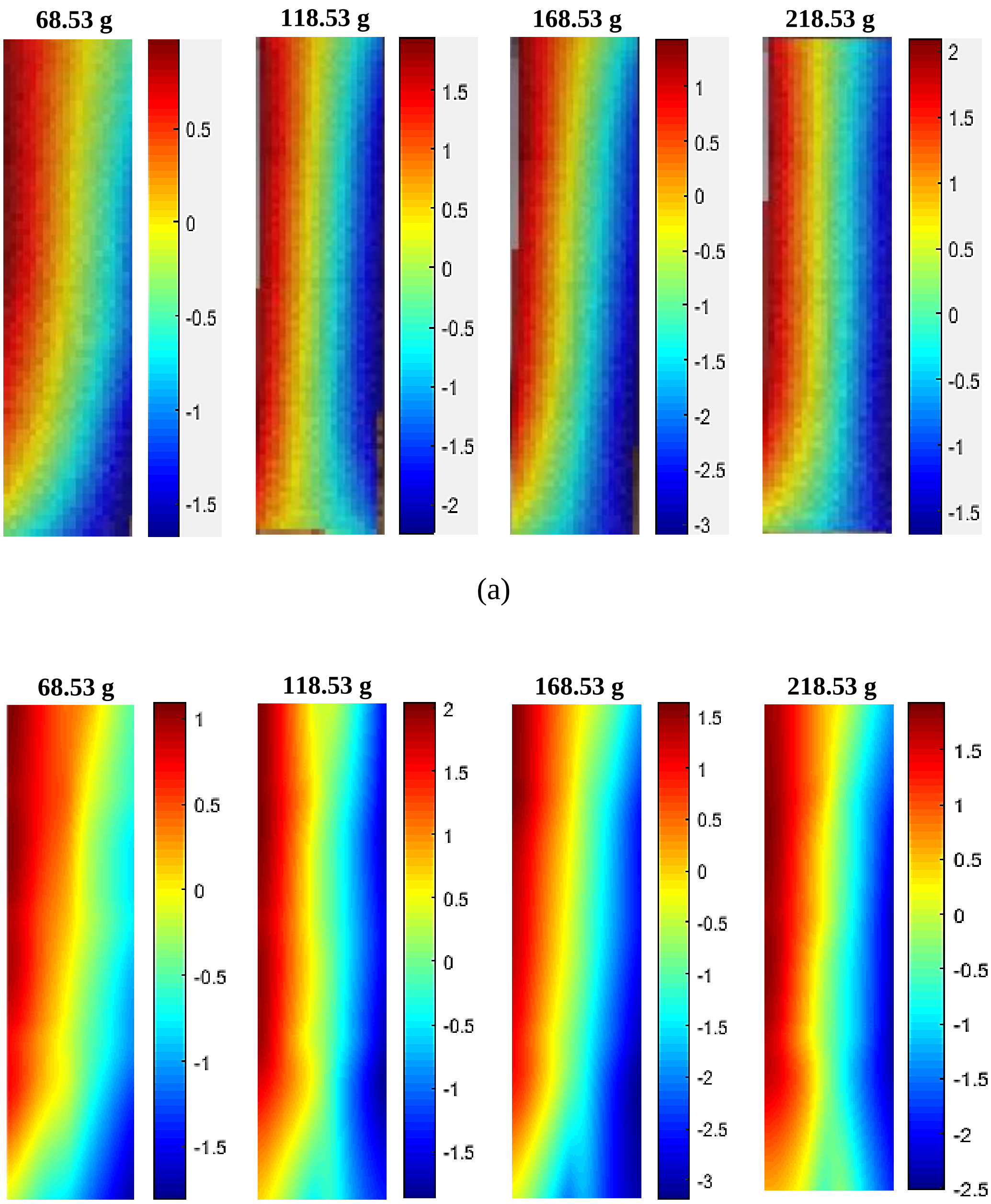
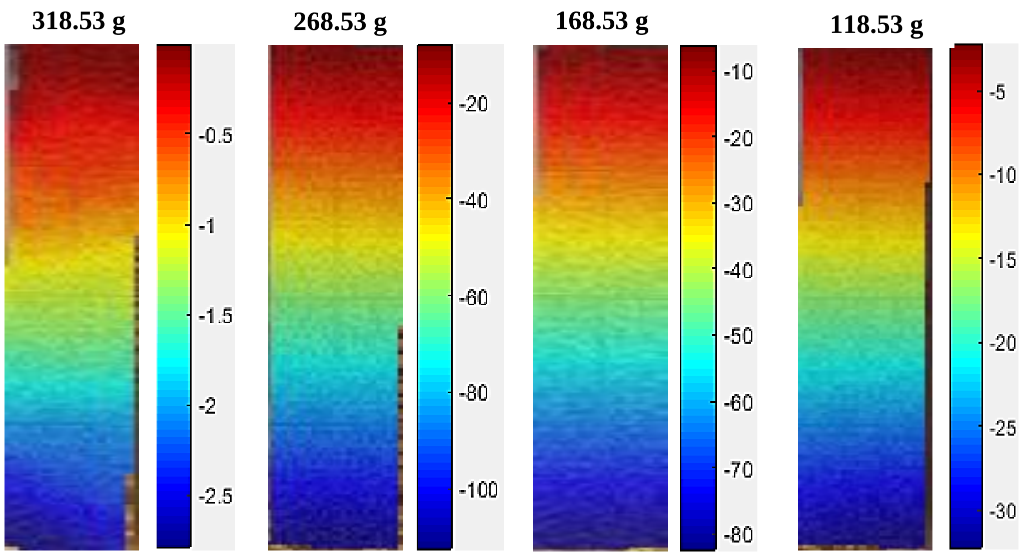
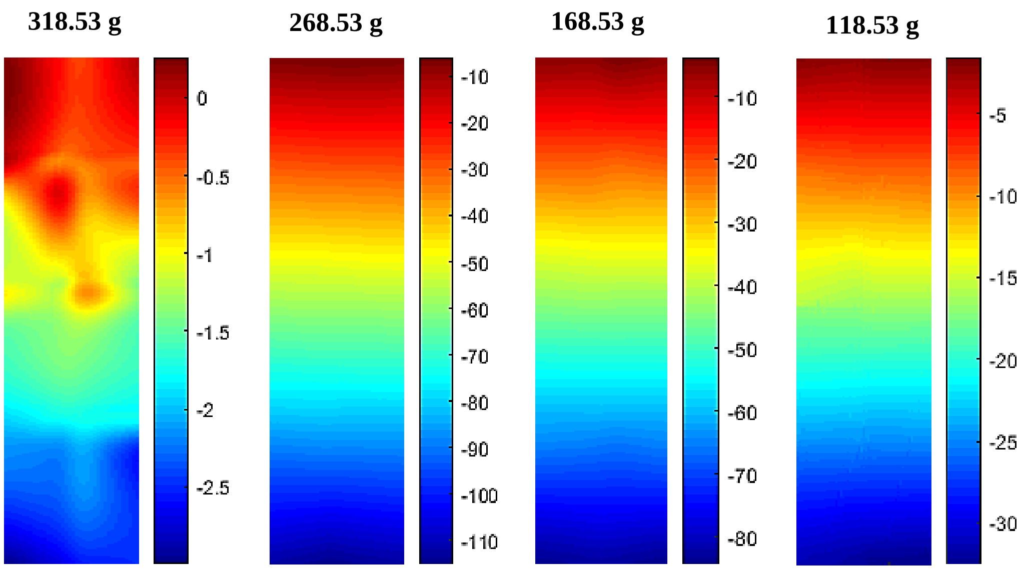


Figure 65: Graphical representation of horizontal displacement during loading process obtained using
(a) Ncorr and (b) proposed technique – experiment 1

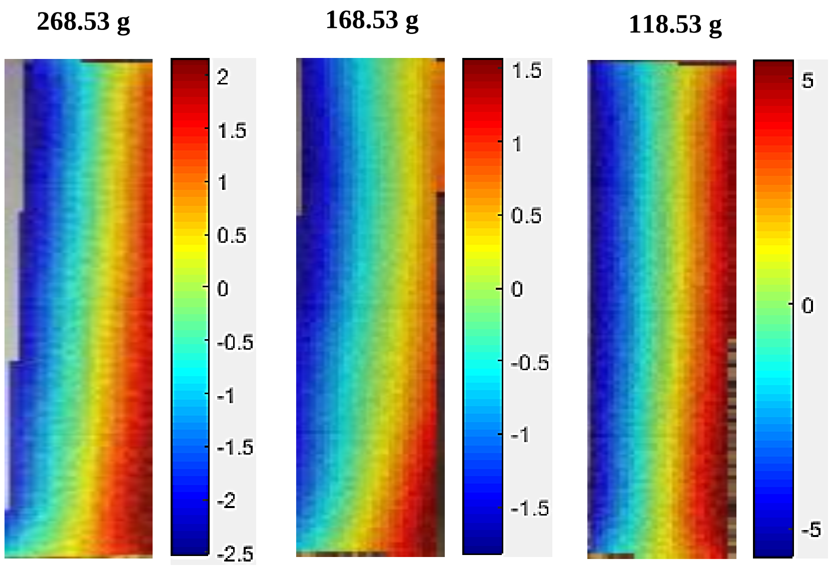


(a)

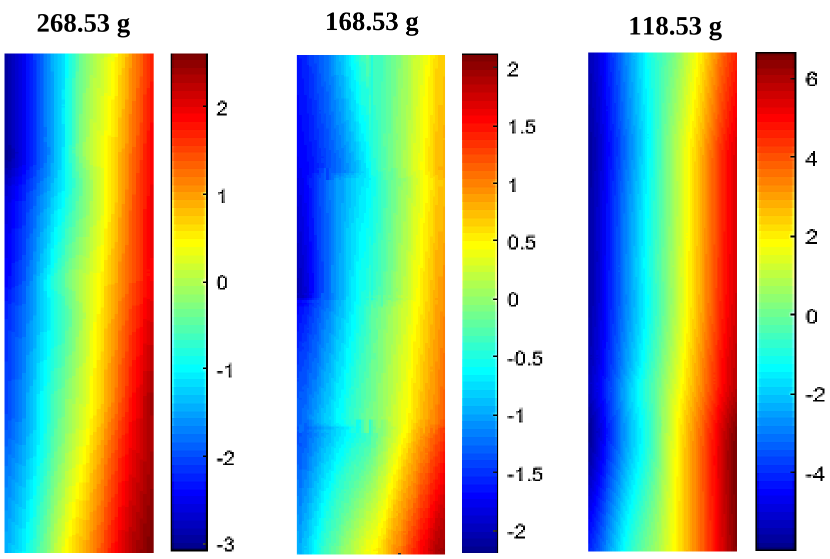


(b)

Figure 66: Graphical representation of vertical displacement during unloading process obtained using
(a) Ncorr and (b) proposed technique – experiment 1

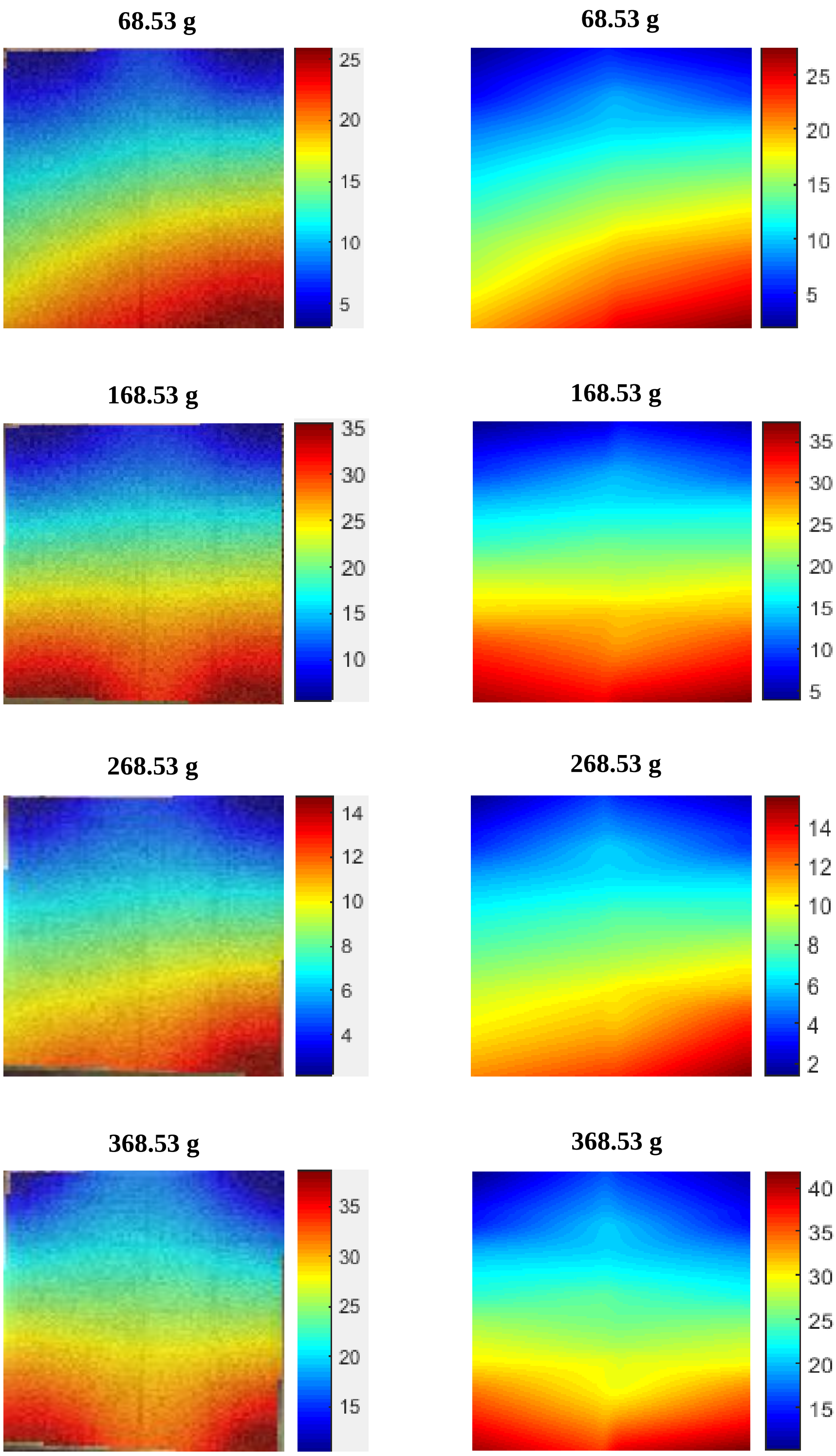


(a)



(b)

Figure 67: Graphical representation of horizontal displacement during unloading process obtained using
(a) Ncorr and (b) proposed technique – experiment 1



(a) (b)

Figure 68: Graphical representation vertical displacement during loading process obtained using
(a) Ncorr and (b) proposed technique – experiment 4

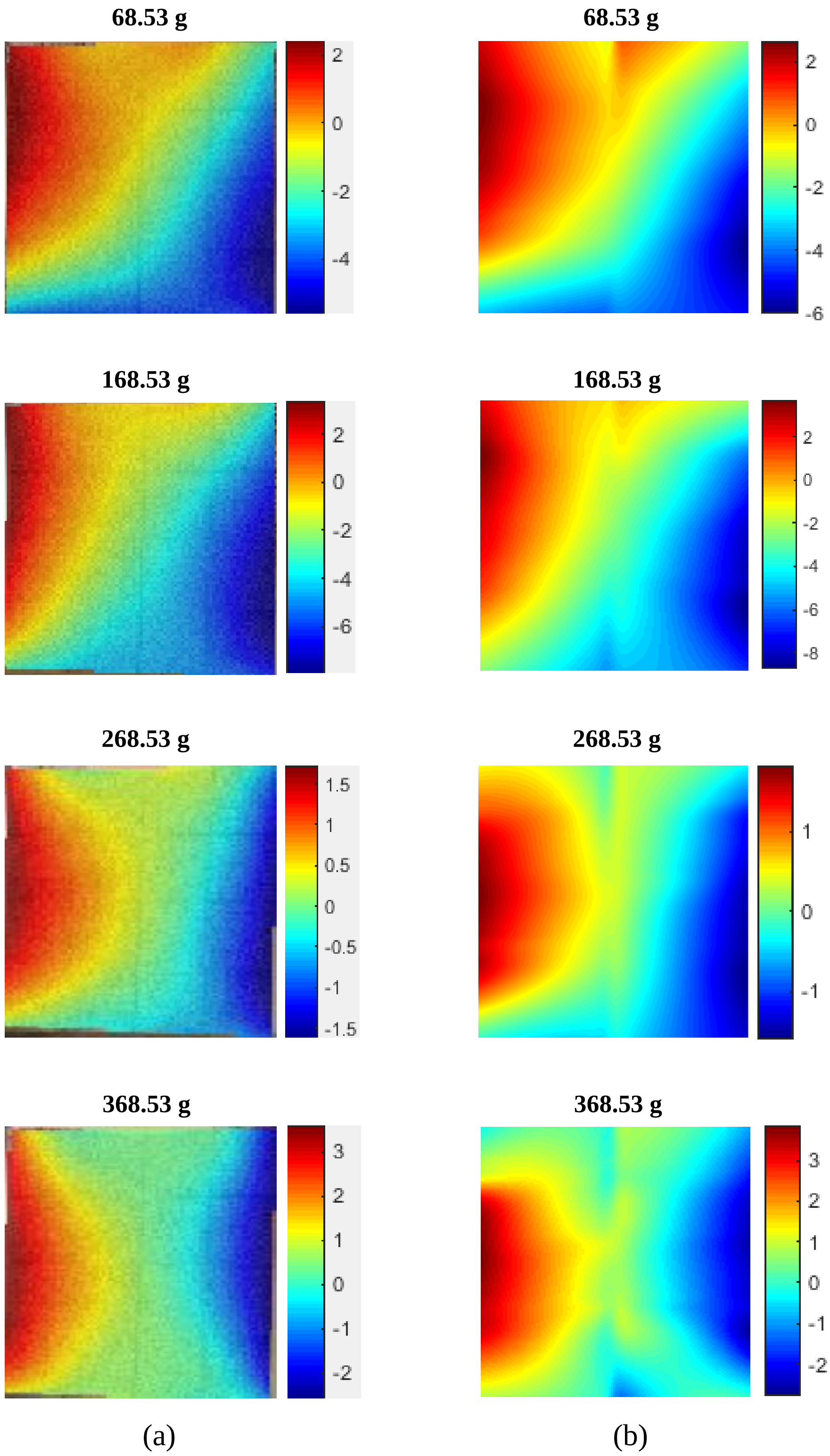


Figure 69: Graphical representation horizontal displacement during loading process obtained using
(a) Ncorr and (b) proposed technique – experiment 4

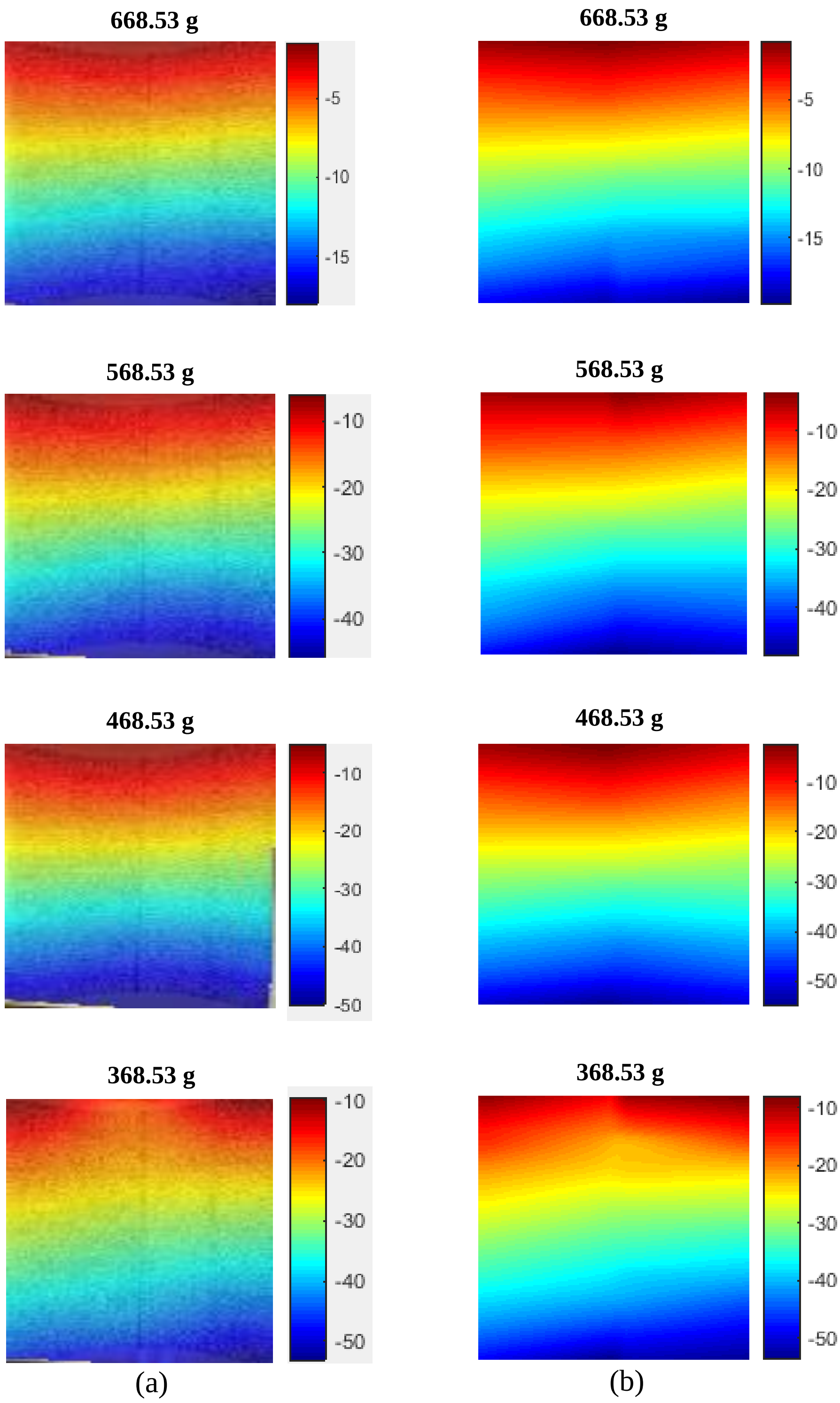


Figure 70: Graphical representation vertical displacement during unloading process obtained using
(a) Ncorr and (b) proposed technique – experiment 4

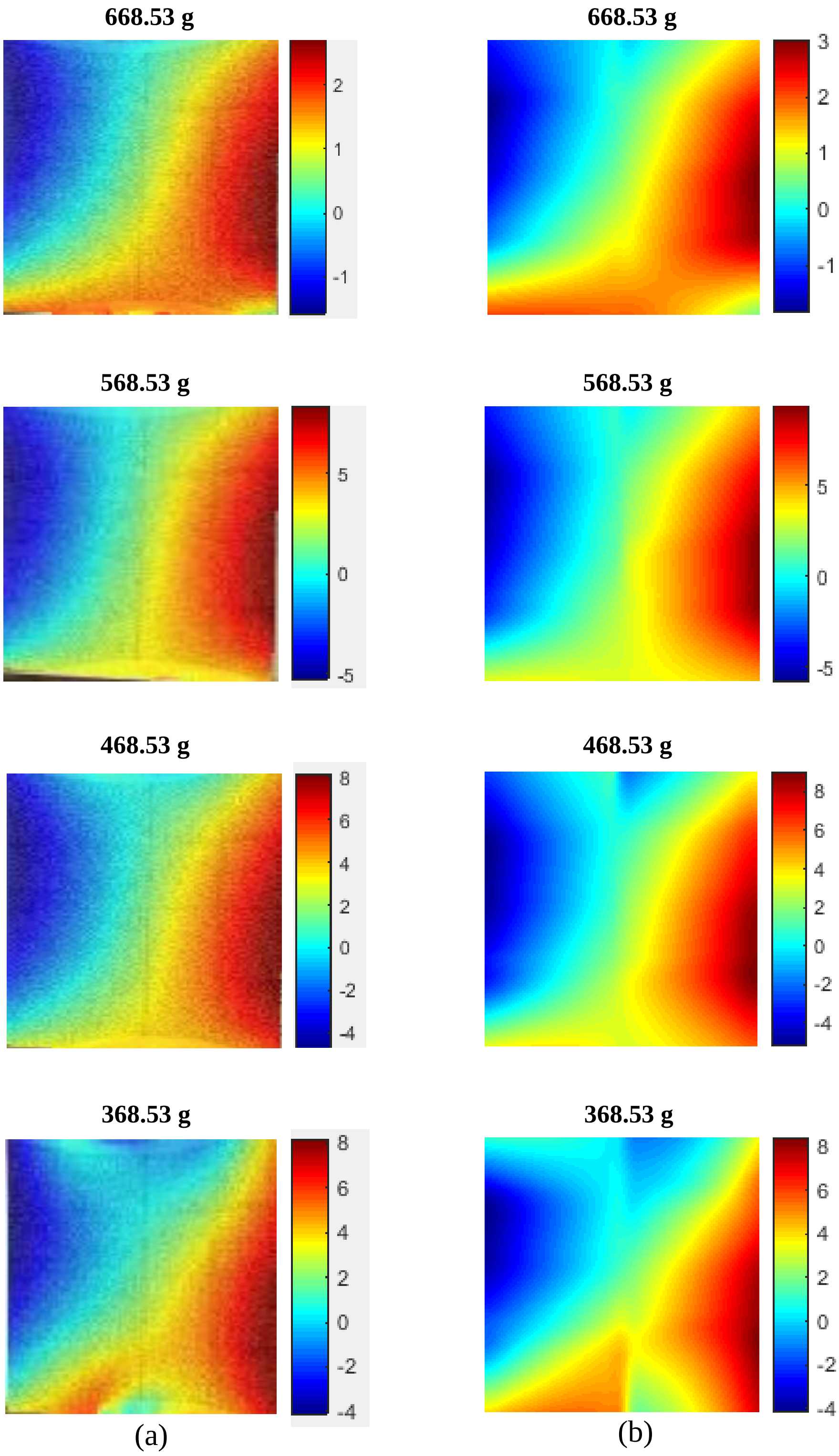


Figure 71: Graphical representation horizontal displacement during unloading process obtained using
(a) Ncorr and (b) proposed technique – experiment 4

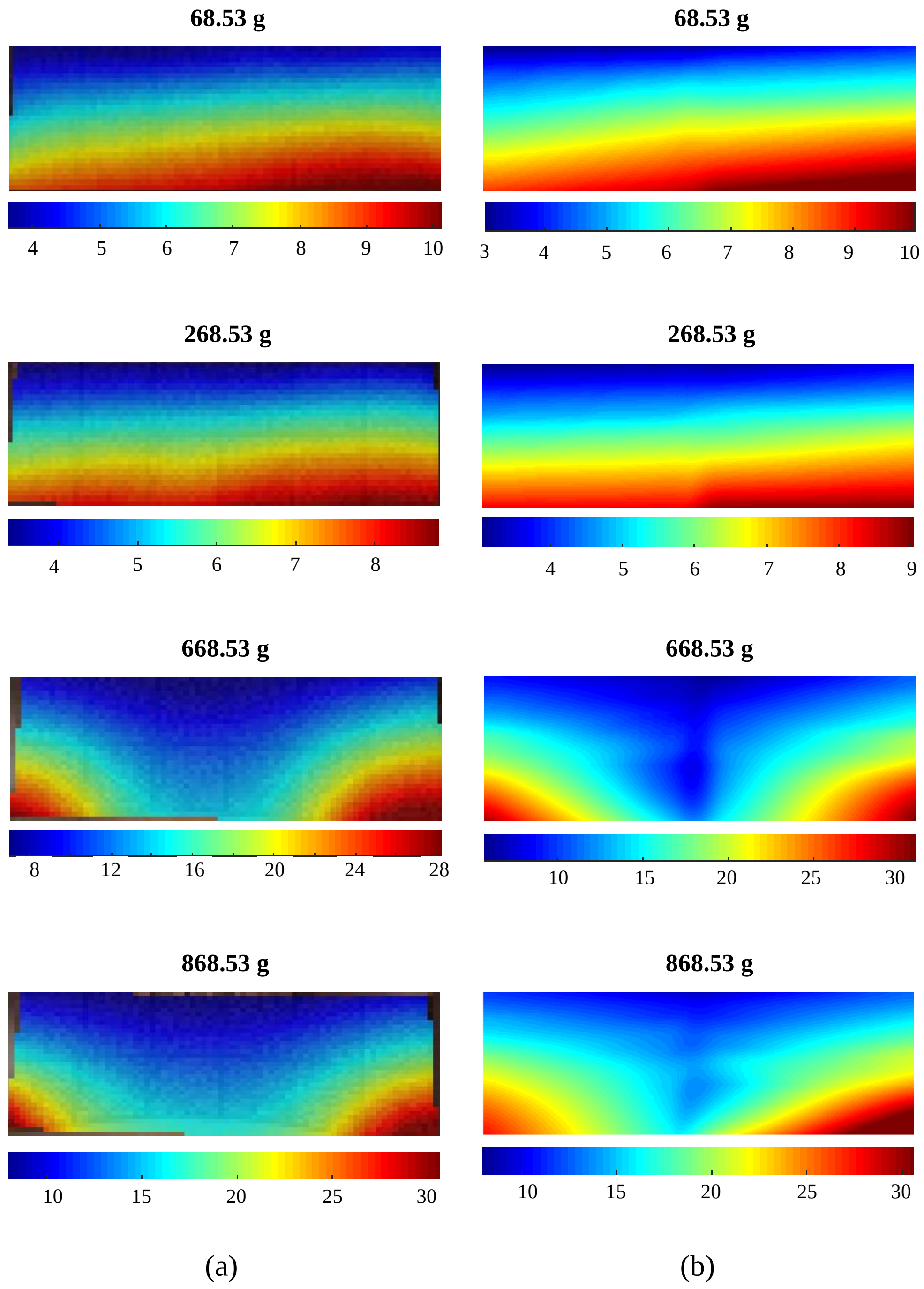


Figure 72: Graphical representation vertical displacement during loading process obtained using
(a) Ncorr and (b) proposed technique – experiment 5

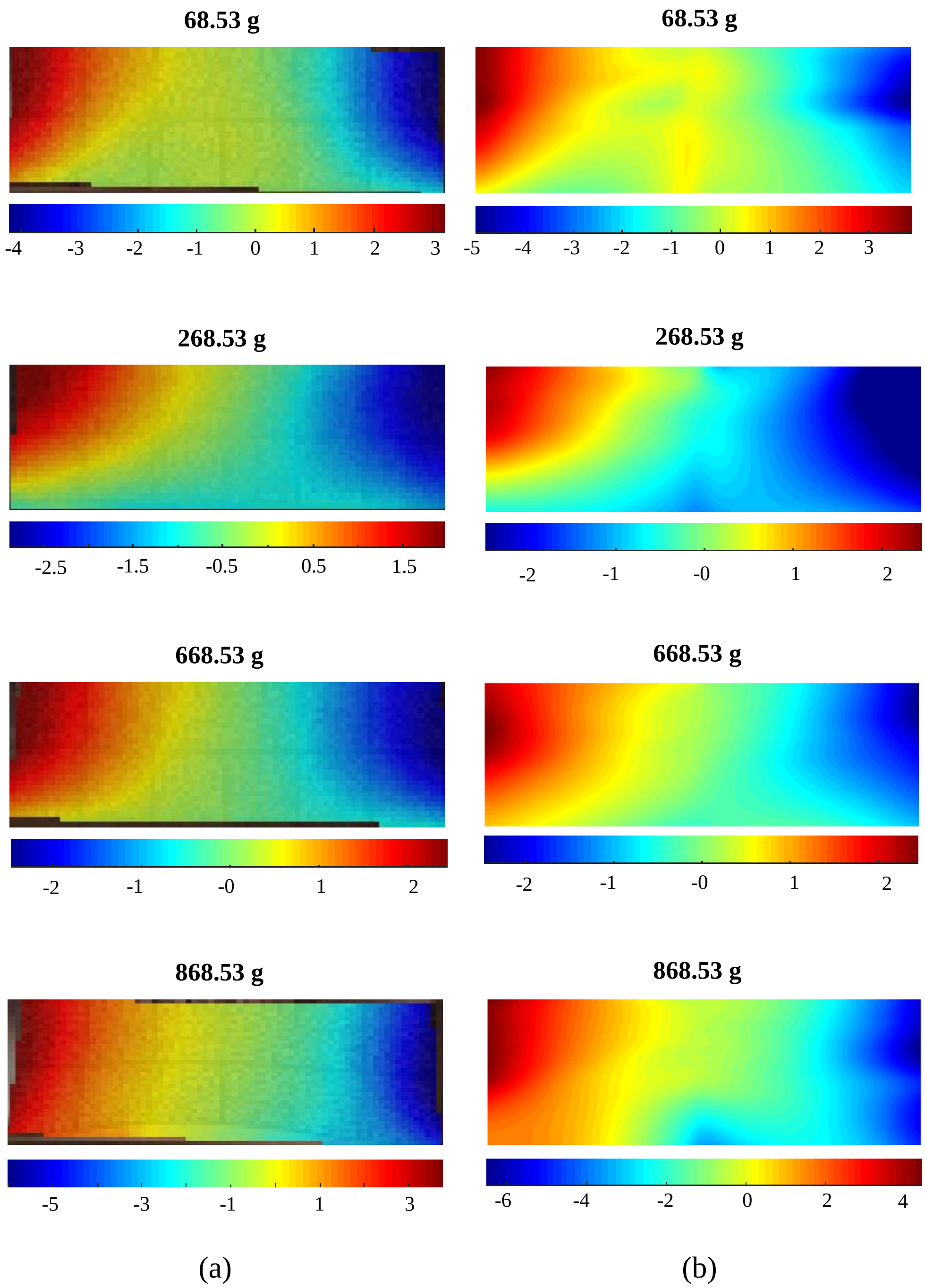


Figure 73: Graphical representation horizontal displacement during loading process obtained using
(a) Ncorr and (b) proposed technique – experiment 5

4.2. Discussion

The preceding chapter presents a tabular comparison of the findings generated by the proposed system and measured deformation values using a vernier calliper that was included in the experimental setup for validation purposes. To eliminate observational mistakes during the measurement, the vernier calliper was attached to the tensile testing apparatus through a moveable head that was fixed to a vertical groove, allowing the measurement initial point to be altered based on the deformation of the latex specimen attached. Furthermore, to improve precision, the lower jaws of the vernier calliper for outside measurement were extended by connecting sharp needles exactly parallel to the jaws' edges. Thus, rather of only capturing the positions of both edges' corner points, the bottom and top edges' locations might be precisely recorded. As a consequence of these improvements, the maximum percentage errors for the difference between measured and estimated deformations for both displacement and strain were maintained below 4%.

The preceding chapter also includes full-field displacement and strain maps in the form of colour maps generated by the proposed system. The pictures of the colour plots are given in proportion to the actual dimension of the specimen, allowing the change in displacement in terms of colour variation and total specimen dimension to be easily understood. The following is the rationale for the colour variation in the plots for the 60 mm × 20 mm x 0.21 mm long narrow specimen. The specimens were subjected to a uniaxial tensile test, as described in Chapter 3.1. The specimens were fastened at the top edge and subjected to vertical downward force from the bottom edge. In each stage of the loading phase, the specimens were supposed to exhibit a gradual increase in vertical displacement from top to bottom and an increasing maximum displacement at the bottom edge at each step in the loading phase as expected in a uniaxial tensile test. The nature of the maps is intended to stay the same during the unloading phase, but the maximum displacement value at the bottom edge at each step should decrease as it approaches the undeformed configuration. The results given in figures 24, 25, 34, 35, 44, and 45 are consistent with the aforementioned rationale.

When it comes to the vertical strain plots of these specimens, it is expected to have a uniform strain value across the specimen at each loading step, with the value of the uniform strain gradually increasing and decreasing at each step of the loading and unloading phases, respectively. Figures 26, 27, 36, 37, 46, and 47 illustrate the same behaviour. Even though the plots during the loading phase in all three experiments with long narrow specimens (figures 26, 36, and 46) contained scattering of tiny patches with non-uniform strain values, the degree of variation is relatively low, as can be seen from the respective colour bars. The vertical strains in the unloading phase display the same behaviour as the loading phase, however the frequency of dispersion of tiny patches with non-uniform strain values is much more than in the loading phase. All three experiments exhibit this behaviour (figures, 27, 37 and 47). In the unloading phase, the amount of deformation recovery is low relative to the loading phase, which might explain this. The stress-strain curve for rubber during the loading and unloading phase is shown in Figure 74. It can be observed that for a given amount of force/stress shift, the displacement/strain changes are less during the unloading phase than during the loading phase. As hyperelastic specimens recover at a slower rate during the unloading phase, the larger dispersion of small patches with non-uniform strain may be owing to semipermanent stresses caused by this impact.

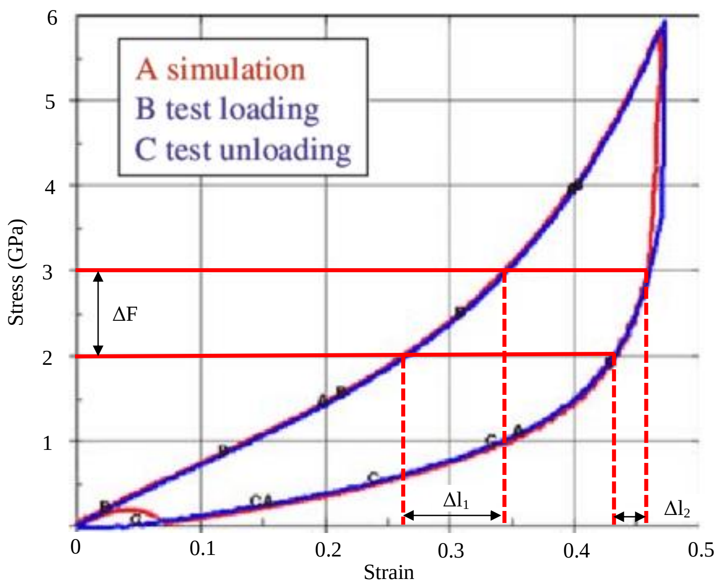


Figure 74: Stress – strain curve of rubber [97]

The horizontal displacement plots generated by the proposed system are depicted in Figures 28, 29, 38, 39, 48, and 49. As a result of the Poisson effect, the hyperlesatic specimen tends to compress in the direction (horizontal) perpendicular to the direction of force application. A vertical axis of symmetry runs through the specimen. Since it tends to shrink from the left and right edges, the effect of compression is lower in the middle region of the specimen than at the edges. For a perfectly vertically loaded scenario, the specimens should have around equal amounts of larger displacement values at the left and right-side edges and nearly less uniform displacement in the middle region. Except for one aspect, the relevant colormaps display this behaviour. The variation in displacement is supposed to be in the horizontal direction. However, the colour variation in the images is tilted in all three experiments. This might be because of any irregularities that occurred when loading and unloading the specimen. In addition, the respective horizontal strain plots exhibit the same trend, except the direction of colour change is inclined rather than horizontal.

The shear strain, as shown in figures 32, 33, 42, 43, 52, and 53, is quite minimal when compared to the strain in the horizontal and vertical directions, as it should be in a uniaxial test. However, strain concentrations may be noticed around the top left and bottom right corners, which may be due to the eccentricities during the load application. The displacement and strain plots generated for different dimensions specimens included in the study show a similar pattern. The degree of influence from the Poisson effect, on the other hand, varies depending on the geometry of the specimen, and so does the variation of strain.

The chapter 4.1 also provides a comparison of displacement strain plots acquired using the proposed system with the Ncorr software for 2D analysis to confirm the system's reliability beyond experimental validation. The specifics of the 2D analysis were discussed in detail in the preceding chapter. Figures 64-73 provides comparison of the results of the Ncorr and suggested techniques in terms of pixels. The comparison indicates that the colour maps are nearly identical for the majority of the loading processes. Furthermore, the range of the colormaps, which was determined automatically by the algorithms based on the maximum value discovered during the computations, was almost the same for both outputs. The resemblance of contours in the colour plots is more visible in the 60 mm x 60 mm square specimen, demonstrating that the approach used in the proposed system for deformation calculation is precise enough for the measurements.

5. CONCLUSIONS AND FUTURE WORKS

5.1. Conclusions

This thesis has presented a novel approach for deformation measurement in 3D space. The system was designed to use mobile phone cameras for image acquisition, which was sufficient for comprehensive analysis of the specimen images while also keeping the system's cost inexpensive when compared to commercial DIC systems. Furthermore, the algorithm constructed with MATLAB tools was capable of precise 3D reconstruction and deformation measurement, as evidenced by the results being compared to experimental validation.

The prime interest of this study has been on confirming the aptness of the proposed technique in capturing the surface deformation information of hyperelastic material. The results obtained using the proposed method were validated against the experimental measurements. And it is proven that the system is capable of producing precise full-field displacement and strain maps with an apparent accuracy greater than 96 %.

Furthermore, the findings were compared to those acquired using a commonly used DIC measuring tool to ensure that the deformation maps produced were accurate. The resemblance of the contour plots in the comparison verified the reliability of the technique. The predicted variation in vertical displacement is accurately represented, as can be seen in the contour plots. In addition, except for a few locations, the vertical stresses were consistent across the RoI, as expected. This might be due to the specimen's uneven thickness, as evidenced by the fact that the strain plots in subsequent loading stages exhibit the same concentrated spots. Furthermore, the alteration of the concentrated spots is progressive in subsequent loading steps, indicating that it has been captured correctly. The horizontal displacement maps show that the central area of the specimen has practically zero or lesser displacement values, while the left and right boundaries of the specimen have positive and negative displacement values that are almost equal in value or close to value, as predicted. However, the contours vary in an inclined rather than a horizontal direction. This might have happened as a result of the eccentric loading. This assumption is reinforced by the fact that the inclined nature of the contour is apparent in all loading steps. As predicted, the horizontal and shear strains produced are insignificant in comparison to the vertical strain.

The resolution of the cameras used in the system, as well as the size and contrast nature of the speckle compared to the specimen's surface, have a significant impact on the accuracy of the results. This enhances the quality of feature detection while also allowing for improved reconstruction. The transformation that happened in the 2D image may eventually be computed for smaller subsets, allowing for more exact deformation capture.

5.2. Future works

Some components of this technique that need to be investigated further were identified during this research and are listed below.

- The text-based interface restricts the general users' access to this approach. The development of a graphical user interface for this technique might make this available to everyone.
- This method can only generate deformation maps for a rectangular region. It may be improved further to enable creating areas with polygons and analysing regions with voids easier.
- It's possible that edge image points that are in depth relative to the RoI will be included erroneously. This might have an impact on the 3D reconstruction's fitted surface profile. Therefore, the RoI boundary inclusion algorithm should be enhanced.
- This technique may be advanced even further by incorporating algorithms that allow deformation to be quantified in the material direction rather than the global direction.
- Strategies for improvement of accuracy of the results further can be implemented.
- This tool may be enhanced even further by integrating intelligent automation process, which analyses data and augments human decision-making in order to improve its own processes and outcomes, maximize resources, and boost efficiency.

REFERENCES

[1] D. Feng and M. Q. Feng, “Experimental validation of cost-effective vision-based structural health monitoring,” *Mechanical Systems and Signal Processing*, vol. 88, pp. 199–211, May 2017, doi: 10.1016/J.YMSSP.2016.11.021.

[2] C. Psaltis and C. Ioannidis, “AN AUTOMATIC TECHNIQUE FOR ACCURATE NON-CONTACT STRUCTURAL DEFORMATION MEASUREMENTS”.

[3] G. A. Stephen, J. M. W. Brownjohn, and C. A. Taylor, “Measurements of static and dynamic displacement from visual monitoring of the Humber Bridge,” *Engineering Structures*, vol. 15, no. 3, pp. 197–208, Jan. 1993, doi: 10.1016/0141-0296(93)90054-8.

[4] L. J. Wu, F. Casciati, and S. Casciati, “Dynamic testing of a laboratory model via vision-based sensing,” *Engineering Structures*, vol. 60, pp. 113–125, Feb. 2014, doi: 10.1016/J.ENGSTRUCT.2013.12.002.

[5] D. Feng and M. Q. Feng, “Model Updating of Railway Bridge Using In Situ Dynamic Displacement Measurement under Trainloads,” *Journal of Bridge Engineering*, vol. 20, no. 12, p. 04015019, Dec. 2015, doi: 10.1061/(ASCE)BE.1943-5592.0000765.

[6] M. Q. Feng, Y. Fukuda, D. Feng, and M. Mizuta, “Nontarget Vision Sensor for Remote Measurement of Bridge Dynamic Response,” *Journal of Bridge Engineering*, vol. 20, no. 12, p. 04015023, Dec. 2015, doi: 10.1061/(ASCE)BE.1943-5592.0000747.

[7] C. A. Santos, C. O. Costa, and J. P. Batista, “Calibration methodology of a vision system for measuring the displacements of long-deck suspension bridges,” *Structural Control and Health Monitoring*, vol. 19, no. 3, pp. 385–404, Apr. 2012, doi: 10.1002/STC.438.

[8] J. H. Lee, H. N. Ho, M. Shinozuka, and J. J. Lee, “An advanced vision-based system for real-time displacement measurement of high-rise buildings,” *Smart*

Materials and Structures, vol. 21, no. 12, p. 125019, Nov. 2012, doi: 10.1088/0964-1726/21/12/125019.

[9] H. Jeon *et al.*, "A paired visual servoing system for 6-DOF displacement measurement of structures," *Smart Materials and Structures*, vol. 20, no. 4, p. 045019, Mar. 2011, doi: 10.1088/0964-1726/20/4/045019.

[10] H. Zhang, S. Hu, and X. Zhang, "SIFT flow for large-displacement object tracking," *Applied Optics, Vol. 53, Issue 27, pp. 6194-6205*, vol. 53, no. 27, pp. 6194–6205, Sep. 2014, doi: 10.1364/AO.53.006194.

[11] P. Kohut *et al.*, "Monitoring of a civil structure's state based on noncontact measurements:," *http://dx.doi.org/10.1177/1475921713487397*, vol. 12, no. 5–6, pp. 411–429, May 2013, doi: 10.1177/1475921713487397.

[12] D. Ribeiro, R. Calçada, J. Ferreira, and T. Martins, "Non-contact measurement of the dynamic displacement of railway bridges using an advanced video-based system," *Engineering Structures*, vol. 75, pp. 164–180, Sep. 2014, doi: 10.1016/J.ENGSTRUCT.2014.04.051.

[13] Y. Fukuda, M. Q. Feng, and M. Shinozuka, "Cost-effective vision-based system for monitoring dynamic response of civil engineering structures," *Structural Control and Health Monitoring*, vol. 17, no. 8, pp. 918–936, Dec. 2010, doi: 10.1002/STC.360.

[14] J. Baqersad, P. Poozesh, C. Niezrecki, and P. Avitabile, "Photogrammetry and optical methods in structural dynamics – A review," *Mechanical Systems and Signal Processing*, vol. 86, pp. 17–34, Mar. 2017, doi: 10.1016/J.YMSSP.2016.02.011.

[15] Y. Z. Song, C. R. Bowen, A. H. Kim, A. Nassehi, J. Padget, and N. Gathercole, "Virtual visual sensors and their application in structural health monitoring:," *http://dx.doi.org/10.1177/1475921714522841*, vol. 13, no. 3, pp. 251–264, Feb. 2014, doi: 10.1177/1475921714522841.

[16] Y. F. Ji and C. C. Chang, "Nontarget Image-Based Technique for Small Cable Vibration Measurement," *Journal of Bridge Engineering*, vol. 13, no. 1, pp. 34–42, Jan. 2008, doi: 10.1061/(ASCE)1084-0702(2008)13:1(34).

[17] J. J. Lee and M. Shinozuka, "A vision-based system for remote sensing of bridge displacement," *NDT & E International*, vol. 39, no. 5, pp. 425–431, Jul. 2006, doi: 10.1016/J.NDTEINT.2005.12.003.

[18] G. Busca, A. Cigada, P. Mazzoleni, and E. Zappa, "Vibration Monitoring of Multiple Bridge Points by Means of a Unique Vision-Based Measuring System," *Experimental Mechanics*, vol. 54, no. 2, pp. 255–271, Feb. 2014, doi: 10.1007/S11340-013-9784-8/FIGURES/16.

[19] S. W. Kim, B. G. Jeon, N. S. Kim, and J. C. Park, "Vision-based monitoring system for evaluating cable tensile forces on a cable-stayed bridge:," *http://dx.doi.org/10.1177/1475921713500513*, vol. 12, no. 5–6, pp. 440–456, Dec. 2013, doi: 10.1177/1475921713500513.

[20] H. Schreier, J. J. Orteu, and M. A. Sutton, *Image correlation for shape, motion and deformation measurements: Basic concepts, theory and applications*. Springer US, 2009. doi: 10.1007/978-0-387-78747-3.

[21] "HISTORY OF PHOTOGRAMMETRY EARLY DEVELOPMENTS 1."

[22] I. Yamaguchi, "Speckle Displacement and Decorrelation in the Diffraction and Image Fields for Small Object Deformation," *Optica Acta: International Journal of Optics*, vol. 28:10, pp. 1359–1376, 1981, [Online]. Available: http://dx.doi.org/10.1080/713820454

[23] W. H. Peters and W. F. Ranson, "Digital imaging techniques in experimental stress analysis," *Optical Engineering*, vol. 21(3), no. 3, pp. 427–431, 2013, doi: https://doi.org/10.1117/12.7972925.

[24] M. A. Sutton, J. Walters, W. H. Peters, W. F. Ranson, and S. R. Mcneil1, "Determination of displacements using an improved digital correlation method."

[25] F. Devernay and O. Faugeras, "Computing Differential Properties of 3-D Shapes from Stereoscopic Images without 3-D Models Computing Differential Properties of 3-D Shapes from Stereo-scopic Images without 3-D Models." [Online]. Available: https://hal.inria.fr/inria-00074369

[26] Z. Wang, M. Vo, H. Kieu, and T. Pan, "Automated fast initial guess in digital image correlation," *Strain*, vol. 50, no. 1, pp. 28–36, Feb. 2014, doi: 10.1111/str.12063.

[27] G. Quino *et al.*, "Speckle patterns for DIC in challenging scenarios: Rapid application and impact endurance," *Measurement Science and Technology*, vol. 32, no. 1, Jan. 2021, doi: 10.1088/1361-6501/abaae8.

[28] M. A. Sutton, J.-J. Orteu, and H. W. Achreier, *Image Correlation for Shape, Motion and Deformation Measurements*. New York, USA: springer, 2009.

[29] 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.

[30] G. Lionello and L. Cristofolini, "A practical approach to optimizing the preparation of speckle patterns for digital-image correlation," *Measurement Science and Technology*, vol. 25, no. 10, Oct. 2014, doi: 10.1088/0957-0233/25/10/107001.

[31] Y. Barranger, P. Doumalin, J. C. Dupré, and A. Germaneau, "Digital image correlation accuracy: Influence of kind of speckle and recording setup," in *EPJ Web of Conferences*, Jun. 2010, vol. 6. doi: 10.1051/epjconf/20100631002.

[32] Z. Chen, C. Quan, F. Zhu, and X. He, "A method to transfer speckle patterns for digital image correlation," *Measurement Science and Technology*, vol. 26, no. 9, Sep. 2015, doi: 10.1088/0957-0233/26/9/095201.

[33] A. P. Ilyin, A. v. Mostovshchikov, F. A. Gubarev, I. D. Sytnik, L. Li, and F. A. Gubarev, "Laser speckle correlation technique application for study of aluminum nanopowder combustion," *Applied Optics, Vol. 59, Issue 5, pp. 1263-1270*, vol. 59, no. 5, pp. 1263–1270, Feb. 2020, doi: 10.1364/AO.380295.

[34] J. N. Petzing and J. R. Tyrer, "Recent developments and applications in electronic speckle pattern interferometry:," *http://dx.doi.org/10.1243/0309324981512887*, vol. 33, no. 2, pp. 153–169, Dec. 2005, doi: 10.1243/0309324981512887.

[35] J. D. Briers, "Speckle techniques," *Optical measurement methods in biomechanics*, pp. 76–98, Aug. 1997, doi: 10.1007/978-0-585-35228-2_5.

[36] Z. Chen, C. Quan, F. Zhu, and X. He, "A method to transfer speckle patterns for digital image correlation," *Measurement Science and Technology*, vol. 26, no. 9, Sep. 2015, doi: 10.1088/0957-0233/26/9/095201.

[37] X. Zhong, X. Wang, N. Cooley, P. Farrell, and B. Moran, "Application of laser speckle pattern analysis for plant sensing," in *PIAGENG 2013: Image Processing and Photonics for Agricultural Engineering*, Mar. 2013, vol. 8761, p. 876102. doi: 10.1117/12.2019463.

[38] S. Cooreman and H. Sol, "Study and generation of optimal speckle patterns for DIC," 2007. [Online]. Available: https://www.researchgate.net/publication/267678766

[39] S. Yaofeng and J. H. L. Pang, "Study of optimal subset size in digital image correlation of speckle pattern images," *Optics and Lasers in Engineering*, vol. 45, no. 9, pp. 967–974, Sep. 2007, doi: 10.1016/J.OPTLASENG.2007.01.012.

[40] Z. Li, P. Zhong, X. Tang, J. Ling, J. Chen, and G. He, "Effect of the speckle size on the quality of speckle pattern in DSPI system," *IEEE Access*, vol. 7, pp. 115010–115022, 2019, doi: 10.1109/ACCESS.2019.2935754.

[41] K. Yamahai, M. Zhou, T. Yoshimura, and Z. Liyan, "Optimum determination of speckle size to be used in electronic speckle pattern interferometry," *Applied Optics, Vol. 34, Issue 1, pp. 87-91*, vol. 34, no. 1, pp. 87–91, Jan. 1995, doi: 10.1364/AO.34.000087.

[42] G. Lionello and L. Cristofolini, "A practical approach to optimizing the preparation of speckle patterns for digital-image correlation," *Measurement*

Science and Technology, vol. 25, no. 10, p. 107001, Sep. 2014, doi: 10.1088/0957-0233/25/10/107001.

[43] J. L. W. Carter, M. D. Uchic, and M. J. Mills, “Impact of speckle pattern parameters on DIC strain resolution calculated from in-situ SEM experiments.”

[44] D. Feng and M. Q. Feng, “Vision-based multipoint displacement measurement for structural health monitoring,” *Structural Control and Health Monitoring*, vol. 23, no. 5, pp. 876–890, May 2016, doi: 10.1002/STC.1819.

[45] X. W. Ye, Y. Q. Ni, T. T. Wai, K. Y. Wong, X. M. Zhang, and F. Xu, “A vision-based system for dynamic displacement measurement of long-span bridges: algorithm and verification,” *Smart Structures and Systems*, vol. 12, no. 3_4, pp. 363–379, Sep. 2013, doi: 10.12989/SSS.2013.12.3_4.363.

[46] S. Silva, “A vision system for vibration monitoring of civil engineering structures Related papers,” 2010, doi: 10.1111/j.1747-1567.2010.00653.x.

[47] Y. Z. Song, C. R. Bowen, A. H. Kim, A. Nassehi, J. Padget, and N. Gathercole, “Virtual visual sensors and their application in structural health monitoring:,” *http://dx.doi.org/10.1177/1475921714522841*, vol. 13, no. 3, pp. 251–264, Feb. 2014, doi: 10.1177/1475921714522841.

[48] D. Ribeiro, R. Calçada, J. Ferreira, and T. Martins, “Non-contact measurement of the dynamic displacement of railway bridges using an advanced video-based system,” *Engineering Structures*, vol. 75, pp. 164–180, Sep. 2014, doi: 10.1016/J.ENGSTRUCT.2014.04.051.

[49] “Template Matching - an overview | ScienceDirect Topics.” https://www.sciencedirect.com/topics/engineering/template-matching (accessed Aug. 04, 2021).

[50] C. Xiu and X. Pan, “Tracking algorithm based on the improved template matching,” *Proceedings of the 29th Chinese Control and Decision Conference, CCDC 2017*, pp. 483–486, Jul. 2017, doi: 10.1109/CCDC.2017.7978142.

[51] Y. M. Fouda, “A Robust Template Matching Algorithm Based on Reducing Dimensions,” *Journal of Signal and Information Processing*, vol. 06, no. 02, pp. 109–122, 2015, doi: 10.4236/jsip.2015.62011.

[52] A. Sibiryakov, “Fast and high-performance template matching method,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1417–1424. doi: 10.1109/CVPR.2011.5995391.

[53] J. Jiao, X. Wang, Z. Deng, J. Cao, and W. Tang, “A fast template matching algorithm based on principal orientation difference,” *International Journal of Advanced Robotic Systems*, vol. 15, no. 3, May 2018, doi: 10.1177/1729881418778223.

[54] D. T. Nguyen, W. Li, and P. Ogunbona, “An improved template matching method for object detection,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010, vol. 5996 LNCS, no. PART 3, pp. 193–202. doi: 10.1007/978-3-642-12297-2_19.

[55] “MATLAB Documentation R2019b,” 2019.

[56] “PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing.” https://gfx.cs.princeton.edu/pubs/Barnes_2009_PAR/ (accessed Aug. 04, 2021).

[57] C. Harris and M. Stephens, “A Combined Corner and Edge Detector,” pp. 23.1-23.6, Apr. 2013, doi: 10.5244/C.2.23.

[58] E. Rosten and T. Drummond, “Machine Learning for High-Speed Corner Detection,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3951 LNCS, pp. 430–443, 2006, doi: 10.1007/11744023_34.

[59] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *International Journal of Computer Vision 2004 60:2*, vol. 60, no. 2, pp. 91–110, Nov. 2004, doi: 10.1023/B:VISI.0000029664.99615.94.

[60] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool, “Speeded-Up Robust Features (SURF),” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, Jun. 2008, doi: 10.1016/J.CVIU.2007.09.014.

[61] A. Neubeck and L. van Gool, “Efficient non-maximum suppression,” *Proceedings - International Conference on Pattern Recognition*, vol. 3, pp. 850–855, 2006, doi: 10.1109/ICPR.2006.479.

[62] S. Leutenegger, M. Chli, and R. Y. Siegwart, “BRISK: Binary Robust invariant scalable keypoints,” *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2548–2555, 2011, doi: 10.1109/ICCV.2011.6126542.

[63] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, “KAZE Features,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7577 LNCS, no. PART 6, pp. 214–227, 2012, doi: 10.1007/978-3-642-33783-3_16.

[64] X. Bai, X. Ning, and L. Wang, “Analysis and comparison of feature detection and matching algorithms for rovers vision navigation,” *2012 the 8th IEEE International Symposium on Instrumentation and Control Technology, ISICT 2012 - Proceedings*, pp. 66–71, 2012, doi: 10.1109/ISICT.2012.6291628.

[65] T. Mouats, N. Aouf, D. Nam, and S. Vidas, “Performance Evaluation of Feature Detectors and Descriptors Beyond the Visible,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 92, no. 1, pp. 33–63, Sep. 2018, doi: 10.1007/s10846-017-0762-8.

[66] D. Lecompte, J. Vantomme, and H. Sol, “Crack detection in a concrete beam using two different camera techniques,” *Structural Health Monitoring*, vol. 5, no. 1, pp. 59–68, Mar. 2006, doi: 10.1177/1475921706057982.

[67] B. Y. Lee, Y. Y. Kim, S. T. Yi, and J. K. Kim, “Automated image processing technique for detecting and analysing concrete surface cracks,” *Structure and Infrastructure Engineering*, vol. 9, no. 6, pp. 567–577, 2013, doi: 10.1080/15732479.2011.593891.

[68] J. F. Destrebecq, E. Toussaint, and E. Ferrier, “Analysis of Cracks and Deformations in a Full Scale Reinforced Concrete Beam Using a Digital Image Correlation Technique,” *Experimental Mechanics*, vol. 51, no. 6, pp. 879–890, Jul. 2011, doi: 10.1007/s11340-010-9384-9.

[69] M. Küntz, M. Jolin, J. Bastien, F. Perez, and F. Hild, “Digital image correlation analysis of crack behavior in a reinforced concrete beam during a load test,” *Canadian Journal of Civil Engineering*, vol. 33, no. 11, pp. 1418–1425, Nov. 2006, doi: 10.1139/L06-106.

[70] S. L. Kumar, H. B. Aravind, and N. Hossiney, “Digital image correlation (DIC) for measuring strain in brick masonry specimen using Ncorr open source 2D MATLAB program,” *Results in Engineering*, vol. 4, Dec. 2019, doi: 10.1016/j.rineng.2019.100061.

[71] 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.

[72] S. Gokhan Ozkaya, M. Baygin, M. Alperen Ozdemir, and I. Kazaz, “Image Processing Based Analysis of the Compressive Strength for the Stones Used In Historical Masonry Structures,” *International Journal of Computer Science and Software Engineering (IJCSSE)*, vol. 6, no. 10, 2017, [Online]. Available: www.IJCSSE.org

[73] T. Ramos *et al.*, “2D and 3D Digital Image Correlation in Civil Engineering - Measurements in a Masonry Wall,” in *Procedia Engineering*, 2015, vol. 114, pp. 215–222. doi: 10.1016/j.proeng.2015.08.061.

[74] A. W. Yu and C. M. Zhu, “Application of DIC system in the tensile test of medium and heavy plate,” *Advanced Materials Research*, vol. 819, pp. 212–215, 2013, doi: 10.4028/www.scientific.net/AMR.819.212.

[75] A. D. Evans, A. Das, and N. Croft, “Implementing Digital Image Correlation for Determining the Tensile Characteristics of Post-Processed Thin Sheet Metal”, doi: 10.1051/matecconf/201.

[76] Y. H. Wang *et al.*, “Whole field sheet-metal tensile test using digital image correlation,” *Experimental Techniques*, vol. 34, no. 2, pp. 54–59, Mar. 2010, doi: 10.1111/j.1747-1567.2009.00483.x.

[77] C. O. Paulsen, E. Fagerholt, T. Børvik, and I. Westermann, “Comparing in situ DIC results from an etched surface with a gold speckled surface,” *Metals*, vol. 9, no. 8, 2019, doi: 10.3390/met9080820.

[78] Y. Chen, C. Ji, C. Zhang, and S. Sun, “The application of DIC technique to evaluate residual tensile strength of aluminum alloy plates with multi-site damage of collinear and non-collinear cracks,” *Metals*, vol. 9, no. 2, Feb. 2019, doi: 10.3390/met9020118.

[79] P. Konopík, J. Džugan, and M. Rund, “DYNAMIC TENSILE AND MICRO-TENSILE TESTING USING DIC METHOD.”

[80] L. Farge, S. André, and J. Boisse, “Use of Digital Image Correlation to study the effect of temperature on the development of plastic instabilities in a semi-crystalline polymer.”

[81] X. Poulain, L. W. Kohlman, W. Binienda, G. D. Roberts, R. K. Goldberg, and A. A. Benzerga, “Determination of the intrinsic behavior of polymers using digital image correlation combined with video-monitored testing,” *International Journal of Solids and Structures*, vol. 50, no. 11–12, pp. 1869–1878, Jun. 2013, doi: 10.1016/j.ijsolstr.2013.01.041.

[82] M. Aydin and Ö. Öz, “APPLICATION OF DIGITAL IMAGE CORRELATION TECHNIQUE TO TENSILE TEST FOR PRINTED PLA SPECIMENS,” 2018.

[83] M. Jerabek, Z. Major, and R. W. Lang, “Strain determination of polymeric materials using digital image correlation,” *Polymer Testing*, vol. 29, no. 3, pp. 407–416, May 2010, doi: 10.1016/j.polymertesting.2010.01.005.

[84] A. Fuad and A. Ghani, “Digital image correlation technique in measuring deformation and failure of composite and adhesive Hybrid Composite Characterization CFRP/GFRP Under Static Loading View project DIGITAL

IMAGE CORRELATION TECHNIQUE IN MEASURING DEFORMATION AND FAILURE OF COMPOSITE AND ADHESIVE," vol. 11, no. 22, 2016, [Online]. Available: www.arpnjournals.com

[85] G. Szebényi and V. Hliva, "Detection of delamination in polymer composites by digital image correlation-experimental test," *Polymers*, vol. 11, no. 3, Mar. 2019, doi: 10.3390/polym11030523.

[86] L. C. S. Nunes, "Mechanical characterization of hyperelastic polydimethylsiloxane by simple shear test," *Materials Science and Engineering A*, vol. 528, no. 3, pp. 1799–1804, Jan. 2011, doi: 10.1016/j.msea.2010.11.025.

[87] S. P. Mates, R. Rhorer, A. Forster, R. K. Everett, K. E. Simmonds, and A. Bagchi, "Modeling and DIC Measurements of Dynamic Compression Tests of a Soft Tissue Simulant."

[88] L. Chevalier, S. Calloch, F. Hild, and Y. Marco, "Digital image correlation used to analyze the multiaxial behavior of rubber-like materials," *European Journal of Mechanics, A/Solids*, vol. 20, no. 2, pp. 169–187, 2001, doi: 10.1016/S0997-7538(00)01135-9.

[89] Z. Li and Z. Liu, "An algorithm for obtaining real stress field of hyperelastic materials based on digital image correlation system," *International Journal of Computational Materials Science and Engineering*, vol. 6, no. 4, Dec. 2017, doi: 10.1142/S2047684118500033.

[90] J. Zapata-Farfan *et al.*, "Low-energy/pulse response and high-resolution-CMOS camera for spatiotemporal femtosecond laser pulses characterization @ 1.55 μ m," *Review of Scientific Instruments*, vol. 90, no. 4, Apr. 2019, doi: 10.1063/1.5071447.

[91] O. Stankiewicz, G. Lafruit, and M. Domański, "Multiview video: Acquisition, processing, compression, and virtual view rendering," *Academic Press Library in Signal Processing: Image and Video Processing and Analysis and Computer Vision*, vol. 6, pp. 3–74, Jan. 2018, doi: 10.1016/B978-0-12-811889-4.00001-4.

[92] "MATLAB Documentation."

[93] N. R. Raajan, M. Ramkumar, B. Monisha, C. Jaiseeli, and S. P. Venkatesan, "Disparity Estimation from Stereo Images," *Procedia Engineering*, vol. 38, pp. 462–472, Jan. 2012, doi: 10.1016/J.PROENG.2012.06.057.

[94] R. A. Hamzah and H. Ibrahim, "Literature Survey on Stereo Vision Disparity Map Algorithms," 2016, doi: 10.1155/2016/8742920.

[95] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, Feb. 2008, doi: 10.1109/TPAMI.2007.1166.

[96] L. R. Ramírez-Hernández *et al.*, "Improve three-dimensional point localization accuracy in stereo vision systems using a novel camera calibration method," *International Journal of Advanced Robotic Systems*, vol. 17, no. 1, Jan. 2020, doi: 10.1177/1729881419896717.

[97] S. Kolling, P. A. du Bois, D. J. Benson, and W. W. Feng, "A tabulated formulation of hyperelasticity with rate effects and damage," *Computational Mechanics*, vol. 40, no. 5, pp. 885–899, 2007, doi: 10.1007/S00466-006-0150-X.

APPENDIX A

MATLAB Script

Step 1 – 3D Reconstruction

```
% Load image pairs
right=imread('E:\ResearchWork\FinalExperiments\Experiment7\Specimen\
Left\ImageLeft25.jpg');
left=imread('E:\ResearchWork\FinalExperiments\Experiment7\Specimen\R
ight\ImageRight25.jpg');
right2=imread('E:\ResearchWork\FinalExperiments\Experiment7\Specimen
\Left\ImageLeft26.jpg');
left2=imread('E:\ResearchWork\FinalExperiments\Experiment7\Specimen\
Right\ImageRight26.jpg');

% Rectification
[rectleft, rectright] =rectifyStereoImages (left, right,
stereoParams,'OutputView','full');
[rectleft2, rectright2] =rectifyStereoImages (left2, right2,
stereoParams,'OutputView','full');

% Stereo anaglyph
imtool(stereoAnaglyph(rectleft, rectright));
imtool(stereoAnaglyph(rectleft2, rectright2));

% Conversion to grey scale
frameLeftGray=rgb2gray(rectleft);
frameRightGray=rgb2gray(rectright);
frameLeftGray2=rgb2gray(rectleft2);
frameRightGray2=rgb2gray(rectright2);

% Disparity map generation
disparityMap = disparitySGM(frameLeftGray,frameRightGray,
'DisparityRange',[min max]);
figure;
imshow(disparityMap,[min max]);
hold on;
title('Disparity Map');
colormap jet
colorbar
caxis([min max])

disparityMap2 = disparitySGM(frameLeftGray2,frameRightGray2,
'Disparity Range',[min max]);
figure;
imshow(disparityMap,[min max]);
hold on;
title ('Disparity Map-2');
colormap jet
colorbar
caxis([min max])

% 3D reconstruction
point3D1 = reconstructScene(disparityMap, stereoParams);
point3D2 = reconstructScene(disparityMap2, stereoParams);
```

```
figure;
pcshow(point3D1, rectleft, 'VerticalAxis', 'Y', 'VerticalAxisDir',
'Down');
xlabel('X (mm)');
ylabel('Y (mm)');
zlabel('Z (mm)');
set(gca, 'CameraViewAngle',10, 'CameraUpVector',[0 -1 0],...
    'CameraPosition',[16500 -13852 -49597], 'DataAspectRatio',[1 1
1]);
title('Reconstructed 3-D Scene');
hold on;

figure;
pcshow(point3D2, rectleft2, 'VerticalAxis', 'Y', 'VerticalAxisDir',
'Down');
xlabel('X (mm)');
ylabel('Y (mm)');
zlabel('Z (mm)');
set(gca, 'CameraViewAngle',10, 'CameraUpVector',[0 -1 0],...
    'CameraPosition',[16500 -13852 -49597], 'DataAspectRatio',[1 1
1]);
title('Reconstructed 3-D Scene-2');
```

Step 2 – Define Region of Interest and Remove NaN Points

```
% Defining ROI1
[ROI1,X1,Y1]=roipoly(rectleft);

% ROI 1
point3DExt1=point3D1(min(Y1):max(Y1),min(X1):max(X1),:);

% Removing NaN in point3DExt1
for q=1:5
      point3DExt1(:,:,1)=fillmissing(fillmissing(point3DExt1(:,:,1),
      'movmean',20),'movmean',20,2);
      point3DExt1(:,:,2)=fillmissing(fillmissing(point3DExt1(:,:,2),
      'movmean',20),'movmean',20,2);
      point3DExt1(:,:,3)=fillmissing(fillmissing(point3DExt1(:,:,3),
      'movmean',20),'movmean',20,2);
end

% Random check for the NaN points in point3DExt1
sum(sum(isnan(point3DExt1)))

% Display ROI - point3DExt1 without NaNs
figure;
pcshow(point3DExt1, rectleft(min(Y1):max(Y1),min(X1):max(X1)),
'VerticalAxis', 'Y', 'VerticalAxisDir', 'Down');
xlabel('X (mm)');
ylabel('Y (mm)');
zlabel('Z (mm)');
set(gca, 'CameraViewAngle',10, 'CameraUpVector',[0 -1 0],...
    'CameraPosition',[16500 -13852 -49597], 'DataAspectRatio',[1 1
1]);
title('point3DExt1');
hold on;
```

```
% Defining ROI2
[ROI2,X2,Y2]=roipoly(rectleft2);

% ROI 2
point3DExt2=point3D2(min(Y2):max(Y2),min(X2):max(X2),:);

% Removing nan in point3DExt2
for q=1:5
      point3DExt2(:,:,1)=fillmissing(fillmissing(point3DExt2(:,:,1),
      'movmean',20),'movmean',20,2);
      point3DExt2(:,:,2)=fillmissing(fillmissing(point3DExt2(:,:,2),
      'movmean',20),'movmean',20,2);
      point3DExt2(:,:,3)=fillmissing(fillmissing(point3DExt2(:,:,3),
      'movmean',20),'movmean',20,2);
end

% Random check for the NaN points in point3DExt2
sum(sum(isnan(point3DExt2)))

% Display ROI - point3DExt2 without NaNs
figure;
pcshow(point3DExt2, rectleft2(min(Y2):max(Y2),min(X2):max(X2)),
'VerticalAxis', 'Y', 'VerticalAxisDir', 'Down');
xlabel('X (mm)');
ylabel('Y (mm)');
zlabel('Z (mm)');
set(gca, 'CameraViewAngle',10, 'CameraUpVector',[0 -1 0],...
    'CameraPosition',[16500 -13852 -49597], 'DataAspectRatio',[1 1
1]);
title('point3DExt2');
hold on;
```

Step 3 – Smooth out extracted point cloud – 1

```
% Smooth out x coordinates
r=size(point3DExt1,1);
c=size(point3DExt1,2);
SmoothX=point3DExt1(:,:,1);
z=SmoothX;
x=meshgrid(1:c,1:r);
y=meshgrid(1:r,1:c)';
[xo, yo, zo]=prepareSurfaceData(x,y,z);
surffit=fit([xo,yo],zo,'poly33','normalize','on');
newz=surffit(xo,yo);
newz=reshape(newz,r,c);
SmoothX(1:r,1:c)=newz;

% Display the smoothened x coordinates
figure;
colormap hsv
surf(SmoothX,
'FaceColor','interp','EdgeColor','none','FaceLighting','gouraud');
daspect([1 1 0.1])
axis tight
camlight left
```

```
title('Smoothened x coordinate - point3DExt1')
set(gcf,'color','w');
view (-40,15)
hold on

% Smooth out y coordinates
r=size(point3DExt1,1);
c=size(point3DExt1,2);
SmoothY=point3DExt1(:,:,2);
z=SmoothY;
x=meshgrid(1:c,1:r);
y=meshgrid(1:r,1:c)';
[xo, yo, zo]=prepareSurfaceData(x,y,z);
surffit=fit([xo,yo],zo,'poly33','normalize','on');
newz=surffit(xo,yo);
newz=reshape(newz,r,c);
SmoothY(1:r,1:c)=newz;

% Display the smoothened y coordinates
figure;
colormap hsv
surf(SmoothY,
'FaceColor','interp','EdgeColor','none','FaceLighting','gouraud');
daspect([1 1 0.1])
axis tight
camlight left
title('Smoothened y coordinate - point3DExt1')
set(gcf,'color','w');
view (-40,15)
hold on

% Smooth out z coordinates
r=size(point3DExt1,1);
c=size(point3DExt1,2);
SmoothZ=point3DExt1(:,:,3);
z=SmoothZ;
x=meshgrid(1:c,1:r);
y=meshgrid(1:r,1:c)';
[xo, yo, zo]=prepareSurfaceData(x,y,z);
surffit=fit([xo,yo],zo,'poly22','normalize','on');
newz=surffit(xo,yo);
newz=reshape(newz,r,c);
SmoothZ(1:r,1:c)=newz;

% Display the smoothened z coordinates
figure;
colormap hsv
surf(SmoothZ,
'FaceColor','interp','EdgeColor','none','FaceLighting','gouraud');
daspect([1 1 0.01])
axis tight
camlight left
set(gcf,'color','w');
view (-40,15)
hold on

% Assign smoothened coordinates
```

```
newpoint3DExt1=nan(size(point3DExt1,1),size(point3DExt1,2),3);
newpoint3DExt1(:,:,1)=SmoothX;
newpoint3DExt1(:,:,2)=SmoothY;
newpoint3DExt1(:,:,3)=SmoothZ;

% Display the smoothened extracted point cloud - 1
figure;
pcshow(newpoint3DExt1, rectleft(min(Y1):max(Y1),min(X1):max(X1)),
'VerticalAxis', 'Y', 'VerticalAxisDir', 'Down');
xlabel('X (mm)');
ylabel('Y (mm)');
zlabel('Z (mm)');
set(gca, 'CameraViewAngle',10, 'CameraUpVector',[0 -1 0],...
    'CameraPosition',[16500 -13852 -49597], 'DataAspectRatio',[1 1
1]);
title('newpoint3DExt1');
hold on;

% Merge the extracted region into the entire point cloud
newpoint3D1=nan(size(point3D1,1), size(point3D1,2),3);
newpoint3D1(min(Y1):max(Y1), min(X1):max(X1),:)=newpoint3DExt1;
```

Step 4 – Smooth out extracted point cloud – 2

```
% Smooth out x coordinates
r=size(point3DExt2,1);
c=size(point3DExt2,2);
SmoothX=point3DExt2(:,:,1);
z=SmoothX;
x=meshgrid(1:c,1:r);
y=meshgrid(1:r,1:c)';
[xo, yo, zo]=prepareSurfaceData(x,y,z);
surffit=fit([xo,yo],zo,'poly33','normalize','on');
newz=surffit(xo,yo);
newz=reshape(newz,r,c);
SmoothX(1:r,1:c)=newz;

% Display the smoothened x coordinates
figure;
colormap hsv
surf(SmoothX,
'FaceColor','interp','EdgeColor','none','FaceLighting','gouraud');
daspect([1 1 0.1])
axis tight
camlight left
title('Smoothened x coordinate - point3DExt1')
set(gcf,'color','w');
view (-40,15)
hold on

% Smooth out y coordinates
r=size(point3DExt2,1);
c=size(point3DExt2,2);
SmoothY=point3DExt2(:,:,2);
z=SmoothY;
x=meshgrid(1:c,1:r);
```

```
y=meshgrid(1:r,1:c)';
[xo, yo, zo]=prepareSurfaceData(x,y,z);
surffit=fit([xo,yo],zo,'poly33','normalize','on');
newz=surffit(xo,yo);
newz=reshape(newz,r,c);
SmoothY(1:r,1:c)=newz;

% Display the smoothened y coordinates
figure;
colormap hsv
surf(SmoothY,
'FaceColor','interp','EdgeColor','none','FaceLighting','gouraud');
daspect([1 1 0.1])
axis tight
camlight left
title('Smoothened y coordinate - point3DExt1')
set(gcf,'color','w');
view (-40,15)
hold on

% Smooth out z coordinates
r=size(point3DExt2,1);
c=size(point3DExt2,2);
SmoothZ=point3DExt2(:,:,3);
z=SmoothZ;
x=meshgrid(1:c,1:r);
y=meshgrid(1:r,1:c)';
[xo, yo, zo]=prepareSurfaceData(x,y,z);
surffit=fit([xo,yo],zo,'poly22','normalize','on');
newz=surffit(xo,yo);
newz=reshape(newz,r,c);
SmoothZ(1:r,1:c)=newz;

% Display the smoothened z coordinates
figure;
colormap hsv
surf(SmoothZ,
'FaceColor','interp','EdgeColor','none','FaceLighting','gouraud');
daspect([1 1 0.01])
axis tight
camlight left
set(gcf,'color','w');
view (-40,15)
hold on

% Assign smoothened coordinates
newpoint3DExt2=nan(size(point3DExt2,1),size(point3DExt2,2),3);
newpoint3DExt2(:,:,1)=SmoothX;
newpoint3DExt2(:,:,2)=SmoothY;
newpoint3DExt2(:,:,3)=SmoothZ;

% Display the smoothened extracted point cloud - 2
figure;
pcshow(newpoint3DExt2, rectleft2(min(Y2):max(Y2),min(X2):max(X2)),
'VerticalAxis', 'Y', 'VerticalAxisDir', 'Down');
xlabel('X (mm)');
ylabel('Y (mm)');
```

```
zlabel('Z (mm)');
set(gca, 'CameraViewAngle',10, 'CameraUpVector',[0 -1 0],...
    'CameraPosition',[16500 -13852 -49597], 'DataAspectRatio',[1 1
1]);
title('newpoint3DExt2');
hold on;

% Merge the extracted region into the entire point cloud
newpoint3D2=nan(size(point3D2,1),size(point3D2,2),3);
newpoint3D2(min(Y2):max(Y2),min(X2):max(X2),:)=newpoint3DExt2;
```

Step 5 – Computation of vertical displacement

```
% Define start and end point of ROI
St1=[min(X1),min(Y1)];
End1=[max(X1),max(Y1)];
St2=[min(X2),min(Y2)];
End2=[max(X2),max(Y2)];

% Define rows, columns and initial points for iteration
rows=3;
cols=3;
sz1=size(rectleft,1);
sz2=size(rectleft,2);
totalDis_y=nan(sz1,sz2,(rows*cols));
stA=St1(1,1);
stB=St1(1,2);

% Iteration for the computation of vertical displacements in subsets
count=1;
for i=1:rows
      for j=1:cols
            a=stA;
            b=stB;
            width=(End1(1,1)-St1(1,1))/cols;
            height=floor((End1(1,2)-St1(1,2))/rows);

            % Define subset for geometric transformation estimation
            boxPolygonk=polyshape([a a a+width a+width],[b b+height
            b+height b]);

            % Detect & extract features from deformed & undeformed
            images
            boxPoints=detectKAZEFeatures(frameLeftGray,'ROI',[a,b,wi
            dth,height]);
            scenePoints = detectKAZEFeatures(frameLeftGray2);
            [boxFeatures, boxPoints] =extractFeatures(frameLeftGray,
            boxPoints);
            [sceneFeatures, scenePoints]
            =extractFeatures(frameLeftGray2, scenePoints);

            % Matchfeatures
            boxPairs = matchFeatures(boxFeatures, sceneFeatures);

            % Display the matched features
```

```
        matchedBoxPoints = boxPoints(boxPairs(:, 1), :);
        matchedScenePoints = scenePoints(boxPairs(:, 2), :);

        % Finding the geomatric transformation
        [tform, inlierBoxPoints, inlierScenePoints] = ...
        estimateGeometricTransform(matchedBoxPoints,matchedScene
        Points,'projective','Confidence',99,'MaxDistance',1.5);

        % Create NaN matrix to store the displacement values
        matDis_y=NaN(sz1,sz2);

        % Calculation of displacement
        inc=3;
        for r=1:inc:height
              aint=a;
              for c=1:inc:width
                    box=3;
                    boxPolygon=polyshape([aint aint aint+box
                    aint+box],[b b+box b+box b]);
                    newBoxPolygon =
                    polyshape(transformPointsForward(tform,
                    boxPolygon.Vertices));
                    [cenx,ceny]=centroid(boxPolygon);
                    [cennewx,cennewy]=centroid(newBoxPolygon);

                    % Coordinates from point cloud 1
                    x1=newpoint3D1(int16(ceny),int16(cenx),1);
                    y1=newpoint3D1(int16(ceny),int16(cenx),2);
                    z1=newpoint3D1(int16(ceny),int16(cenx),3);

                    % Coordinates from point cloud 2
                    x1c=newpoint3D2(int16(cennewy),int16(cennewx
                    ),1);
                    y1c=newpoint3D2(int16(cennewy),int16(cennewx
                    ),2);
                    z1c=newpoint3D2(int16(cennewy),int16(cennewx
                    ),3);

                    if y1c>y1
                          dy=sqrt((y1c-y1)^2+(z1c-z1)^2);
                    else
                          dy=-1.*sqrt((y1c-y1)^2+(z1c-z1)^2);
                    end

                    matDis_y(int16(cennewy),int16(cennewx))=dy;
                    aint=aint+inc;
              end
              b=b+inc;
          end
          totalDis_y(:,:,(count))=matDis_y;
          stA=stA+width;
          count=count+1;
    end
    stA=St1(1,1);
    stB=stB+height;
end
```

```
% Concatenate and get the mean of the matrices
majorDis_y=nanmean(totalDis_y,3);

% Remove NaN points
nanmat_y=majorDis_y(min(Y2):max(Y2),min(X2):max(X2));
newMajorDis_y=fillmissing(fillmissing(nanmat_y,'movmean',20),'movmea
n',20,2);
for q=1:4
      newMajorDis_y=fillmissing(fillmissing(newMajorDis_y,'movmean',
      20),'movmean',20,2);
end

% Random check for NaN points
sum(sum(isnan(newMajorDis_y)))

% Display the displacement
figure;
imagesc(newMajorDis_y);
hold on;
axis('equal');
caxis([-2 16]);
title('Displacement Y');
colormap jet
colorbar
set(gcf,'color','w');
```

Step 6 – Computation of horizontal displacement

```
% Define start and end point of ROI
St1=[min(X1),min(Y1)];
End1=[max(X1),max(Y1)];
St2=[min(X2),min(Y2)];
End2=[max(X2),max(Y2)];

% Define rows, columns and initial points for iteration
rows=3;
cols=3;
sz1=size(rectleft,1);
sz2=size(rectleft,2);
totalDis_x=nan(sz1,sz2,(rows*cols));
stA=St1(1,1);
stB=St1(1,2);

% Iteration for the computation of horizontal displacements in
subsets
count=1;
for i=1:rows
      for j=1:cols
            a=stA;
            b=stB;
            width=(End1(1,1)-St1(1,1))/cols;
            height=floor((End1(1,2)-St1(1,2))/rows);

            % Define subset for geometric transformation estimation
```

```
boxPolygonk=polyshape([a a a+width a+width],[b b+height
b+height b]);

% Detect & Extract features from deformed & undeformed
images
boxPoints=
detectKAZEFeatures(frameLeftGray,'ROI',[a,b,width,height
]);
scenePoints = detectKAZEFeatures(frameLeftGray2);
[boxFeatures, boxPoints] =
extractFeatures(frameLeftGray,boxPoints);
[sceneFeatures, scenePoints] =
extractFeatures(frameLeftGray2,scenePoints);

% Matchfeatures
boxPairs = matchFeatures(boxFeatures, sceneFeatures);

% Display the matched features
matchedBoxPoints = boxPoints(boxPairs(:, 1), :);
matchedScenePoints = scenePoints(boxPairs(:, 2), :);

% Finding the geometric transformation
[tform, inlierBoxPoints, inlierScenePoints] =
estimateGeometricTransform(matchedBoxPoints,
matchedScenePoints,
'projective','Confidence',99,'MaxDistance',1.5);

% Create NaN matrix to store the displacement values
matDis_x=NaN(sz1,sz2);

% Calculation of displacement
inc=3;
for r=1:inc:height
      aint=a;
      for c=1:inc:width
            box=3;
            boxPolygon=polyshape([aint aint aint+box
            aint+box],[b b+box b+box b]);
            newBoxPolygon =
            polyshape(transformPointsForward(tform,
            boxPolygon.Vertices));
            [cenx,ceny]=centroid(boxPolygon);
            [cennewx,cennewy]=centroid(newBoxPolygon);

            % Coordinates from point cloud - 1
            x1=newpoint3D1(int16(ceny),int16(cenx),1);
            y1=newpoint3D1(int16(ceny),int16(cenx),2);
            z1=newpoint3D1(int16(ceny),int16(cenx),3);

            % Coordinates from point cloud - 2
            x1c=newpoint3D2(int16(cennewy),int16(cennewx
            ),1);
            y1c=newpoint3D2(int16(cennewy),int16(cennew
            ),2);
            z1c=newpoint3D2(int16(cennewy),int16(cennewx
            ),3);
```

```
                        if x1c>x1
                              dx=sqrt((x1c-x1)^2+(z1c-z1)^2);
                        else
                              dx=-1.*sqrt((x1c-x1)^2+(z1c-z1)^2);
                        end

                        matDis_x(int16(cennewy),int16(cennewx))=dx;
                        aint=aint+inc;
                  end
                  b=b+inc;
            end
            totalDis_x(:,:,(count))=matDis_x;
            stA=stA+width;
            count=count+1;
      end
      stA=St1(1,1);
      stB=stB+height
end

% Concatenate and get the mean of the matrices
majorDis_x=nanmean(totalDis_x,3);

% Remove NaN points
nanmat_x=majorDis_x(min(Y2):max(Y2),min(X2):max(X2));
newMajorDis_x=fillmissing(fillmissing(nanmat_x,'movmean',20),'movmea
n',20,2);
for q=1:4
      newMajorDis_x=fillmissing(fillmissing(newMajorDis_x,'movmean',
      20),'movmean',20,2);
end

% Random check for NaN points
sum(sum(isnan(newMajorDis_x)))

% Display the displacement
figure;
imagesc(newMajorDis_x);
hold on;
axis('equal');
% caxis([-1.4 1.4]);
title('Displacement X');
colormap jet
colorbar
set(gcf,'color','w');
```

Step 7 – Computation of vertical strain

```
% Smooth out vertical difference in y coordinates
r=size(newpoint3DExt2,1);
c=size(newpoint3DExt2,2);
Ycoord=imresize(newpoint3DExt1(:,:,2),[r c]);
DeltaY=diff(Ycoord);
x=meshgrid(1:c,1:r-1);
y=meshgrid(1:r-1,1:c)';
[xo, yo, zo]=prepareSurfaceData(x,y,DeltaY);
```

```
surffit=fit([xo,yo],zo,'poly11','normalize','on');
newDeltaY_raw=surffit(xo,yo);
newDeltaY=reshape(newDeltaY_raw,r-1,c);

% Savitzky-Golay smoothing filter for vertical displacement field
r=size(newpoint3DExt2,1);
c=size(newpoint3DExt2,2);
Dis_y=newMajorDis_y;

% Smoothened by 80x20 window @ 5-pixel interval
for i= 1:5:r-80
    for j = 1:5:c-20
        if i+5>r-80
            p=r-i;
        else
            p=80;
        end

        if j+5>c-20
            q=c-j;
        else
            q=20;
        end

        z=Dis_y(i:i+p,j:j+q);
        x=meshgrid(1:q+1,1:p+1);
        y=meshgrid(1:p+1,1:q+1)';
        [xo, yo, zo]=prepareSurfaceData(x,y,z);
        surffit=fit([xo,yo],zo,'poly22','normalize','on');
        newz=surffit(xo,yo);
        newz=reshape(newz,p+1,q+1);
        Dis_y(i:i+p,j:j+q)=newz;
    end
end

% Display smoothened vertical displacement
figure;
imagesc(Dis_y);
hold on;
axis('equal');
caxis([0 11]);
title('Smoothed Displacement field - Y');
colormap jet
colorbar('southoutside')
set(gca,'XColor', 'none','YColor','none')
set(gcf,'color','w');

% Savitzky-Golay smoothing filter for difference(vertical) in
vertical displacement field
r=size(newpoint3DExt2,1)-1;
c=size(newpoint3DExt2,2);
Del_Dis_y_ydir=diff(Dis_y);

% Smoothened by 80x20 window @ 5-pixel interval
for i= 1:5:r-80
    for j = 1:5:c-20
        if i+5>r-80
```

```
            p=r-i;
        else
            p=80;
        end

        if j+5>c-20
            q=c-j;
        else
            q=20;
        end

        z=Del_Dis_y_ydir(i:i+p,j:j+q);
        x=meshgrid(1:q+1,1:p+1);
        y=meshgrid(1:p+1,1:q+1)';
        [xo, yo, zo]=prepareSurfaceData(x,y,z);
        surffit=fit([xo,yo],zo,'poly22','normalize','on');
        newz=surffit(xo,yo);
        newz=reshape(newz,p+1,q+1);
        Del_Dis_y_ydir(i:i+p,j:j+q)=newz;
    end
end


% Display the vertical strain field
figure;
imagesc(Del_Dis_y_ydir./newDeltaY);
axis('equal')
caxis([0 0.45])
hold on;
title('Strain yy');
colormap jet;
colorbar('southoutside')
set(gca,'XColor', 'none','YColor','none')
set(gcf,'color','w');
```

Step 8 – Computation of horizontal strain

```
% Smooth out difference(horizontal) in x coordinates
r=size(newpoint3DExt2,1);
c=size(newpoint3DExt2,2);
Xcoord=imresize(newpoint3DExt1(:,:,1),[r c]);
DeltaX=diff(Xcoord,1,2);
x=meshgrid(1:c-1,1:r);
y=meshgrid(1:r,1:c-1)';
[xo, yo, zo]=prepareSurfaceData(x,y,DeltaX);
surffit=fit([xo,yo],zo,'poly11','normalize','on');
newDeltaX_raw=surffit(xo,yo);
newDeltaX=reshape(newDeltaX_raw,r,c-1);

% Savitzky-Golay smoothing filter for horizontal displacement field
r=size(newpoint3DExt2,1);
c=size(newpoint3DExt2,2);
Dis_x=newMajorDis_x;

% Smoothened by 80x20 window @ 5-pixel interval
for i= 1:5:r-80
```

```
    for j = 1:5:c-20
        if i+5>r-80
            p=r-i;
        else
            p=80;
        end

        if j+5>c-20
            q=c-j;
        else
            q=20;
        end

        z=Dis_x(i:i+p,j:j+q);
        x=meshgrid(1:q+1,1:p+1);
        y=meshgrid(1:p+1,1:q+1)';
        [xo, yo, zo]=prepareSurfaceData(x,y,z);
        surffit=fit([xo,yo],zo,'poly22','normalize','on');
        newz=surffit(xo,yo);
        newz=reshape(newz,p+1,q+1);
        Dis_x(i:i+p,j:j+q)=newz;
    end
end

% Display smoothened horizontal displacement
figure;
imagesc(Dis_x);
hold on;
axis('equal');
caxis([-3.5 2]);
title('Smoothed Displacement field - X');
colormap jet
colorbar('southoutside')
set(gca,'XColor', 'none','YColor','none')
set(gcf,'color','w');

% Savitzky-Golay smoothing filter for difference(horizontal) in
horizontal displacement field
r=size(newpoint3DExt2,1);
c=size(newpoint3DExt2,2)-1;
Del_Dis_x_xdir=diff(Dis_x,1,2);

% Smoothened by 80x20 window @ 5-pixel interval
for i= 1:5:r-80
    for j = 1:5:c-20
        if i+5>r-80
            p=r-i;
        else
            p=80;
        end

        if j+5>c-20
            q=c-j;
        else
            q=20;
        end
```

```
        z=Del_Dis_x_xdir(i:i+p,j:j+q);
        x=meshgrid(1:q+1,1:p+1);
        y=meshgrid(1:p+1,1:q+1)';
        [xo, yo, zo]=prepareSurfaceData(x,y,z);
        surffit=fit([xo,yo],zo,'poly22','normalize','on');
        newz=surffit(xo,yo);
        newz=reshape(newz,p+1,q+1);
        Del_Dis_x_xdir(i:i+p,j:j+q)=newz;
    end
end

% Display the horizontal strain field
figure;
imagesc(Del_Dis_x_xdir./newDeltaX);
axis('equal')
caxis([-0.9 .2]);
hold on;
title('Strain xx');
colormap jet;
colorbar('southoutside')
set(gca,'XColor', 'none','YColor','none')
set(gcf,'color','w');
```

Step 9 – Computation of shear strain

```
% Savitzky-Golay smoothing filter for difference(vertical) in
horizontal displacement field
r=size(newpoint3DExt2,1)-1;
c=size(newpoint3DExt2,2);
Del_Dis_x_ydir=diff(Dis_x);

% Smoothened by 80x20 window @ 5-pixel interval
for i= 1:5:r-80
    for j = 1:5:c-20
        if i+5>r-80
            p=r-i;
        else
            p=80;
        end

        if j+5>c-20
            q=c-j;
        else
            q=20;
        end

        z=Del_Dis_x_ydir(i:i+p,j:j+q);
        x=meshgrid(1:q+1,1:p+1);
        y=meshgrid(1:p+1,1:q+1)';
        [xo, yo, zo]=prepareSurfaceData(x,y,z);
        surffit=fit([xo,yo],zo,'poly22','normalize','on');
        newz=surffit(xo,yo);
        newz=reshape(newz,p+1,q+1);
        Del_Dis_x_ydir(i:i+p,j:j+q)=newz;
    end
```

```
end
DXbyDy=Del_Dis_x_ydir./newDeltaY;

% Savitzky-Golay smoothing filter for difference(horizontal) in
vertical displacement field

r=size(newpoint3DExt2,1);
c=size(newpoint3DExt2,2)-1;
Del_Dis_y_xdir=diff(Dis_y,1,2);

% Smoothened by 80x20 window @ 5-pixel interval
for i= 1:5:r-80
    for j = 1:5:c-20
        if i+5>r-80
            p=r-i;
        else
            p=80;
        end

        if j+5>c-20
            q=c-j;
        else
            q=20;
        end

        z=Del_Dis_y_xdir(i:i+p,j:j+q);
        x=meshgrid(1:q+1,1:p+1);
        y=meshgrid(1:p+1,1:q+1)';
        [xo, yo, zo]=prepareSurfaceData(x,y,z);
        surffit=fit([xo,yo],zo,'poly22','normalize','on');
        newz=surffit(xo,yo);
        newz=reshape(newz,p+1,q+1);
        Del_Dis_y_xdir(i:i+p,j:j+q)=newz;
    end
end
DYbyDx=Del_Dis_y_xdir./newDeltaX;

% Computation of shear strain
ShearStr=(DYbyDx(2:size(newpoint3DExt2,1),:))+(DXbyDy(:,2:size(newpo
int3DExt2,2)));

% Display the shear strain field
figure;
imagesc(ShearStr);
axis('equal')
caxis([-.45 0.3]);
hold on;
title('Shear Strain');
colormap jet
colorbar('southoutside')
set(gca,'XColor', 'none','YColor','none')
set(gcf,'color','w');
```

Step 10 – Addition of incremental displacement and strain fields

```
% Addition of incremental vertical displacement fields
TotaDisY1=Dis_y_1;
TotaDisY2=Dis_y_2 + imresize(Dis_y_1,[size(Dis_y_2,1)
size(Dis_y_2,2)]); % and so on

% Display the final vertical displacement fields
figure;
imagesc(TotaDisY1);
axis('equal')
hold on;
title('Dis-Y-1st');
colormap jet;
colorbar
set(gca,'XColor', 'none','YColor','none')
set(gcf,'color','w');

% Addition of incremental horizontal displacement fields
TotaDisX1=Dis_x_1;
TotaDisX2=Dis_x_2 + imresize(Dis_x_1,[size(Dis_x_2,1)
size(Dis_x_2,2)]); % and so on

% Display the final horizontal displacement fields
figure;
imagesc(TotaDisX1);
axis('equal')
hold on;
title('Dis-X-1st');
colormap jet;
colorbar
set(gca,'XColor', 'none','YColor','none')
set(gcf,'color','w');

% Addition of incremental vertical strain fields
TotaStrY1=Str_yy_1;
TotaStrY2=Str_yy_2 + imresize(Str_yy_1,[size(Str_yy_2,1)
size(Str_yy_2,2)]); % and so on

% Display the final vertical strain fields
figure;
imagesc(TotaStrY1);
axis('equal')
hold on;
title('Str-Y-1st');
colormap jet;
colorbar
set(gca,'XColor', 'none','YColor','none')
set(gcf,'color','w');

% Addition of incremental horizontal strain fields
TotaStrX1=Str_xx_1;
TotaStrX2=Str_xx_2 + imresize(Str_xx_1,[size(Str_xx_2,1)
size(Str_xx_2,2)]); % and so on

% Display the final horizontal strain fields
```

```
figure;
imagesc(TotaStrX1);
axis('equal')
hold on;
title('Str-X-1st');
colormap jet;
colorbar
set(gca,'XColor', 'none','YColor','none')
set(gcf,'color','w');

% Addition of incremental shear strain fields
TotaShr1=ShearStr_1;
TotaShr2=ShearStr_2 + imresize(ShearStr_1,[size(ShearStr_2,1)
size(ShearStr_2,2)]); % and so on

% Display the final shear strain fields
figure;
imagesc(TotaShr1);
axis('equal')
hold on;
title('Shear-1st');
colormap jet;
colorbar
set(gca,'XColor', 'none','YColor','none')
set(gcf,'color','w');
```

Step 11 – Generation of displacement fields in pixels (2D analysis for comparison)

```
% Load image pairs
left=imread('E:\ResearchWork\FinalExperiments\Experiment7\Specimen\L
eft\ImageLeft25.jpg');
left2=imread('E:\ResearchWork\FinalExperiments\Experiment7\Specimen\
Left\ImageLeft26.jpg');

% Define Region of interest
[ROI1,X1,Y1]=roipoly(left);
[ROI2,X2,Y2]=roipoly(left2);

% Define start and end point of ROI
St1=[min(X1),min(Y1)];
End1=[max(X1),max(Y1)];
St2=[min(X2),min(Y2)];
End2=[max(X2),max(Y2)];

% Define rows, columns and other parameters for iteration
rows=4;
cols=2;
sz1=size(left,1);
sz2=size(left,2);
totalDis_y=nan(sz1,sz2,(rows*cols));
totalDis_x=nan(sz1,sz2,(rows*cols));
stA=St1(1,1);
stB=St1(1,2);
```

```
% Iteration for the computation of vertical and displacements in
subsets
count=1;
for i=1:rows
      for j=1:cols
            a=stA;
            b=stB;
            width=(End1(1,1)-St1(1,1))/cols;
            height=floor((End1(1,2)-St1(1,2))/rows);

            % Define subset for geometric transformation estimation
            boxPolygonk=polyshape([a a a+width a+width],[b b+height
            b+height b]);

            % Detect & Extract features from deformed & undeformed
            images
            boxPoints =
            detectKAZEFeatures(rgb2gray(left),'ROI',[a,b,width,heigh
            t]);
            scenePoints = detectKAZEFeatures(rgb2gray(left2));
            [boxFeatures, boxPoints] =
            extractFeatures(rgb2gray(left), boxPoints);
            [sceneFeatures, scenePoints] =
            extractFeatures(rgb2gray(left2), scenePoints);

            % Match features
            boxPairs = matchFeatures(boxFeatures, sceneFeatures);

            % Display the matched features
            matchedBoxPoints = boxPoints(boxPairs(:, 1), :);
            matchedScenePoints = scenePoints(boxPairs(:, 2), :);

            % Finding the geomatric transformation
            [tform, inlierBoxPoints, inlierScenePoints] = ...
            estimateGeometricTransform(matchedBoxPoints,
            matchedScenePoints,
            'projective','Confidence',99,'MaxDistance',1.5);

            % Finding the displacements
            matDis_y=NaN(sz1,sz2);
            matDis_x=NaN(sz1,sz2);

            inc=3;
            for r=1:inc:height
                  aint=a;
            for c=1:inc:width
                  box=3;
                  boxPolygon=polyshape([aint aint aint+box
                  aint+box],[b b+box b+box b]);
                  newBoxPolygon =
                  polyshape(transformPointsForward(tform,
                  boxPolygon.Vertices));

                  [cenx,ceny]=centroid(boxPolygon);
                  [cennewx,cennewy]=centroid(newBoxPolygon);
```

```
                  x1=cenx;
                  y1=ceny;
                  x1c=cennewx;
                  y1c=cennewy;

                  dy = y1c-y1;
                  dx = (x1c-x1);

                  matDis_y(int16(cennewy),int16(cennewx))=dy;
                  matDis_x(int16(cennewy),int16(cennewx))=dx;
                  aint=aint+inc;
            end
            b=b+inc;
        end
        totalDis_y(:,:,(count))=matDis_y;
        totalDis_x(:,:,(count))=matDis_x;
        stA=stA+width;
        count=count+1;
    end
    stA=St1(1,1);
    stB=stB+height;
end

% Concatenate and get the mean of the matrices
majorDis_y=nanmean(totalDis_y,3);
majorDis_x=nanmean(totalDis_x,3);
r = int16(max(Y2)- min(Y2));
c = int16(max(X2)- min(X2));

% Remove NaN points
nanmat_y=majorDis_y(int16(min(Y2)):int16((min(Y2))+r-
1),int16(min(X2)):int16(min(X2))+c-1);
newMajorDis_y=fillmissing(fillmissing(nanmat_y,'movmean',20),'movmea
n',20,2);

nanmat_x=majorDis_x(int16(min(Y2)):int16((min(Y2))+ r-
1),int16(min(X2)):int16(min(X2))+c-1);
newMajorDis_x=fillmissing(fillmissing(nanmat_x,'movmean',20),'movmea
n',20,2);

for q=1:4
      newMajorDis_y=fillmissing(fillmissing(newMajorDis_y,'movmean',
      20),'movmean',20,2);
      newMajorDis_x=fillmissing(fillmissing(newMajorDis_x,'movmean',
      20),'movmean',20,2);
end

% Random check for the NaN points
sum(sum(isnan(newMajorDis_y)))
sum(sum(isnan(newMajorDis_x)))

% Savitzky-Golay smoothing filter for vertical displacement field
Dis_y=newMajorDis_y;

% Smoothened by 80x20 window @ 5-pixel interval
for i= 1:5:r-80
```

```
    for j = 1:5:c-20
        if i+5>r-80
            p=r-i;
        else
            p=80;
        end

        if j+5>c-20
            q=c-j;
        else
            q=20;
        end

        z=Dis_y(i:i+p,j:j+q);
        x=meshgrid(1:q+1,1:p+1);
        y=meshgrid(1:p+1,1:q+1)';
        [xo, yo, zo]=prepareSurfaceData(x,y,z);
        surffit=fit([xo,yo],zo,'poly22','normalize','on');
        newz=surffit(xo,yo);
        newz=reshape(newz,p+1,q+1);
        Dis_y(i:i+p,j:j+q)=newz;
    end
end

% Display smoothened vertical displacement
figure;
imagesc(Dis_y);
hold on;
axis('equal');
caxis([5 30]);
title('Smoothed Displacement field - Y');
colormap jet
colorbar
set(gca,'XColor', 'none','YColor','none')
set(gcf,'color','w');

% Savitzky-Golay smoothing filter for horizontal displacement field
Dis_x=newMajorDis_x;

% Smoothened by 80x20 window @ 5-pixel interval
for i= 1:5:r-80
    for j = 1:5:c-20
        if i+5>r-80
            p=r-i;
        else
            p=80;
        end

        if j+5>c-20
            q=c-j;
        else
            q=20;
        end

        z=Dis_x(i:i+p,j:j+q);
        x=meshgrid(1:q+1,1:p+1);
```

```
        y=meshgrid(1:p+1,1:q+1)';
        [xo, yo, zo]=prepareSurfaceData(x,y,z);
        surffit=fit([xo,yo],zo,'poly22','normalize','on');
        newz=surffit(xo,yo);
        newz=reshape(newz,p+1,q+1);
        Dis_x(i:i+p,j:j+q)=newz;
    end
end

% Display smoothened horizontal displacement
figure;
imagesc(Dis_x);
hold on;
axis('equal');
caxis([-5 4]);
title('Smoothed Displacement field - X');
colormap jet
colorbar
set(gca,'XColor', 'none','YColor','none')
set(gcf,'color','w');
```