university of groningen

faculty of science and engineering

Computer-vision aided structural vibration tracking and analysis

Exploring the potential of CoTracker, a novel video-based algorithm for full-field vibration analysis

Justin de Groot

university of groningen

faculty of science and engineering

University of Groningen

Computer-vision aided Structural Vibration Tracking and Analysis

Master's Research Project

To fulfill the requirements for the degree of
Master of Science in Industrial Engineering and Management
at University of Groningen under the supervision of
Dr. L. (Liangliang) Cheng
and
Dr. M. (Mauricio) Muñoz Arias

Justin de Groot
(s3729737)

July 16, 2024

Preface

As the end of an academic career approaches, it is fitting to reflect on the past years. These years have been a journey of self-discovery, encompassing academic, social, and professional growth. Over these years, I have discovered my affinity for the industrial environment. The dynamic interplay between theoretical knowledge, ranging from management to complex engineering challenges, has fueled my desire to apply this knowledge in practice.

This research represents a significant first step towards future endeavors. It has provided me with invaluable experience in conducting relevant academic research, always with an eye towards the commercial applicability of the findings. I would like to express my gratitude to several individuals for their support during this research.

I would like to begin by expressing my gratitude to Kun Xie for his dedication and guidance at the start of this research. His support was vital in helping me understand the fundamental technical concepts. Furthermore, I am thankful to Yanxin Si for her assistance with the DMD research.

Specifically, I want to express my gratitude to my primary supervisor, Liangliang Cheng, for his extensive involvement. His engagement significantly contributed to the progress of this research. Lastly, I would like to thank my secondary supervisor, Mauricio Muñoz Arias, for his critical review of the results and feedback on the research design.

Finally, I am very excited that parts of this research contributed to the publication titled 'Camera-Based Dynamic Vibration Analysis Using Transformer-Based Model CoTracker and Dynamic Fashion Decomposition,' which appeared on May 30, 2024 (https://doi.org/10.3390/s24113541). I am delighted to have participated in this research.

Justin de Groot,
Groningen, July 16, 2024

Summary

The results in this research prove the promising application of CoTracker as a full-field vision-based SHM algorithm. In this section, key results of the research are summarized and categorized by research questions.

What is the effect of camera calibration in terms of accurate displacement tracking?

Section 2.3 provides an extensive overview of the theory behind camera distortion and how extrinsic and intrinsic camera properties can be used to account for these distortions. It also explaines how distortion parameters are used to determine the realistic size of pixels in the world-coordinate system. Section 4.1 presents the results of camera calibration by using a chessboard. This is executed before the video analysis is done by either CoTracker or edge detection methods to validate if the distortion in the images is within an acceptable range. The distortion, which is expressed as the mean projection error (MPE), should not exceed a value of 0.5, indicating that the MPE remains within the original pixel. For an accurate calibration, at least 10 images are required. This research used 196 images, of which 188 are accepted and used for calibration. This ensures that, in the context of this experiment, minimal distortion can be achieved. The minimal and maximal distortion are found at 0.144 and 0.631 MPE, respectively. Two images with a respective MPE of 0.144 and 0.475 are used to extract the effect of adequate camera calibration. By identifying identical points in both images, the difference in the extracted pixel size can be determined. The true size of the squares of the chessboard are manually measured to be 17.5 mm. The extracted size of the squares after camera calibration are 17.50007 mm and 17.24176 mm, respective for both images. From this square size, the pixel size is calculated by dividing the square size by the number of pixels between the two square corners. This results in a pixel size of 0.219 mm and 0.203 mm for the respective images. This shows that, in this research, adequate camera calibration will result in a maximum displacement error of ±0.016 mm, caused by image distortion. In the context of this research, this error margin can be neglected.

What is the best strategy to adequately track points in a video for measuring vibrations using CoTracker?

Answering this question validates the optimal method for determining how to initialize the points in the video that are tracked by CoTracker. Section 4.2 elaborates on the results of two experiments. The first experiment is a single harmonic excitation with a constant amplitude. The data from an accelerometer and CoTracker are initially compared to validate if CoTracker is able to accurately determine the displacement of a specific region of interest (ROI) on a cantilever beam. To compare these results, the acceleration measured by CoTracker is extracted from the extracted displacement. It is found that the data from the accelerometer and CoTracker have a correlation of 0.9918 with a standard deviation of 0.4189. The mean difference between the data approximates zero, which implies a strong agreement between the data of the accelerometer and CoTracker, showing the ability of CoTracker to adequately track the structural vibrations.

To validate the best segmentation technique, a different experiment is conducted. By using an impact hammer, a different displacement is generated compared to the first experiment. The

three segmentation strategies used in this research are found to provide similar results . However, for CoTracker to be a full-field measurement algorithm, joint-tracking is preferred. This not only allows tracking the displacement of the entire structure, but also ensures that modal analysis is more accurate.

What is the performance of CoTracker compared to common edge detection methods that are used for vision-based SHM measurements in terms of tracking the displacement?

To compare the performance of CoTracker with existing algorithms used in video-based vibration analysis, several edge detection methods are used. In this analysis, two regions of the beam are evaluated and are represented by a green and red area. The green area is chosen with high efforts on precision to acquire the best results using edge detection, as the red area is chosen arbitrarily to simulate characteristics of full-field measurement. From the results, described in Subsection 4.2.3, two conclusions are made.

When comparing the potential of edge detection with CoTracker in the high precision area, it is found that the correlation between all edge detection methods and CoTracker is higher than 0.96 with a standard deviation less than 0.865 and almost a zero mean difference, indicating a strong agreement on the extracted displacement.

Since CoTracker is evaluated for its potential as a full-field measurement algorithm, an arbitrary area on the beam, indicated in red, is also analyzed. This analysis shows the limitations of edge detection and validates that CoTracker outperforms each edge detection method significantly. When applying constant parameters for edge detection, the mean correlation between each edge detection method and CoTracker is 0.3460 with a mean standard deviation of 4.48 mm, indicating different displacement extraction. As CoTracker is able to extract accurate displacement over the entire beam, it can be concluded that CoTracker performs better in tracking displacement for full-field analysis.

What is the performance of CoTracker compared to common edge detection methods that are used for vision-based SHM measurements in terms of modal parameter extraction?

To finally extract the potential of CoTracker as a full-field measurement algorithm for vision-based SHM, the performance of full-field point tracking is compared with the performance of several edge detection methods. To ensure computational efficiency, 20 points are evaluated on the beam for both CoTracker and edge detection. The first result is that CoTracker is able to identify natural frequencies within 1% of the true natural frequencies found by an accelerometer. To obtain meaningful results for edge detection, a 10% error margin is considered to find the true natural frequencies.

For the damping ratios, CoTracker provides damping ratios of 0.0013 and 0.0024 for the first and second mode, respectively. The correctness of these values are validated by the corresponding mode shape. The mode shapes are identified by a time-embedding DMD algorithm, which shows that the mode shapes by CoTracker are close to the analytical solution proposed in this research, and are also confirmed by the results in [10] which validates the mode shape with a FEM analysis. Additionally, it is found that all edge detection methods are not able to extract the mode shape by using a time-embedded DMD algorithm. Hence, it can be concluded that CoTracker has great potential as a full-field vision-based SHM algorithm compared to several edge detection methods that are widely used for structural vibration analysis.

Abstract

Keywords – Structural vibration; deep-learning; CoTracker; dynamic mode decomposition (DMD); video-based measurement.

Structural health monitoring (SHM) traditionally relies on accelerometers, but these come with practical limitations due to localized measurements and significant operational expenses. A new trend in SHM is non-contact video-based vibration analysis, allowing for comprehensive field analysis. This study investigates a novel deep-learning model named CoTracker for its efficacy as a comprehensive non-contact SHM tool. By recording the vibrations of a cantilever beam using a camera, various structural parameters are evaluated. A time-embedding DMD algorithm is employed for this analysis, and the results are then compared with an analytical solution and a FEM study by Cheng et al [10]. Findings reveal that CoTracker excels in measuring structural vibrations, showing a correlation of > 0.99 with accelerometer readings. Moreover, CoTracker identifies natural frequencies with a 0.7% error compared to accelerometer data. Both the analytical solution and FEM study affirm that CoTracker can extract the first two modal shapes, underscoring its significant potential as a full-field non-contact SHM algorithm.

Contents

List of Figures

List of Tables

Abbreviations

DMD Dynamic Mode Decomposition. 14

DOF Degree of Freedom. 14, 16

FFT Fast Fourier Transform. 13

FRF Frequency Response Function. 14

MPE Mean Projection Error. 25

POD Proper Orthogonal Decomposition. 15

ROI Region of Interest. 19

SHM Structural Health Monitoring. 1

SVD Singular Value Decomposition. 15

1. Introduction

According to the 2021 *Infrastructure Report Card*, which is designed by the American Society of Civil Engineers, there are about 617,000 bridges across the United States. 7.5% of these bridges are estimated to be structurally deficient, which implies that maintenance is required to keep the bridges operational. The financial means for this maintenance are estimated at 125 billion dollars, which emphasizes the financial magnitude of this problem and the importance of well-defined structural health monitoring mechanisms [3]. In today's industrial landscape, structural health monitoring (SHM) plays not only a pivotal role in civil engineering, but also across various sectors, ranging from aerospace to mechanical engineering [19]. Common methods that are employed in SHM are often involved with significant costs, particularly concerning access to remote or hard-to-reach structures and the maintenance of sensors [1, 29, 30].

Typically, SHM systems are composed by three key elements. The first element is at least one sensor that is attached to the structure that is observed. The purpose of the sensors is to measure local conditions in terms of, for example, humidity, wind speed, and temperature [6]. The second element are data handling facilities, such as computers that are able to store the data obtained from the sensors. The final element consists of mathematical algorithms that compare real-time data with historical data to detect changes in the structure [22]. To address a solution that aims to reduce the operational costs involved in SHM, efforts are focused on non-contact measurement technologies [16, 31, 40]. The main idea is that non-contact vibration measurements reduce the need for physical maintenance, consequently reducing the risks and monetary intensives associated with physical inspections.

There are several non-contact methods in SHM, for instance, magnetic methods, which exploit the magnetic properties of certain materials. The base principle is that the magnetic materials in a structure exhibit a characteristic magnetic field. If the structural integrity of the material is compromised, the characteristics of the magnetic field will change [18]. Another non-contact SHM approach is radar-vibration based, where a known frequency radio signal is transmitted to a structural region of interest. Then, the Doppler shift is used to determine if an object is moving away from the signal transmitter or if the object is approaching the signal transmitter. This results in an estimation of the displacement of structural objects, which has shown that vibrations can successfully be extracted [36, 41]. The downside on this method is that it is optimal for slow moving objects. The second most researched topic, according to the extent of the literature, is a non-contact approach based on guided waves [57]. Massery et al. [33] explained how guided ultrasonic waves are able to detect small damages in an aluminum sample by analyzing changes in the guided wave signal response. However, the most promising non-contact method is *vision-based* SHM [57].

The main advantages of vision-based SHM compared to alternatives are the application of low-cost and easy to setup equipment that are able to extract displacements of any points (pixels) on the structure from one video measurement. Although many studies are limited to experimental research in a laboratory, usually using high-contrast images [34, 35, 45, 60], efforts are also focused on field experiments, simulating real-life examples for the potential application of video-based SHM [20]. A practical example is the research of Shariati et al. [51] who conducted experiments on a pedestrian bridge. They showed that vision-based non-contact SHM is possible by extracting the frequency-domain characteristics of the bridge using different vision sensors. Another example is the research by Whabeh et al. [58], who developed a camera system that was pointed on a 1410m bridge in the USA. They installed red LED lights to set markers that were tracked using point-tracking algorithms, i.e., extracting the displacement of the red LED lights.

A common strategy in vision-based non-contact SHM is the application of edge detection algorithms [23, 44]. However, deep-learning based edge detection is also proving its potential [24]. A newly designed algorithm in the field of deep-learning image processing is called *CoTracker*, which is the result of a combined effort of Meta AI and the Visual Geometry Group from the University of Oxford [27]. This algorithm can track the displacement of any pixel in a video. By jointly tracking points, CoTracker is able to track the displacement of points in a video with higher accuracy and robustness than alternative algorithms, such as TAPIR (by a collaboration of Google DeepMind, the University College of London, and the University of Oxford) [15] and PIPs++ (Stanford University) [62], that track points individually and ignore the correlation between points. By integrating point correlation, CoTracker has proven that it provides a solution for a current difficult-to-solve problem, which is the existence of occlusions in a video. An occlusion occurs when an object hides a part of another object and therefore hides the points of the object in the background. CoTracker has proven that its algorithm is able to deal with a certain degree of occlusions in a way that has not been achieved yet [27].

As mentioned, the goal of SHM is to monitor structural conditions by analyzing potential structural defects. Cheng et al. [9] explained how vibration analysis can extract intrinsic structural parameters. For example, natural frequencies, damping ratios, and corresponding mode shapes. A change of these parameters indicates an intrinsic change of the structure, potentially the result of a defect. A method that has been applied successfully to extract intrinsic structural parameters is the data-driven model dynamic mode decomposition (DMD) [49, 50]. DMD is originally applied in the field of fluid mechanics, but is more frequently applied in other fields. Examples of recent applications are smart energy grids, financial markets forecasting, and different image processing techniques [2].

At this moment, research has been published where CoTracker has been adopted in facial feature tracking for ballistocardiography [8], and vessel trajectory estimation for reducing vessel-bridge collision risks [38]. The research in this paper investigates the potential application of CoTracker in the field of vision-based non-contact SHM by using an experimental setup. The final goal is to explore the dynamic full-field behaviors of a beam structure using CoTracker to facilitate SHM engineering in future applications.

1.1 Research Objective

The goal of this research is to extract modal parameters of a cantilever beam which dynamics are captured using a camera. By integrating CoTracker as a point tracking algorithm, the results of CoTracker are compared with several widely adopted edge detection methods for video-based vibration analysis. To do this, an initial validation of CoTracker is performed using data from an accelerometer located on the beam. After exploring the potential of CoTracker to track object displacement, modal analysis is performed to extract structural parameters from the cantilever beam by using a data-driven model, which is a time-embedding DMD algorithm.

1.2 Research Questions

To structure the project, several research questions have been developed. The following is a list of these questions.

1. What is the effect of camera calibration in terms of accurate displacement tracking?
2. What is the best strategy to adequately track points in a video for measuring vibrations using CoTracker?
3. What is the performance of CoTracker compared to common edge detection methods that are used for vision-based SHM measurements in terms of tracking the displacement?
4. What is the performance of CoTracker compared to common edge detection methods that are used for vision-based SHM measurements in terms of modal parameter extraction?

1.3 Research Outline

This report contains several chapters that elaborate on specific parts of the research. Chapter 2 describes the theory used in this research. Chapter 3 describes the experimental setup and the methods and tools that are used to conduct the experiments and explains the research strategies that are taken. The results of the experiments are presented in Chapter 4, as Chapter 5 provides the conclusions that can be drawn from the results. Finally, recommendations for additional future research are given in Chapter 6.

2. Preliminaries

Recent studies on contactless SHM have shown that vision-based monitoring shows great potential as an alternative to the traditional physical installations that are used for SHM. The reason is the application of low-cost and non-professional equipment that allows vision-based SHM, which can reduce the need for physical inspections, subsequently reducing the risks associated when inspections are performed in difficult-to-access locations [53, 63]. This chapter elaborates on the state-of-the-art theory that forms the fundament of today's vision-based SHM. Section 2.1 explains the edge detection methods frequently used for vision-based SHM. Section 2.2 explains the algorithm of CoTracker. Section 2.3 covers how potential distortions in an image are dealt with to acquire true displacements of the tracked points. Finally, Section 2.4 covers the theory of data analysis. Specifically, a frequency analysis is performed to extract structural parameters from a cantilever beam.

2.1 Edge Detection

A frequently applied method for vision-based vibration analysis is edge detection [5, 21, 31]. An example of an application of edge detection in structural vibration analysis is described by Javed et al. [25], who describe how vision-based SHM is successfully applied to monitoring the vibrations of a rotating cylindrical structure. This section explains the principles and differences of widely used edge detection methods. Figure 2.1 shows the beam used in this research, including the accelerometer that generates the reference data.

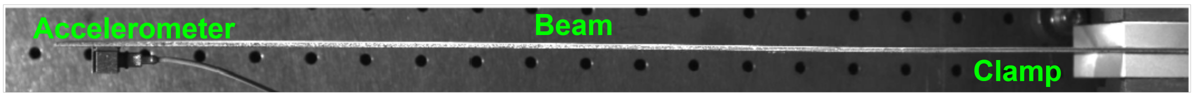
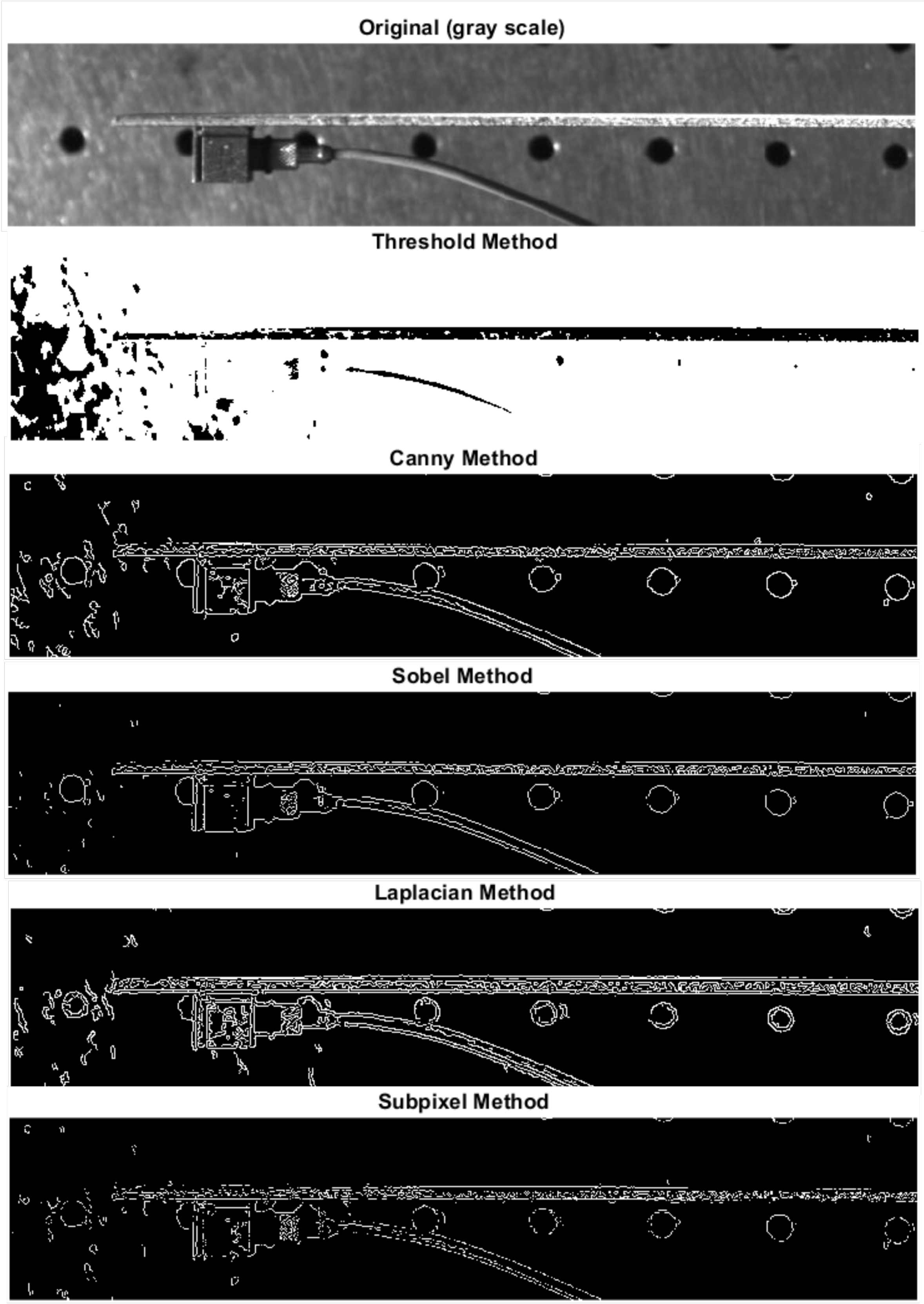


Figure 2.1: Beam. *This beam is used during the experiments. An accelerometer generates reference data. The beam is fixed on one end by a clamp.*

A widely used method for image processing is edge detection, especially utilized in the field of image feature extraction, which is the identification of characteristic shapes in an image. Common edge detection algorithms are described by different literature, but the most used algorithms are the Canny algorithm, the Sobel algorithm, the Laplacian method, and a sub-pixel algorithm [12, 28, 54]. Additionally, a threshold method is used to extract the characteristic edges in the image. Figure 2.2 shows a visualization of the differences between the edge detection algorithms. To apply the algorithms, all images are first converted from their RGB-colors to a gray-scale image, as presented in Figure 2.1. As can be seen in the image, one end of the beam is clamped and fixed, whereas the other end is free to move. An accelerometer is placed at the free end of the beam to capture reference data, which is compared with the data from CoTracker and the edge detection methods. Below, the theory of several edge detection methods is explained.



Original (gray scale)

Threshold Method

Canny Method

Sobel Method

Laplacian Method

Subpixel Method

Figure 2.2: Edge detection comparison. *Visualizing the canny method, sobel method, Laplacian method, sub-pixel method and threshold method that are used for vibration analysis.*

2.1.1 Threshold Method

The threshold method is based on the intensity of each pixel P in the original gray-scale image. Using trial-and-error and setting a variable threshold value T, only pixels with an intensity $P(x_i, y_i) < T$ are visualized. This gives that the pixels with an intensity $P(x_i, y_i) \geq T$ are displayed in black, as the pixels with an intensity $P(x_i, y_i) < T$ are displayed in white, as visualized in Figure 2.2. This results in an image in which each pixel is given as $P \in (0, 1)$, where $P = 0$ is a black pixel and $P = 1$ is a white pixel. In this example, $T = 140$, but can be changed to optimize results.

2.1.2 Canny Method

The Canny algorithm, developed by John F. Canny [7], is one of the most widely used edge detection methods [28]. The algorithm consists of five computational steps that allow for accurate edge detection, even in noisy environments.

Step 1
The first step is to remove initial noise present in the image. This is done by using the two-dimensional Gaussian filter $H(x)$ [46], which is given by

$$H(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{2.1}$$

and applied on the original image, using

$$G(x, y) = I(x, y) * H(x, y, \sigma) \tag{2.2}$$

where x and y represent the horizontal and vertical position of an edge, and σ equals the standard deviation of the filter. $I(x, y)$ represents the original image in gray-scale. The value of σ used in Matlab is $\sigma = 0.05$. The main idea of applying the filter is to smoothen high frequencies that represent sudden changes in intensity of the pixel colors. These high frequencies are the result of an identified edge, but also arise by the existence of noise. By applying the Gaussian filter, the appearance of high frequencies due to noise is limited.

Step 2
After applying the Gaussian filter to the image, the gradient of the image is computed [39]. The gradient of an image represents how the color of individual pixels changes. Tracking specific changes in the gradient allows tracking specific points in an image. The gradient is found using

$$|G| = \sqrt{G_x^2 + G_y^2} \tag{2.3}$$

where G_x is the gradient in the x-direction and G_y is the gradient in the y-direction. The horizontal and vertical gradients are obtained using an integrated version of the Sobel method, which approximates the pixels in which the gradient is maximum [47]. The direction of the gradient is given by $\theta = \arctan(G_y/G_x)$. Figure 2.3 visualizes how the gradient is determined.

Step 3
After applying step 2, thick edges remain in the processed image. However, only local maxima should be used for the edges. Consequently, step 3 suppresses the pixels that are not a local maximum by setting a threshold σ which results in thinning the initial edges

Step 4
After the first filtering steps, a second threshold method is used to eliminate potential erroneous

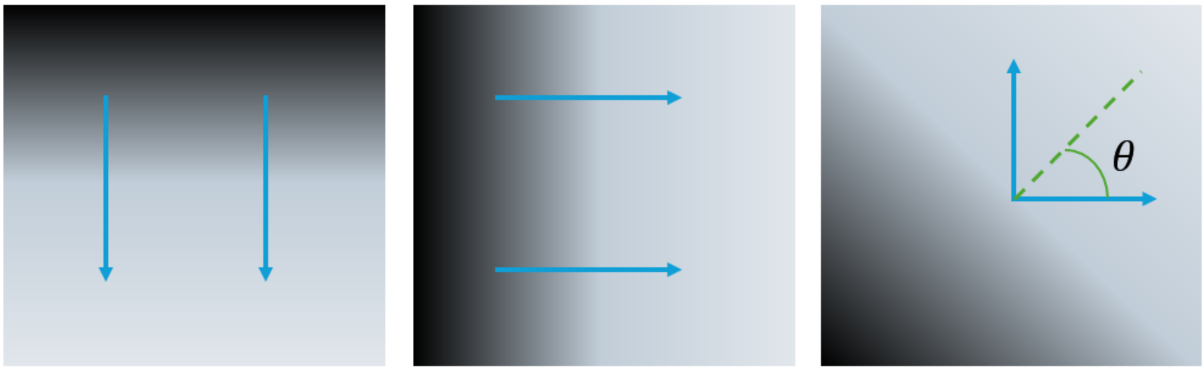


Figure 2.3: Gradient of an image. *The gradient of an image is determined by the change of the color of a specified range of pixels.*

edges. Edges that exceed the threshold are immediately disregarded, while accurate edges remain valid.

Step 5
A final filter is applied over the edges. Only the pixels that have a strong connection, implying that the difference in the gradient is not exceeding a particular threshold, remain valid as a detected edge. The Canny algorithm is therefore a combination between a Gaussian approach, which filters out potential noise, and a gradient-based approach, which is used by setting a threshold value σ.

2.1.3 Sobel Method

The Sobel method is developed by Irwin Sobel and Gary Feldman [52], and detects edges based on local maxima in the image gradient. This gradient is found using Equation 2.3. Therefore, this algorithm is different from the Canny algorithm, which is based not only on the gradient but also on the Gaussian filtering algorithm. The main idea of the Sobel algorithm is that only the local maxima of the gradient are considered edges. The Canny method uses a threshold method to distinguish between the accepted and rejected gradients.

2.1.4 Laplacian Method

The Laplacian algorithm is based on the Gaussian filtering approach similar to the Canny algorithm. However, the Laplacian algorithm computes the Laplacian of the Gaussian operator from Equation 2.1. This gives that the Laplacian of Gaussian (LoG) operator equals

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right]\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{2.4}$$

To extract the edges corresponding to the characteristic shapes in the image, this algorithm is often combined with zero-crossing based on the two-dimensional LoG function as given in Equation 2.4. This implies that the algorithm identifies points in the image where the LoG changes its sign (crossing zero), which indicates the presence of an edge or a swift change in the intensity [13].

2.1.5 Sub-pixel Method

A sub-pixel method is used to find the edges based on interpolation of the pixels in an image. Interpolation is mostly applied using different methods; bilinear interpolation, spline interpolation, and bicubic interpolation [26]. Liu et al. [37] tested the application of the different methods

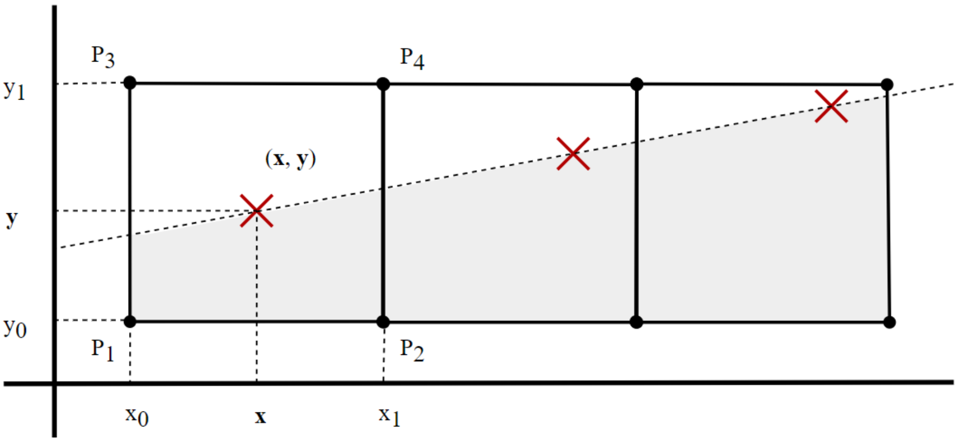


***Figure 2.4:* Bilinear interpolation.** *By finding the pixel values of the points indicated by the red cross, it is possible to determine edges that are not based on the pixel boundaries. Instead, properties are found on sub-pixel level.*

and found that the most appropriate method depends on the shape of the edges. Images in which the structures have significant bends in their edges should be interpolated using a (bi-)cubic algorithm, as images showing structures without significant bends should be linearly interpolated. Since the structures in the experiments of this research are limited to linear shapes, only bilinear interpolation is considered. Figure 2.4 gives an example of bilinear interpolation. The interpolation is done using the following procedure. Firstly, linear interpolation in the x-direction is performed using four known points that represent known points in an image, which are $P_1 = (x_0, y_0)$, $P_2 = (x_1, y_0)$, $P_3 = (x_0, y_1)$, and $P_4 = (x_1, y_1)$. Then, interpolation in the x-direction is given by

$$\begin{aligned} f(x, y_0) &= \frac{x_1 - x}{x_1 - x_0} f(P_1) + \frac{x - x_0}{x_1 - x_0} f(P_2) \\ f(x, y_1) &= \frac{x_1 - x}{x_1 - x_0} f(P_3) + \frac{x - x_0}{x_1 - x_0} f(P_4) \end{aligned} \tag{2.5}$$

After this, interpolation in the y-direction is done using

$$f(x, y) = \frac{1}{(x_1 - x_0)(y_1 - y_0)} [x_1 - x \quad x - x_0] \begin{bmatrix} f(P_1) & f(P_3) \\ f(P_2) & f(P_4) \end{bmatrix} \begin{bmatrix} y_1 - y \\ y - y_0 \end{bmatrix} \tag{2.6}$$

As bilinear interpolation is applied, the pixels are now divided by a scaling factor. This implies that it possible to more accurately determine the intensity of each sub-divided pixel, allowing to track edges beyond the initial pixel dimensions.

2.1.6 Discussion of Edge Detection Results

Figure 2.2 shows the difference between each edge detection method. As the threshold method sets pixel with an intensity lower than 140 to black and higher than 140 as white. Hence, this method highly depends on the light intensity in the room. As seen in Figure 2.2, the beam is recognized, but the sensor is not. The Canny method detects edges based on a two-dimensional Gaussian filter and a manual threshold, implying that only sudden changes in pixel intensity are marked as an edge. The Sobel method detects edges similar to the Canny method, but does not consider the Gaussian filter which limits potential noise. Therefore, images that are less strong could also be detected, but also vice verse. In Figure 2.2 it can be seen that less edges are detected. The Laplacian Method is similar to the Canny approach and adopts a combination of zero-crossing and the Laplacian of the Gaussian filter. A swift change of pixel intensity will be marked as an edge. Finally, the Sub-Pixel Method uses bilinear interpolation to determine edges within pixel-boundaries. This results in smaller edges, as visualized in Figure 2.2.

2.2 CoTracker

A novel method that combines the principles of the previous sections is the CoTracker algorithm, which is the result of a collaboration between researchers from Meta AI and the Visual Geometry Group from the University of Oxford [27]. CoTracker shows a great potential in tracking points in a video. This section explains the working principles of CoTracker to understand the main difference between the performance of this algorithm and those mentioned before. The fundamental innovation of CoTracker is the principle of tracking the points in the video based on the correlation between the tracked points. Hence, a neural tracker is integrated in the algorithm which is able to track multiple points simultaneously. Figure 2.5 visualizes the architecture of the CoTracker algorithm. The algorithm of CoTracker is explained below.

2.2.1 Initializing an Image

Compared to the 3D real-world coordinates, an image coordinate system has only two dimensions; a height H and a width W of the image. Consider a video V with a duration of length T, then the video can be described using $V = (I_t)_{t=1}^{T}$, where $I_t \in \mathbb{R}^{3\times H\times W}$ describes a frame I_t in the video, consisting of a three-dimensional RGB color frame. The main goal is to predict the tracks of N points, where the point is given by $P_i^t = (x_i^t, y_i^t) \in \mathbb{R}^2$, where $t = t_i, \ldots, T$ and $i = 1, \ldots, N$. t_i represents the time when the tracks start. For simplicity, it is taken that the measurement of each track starts at $t = 1$, indicating the start of the video.

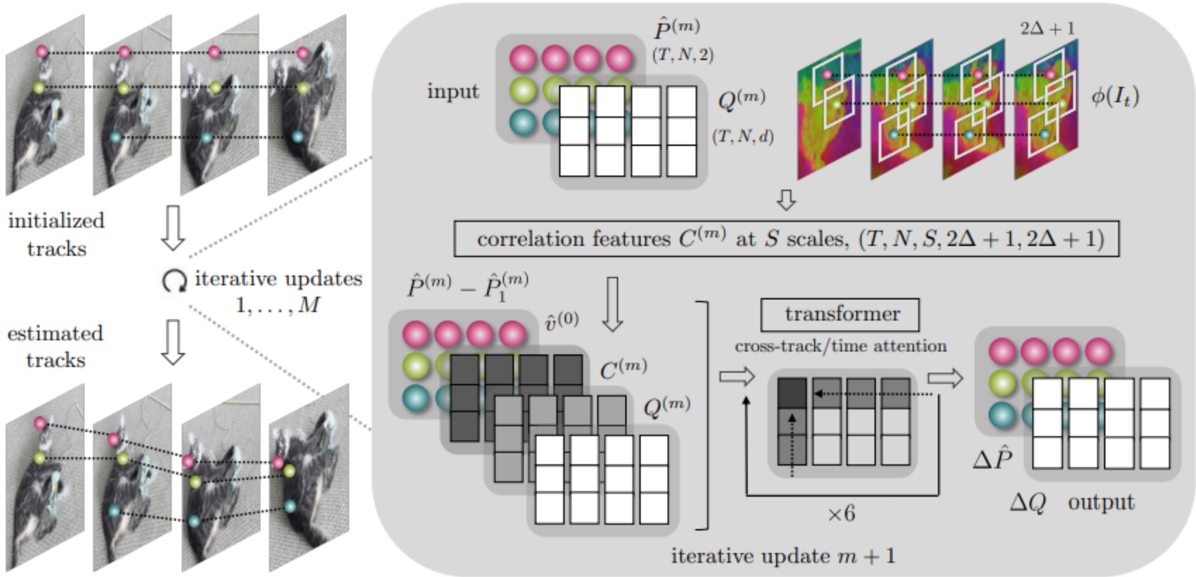


Figure 2.5: Architecture of CoTracker [27]. *Using the convolutional neural network algorithm the features for $\phi(I_t)$ are computed for all frames with sliding windows. To prepare the track features, given by Q, samples are taken from $\phi(I_t)$, considering the starting locations P.*

2.2.2 The Basic Algorithm

Using the CoTracker architecture, visualized in Figure 2.5, the algorithm starts by initializing the locations of the points $(P_{t^i}^i, t^i)_{i=1}^N$ that should be tracked on N number of tracks. These points, called queries, are chosen on a specific region-of-interest to track only desired points in the video. Subsequently, the algorithm validates if these initial points are located inside the current frame. This is done by allocating a visibility flag $v_t^i \in \{0, 1\}$, which indicates if a point in the query is occluded or included in each frame [14]. Then, the neural tracker estimates the point $\hat{P}_t^i = [(\hat{x}_t^i, \hat{y}_t^i), \hat{v}_t^i)]$ of the track and the visibility for the time t. The goal of the neural tracker is to improve the initial estimates of the track. To do this, three main features are used.

1. **Image features**
   To extract d-dimensional image features from each frame in the video a neural network is used. The image features that are extracted are given by $\phi(I_t) \in \mathbb{R}^{d \times H/k \times W/k}$. For efficiency, the resolution is reduced by $k = 4$ to ensure computational efficiency.

2. **Track features**
   The tracks are given by a vector $Q_t^i \in \mathbb{R}^d$, and are initialized by the starting positions of P. Then, the neural tracker estimates the next location of the points, and subsequently makes an estimate of the track.

3. **Correlation features**
   The main innovation compared to existing state-of-the-art algorithms that are able to track points in an image, is that CoTracker can track based on joint-tracking, implying that a correlation between each point is used to better estimate the displacement between each frame. The correlation between the points is given by $C_t^i \in \mathbb{R}^S$ and is based on an optical-flow detection, developed by Teed et al. [55]. The correlation C_t^i is found by comparing the image features $\phi(I_t)$ with the track features Q_t^i around a specific point estimate $\hat{P}_t^i$ using $[C_t^i]_{s\delta} = \langle Q_t^i, \phi_s(I_t)[\hat{P}_t^i/ks + \delta]\rangle$, where $s = 1, \ldots, S$ are the feature scales, and $\delta \in \mathbb{Z}^2$ is an integer offset from the estimated position $\hat{P}_t^i$, and is bounded by $||\delta||_\infty < \Delta$. The initial conditions in the standard algorithm are $S = 4$ and $\Delta = 3$.

4. **Iterated computations**
   The iterations are repeated M times to improve the estimates of the neural tracker. For each iteration, the estimate $\hat{P}_t^i$ is updated $m = 0, 1 \ldots, M$ times according to $\hat{P}^{(m+1)} = \hat{P}^{(m)} + \Delta\hat{P}$ whereas the estimated tracks are updated by $Q^{(m+1)} = Q^{(m)} + \Delta(Q)$. The initial points $\hat{P}^{(0)}$ and $Q^{(0)}$ are set by the initial query points.

The next step is to discover if CoTracker is also able to track vibrations in an experimental setting and allows for modal parameter extraction. This is further explained in Subsection 2.4.2 and Chapter 3.

2.3 Camera Calibration in OpenCV

A common practise in image processing is camera calibration, which is done to extract intrinsic and extrinsic parameters of the camera used for capturing the desired structure. By extracting these camera-specific parameters, it is possible to determine the degree of *radial* and *tangential* distortion in the image. By this way, it is possible to undistort the image, allowing to extract the true world-distance of the measured displacement from pixel displacement.

2.3.1 Mapping Coordinates Systems

An important factor that affects the proper calibration of the camera is the type of camera that is used. The camera calibration provides the *intrinsic* and the *extrinsic* parameters of the camera, allowing to eliminate the distortion of the image [59]. The main principle of extracting the extrinsic parameters of the camera is to acquire the true-world coordinates of an object in the image when the pixel coordinates are known. It is therefore important to understand how the coordinate mapping between the world coordinates, the camera transformation, and the pixel coordinates is established. Figure 2.6 visualizes the translation from the pixel coordinate system to the world coordinate system using intrinsic and extrinsic camera properties.

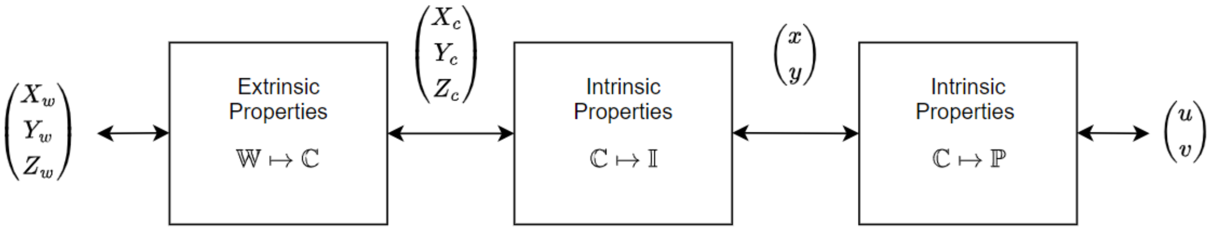

$\begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix}$
Extrinsic Properties
$\mathbb{W} \mapsto \mathbb{C}$
$\begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix}$
Intrinsic Properties
$\mathbb{C} \mapsto \mathbb{I}$
$\begin{pmatrix} x \\ y \end{pmatrix}$
Intrinsic Properties
$\mathbb{C} \mapsto \mathbb{P}$
$\begin{pmatrix} u \\ v \end{pmatrix}$

Figure 2.6: Coordinate mapping. *By using the extrinsic and extrinsic properties of a camera, true world-distances can be derived from pixel distance of the 2D-image.*

Suppose the world coordinates are given as

$$\mathbb{W} = \begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} \tag{2.7}$$

where $X_w, Y_w, Z_w \in \mathbb{R}$. Then, the world coordinates in the camera system can be represented as

$$\mathbb{C} = \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} \tag{2.8}$$

where $X_c, Y_c, Z_c \in \mathbb{R}$. The camera coordinate system is subsequently mapped in the image by the physical coordinates (x, y) with $x, y \in \mathbb{R}$. Finally, the pixel coordinate system is represented by

$$\mathbb{P} = \begin{pmatrix} u \\ v \end{pmatrix} \tag{2.9}$$

where $u, v \in \mathbb{R}^+$. The process of transforming the world coordinates $\mathbb{W}$ into the camera coordinate system is given by

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = R \begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} + t \tag{2.10}$$

where $R \in \mathbb{R}^{3\times3}$ is the rotation matrix, describing how the world-coordinate system is mapped to the camera-coordinate system. $t \in \mathbb{R}^{3\times1}$ is the translation matrix, indicating a potential shift parallel to the (X_w, Y_w, Z_w)-direction. After this transformation the camera coordinates $\mathbb{C}$ can expressed as physical coordinates on the image plane using

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} X_c/Z_c \\ Y_c/Z_c \end{pmatrix} \tag{2.11}$$

This computation transforms the 3D camera coordinates into 2D image coordinates. Since the goal of camera calibration is to extract intrinsic and the extrinsic parameters to eliminate image distortion, the distortion parameters should be used to accurately map the camera coordinate system into the physical image coordinate system.

2.3.2 Tangential and Radial Distortion

The main reason for the distorted images is explained by tangential and radial distortion [4]. Radial distortion deforms the image in such a way that straight lines appear to be curved, as tangential distortion is affected by the angle of the camera onto the object. Figure 2.7 shows how distortion affects an image. *Radial* distortion can be described by

$$\begin{pmatrix} x_{distorted} \\ y_{distorted} \end{pmatrix} = \begin{pmatrix} x(1 + k_1 r^2 + k_2 r^4 + \cdots + k_n r^{2n}) \\ y(1 + k_1 r^2 + k_2 r^4 + \cdots + k_n r^{2n}) \end{pmatrix} \tag{2.12}$$

where k_n represents the n^{th} radial distortion coefficient. *Tangential* distortion is given by

$$\begin{pmatrix} x_{distorted} \\ y_{distorted} \end{pmatrix} = \begin{pmatrix} x + [2p_1 xy + p_2(r^2 + 2x^2)] \\ y + [p_1(r^2 + 2y^2) + 2p_2 xy] \end{pmatrix} \tag{2.13}$$

In these equations, $k_1, k_2, k_3 \in \mathbb{R}$ represents the radial distortion coefficients and $p_1, p_2 \in \mathbb{R}$ represent the tangential distortion coefficients, as $r \in \mathbb{R}$ is given as $r = \sqrt{(x_d - x_c)^2 + (y_d - y_c)^2}$. Here, (x_d, y_d) is given in Equation 2.14, and (x_c, y_c) are the coordinates of the center of distortion [11]. Given the coordinate system transformation in Equation 2.11, it is possible to determine the transformation of the camera coordinate system to the physical coordinate system on the image plane using the distortion equations in Equation 2.12 and Equation 2.13. Let the coordinate system (x_d, y_d) describe the physical coordinate system on the image plane, and taking that OpenCV uses $n = 2$ for the number of radial distortion coefficients, (x_d, y_d) is given by

$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} = (1 + k_1 r^2 + k_2 r^4) \times \begin{pmatrix} X_c/Z_c \\ Y_c/Z_c \end{pmatrix} + \begin{pmatrix} 2p_1 xy + p_2(r^2 + 2x^2) \\ p_1(r^2 + 2y^2) + 2p_2 xy \end{pmatrix} \tag{2.14}$$

The final step for the transformation of the world coordinate system to the pixel coordinate system is to understand how the physical coordinate system on the image plane ($\mathbb{I}$) is transformed to the pixel coordinate system ($\mathbb{P}$). This happens in the following procedure. Take the physical distorted coordinate system (x_d, y_d), then the pixel coordinate system is represented as

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} f_x x_d + u_0 \\ f_y y_d + v_0 \end{pmatrix} \tag{2.15}$$

In this equation, f_x and f_y represent the focal length of the camera and $(u_0, v_0)^T$ is the base point of the image. The focal length of the camera f is given by

$$\begin{cases} f_x = s \cdot f/\Delta x \\ f_y = f/\Delta y \end{cases}$$

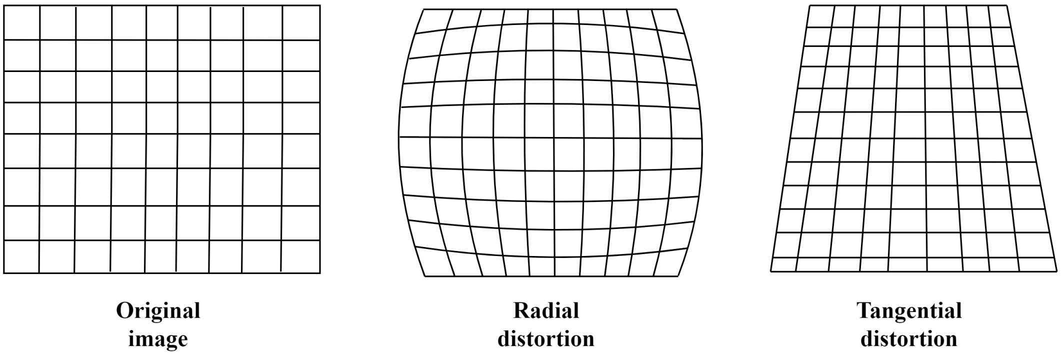


Figure 2.7: Distortion. *The theoretical grid of an original image is shown on the left. Radial distortion causes straight edges to appear bent, while tangential distortion makes some points in the image seem farther or nearer than they actually are.*

Δx is the distance in mm/pixel between two pixel points on the image in horizontal direction. The same reasoning is valid for Δy in vertical direction. s is a scale factor that compensates for uncertainties that arise from sampling in the horizontal direction in computer-generated images during image processing. Finally, it is possible to determine the immediate mapping of the world coordinate system to the pixel coordinate system, and vice verse, by using

$$sp = A(R\,|\,t)P \tag{2.16}$$

where p equals the pixel coordinate system $p = (u, v)^T$, A represents a matrix of intrinsic parameters, and $(R\,|\,t)$ is the matrix containing the extrinsic parameters, s represents a scaling factor to adapt to uncertainties, and P are the homogeneous coordinates of the space points $P = (X, Y, Z, 1)^T$. Understanding how coordinate system mapping is computed allows one to extract the true displacement in real-world distances based on the pixel displacement found in image analysis.

2.4 Frequency Analysis

In order to compare the performance of the edge detection methods from Section 2.1 with the performance of CoTracker, it is required to do vibration analysis. This section explains the principles behind the methods that are used to analyze the data from the experiments.

2.4.1 The Fast Fourier Transform (FFT)

A method that is frequently used for digital signal processing is the Fast Fourier Transform (FFT), which plays a crucial role in applications such as radar technology, seismic data processing, medical electrical technologies, but also telecommunications and image processing [17, 32]. The FFT computes the Direct Fourier Transform (DFT) or the inverse DFT (IDFT). The main principle of the Fourier transformation is to map data, which is obtained in the time domain, to the frequency domain.

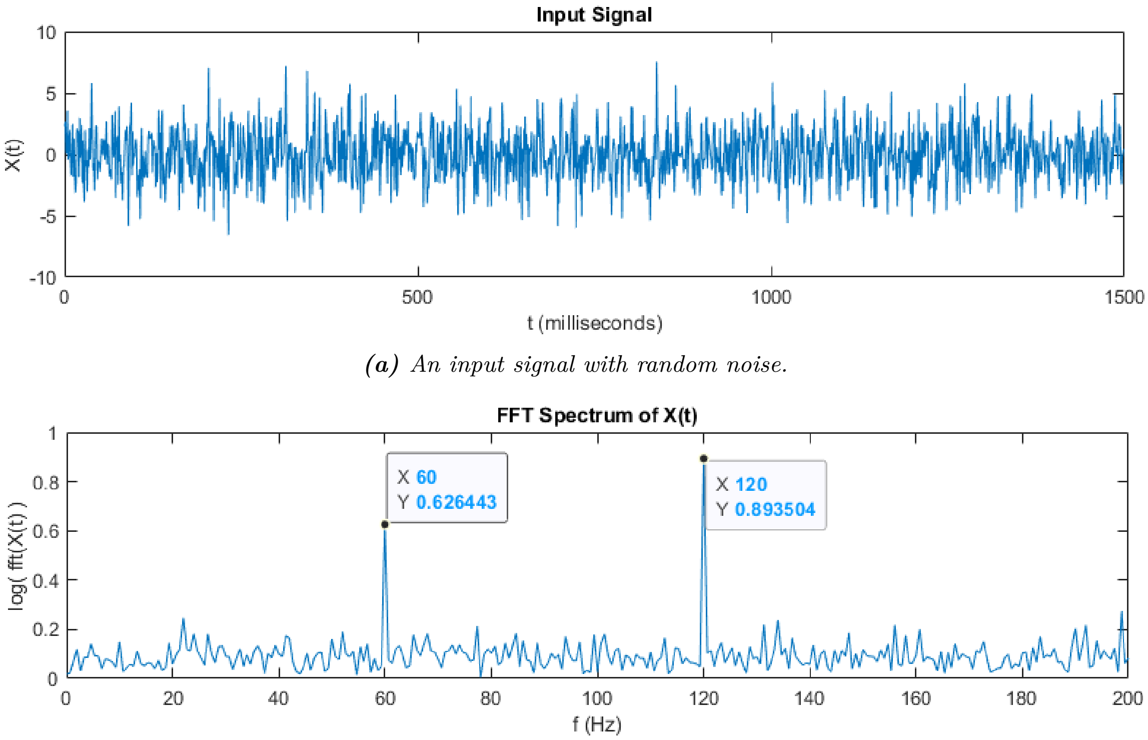
The DFT algorithm computes a frequency range of the same length as the number of samples that have been taken during an experiment. The sampling period, which is the time between each sample, is given by

$$\Delta t = 1/f_s \tag{2.17}$$

where f_s represents the number of samples that are taken each second, also known as sampling frequency or sampling rate in Hertz. Suppose that N represents the number of samples that have been taken during an experiment, than the DFT is given by

$$X_k = \sum_{i=0}^{N-1} x_m e^{-i2\pi km/n} \tag{2.18}$$

where X_k is the amplitude of the k^{th} frequency component in the signal, x_m is the discrete-time signal, and $e^{-i2\pi km/n}$ is a complex exponential function, which includes the phase shift and frequency of the signal that are found in X_k. A theoretical example that illustrates the FFT is given in Figure 2.8. This example illustrates that time-domain system parameters can be extracted using the FFT. An important consideration is the sample frequency (f_s). By increasing the sampling frequency, the computational efforts increase, but allows to extract higher natural frequencies. According to the Nyquist Sampling Theorem, the sampling frequency should be at least twice the maximum natural frequency that is to be observed, i.e. $f_s \geq 2 \times f_{n,max}$, where $f_{n,max}$ is the maximum frequency that can be observed.



(a) An input signal with random noise.

(b) The FFT, indicating two dominant frequencies in the input signal.

Figure 2.8: Example of the FFT. *Assume a signal given by $x(t) = 0.3\sin(2\pi 60t) + 0.8\sin(2\pi 120t)$ which is disrupted by random noise. The sampling frequency is $f_s = 1000$ Hz. This implies that the sampling period is given as before by $\Delta t = 1/f_s$. The length of the signal is $L = 1500$ samples. Then, by plotting the logarithmic of the FFT it is evident that the first and second natural frequencies are found at 60 Hz and 120 Hz, respectively, as given in the input signal $x(t)$.*

2.4.2 Modal Parameter Extraction using Time-Embedding DMD

Frequency analysis allows extracting modal parameters that belong to a dynamic structure. This research focuses on the extraction of three modal parameters. That is, the natural frequency (f_n), the damping ratio (ζ_n), and the mode shape (Φ_n). Here, n depicts the degree of freedom (DOF). The extraction of modal parameters is typically done by using frequency response functions (FRFs), which are given in the frequency domain, or by analyzing time responses [42, 61]. Methods in the frequency domain are widely preferred over methods in the time domain, since gathering the number of data samples that are required for systems with a low natural frequency become a practical problem [48].

To find the modal parameters of the dynamic system, a method based on data-driven experimental modal analysis is used by implementing a time-embedding dynamic mode decomposition (DMD) algorithm [49]. The standard algorithm of the DMD method is as follows. Consider a discrete-time signal that represents the displacement of a specific point on a structure, which represents the dynamic system. The data set representing this discrete-time signal is given by $x(t) \in \mathbb{R}^m$, where m is the sample size. Then, a snapshot is taken from $x(t)$, given by

$$X \triangleq \begin{pmatrix} x_1, \dots, x_{m-n} \\ x_2, \dots, x_{m-n+1} \\ x_3, \dots, x_{m-n+2} \\ \vdots \\ x_n, \dots, x_{m-1} \end{pmatrix} \tag{2.19}$$

for $t_{j+1} = t_j + \Delta t$. Here, $x_j \triangleq x(t_j)$ for $j = 1, \ldots, n$. Then there exists also a matrix

$$Y \triangleq \begin{pmatrix} x_2, \ldots, x_{m-n+1} \\ x_3, \ldots, x_{m-n+2} \\ x_4, \ldots, x_{m-n+3} \\ \vdots \\ x_{n+1}, \ldots, x_m \end{pmatrix} \tag{2.20}$$

where n is the number of time-shifted snapshots that are used. It is assumed that there is a linear transformation between x_j and x_{j+1}, which is represented by matrix A as

$$x_{j+1} = Ax_j \tag{2.21}$$

Consequently, it can be concluded that

$$Y = AX \tag{2.22}$$

from which can be derived that

$$A = YX^{\dagger} \tag{2.23}$$

$X^{\dagger}$ represents the Moore-Penrose pseudo inverse matrix of the snapshot matrix X. To extract the natural frequency (f_n) and damping ratio (ζ) using DMD, singular value decomposition (SVD) is applied to the snapshot matrix X. By applying SVD, three matrices are extracted by using the fact that $X = USV^*$, where * indicates the Hermitian transpose of a matrix. From the three matrices $U \in \mathbb{C}^{m\times r}$, $S \in \mathbb{R}^{r\times r}$, and $V \in \mathbb{C}^{n\times r}$, where r is the rank of X, matrix U used to extract proper orthogonal decomposition (POD) modes [56].

The next step is to extract the DMD eigenvalues of a projected matrix A onto the POD modes. To do this, the projected matrix A is denoted as

$$\tilde{A} \triangleq U^*AU = U^*YVS^{-1}U^*U = U^*YVS^{-1} \tag{2.24}$$

Now, the DMD eigenvalues can be found by computing the eigenvalues of $\tilde{A}$, i.e. $\tilde{A}w = \mu w$, where μ indicates the DMD eigenvalue and w represents its eigenvector. The exact DMD mode that corresponds to μ is defined as $\Phi = YVS^{-1}w$.

To extract the natural frequencies f_n and the damping ratios ζ_n from the discrete-time to the continuous time, the following is assumed:

- Take a continuous time-system $\dot{\mathrm{x}} = \mathcal{A}\mathrm{x}$, where $\mathcal{A}$ is the continuous representation of the discrete-time variable A, which is given in Equation 2.21.
- The solution of the continuous system is given by [48]

$$\mathrm{x}(t) = \sum_{i=1}^{m} e^{s_i t}\varphi_i p_{0,i} \tag{2.25}$$

  in which s_i is the vector of eigenvalues and φ_i contains the eigenvectors of $\mathcal{A}$. $p_{0,i}$ contains the modal coordinates $p_0 = \Psi \mathrm{x}_0$, in which $\Psi = (\varphi_1, \ldots, \varphi_m)$.
- The solution of the discrete-time signal is represented by

$$\mathrm{x}_k = \sum_{i=1}^{m} \mu_i^k \Phi_i q_{0,i} \tag{2.26}$$

  in which $q_{0,i}$ equals the modal coordinate at t_j (discrete time.

- Assume $p_{0,i} \approx q0, i$, and $\varphi \approx \Phi$.

Under the assumptions above, it can be seen that the difference between Equation 2.25 and Equation 2.26 results in the following equation:

$$e^{s_i kt} = \mu_i^k \tag{2.27}$$

Now, taking $t = t_k = k\Delta t$ and that $s_i = \log(\mu_i)/\Delta t$ where s_i represents the eigenvalue of the continuous-time signal obtained from the same (discrete-time) system, given in Equation 2.21. This eigenvalue can be found using the DMD eigenvalue μ and the sample period Δt.

$$s_i = -\zeta_i \omega_i \pm j w_i \sqrt{1 - \zeta_i^2} \tag{2.28}$$

where ω_i is the natural frequency and ζ_i is the modal damping ratio. Then, the undamped natural frequency f_n and damping ratio ζ_i can be found using

$$\begin{aligned} f_i &= |s_i|/2\pi \\ \zeta_i &= Re(s_i)/|s_i| \end{aligned} \tag{2.29}$$

The preceding literature outlines the traditional DMD algorithm, which has demonstrated its effectiveness across various domains, such as in the examination of flows around high-speed trains [56], as well as in the analysis of laminar axisymmetric jet flows [50]. Nevertheless, when utilizing data with significant noise, DMD might fail to yield meaningful results. Consequently, fake and numerical eigenvalues induced by noise are eliminated. The subsequent section details this procedure.

As indicated by Equation 2.27, the connection between the continuous and discrete time signals is represented by the vector of DMD eigenvalues μ_i and the continuous time eigenvalues s_i. Thus, by recording the index k of the eigenvalues μ_i that lie within a designated tolerance ϵ of the natural frequencies f_n^i, i.e., $\epsilon_1 \leq f_n^i \leq \epsilon_2$ where $i \in \mathbb{N}$ denotes the i^{th} natural frequency identified using FFT, the corresponding damping ratios ζ_i can be determined. This leads to the identification of DMD frequencies and damping ratios that align with the natural frequencies obtained via FFT. By subsequently locating the eigenvectors w linked to the index k, the corresponding mode shapes Φ for each natural frequency can be calculated using $\Phi_i = YVS^{-1}w_i$. This enhancement is depicted below.

Eliminating noise induced eigenvalues.

1. Compute the standard snapshot matrix X and time-shifted matrix Y from the data set $x(t) \in \mathbb{R}^m$ for a degree of freedom (DOF) n.

$$X \triangleq \begin{pmatrix} x_1, \ldots, x_{m-n} \\ x_2, \ldots, x_{m-n+1} \\ x_3, \ldots, x_{m-n+2} \\ \vdots \\ x_n, \ldots, x_{m-1} \end{pmatrix}$$

$$Y \triangleq \begin{pmatrix} x_2, \ldots, x_{m-n+1} \\ x_3, \ldots, x_{m-n+2} \\ x_4, \ldots, x_{m-n+3} \\ \vdots \\ x_{n+1}, \ldots, x_m \end{pmatrix}$$

2. Compute the SVD of X for rank r, and set $r < m$ manually to remove relative insignificant data (truncated DMD)
$$X = USV^*$$
from which $U_{trunc} = U^{n^2 \times r}$, $S_{trunc} = S^{r \times r}$, and $V_{trunc} = V^{m \times r}$ are found.
3. Define the standard projected matrix using the truncated SVD parameters
$$\tilde{A} \triangleq U^* Y V S_{-1}$$
4. Compute the eigenvectors and eigenvalues of $\tilde{A}$ using
$$\tilde{A}w = \mu w$$
5. Compute the continuous time system eigenvalues using
$$s_i = \log(\mu_i)/\Delta t$$
and find the corresponding DMD natural frequencies
$$f_i = |s_i|/2\pi$$
6. Extract dominant natural frequencies f_n^i using the FFT, where $i \in \mathbb{N}$ denotes the i$^{\text{th}}$ natural frequency, resulting in $f_n = [f_n^1, \dots, f_n^i]$.
7. For each DMD frequency f_i, store the index k which satisfies
$$\epsilon_1 f_n^i \leq f_i \leq \epsilon_2 f_n^i \tag{2.30}$$
where $\epsilon = [\epsilon_1 \quad \epsilon_2]$ gives the upper and lower boundary for the error margin.
8. Find all DMD natural frequencies and damping ratios that correspond to the index k
$$f_{i,filtered}^k = f_i(k) \tag{2.31}$$
$$\zeta_{i,filtered}^k = \zeta_i(k) \tag{2.32}$$

This improvement only finds the DMD damping ratios and natural frequencies that are found within a specific error margin ϵ from the dominant frequencies f_n found using the FFT.

2.4.3 Natural Frequency and Mode Shape: Analytical Solution

To find the theoretical solution for the natural frequency and the mode shape of a cantilever beam, an analytical approach is used [43]. The analytical natural frequency is given by

$$f_i = \frac{\lambda_i^2}{2\pi L_c^2}\sqrt{\frac{EI}{m^*}} \tag{2.33}$$

where i indicates an integer representing the mode, λ_i is the eigenvalue of mode i, β is dimensionless and represents boundary conditions, L_c equals the length of the beam, E is the modulus of

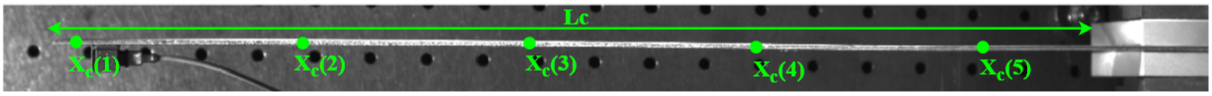


Figure 2.9: Physical Interpretation. *This figure illustrates the physical interpretation of L_c and x_c.*

elasticity, I represents the moment of inertia, and m^* is the mass per unit length. The mode shape is given by

$$y_i\left(\frac{x_c}{L_c}\right) = \cosh\left(\frac{\lambda_i x_c}{L_c}\right) - \cos\left(\frac{\lambda_i x_c}{L_c}\right) - \beta_i\left[\sinh\left(\frac{\lambda_i x_c}{L_c}\right) - \sin\left(\frac{\lambda_i x_c}{L_c}\right)\right] \tag{2.34}$$

in which x_c represents arbitrary points on the beam that are used for the modal analysis. Figure 2.9 shows the physical interpretation of the extrinsic variables L_c and x_c. To obtain the best results for the modal shape, the points x_c should be normally distributed over the beam.

3. Methods and Tools

This chapter describes the tools and methods that are used to generate the desired data from the experiments. Firstly, the experimental setup is discussed and the procedure to execute the experiment is explained. After this, different experimental strategies are covered.

3.1 Experimental Setup

In this research, only one experimental setup is considered which is visualized in Figure 3.1. A cantilever beam of dimensions 420 mm in length, 2 mm in width, and 30 mm in height is used to measure its vibrations. A signal is excited in different experiments using a shaker. To design, generate, and control the excitation on the beam, dSPACE and Simulink are used. To control the light of the environment, a lamp is used. Figure 3.1b visualizes the accelerometer used to obtain reference data to capture the dynamics of the beam. Figure 3.1c shows the high-speed camera that captures images of the dynamics of the beam.

3.2 Conducting the Experiment

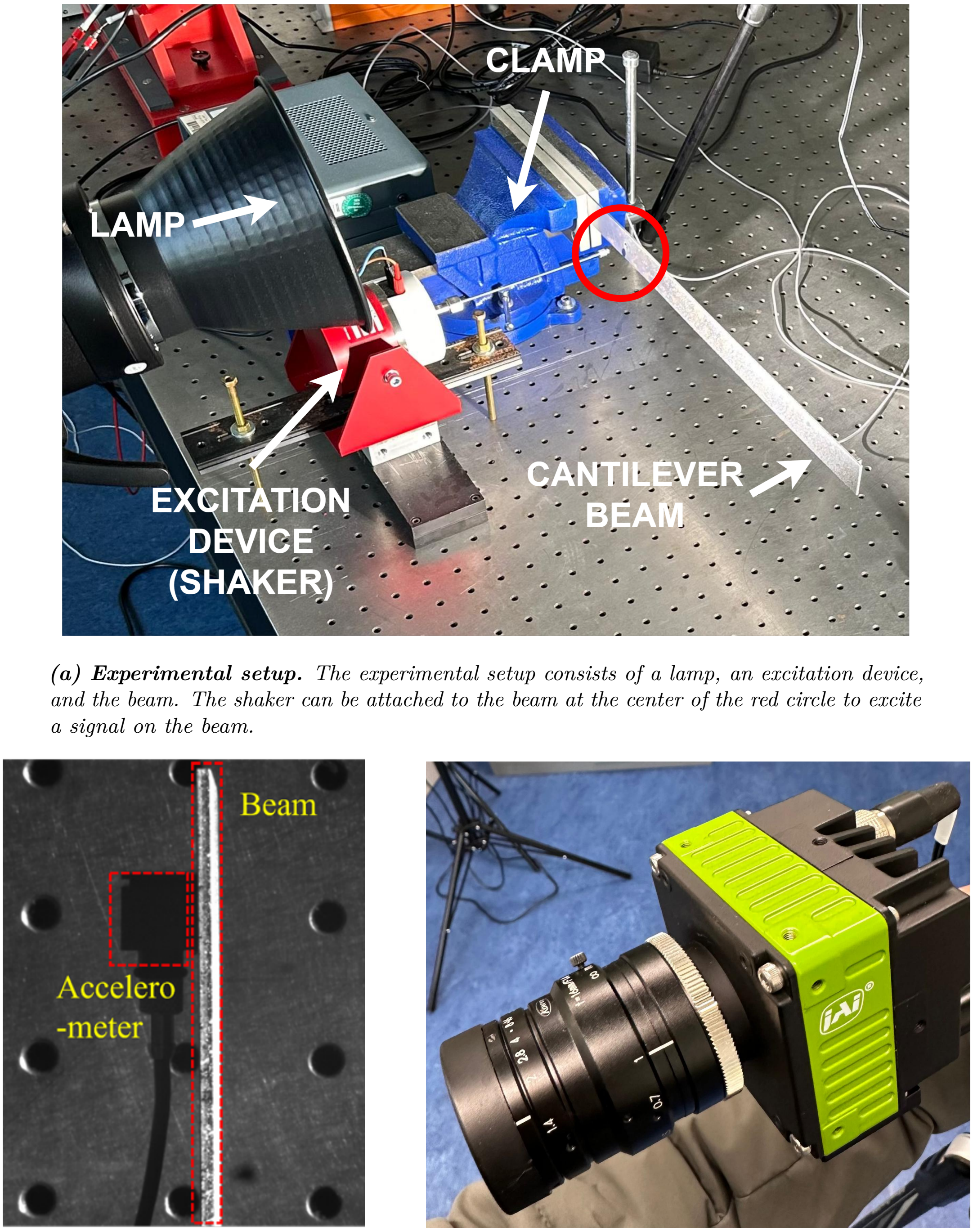
To explore the potential of CoTracker as a camera-based tracking algorithm for vibration measurements, multiple experiments are carried out using different strategies. Each strategy has the following procedure: (1) position the high-speed camera and determine the region of interest (ROI) on the beam, (2) conduct camera calibration to eliminate potential distortions in the images, (3) determine the signal that is excited on the cantilever beam, (4) determine the position of the accelerometer on the beam, (5) conduct the experiment and measure vibrations with CoTracker's algorithm and edge detection, (6) analyze the vibrations using DMD for modal parameter identification, and (7) compare and validate results.

3.2.1 Camera Calibration

To obtain the real-world coordinates from an image coordinate system, the camera calibration tool by Matlab® Camera Calibrator is used to extract the extrinsic and intrinsic camera parameter values. From these camera parameters, it is possible to obtain the real-world coordinate system from the image coordinate system, as explained in Section 2.3. The distortions in the image are generated by capturing images of a chessboard in different orientations, respectively to the camera, where the distance from the camera lens to the chessboard remains constant and equals the distance from the camera lens to the beam.

3.2.2 Generating the Input Signal

To apply a force to the beam, various techniques can be employed. For instance, one can excite a fixed frequency with a constant amplitude, or excite a fixed frequency with a varying amplitude. Moreover, a varying frequency with a constant amplitude can also be utilized. A different technique is to excite the beam with a single impact, generated by a modal hammer. Figure 3.2 shows this hammer. For modal analysis, each excitation provides different modal insights. For instance, a sinusoidal excitation is used to detect mode shapes at a particular frequency. A hammer impact excitation is widely used for modal shape extraction for multiple frequencies.



(a) ***Experimental setup.*** *The experimental setup consists of a lamp, an excitation device, and the beam. The shaker can be attached to the beam at the center of the red circle to excite a signal on the beam.*

(b) ***Accelerometer on the beam****. Accurately measures the output signal resulting from the excitation.*

(c) ***High-Speed Camera.*** *The camera is used to capture the images of the cantilever beam.*

Figure 3.1: Experimental setup. *(a) shows the experimental setup used in the project. (b) shows the position of the accelerometer on the cantilever beam. (c) visualized the high-speed camera used to capture the images that are used to track the vibrations.*

Figure 3.2: Impact Hammer. *A representation of the impact hammer.*

For the scope of this project, two excitation techniques are used. The first option is single harmonic excitation with a constant amplitude. For this experiment, a signal is created in Simulink. To excite the signal on the beam, the shaker is attached to the beam and generates a force, causing the beam to vibrate. This is illustrated in Figure 3.1a. The second signal is excited using a single impact force, generated by a modal hammer. This results in a vibration with an initial high amplitude that slowly starts to fade. The input data from the hammer impact test are used in two ways. First, the results of CoTracker are compared with data from the accelerometer, which is the validation of the potential of CoTracker to track points in an experimental setup. Secondly, the measured vibrations resulting from the hammer impact allow for the extraction of natural frequencies, which are subsequently used for modal parameter extraction using DMD.

3.2.3 Segmentation Mapping for Point Tracking

To validate the performance of CoTracker, different strategies are used to choose the points that should be tracked. In other words, based on different initial segmentation maps the performance

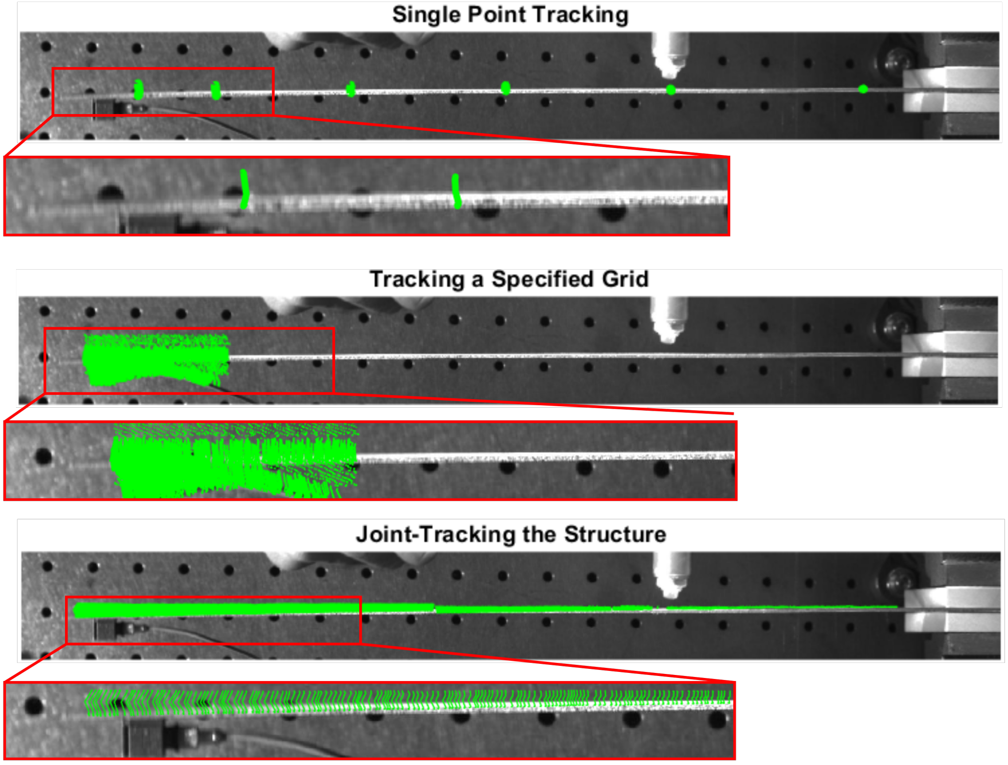


Figure 3.3: Segmentation Strategies. *Different strategies are evaluated for the best performance in tracking displacement.*

of CoTracker can be analyzed for each segment. This is done in three ways, as illustrated in Figure 3.3.

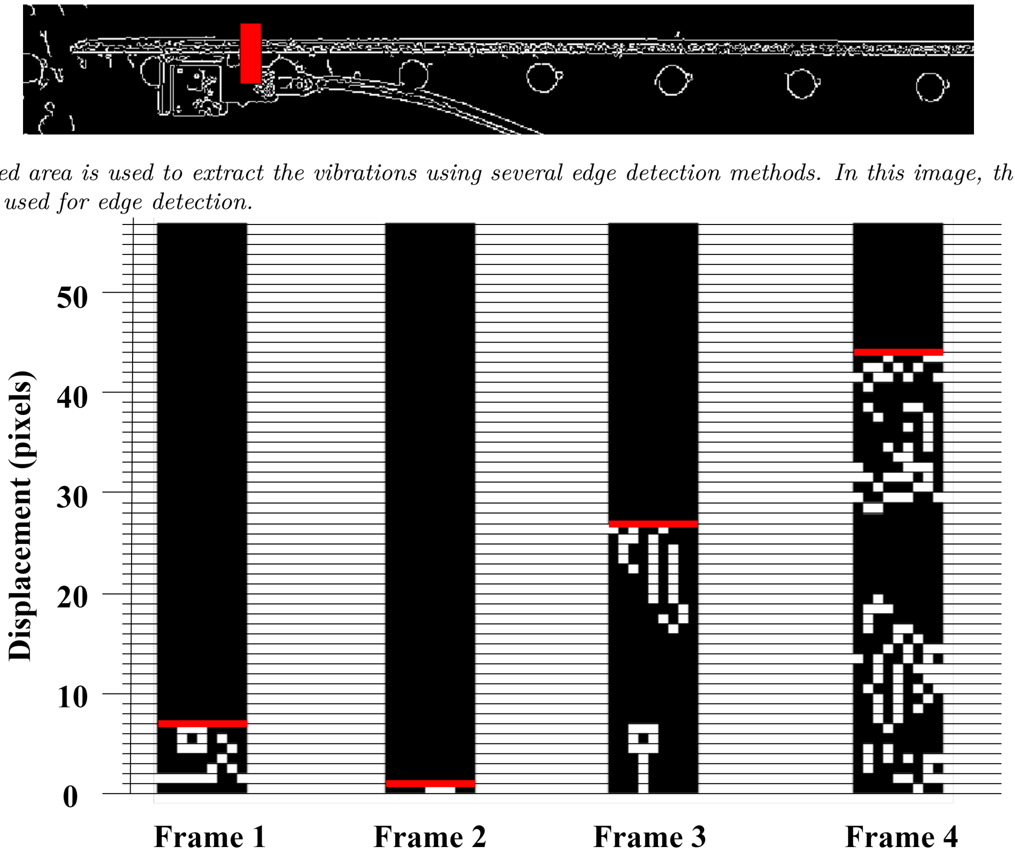
1. *Single Point Tracking*
   This method enables precise selection of points within the image to be analyzed. By choosing specific individual points, the dynamics of the beam is captured partially.

2. *Tracking a Grid*
   Joint-tracking points within a particular region captures the dynamics of this specific region, but also captures both the points that must stay physically stationary and those that are physically dynamic because of structural movement.

3. *Joint-Tracking*
   By monitoring points across the whole structure simultaneously, one can capture the complete dynamics, offering an advantage over the other two methods. However, this method is also computationally intensive when a large number of points are chosen. For this reason, it is worthwhile adapting the segmentation strategy to the desired ROI.

3.2.4 Extracting Vibrations using Edge Detection

Edge detection is widely used to extract vibrations from videos or images of dynamic systems [23, 24, 44]. The basic principle is to track the displacement of an individual pixel by identifying



(a) *The red area is used to extract the vibrations using several edge detection methods. In this image, the Canny method is used for edge detection.*

(b) *Visualization of how the displacement of the edge is found using different frames.*

Figure 3.4: Extracting displacement using edge detection. *By finding the highest point from each frame in the ROI, it is possible to determine the displacement of the edge.*

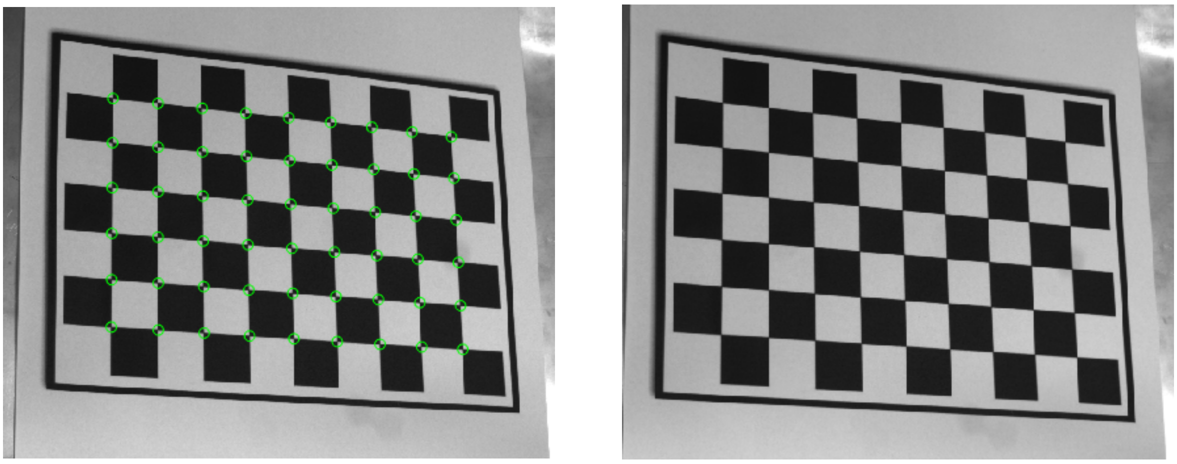
it in every frame of an image. This involves choosing a ROI in the original image that can encompass all vibrations on the structure. Next, the maximum height of the edge within the ROI is identified to determine the edge displacement in each frame. Figure 3.4, illustrated on the following page, demonstrates this process. In four frames, the displacements are 7, 1, 27, and 44 pixels, respectively.

4. Results

This chapter presents the results found in the laboratory experiments. In Section 4.1, the results of camera calibration are discussed. In Section 4.2, the results of CoTracker's performance are shown. These results show the potential of CoTracker as an image processing algorithm to extract vibrations from a dynamic structure in two different experiments. Furthermore, these findings determine the most effective strategy for the segmentation map within the experimental arrangement. This optimal strategy is applied to produce the data discussed in the following section, Section 4.3, where the outcomes of the DMD analysis are shown. In this section, the performance of the standard DMD algorithm is contrasted with that of the filtered algorithm. The derived parameters are presented and confirmed using an analytical method.

4.1 The Effect of Camera Calibration

The first step of the experimental strategy is to position the high-speed camera and determine the ROI on the beam. The second step is to apply camera calibration to extract real-world displacement from pixel displacement in, respectively, the world-coordinate system and the camera-coordinate system. Figure 4.1 shows the effect of applying camera calibration. The left image shows how the checkerboard is initialized. The green points indicate the corners of the squares on the board, in which the location of each point is determined. After this, the calibration is executed using Matlab® Camera Calibrator. For an accurate finding of the distortion coefficients, at least 10 images should be used. For calibration in this research, 196 images were used, of which 8 images were rejected. Therefore, by analyzing 188 images the distortion coefficients are found.



***(a)** Initialization of the calibration using Matlab® Camera Calibrator.*

***(b)** The undistorted image using the extracted distortion parameters.*

***Figure 4.1: Result of Camera Calibration.** The figure on the left shows the initialization of the calibration, in which the corners of the checkerboard are identified. The figure on the right shows the undistorted image.*

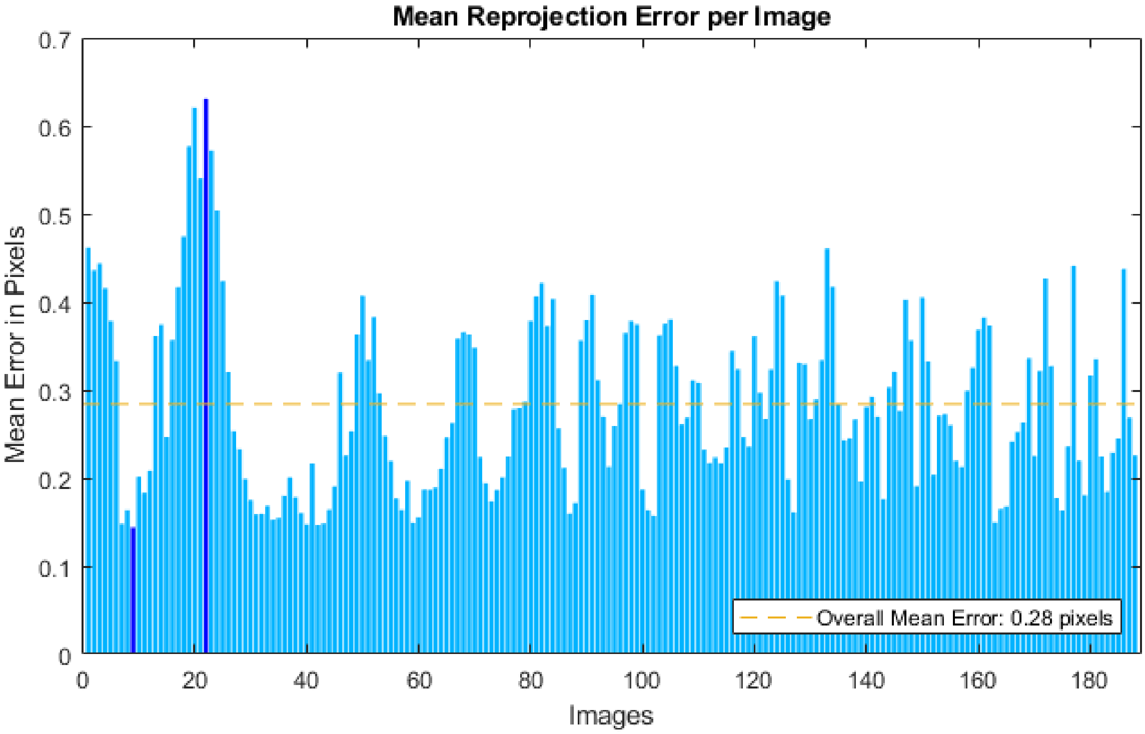


Figure 4.2: Mean Projection Error. *The mean projection error of all images that are used for calibration. The average MPE is 0.28 pixels.*

4.1.1 Mean Projection Error

To determine if the calibration is successfully performed, the mean projection error (MPE) is used. This gives the MPE for all images that are used for the calibration. The result gives an average MPE of 0.28 pixels. To ensure accurate calibration results, it should be true that the average MPE < 0.5 pixels, which indicates that the projection error remains within 50% of the pixel dimensions. As the average MPE $= 0.28 < 0.5$ pixels, it can be concluded that the calibration for this experiment is sufficiently accurate.

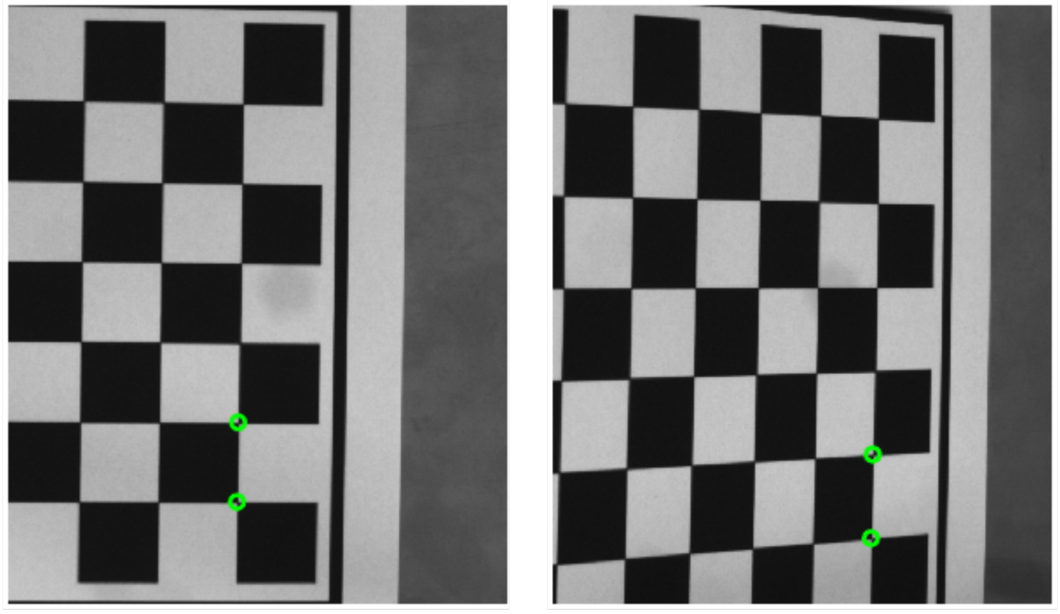
4.1.2 Extracting True Displacement

The next step is to extract the true pixel dimensions from the pixel-coordinate system to the world-coordinate system. This results in the true displacement of the beam, as opposed to the pixel displacement. To illustrate this procedure, the results of two images, image 9 and 18, as highlighted in Figure 4.2, are used and explained below. The MPEs are 0.144 and 0.475 for images 9 and 18, respectively.

As the input of camera calibration is an image, the analysis starts in the image-coordinate system $\mathbb{I}$. Figure 4.3 visualizes two points for both image 9 and image 18. These points are used to determine the distance between the coordinates in the world-coordinate system and the image-coordinate system. Table 4.1 gives the coordinates (x, y) in the image-coordinate system for the selected images.

Table 4.1: Image-Coordinate System.

Image	P_1	P_2
9	$(231.4, 619, 2)$	$(230.2, 699.1)$
18	$(321.7, 651.5)$	$(320.0, 735.6)$



(a) Image 9 **(b) Image 18**

Figure 4.3: Extracted Pixel Size. *The images in (a) and (b) show the same points that are used for extracting pixel size. Image 9 has an MPE of 0.14 pixels, as image 18 has an MPE of 0.47.*

Then using the intrinsic and extrinsic properties that are extracted using camera calibration, it is possible to find the distortion coefficients k described in Equation 2.14. Figure 4.3 presents the points given in Table 4.1. Image 9 has a projection error of 0.14, whereas image 18 has a projection error of 0.47. A measurement with a centimeter tape gives that the actual size of the squares is 17.5 mm. After calibration, it is found that the extracted square size is 17.50007 mm and 17.24176 mm for images 9 and 18, respectively. This implies that the error of the extracted square size in images with a relative high projection error approximates 1,47%, whereas images with a low projection error have no error in extracting the square size. To extract the pixel size, the number of pixels between the two green points is divided by the extracted square size. As the number of pixels between the two points equals 80 and 85 for image 9 and 18, the actual pixel size are found to be 0.219 mm and 0.203 mm, respectively. This implies a difference of 0.016 mm when comparing the pixel size of images with a relative low and high MPE. In other words, the maximum displacement error caused by camera distortion is approximately 0.016 mm.

4.2 Exploring CoTracker's Potential

To find the potential of CoTracker as a full-field vision-based SHM algorithm, two different experiments are used. These experiments are commonly used for modal vibration analysis and consist of an excitation with a fixed amplitude and fixed frequency, and a hammer test. In the first experiment, the data of an accelerometer and CoTracker are compared to validate if CoTracker is capable of capturing the dynamics of the beam. The second experiment is a hammer impact on the beam, causing a single input force that causes an initial displacement and acceleration of the beam. After this, the damping characteristics of the beam cause the vibration to dampen, until the beam is back in its initial position. Additionally, the second experiment validates three segmentation methods to capture dynamics using CoTracker. The optimal segmentation method is evaluated and compared with data from the accelerometer.

4.2.1 Fixed Frequency

This experiment involves an excitation with a constant amplitude and a steady frequency of 20 Hz. CoTracker analyzes 500 points on the beam for 3.3 seconds. The experiment was captured using the high-speed camera, with a frame rate of 150 fps. Figure 4.4 shows a time-domain comparison between the accelerometer data and the acceleration calculated from the displacement determined by CoTracker. The results indicate a strong agreement between the accelerometer and CoTracker data throughout the duration of the experiment. To qualitatively assess these findings, the correlation, difference, and error are calculated. It is found that the correlation between the data of the accelerometer and CoTracker have a correlation of 0.9918, which indicates that they represent nearly an identical acceleration. Secondly, the mean of the difference is -0.0328 with a standard deviation of 0.4189, which confirms that the two data sets are in close proximity to each other.

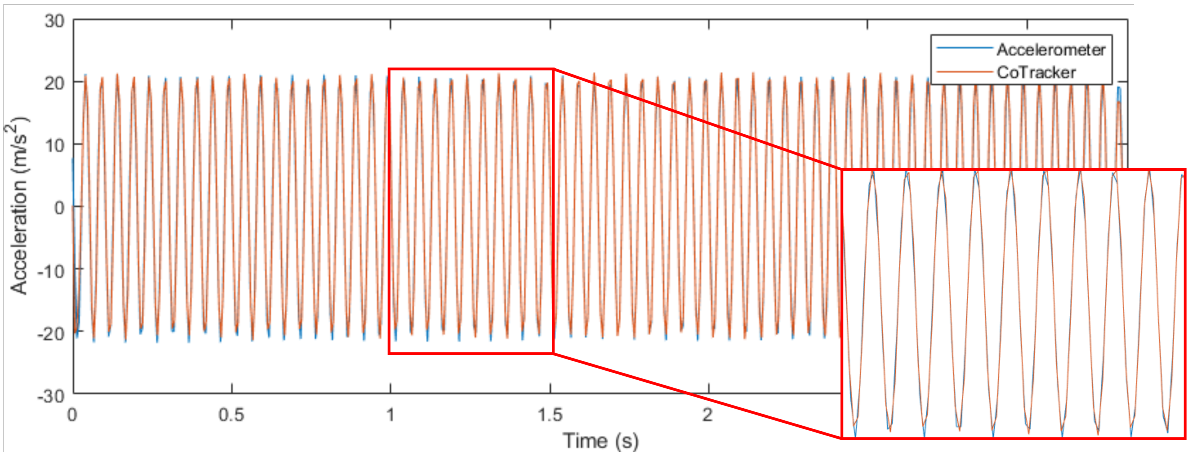


Figure 4.4: Time-Domain. *The time-domain diagram indicates a strong agreement between the data of the accelerometer and CoTracker.*

Additional to the analysis in the time-domain, also a comparison in the frequency-domain is done. Figure 4.5 shows how the data from the accelerometer and CoTracker both identify the dominant 20 Hz frequency. In addition to this, the data from the accelerometer shows a slight peak around 40 Hz. This can be explained due to nonlinear behaviour of the beam.

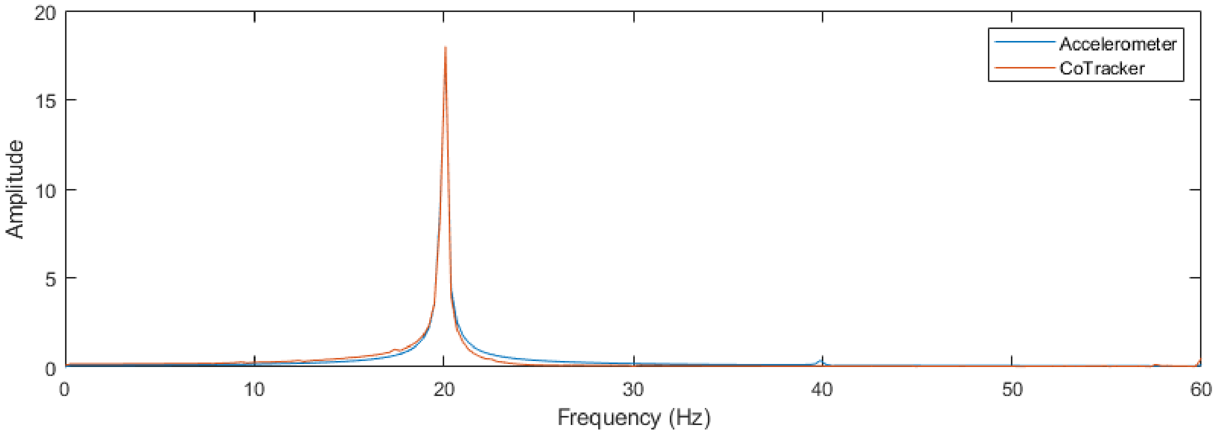


Figure 4.5: Frequency-Domain. *The data of the accelerometer and CoTracker agree on identifying the dominant frequency of 20 Hz.*

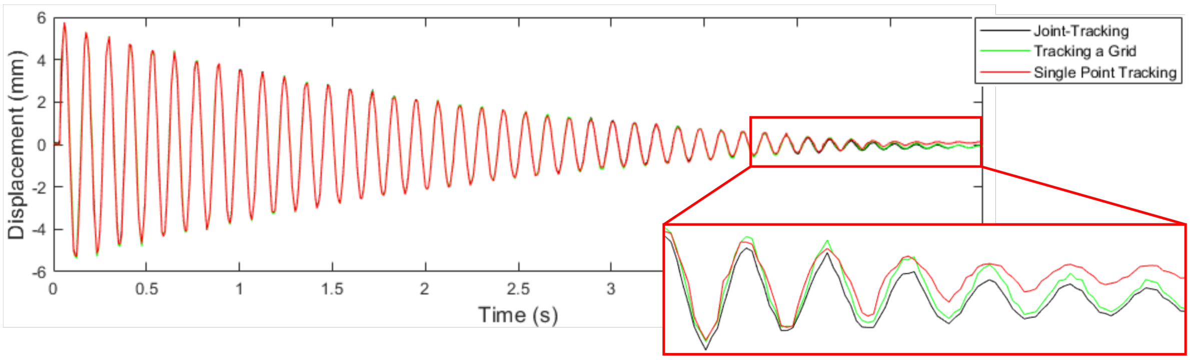


Figure 4.6: Comparing Point Initialization Strategies. *In the time-domain analysis, each strategy shows similar tracking abilities.*

4.2.2 Hammer Impact Test

The second experiment is a hammer impact test. Here, a single impact force is excited on the beam. Subsequently, the dynamics are captured by the high-speed camera with 120 fps and captured 601 frames. This results in a data set with approximately 5 seconds of length. In this experiment, differences between each segmentation method are found to validate the optimal segmentation method. The three segmentation techniques are analyzed in both the time-domain and the frequency-domain. Figure 4.6 illustrates the outcomes of the different methods. As shown, all three methods are capable of tracking the displacement. The discrepancy at the end of the diagram can be attributed to the single point method tracking a slightly different location. The difference between the points is 0.19 mm. Table 4.2 shows the qualitative comparison. The different strategies show a procentual error of 0.7%, caused by the initial difference of the points.

Table 4.2: Difference in Strategies. *The error margin margin between each method can be disregarded.*

		Tracking a Grid	Single Point Tracking	Joint-Tracking
Joint-Tracking	Error (%)	0.7283	0.7239	-
	Standard Deviation	0.0814	0.1131	-
Tracking a Grid	Error (%)	-	0.2624	-0.3344
	Standard Deviation	-	0.1028	0.1131

To verify the outcomes of the various approaches, accelerometer data is utilized. Figure 4.7 demonstrates this comparison. Although the accelerometer data and CoTracker's data show similar behavior, the initial acceleration recorded by the accelerometer is notably greater than that from CoTracker. The reason for this is twofold. Firstly, Co-Tracker measures pixel displacement

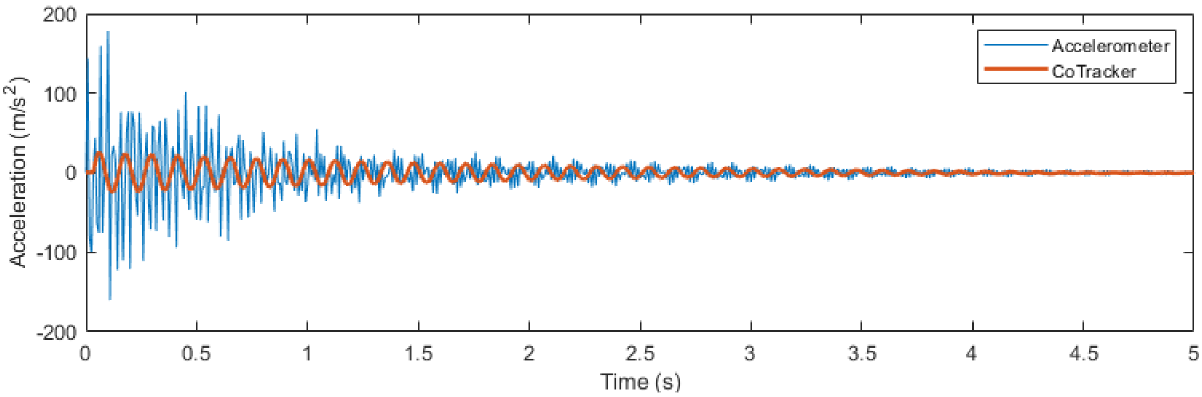


Figure 4.7: Time-Domain Comparison. *The time-domain comparison between the accelerometer and CoTracker shows large differences.*

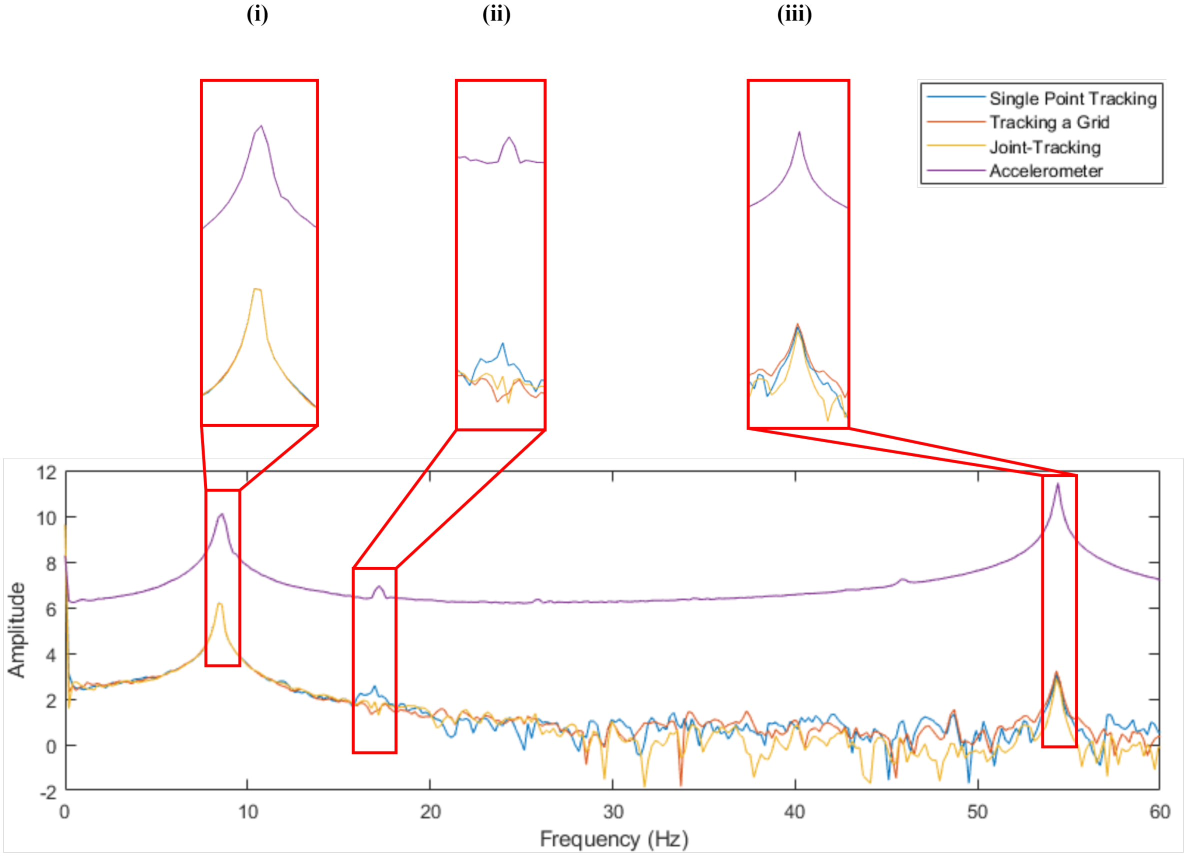


Figure 4.8: Frequency-Domain Comparison. *The frequency-domain analysis shows that CoTracker is able to capture similar dominant frequencies as accelerometer data.*

instead of structural deceleration, which can induce the discrepancy in amplitude. Secondly, the sampling rate of the accelerometer equals 8192 Hz, as the sampling rate of CoTracker equals 120 Hz. This difference enables the accelerometer to capture more frequency components that are present in the structure. Therefore, the comparison of the time-domain diagram indicates a significant disparity between the CoTracker readings and the accelerometer. To confirm CoTracker's capability to detect beam frequencies, a frequency-domain analysis is also performed. Figure 4.8 illustrates the comparison between the outcomes of different segmentation strategies and the data from an accelerometer. The accelerometer data shows four distinctive frequencies at 8.4 Hz, 17.1 Hz, 45.8 Hz, and 54.3 Hz. The second and third frequencies are caused by nonlinear behavior of the beam. It is found that CoTracker is able to identify distinctive natural frequencies and can even locate frequencies caused by nonlinearities.

4.2.3 Edge Detection vs CoTracker

To validate the results of CoTracker with existing algorithms that are widely used for video-based vibration extraction, several edge detection methods are used. This section presents the results of the comparison between edge detection methods and the performance of CoTracker. Figure 4.9 shows two areas that are used to extract the displacement of the beam. The green area indicates a grid with highly promising results, as the red area represents a grid for which edge detection shows its limitations. Figure A.1 in Appendix A presents the results of extracting the displacement using edge detection in time-domain. Note that the displacement is converted from pixel displacement to true displacement using camera calibration. The diagram for the green area shows that both edge detection and CoTracker appear to be able to capture the

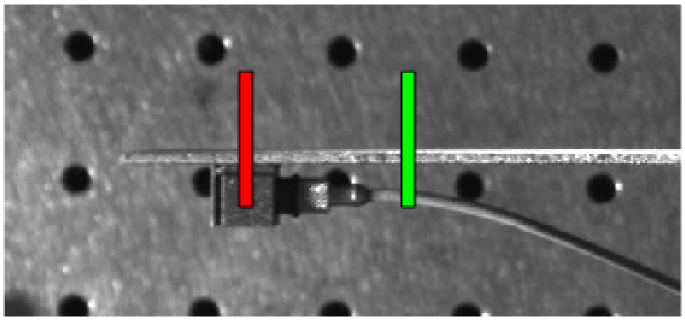


Figure 4.9: Analysis Edge Detection. *The green area represents the grid that gives highly promising results. The red grid shows the limitations of edge detection.*

displacement of the beam. To validate this intuition, Table 4.3 is summarizing the qualitative comparison in terms of the mean difference, the standard deviation, and the correlation. It is found that all edge detection methods have a correlation of more than 0.96 to CoTracker, with a standard deviation lower than 0.865, and a minimal mean difference.

Table 4.3: Qualitative Comparison. *A qualitative comparison between CoTracker and several edge detection methods. The green area indicates a good fit, as the red area shows that edge detection is not suitable as a full-field measurement tool.*

		Canny	Sobel	Laplacian	Subpixel	Threshold
Green area	Mean difference	1.32E-04	9.68E-05	1.35E-04	2.45E-04	2.82E-04
	Standard deviation	0.8616	0.8597	0.8545	0.8645	0.8401
	Correlation	0.9807	0.9808	0.9642	0.9770	0.9947
Red area	Mean difference	2.19E-07	-2.99E-04	1.46E-02	-1.79E-04	3.84E-07
	Standard deviation	2.4257	1.7764	6.4827	2.0679	9.6607
	Correlation	0.4718	0.7223	0.1204	0.3498	0.0658

Figure A.1 also shows the time-domain results for the analysis in the red area. It can be seen that edge detection loses its ability to accurately track the displacement of the edges. Table 4.3 shows that the correlation between each method and CoTracker significantly decreases compared to the analysis in the green area. Whereas the mean error remains close to zero, it can be seen that the standard deviation significantly increases, which implies that local differences between CoTracker and edge detection methods have increased. To extract modal parameters, the analysis should also be done in the frequency domain. Figure A.2 in Appendix A presents these results. For both the green and red area, the first natural frequency is found by CoTracker and edge detection methods and is found at 8.4 Hz. For the second natural frequency of 54.3 Hz, it can be seen that the green area can detect the dominant frequency. The red area shows the limitations of edge detection. Due to a complex homogeneous background, edge detection loses its ability to accurately detect edges, and therefore the displacement of these edges.

4.3 Time-Embedding DMD - Modal Parameter Extraction

This section presents the results obtained by extracting the natural frequency (f_n), the damping ratio (ζ), and the mode shape (Φ). An optimization of the standard DMD algorithm is presented in Subsection 2.4.2. The results of the optimized DMD algorithm are compared with the results of the standard DMD algorithm. After this comparison, the results of the modal parameter extraction are presented. Figure 4.10 shows 20 points that are used for the analysis.

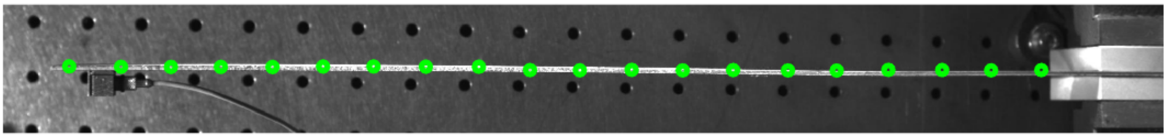


Figure 4.10: Initialization. *For the DMD analysis, 20 uniformly distributed points are chosen on the entire length of the beam.*

4.3.1 Time-Embedding DMD: CoTracker

The data used for this experiment is equal to the data presented in Subsection 4.2.2. Specifically, the displacement data obtained by joint-tracking is used to capture the displacement along the entire beam. First, the results for extracting the natural frequencies are presented. Figure 4.11 illustrates the pseudo-frequency stability diagram of the standard DMD algorithm and the filtered DMD algorithm, which shows the stability of identified natural frequencies. Instead of increasing mode order, as done by Saito et al. [48], this research investigates the stability of the DMD eigenvalue by applying an increasing sampling frequency using a resampling strategy. The idea is if DMD identifies similar eigenvalues at different sampling frequencies, this eigenvalue can be classified as a true eigenvalue. As shown in Figure 4.8, the first and second natural frequencies are found to be 8.4 Hz and 54.3 Hz, respectively. As seen in the diagram on the left, multiple frequencies are identified as natural frequencies. However, this is due to the non-linear behavior of the beam. Hence, the filtered diagram extracts natural frequencies that are within 1.0% of the two real natural frequencies. The diagram on the right shows the extraction of these two frequencies. Table 4.4 summarized the key results. The extraction of the first frequency is compared with accelerometer data. The mean first natural frequency is 8.4609, indicating a difference from the accelerometer of 0.0609. Furthermore, the standard deviation of all the eigenvalues found at the first frequency equals 0.011, indicating that all the eigenvalues remain close to the true eigenvalue embedded in the natural frequency found by the accelerometer. This demonstrates that the DMD algorithm is able to accurately extract the natural frequencies using CoTracker data.

Table 4.4: Accelerometer vs CoTracker. *The time-embedding DMD algorithm shows that CoTracker is able to identify the natural frequencies with high accuracy.*

First nat. freq.	Mean	8.4609
	Difference with accelerometer	-0.0609
	Standard deviation	0.0110
Second nat. freq.	Mean	54.301
	Difference with accelerometer	-0.0010
	Standard deviation	0.0739

Subsequently, using the DMD algorithm, the corresponding eigenvalues μ_i are extracted that are present in $f_i = |s_i|/2\pi$, where $s_i = \log(\mu_i)/\Delta t$, resulting in

$$f_i = \frac{|\log(\mu_i)/\Delta t|}{2\pi} \tag{4.1}$$

Then, reformulating this equation gives the eigenvalue. From Equation 4.2 it can be seen that the eigenvalue is related to the extracted natural frequency f_i.

$$\mu_i = 10^{2\pi f_i \Delta t} \tag{4.2}$$

Based on these eigenvalues, the corresponding damping ratios are extracted. This is done using $\zeta_i = Re(s_i)/|s_i|$, which results in

$$\zeta_i = Re\left(\frac{\log(\mu_i)}{\Delta t}\right) / \left|\frac{\log(\mu_i)}{\Delta t}\right| \tag{4.3}$$

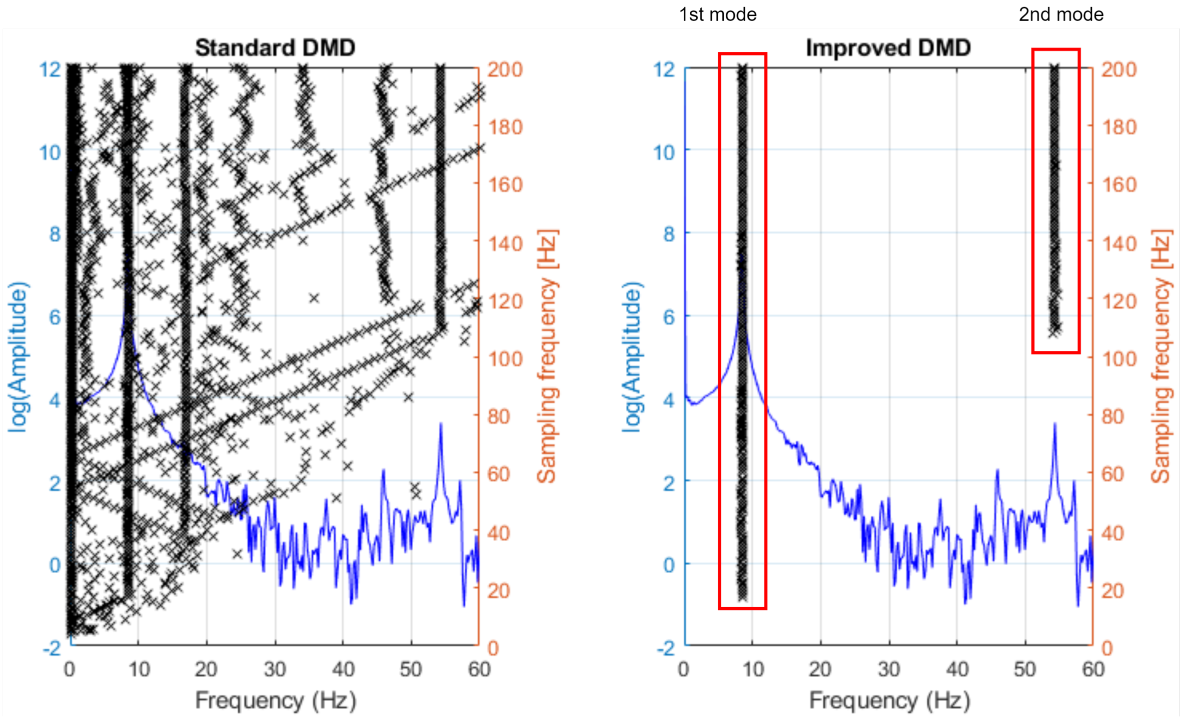


Figure 4.11: ***Natural Frequency.*** *Extracting the natural frequencies using the standard DMD algorithm (left) and the filtered DMD algorithm (right).*

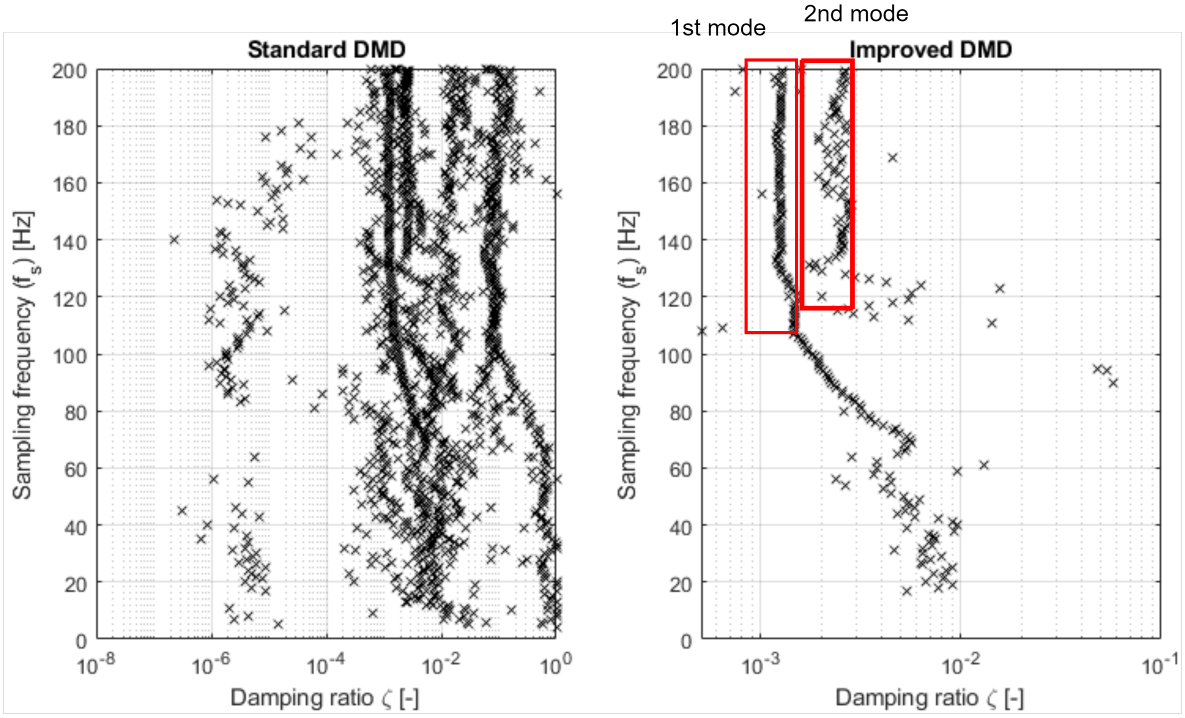


Figure 4.12: ***Damping Ratios.*** *Extracting the damping ratios using the standard DMD algorithm (left) and the filtered DMD algorithm (right).*

Hence, the extracted natural frequency and the damping ratio are related by the eigenvalue μ_i. Figure 4.12 shows the results for the damping ratios. In the left-hand diagram, the results for the standard DMD algorithm are presented. It is difficult to identify individual damping ratios. By applying the filtered DMD algorithm, which only considers eigenvalues that remain within a specified error margin ϵ, it is possible to extract the corresponding damping ratios. This is presented in the diagram on the right. In this diagram, the corresponding damping ratios are more evident in comparison to the standard DMD algorithm. For the first and second natural frequencies, approximations for two damping ratios are extracted at 0.0013 and 0.0023. This shows that the filtered DMD algorithm is better able to extract the corresponding damping ratios than the standard DMD algorithm. Additional to extracting the natural frequencies, the DMD algorithm is also able to extract the corresponding mode shapes. Figure 4.13 presents these results, validated by the analytical approach mentioned in Subsection 2.4.3. As can be seen, the analytical approach and the DMD algorithm show similar mode shapes, indicating that CoTracker is able to extract the first two mode shapes correctly. Additional to the analytical study, Cheng et al. [10] have validated the mode shapes with a FEM analysis. The results from this FEM analysis are presented in Figure 4.14 and compared with the DMD mode shapes in Figure 4.13. It is found that the FEM-analysis closely corresponds to the analytical approach and the DMD results. This implies that CoTracker can accurately identify modal parameters as the natural frequency, the damping ratios, the mode shapes and the corresponding eigenvalues, which underlines the potential of CoTracker as a full-field non-contact SHM algorithm.

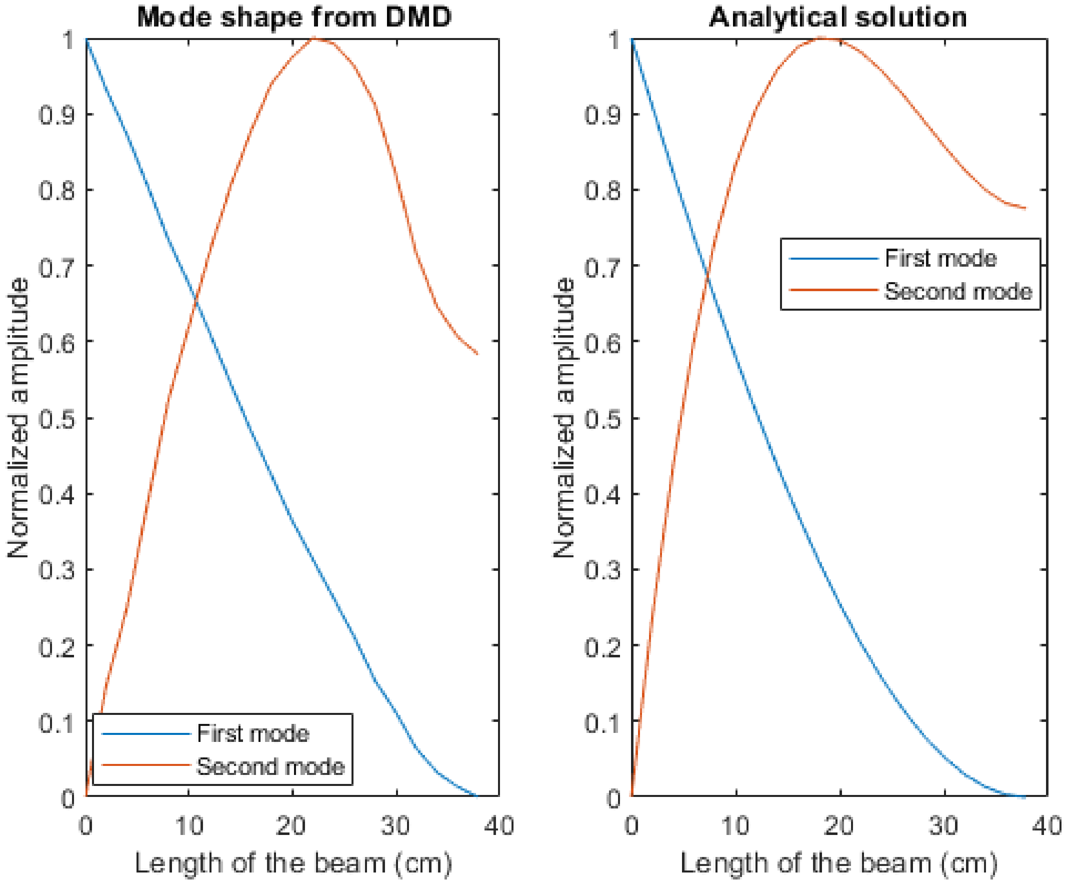


Figure 4.13: Analytical Solution vs DMD. *This figure shows that the mode shapes from the time-embedding DMD algorithm closely correspond to analytical solutions.*

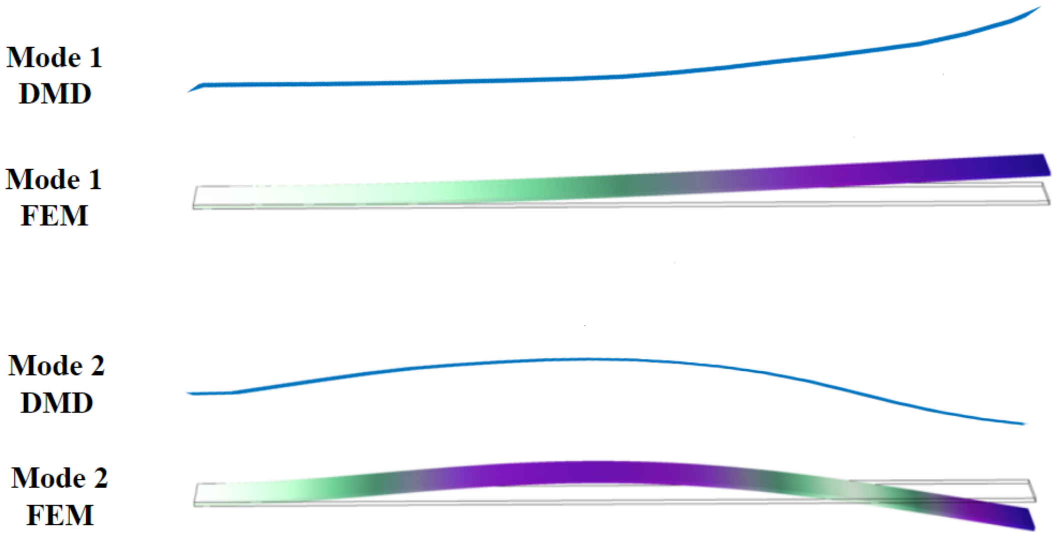


Figure 4.14: FEM-analysis vs DMD. *The FEM study from Cheng et al. [10] shows that the time-embedding DMD algorithm can identify mode shapes correctly using CoTracker data.*

4.3.2 Time-Embedding DMD: Edge Detection

To validate the performance of CoTracker as a full-field vision-based SHM algorithm, several edge detection methods are used. For each method, the mean of the extracted natural frequencies, the difference with the accelerometer, and the standard deviation is determined. For all methods, the error margin ϵ increases from 1% to 10%. The initial error margin of 1% could not provide meaningful results. Table 4.5 shows the results of the comparison with the accelerometer data for extracting the two natural frequencies. It can be seen that the first natural frequency is identified by each edge detection method accurately. Additionally, it is found that edge detection methods can also find the second natural frequency with less accuracy than the first natural frequency. The accuracy of CoTracker compared to the edge detection methods is given in Table 4.4. The difference of CoTracker with the first and second natural frequency equals 0.0609 and 0.0010, respectively. For the edge detection methods it is found that these differences are significantly higher, mostly in extracting the second natural frequency.

Table 4.5: Qualitative Comparison. *A qualitative comparison between the accelerometer data and several edge detection methods.*

		Edge Detection Method				
		Canny	Sobel	Laplacian	Subpixel	Threshold
First nat. freq.	Mean	8.4736	8.4399	8.4442	8.3849	8.4903
	Difference with accelerometer	0.0736	0.0399	0.0442	-0.0151	0.0903
	Standard deviation	0.1042	0.1629	0.1376	0.1582	0.1624
Second nat. freq.	Mean	50.080	52.231	53.527	52.530	54.573
	Difference with accelerometer	-4.220	-2.069	-0.773	-1.770	0.273
	Standard deviation	1.0943	2.7539	3.2228	2.2125	1.5873

Table 4.6 shows the qualitative differences of CoTracker and edge detection compared with accelerometer data. It is found that CoTracker is better in extracting the natural frequencies as edge detection by a significant percentual gain. Additionally, since the error margin ϵ is increased to 10%, edge detection erroneous numerical eigenvalues for the natural frequencies. This implies that these eigenvalues are not actually corresponding to the respective vibration mode. To validate the accuracy of edge detection eigenvalues, the time-embedded DMD algorithm is also applied for all edge detection methods. Appendix B presents the outcomes of the time-embedded

Table 4.6: Accuracy of CoTrackef vs Edge Detection. *CoTracker performs better in extracting the first and second natural frequency compared with several edge detection methods.*

Canny	Sobel	Laplacian	Subpixel	Threshold
0.151%	0.250%	0.199%	0.905%	0.350%
7.773%	3.813%	1.425%	3.261%	0.502%

DMD algorithm across all edge detection techniques. It reveals that, among all methods, only the first natural frequency can be accurately identified. For the second frequency, frequencies within a 10% range are sparingly extracted. Consequently, the damping ratios for the first mode are relatively stable. However, due to the limited extracted poles for the second mode, the corresponding damping ratios remain unstable. This observation is further affirmed by deriving the DMD mode shapes. It becomes evident that no edge detection technique successfully captures the mode shape when applying the time-embedded DMD algorithm. Although edge detection can identify natural frequencies, its accuracy is significantly restricted. Therefore, it is concluded that CoTracker demonstrates higher potential in extracting structural modal parameters, highlighting its application as a comprehensive non-contact SHM algorithm in comparison to conventional edge detection methods commonly employed in structural analysis.

5. Conclusion

This study investigates a newly developed deep-learning algorithm named CoTracker for its potential as a comprehensive vision-based Structural Health Monitoring (SHM) solution. An experimental setup featuring a cantilever beam is used, from which CoTracker extracts structural vibrations. These findings are validated with reference data from an accelerometer placed on the beam. Furthermore, CoTracker's performance in extracting structural vibrations is compared to commonly used edge detection methods. The results of this study suggest that CoTracker is an effective algorithm for extracting structural vibrations in experimental setups, with a correlation of over 0.99 and a margin of error of 0.7% when compared to accelerometer data.

Besides monitoring structural vibrations, a time-embedded DMD algorithm is employed to determine three modal parameters: the first two natural frequencies, the damping ratios, and the associated mode shapes. Utilizing a full-field segmentation approach, mentioned *joint-tracking* in this study, it is possible to monitor all individual pixels on the beam. The time-embedded DMD algorithm has been found to accurately identify natural frequencies, damping ratios, and mode shapes when compared to analytical solutions and a FEM study. Consequently, it can be concluded that CoTracker demonstrates its capability in monitoring structural vibrations and identifying modal parameters by utilizing the comprehensive measurement capabilities of a camera.

6. Future Work

In order to thoroughly investigate the capabilities of CoTracker, further studies are necessary. The following section outlines additional research to consider.

1. *Investigate the effect of uncontrolled environmental conditions.*
   This research uses an experimental setup in a laboratory, where environmental conditions can be controlled. However, for the commercial application of CoTracker it is recommended to validate the ability of structural vibration tracking when environmental conditions are not controlled. For instance, conducting field measurements (capturing video images from a structure outside a controlled environment), or using a highly complex background.

2. *Investigate both micro vibrations and macro vibrations.*
   Structural vibrations might infer both micro vibrations and macro vibrations. To validate the accuracy of CoTracker, both type of vibrations could be investigated on, relatively, small and large structures.

Bibliography

[1] D. Agdas, J. A. Rice, J. R. Martinez, and I. R. Lasa. Comparison of visual inspection and structural-health monitoring as bridge condition assessment methods. *Journal of Performance of Constructed Facilities*, 30(3):04015049, 2016.

[2] S. Akshay, K. Soman, N. Mohan, and S. Sachin Kumar. Dynamic mode decomposition and its application in various domains: An overview. *Applications in Ubiquitous Computing*, pages 121–132, 2021.

[3] American Society of Civil Engineers. *Overview of Bridges.* https://infrastructurereportcard.org/cat-item/bridges-infrastructure/, jan 2022. Acquired on: 15-04-2024.

[4] S. S. Beauchemin and R. Bajcsy. Modelling and removing radial and tangential distortions in spherical lenses. In *Multi-Image Analysis: 10th International Workshop on Theoretical Foundations of Computer Vision Dagstuhl Castle, Germany, March 12–17, 2000 Revised Papers*, pages 1–21. Springer, 2001.

[5] S. S. Beauchemin and J. L. Barron. The computation of optical flow. *ACM computing surveys (CSUR)*, 27(3):433–466, 1995.

[6] J. M. Brownjohn. Structural health monitoring of civil infrastructure. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851):589–622, 2007.

[7] J. F. Canny. Finding edges and lines in images. 1983.

[8] J. Chang and T. E. Nordling. Unsupervised skin feature tracking with deep neural networks. *arXiv preprint arXiv:2405.04943*, 2024.

[9] L. Cheng, A. Cigada, et al. Experimental strain modal analysis for beam-like structure by using distributed fiber optics and its damage detection. *Measurement Science and Technology*, 28(7):074001, 2017.

[10] L. Cheng, J. de Groot, K. Xie, Y. Si, and X. Han. Camera-based dynamic vibration analysis using transformer-based model cotracker and dynamic mode decomposition. *Sensors*, 24(11):3541, 2024.

[11] J. P. De Villiers, F. W. Leuschner, and R. Geldenhuys. Centi-pixel accurate real-time inverse distortion correction. In *Optomechatronic Technologies 2008*, volume 7266, pages 320–327. SPIE, 2008.

[12] P. Dhankhar and N. Sahu. A review and research of edge detection techniques for image segmentation. *International Journal of Computer Science and Mobile Computing*, 2(7):86–92, 2013.

[13] V. M. Dharampal. Methods of image edge detection: A review. *Journal of Electrical & Electronic Systems*, 4(2):5, 2015.

[14] C. Doersch, A. Gupta, L. Markeeva, A. Recasens, L. Smaira, Y. Aytar, J. Carreira, A. Zisserman, and Y. Yang. Tap-vid: A benchmark for tracking any point in a video. *Advances in Neural Information Processing Systems*, 35:13610–13626, 2022.

[15] C. Doersch, Y. Yang, M. Vecerik, D. Gokay, A. Gupta, Y. Aytar, J. Carreira, and A. Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10061–10072, 2023.

[16] W. Du, D. Lei, F. Zhu, P. Bai, and J. Zhang. A non-contact displacement measurement system based on a portable smartphone with digital image methods. *Structure and Infrastructure Engineering*, pages 1–19, 2022.

[17] P. Duhamel and M. Vetterli. Fast fourier transforms: a tutorial review and a state of the art. *Signal processing*, 19(4):259–299, 1990.

[18] A. D. Eslamlou, A. Ghaderiaram, M. Fotouhi, and E. Schlangen. A review on non-destructive evaluation of civil structures using magnetic sensors. In *European Workshop on Structural Health Monitoring*, pages 647–656. Springer, 2022.

[19] C. R. Farrar and K. Worden. An introduction to structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851):303–315, 2007.

[20] D. Feng and M. Q. Feng. Computer vision for shm of civil infrastructure: From dynamic response measurement to damage detection–a review. *Engineering Structures*, 156:105–117, 2018.

[21] T. Gautama and M. Van Hulle. A phase-based approach to the estimation of the optical flow field using spatial filtering. *IEEE transactions on neural networks*, 13(5):1127–1136, 2002.

[22] A. Güemes, A. Fernandez-Lopez, A. R. Pozo, and J. Sierra-Pérez. Structural health monitoring for advanced composite structures: a review. *Journal of Composites Science*, 4(1):13, 2020.

[23] S. Hassani and U. Dackermann. A systematic review of advanced sensor technologies for non-destructive testing and structural health monitoring. *Sensors*, 23(4):2204, 2023.

[24] S.-M. Hou, C.-L. Jia, Y.-B. Wanga, and M. Brown. A review of the edge detection technology. *Sparklinglight Transactions on Artificial Intelligence and Quantum Computing (STAIQC)*, 1(2):26–37, 2021.

[25] A. Javed, H. Lee, B. Kim, and Y. Han. Vibration measurement of a rotating cylindrical structure using subpixel-based edge detection and edge tracking. *Mechanical Systems and Signal Processing*, 166:108437, 2022.

[26] K. Jensen and D. Anastassiou. Subpixel edge localization and the interpolation of still images. *IEEE transactions on Image Processing*, 4(3):285–295, 1995.

[27] N. Karaev, I. Rocco, B. Graham, N. Neverova, A. Vedaldi, and C. Rupprecht. Cotracker: It is better to track together. *arXiv:2307.07635*, 2023.

[28] S. K. Katiyar and P. Arun. Comparative analysis of common edge detection techniques in context of object extraction. *arXiv preprint arXiv:1405.6132*, 2014.

[29] B. Kim, C. Min, H. Kim, S. Cho, J. Oh, S.-H. Ha, and J.-h. Yi. Structural health monitoring with sensor data and cosine similarity for multi-damages. *Sensors*, 19(14):3047, 2019.

[30] S. Komarizadehasl, F. Lozano, J. A. Lozano-Galant, G. Ramos, and J. Turmo. Low-cost wireless structural health monitoring of bridges. *Sensors*, 22(15):5725, 2022.

[31] X. Kong, J. Yi, X. Wang, K. Luo, and J. Hu. Full-field mode shape identification based on subpixel edge detection and tracking. *Applied Sciences*, 13(2):747, 2023.

[32] G. G. Kumar, S. K. Sahoo, and P. K. Meher. 50 years of fft algorithms and applications. *Circuits, Systems, and Signal Processing*, 38:5665–5698, 2019.

[33] C. Lee, D. Kang, and S. Park. Visualization of fatigue cracks at structural members using a pulsed laser scanning system. *Research in Nondestructive Evaluation*, 26(3):123–132, 2015.

[34] 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*, 21(12):125019, 2012.

[35] J. J. Lee and M. Shinozuka. A vision-based system for remote sensing of bridge displacement. *Ndt & E International*, 39(5):425–431, 2006.

[36] C. Li, Z. Peng, T.-Y. Huang, T. Fan, F.-K. Wang, T.-S. Horng, J.-M. Muñoz-Ferreras, R. Gómez-García, L. Ran, and J. Lin. A review on recent progress of portable short-range noncontact microwave radar systems. *IEEE Transactions on Microwave Theory and Techniques*, 65(5):1692–1706, 2017.

[37] T.-H. Liu, G.-Q. Li, X.-N. Nie, H.-J. Wang, D. Zhang, J.-M. Wu, and W. Liu. Enhancement of contour smoothness by substitution of interpolated sub-pixel points for edge pixels. *IEEE Access*, 9:44236–44246, 2021.

[38] W. Luo, Y. Xia, and T. He. Video-based identification and prediction techniques for stable vessel trajectories in bridge areas. *Sensors*, 24(2):372, 2024.

[39] R. Maini and H. Aggarwal. Study and comparison of various image edge detection techniques. *International journal of image processing (IJIP)*, 3(1):1–11, 2009.

[40] P. K. Muralidharan and H. Yanamadala. Comparative study of vision camera-based vibration analysis with the laser vibrometer method, 2021.

[41] H. H. Nassif, M. Gindy, and J. Davis. Comparison of laser doppler vibrometer with contact sensors for monitoring bridge deflection and vibration. *Ndt & E International*, 38(3):213–218, 2005.

[42] O. Omar, N. Tounsi, E.-G. Ng, and M. Elbestawi. An optimized rational fraction polynomial approach for modal parameters estimation from frf measurements. *Journal of mechanical science and technology*, 24:831–842, 2010.

[43] S. Patsias, W. Staszewski, and G. Tomlinson. Image sequences and wavelets for vibration analysis: Part 1: Edge detection and extraction of natural frequencies. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 216(9):885–900, 2002.

[44] J. M. G. Payawal and D.-K. Kim. Image-based structural health monitoring: A systematic review. *Applied Sciences*, 13(2):968, 2023.

[45] 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*, 75:164–180, 2014.

[46] W. Rong, Z. Li, W. Zhang, and L. Sun. An improved canny edge detection algorithm. In *2014 IEEE international conference on mechatronics and automation*, pages 577–582. IEEE, 2014.

[47] J. A. Saif, M. H. Hammad, and I. A. Alqubati. Gradient based image edge detection. *International Journal of Engineering and Technology*, 8(3):153, 2016.

[48] A. Saito and T. Kuno. Data-driven experimental modal analysis by dynamic mode decomposition. *Journal of Sound and Vibration*, 481:115434, 2020.

[49] P. J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

[50] P. J. Schmid. Application of the dynamic mode decomposition to experimental data. *Experiments in fluids*, 50:1123–1130, 2011.

[51] A. Shariati and T. Schumacher. Eulerian-based virtual visual sensors to measure dynamic displacements of structures. *Structural Control and Health Monitoring*, 24(10):e1977, 2017.

[52] I. Sobel, R. Duda, P. Hart, and J. Wiley. Sobel-feldman operator. *Preprint at https://www. researchgate. net/profile/Irwin-Sobel/publication/285159837. Accessed*, 20, 2022.

[53] B. F. Spencer Jr, V. Hoskere, and Y. Narazaki. Advances in computer vision-based civil infrastructure inspection and monitoring. *Engineering*, 5(2):199–222, 2019.

[54] H. Spontón and J. Cardelino. A review of classic edge detectors. *Image Processing On Line*, 5:90–123, 2015.

[55] Z. Teed and J. Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.

[56] J. H. Tu. *Dynamic mode decomposition: Theory and applications.* PhD thesis, Princeton University, 2013.

[57] S. T. Vegas and K. Lafdi. A literature review of non-contact tools and methods in structural health monitoring. *Eng. Technol. Open Access J.*, 4(1):9–50, 2021.

[58] A. M. Wahbeh, J. P. Caffrey, and S. F. Masri. A vision-based approach for the direct measurement of displacements in vibrating systems. *Smart materials and structures*, 12(5):785, 2003.

[59] Y. Wang, Y. Li, and J. Zheng. A camera calibration technique based on opencv. In *The 3rd International Conference on Information Sciences and Interaction Sciences*, pages 403–406. IEEE, 2010.

[60] L.-J. Wu, F. Casciati, and S. Casciati. Dynamic testing of a laboratory model via vision-based sensing. *Engineering structures*, 60:113–125, 2014.

[61] F. B. Zahid, Z. C. Ong, and S. Y. Khoo. A review of operational modal analysis techniques for in-service modal identification. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 42(8):398, 2020.

[62] Y. Zheng, A. W. Harley, B. Shen, G. Wetzstein, and L. J. Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19855–19865, 2023.

[63] A. Zona. Vision-based vibration monitoring of structures and infrastructures: An overview of recent applications. *Infrastructures*, 6(1):4, 2020.

A. Appendix: Accuracy Edge Detection

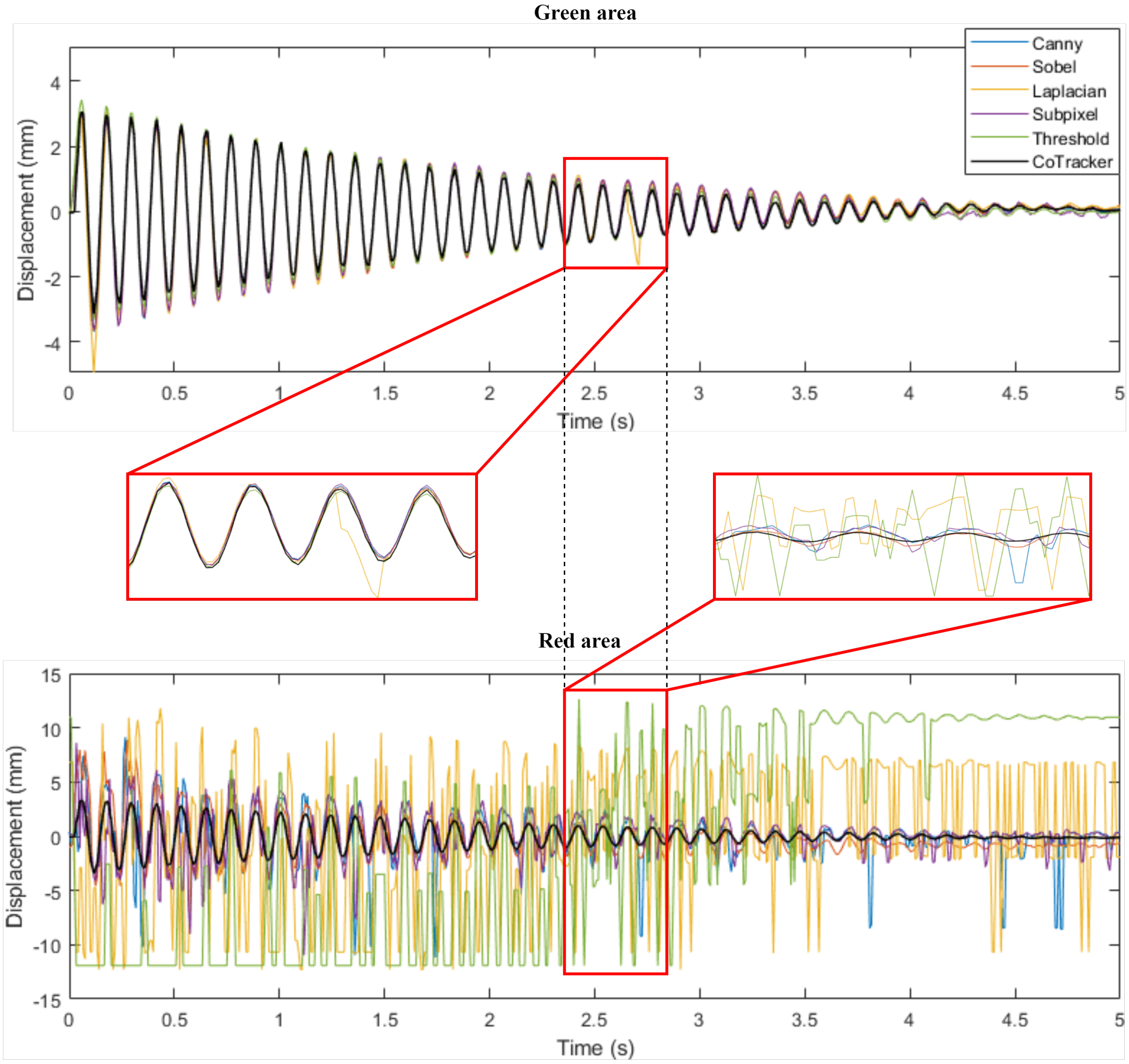


Figure A.1: Edge Detection vs CoTracker Time-Domain. *The analysis in the time-domain provides two conclusions. Edge detection and CoTracker can track displacement with similar results when high precision is used for edge detection (green area). However, for choosing an arbitrary region (red area), the limitations of edge detection become visible.*

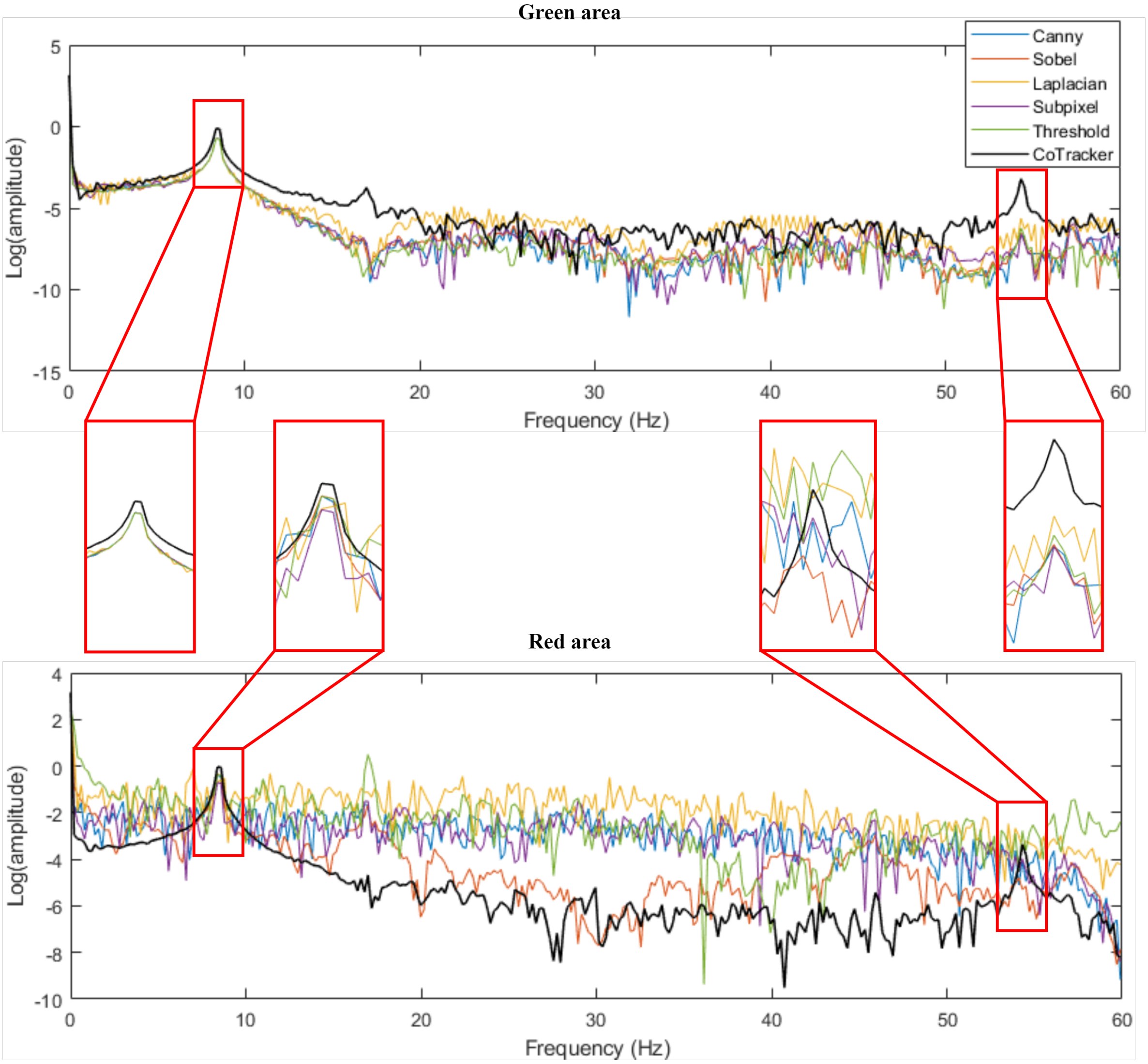


Figure A.2: Edge Detection vs CoTracker Frequency-Domain. *It can be seen that CoTracker (black) shows the correct dominant frequencies at 8.4 Hz and 54.3 Hz. However, edge detection is not able to accurately detect these natural frequencies when an arbitrary region is chosen.*

B. Appendix: Time-Embedding DMD: Edge Detection

B.1 Canny Method

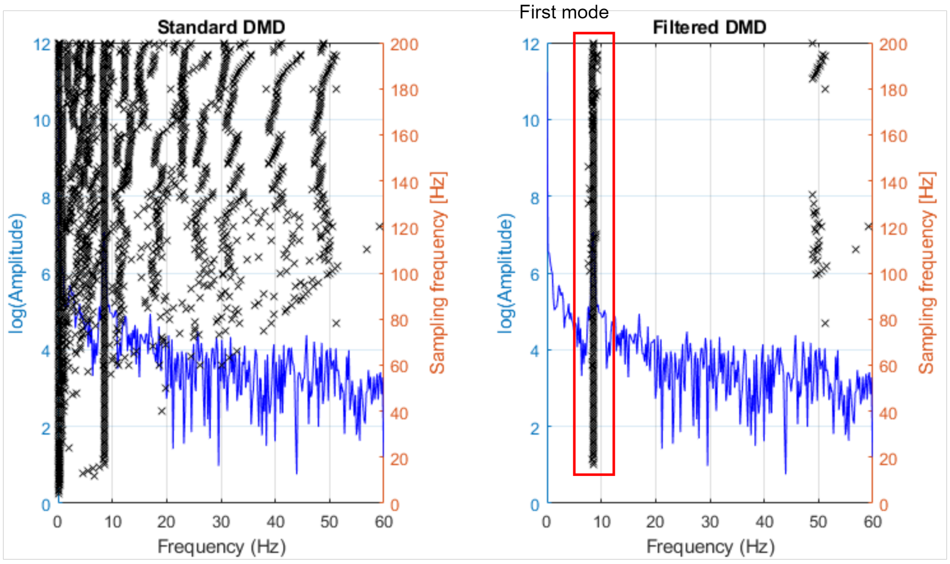


Figure B.1: DMD frequencies Canny Method. *The first natural frequency can be identified.*

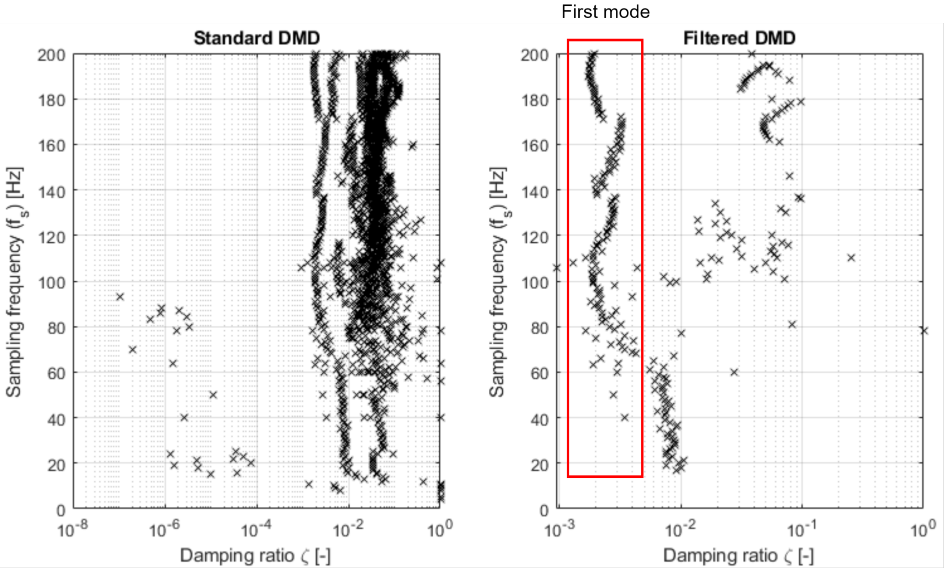


Figure B.2: DMD damping ratios Canny Method. *No stable damping ratios can be extracted.*

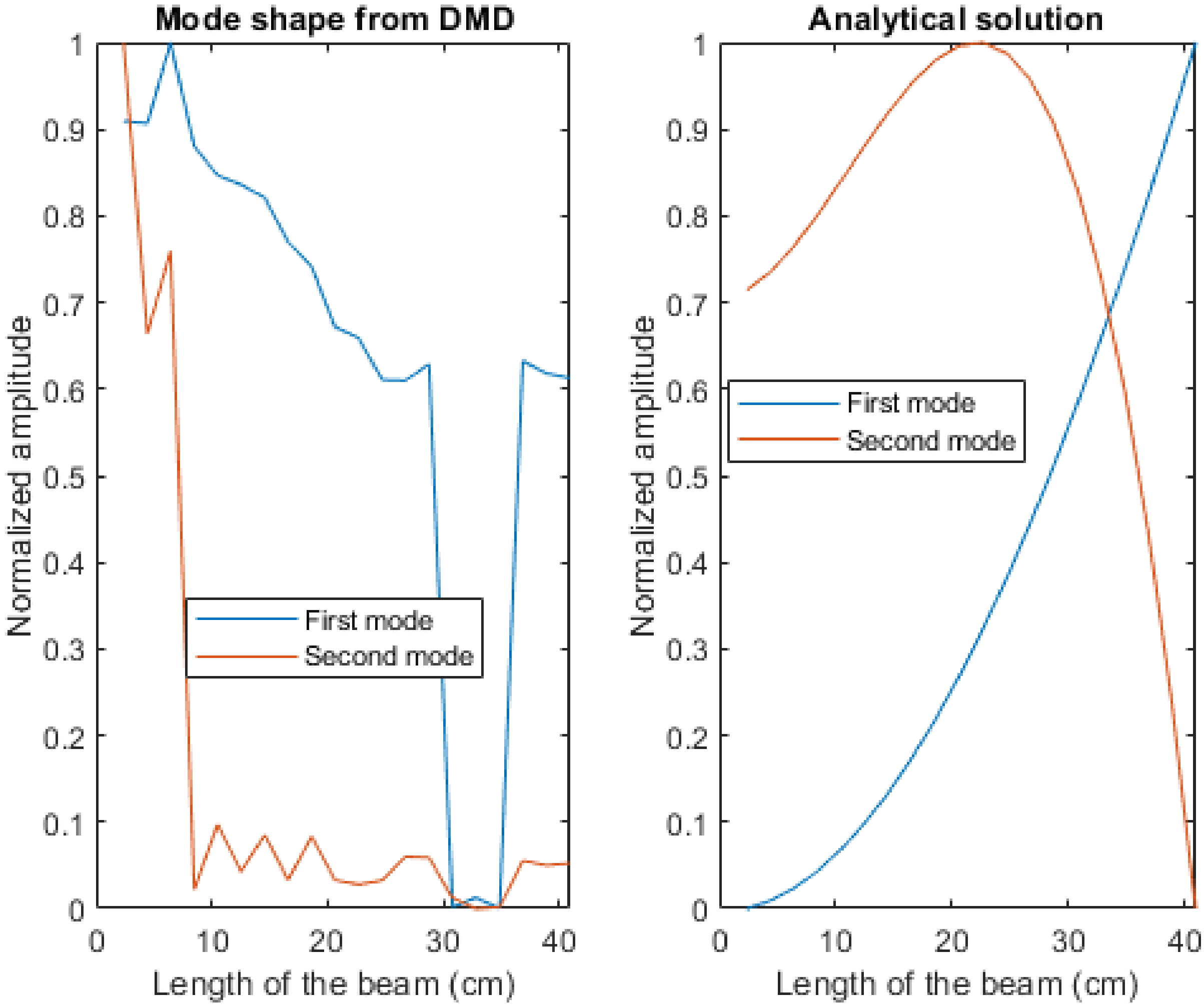


Figure B.3: DMD mode shape Canny Method. *The analytical solution is accepted, but the DMD mode shape is not supported by either the analytical solution or the FEM study.*

B.2 Sobel Method

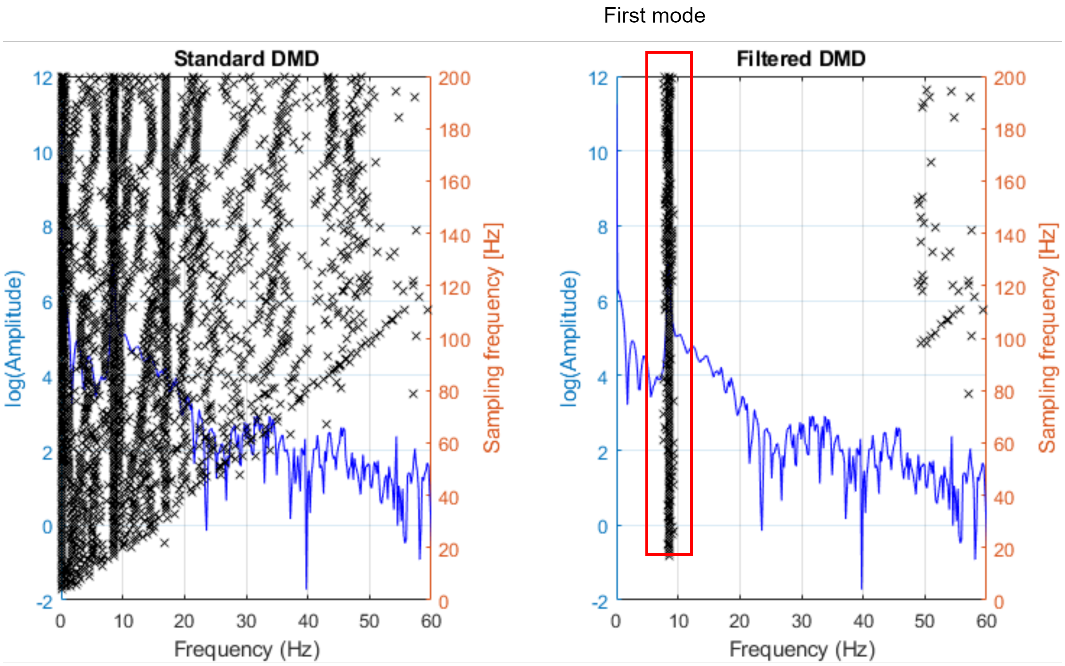


Figure B.4: DMD frequencies Sobel Method. *The first natural frequency can be identified.*

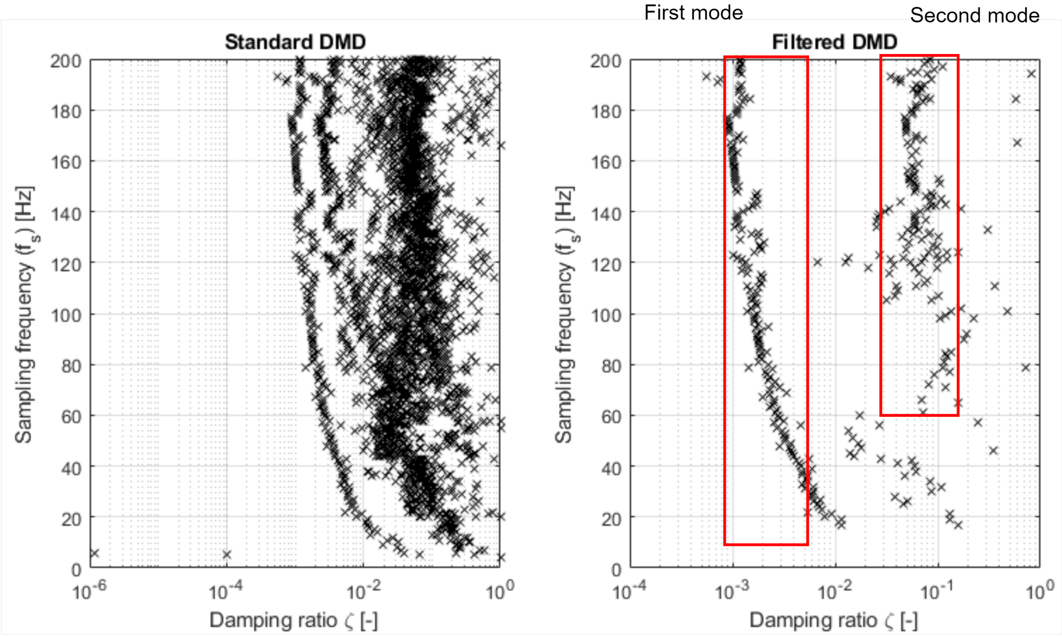


Figure B.5: DMD damping ratios Sobel Method. *The first damping ratio is relative stable, but the second damping ratio cannot be identified correctly.*

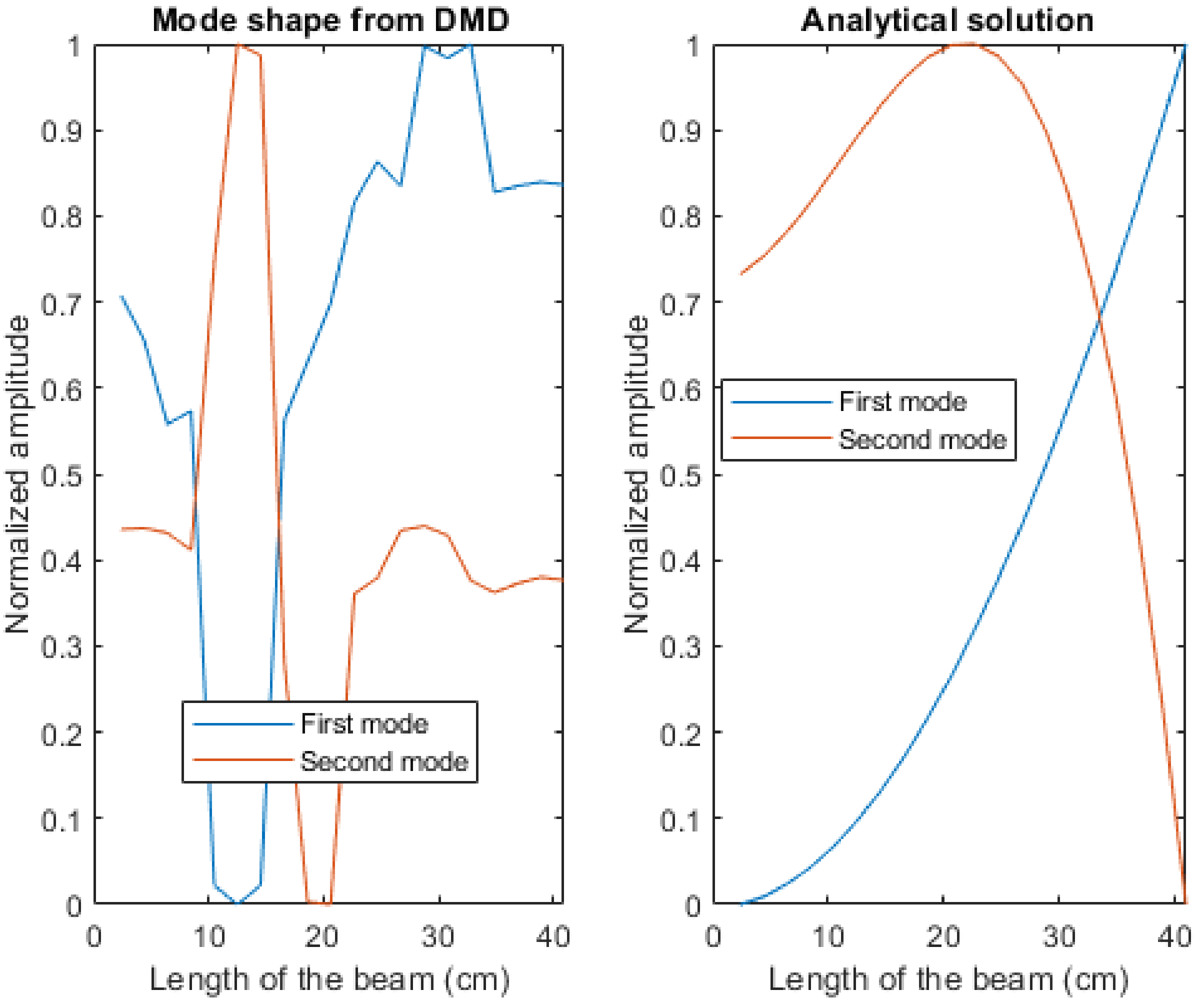


__Figure B.6: DMD mode shape Sobel Method.__ The analytical solution agrees with the FEM study. The DMD mode shape cannot be extracted accurately.

B.3 Laplacian Method

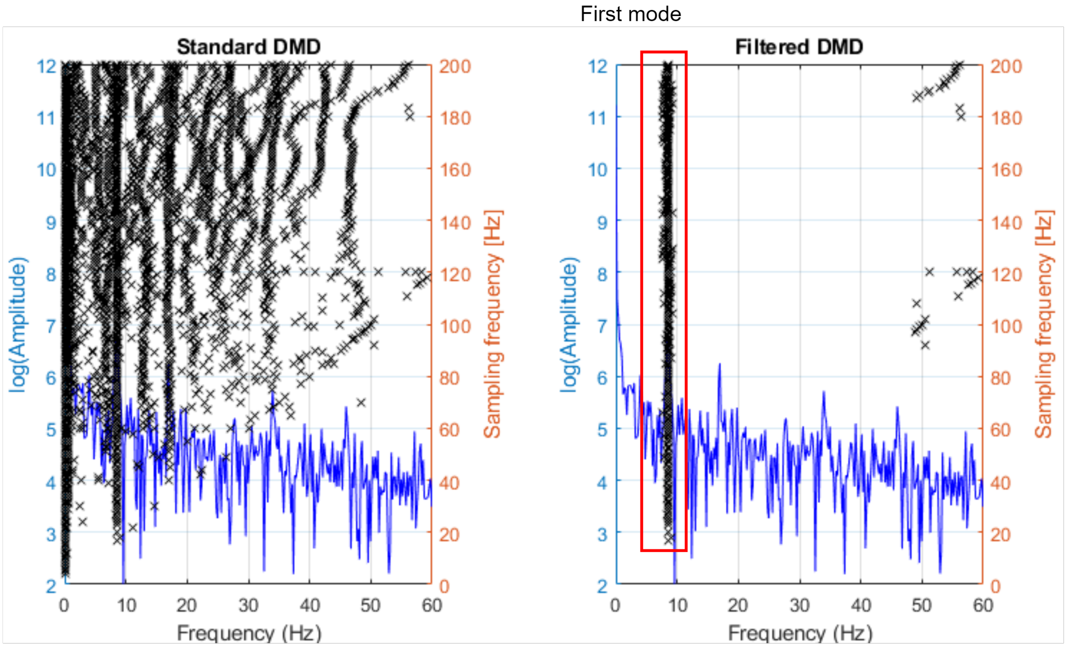


Figure B.7: DMD frequencies Laplacian Method. *The first natural frequency can be identified.*

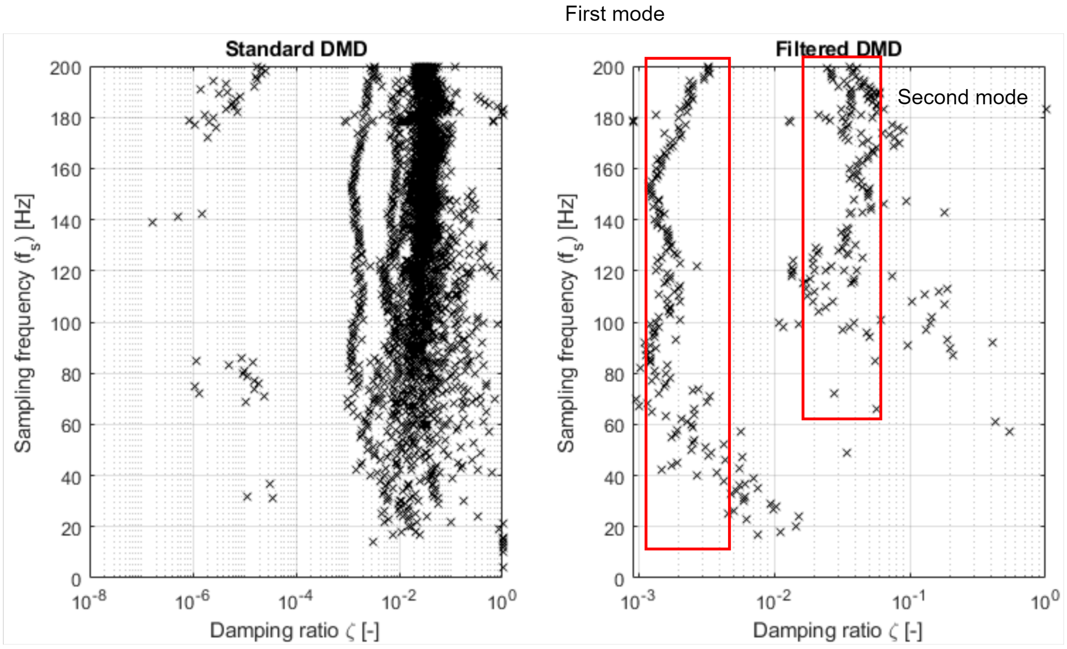


Figure B.8: DMD damping ratios Laplacian Method. *The first damping ratio is relatively stable. The second damping ratio cannot be extracted.*

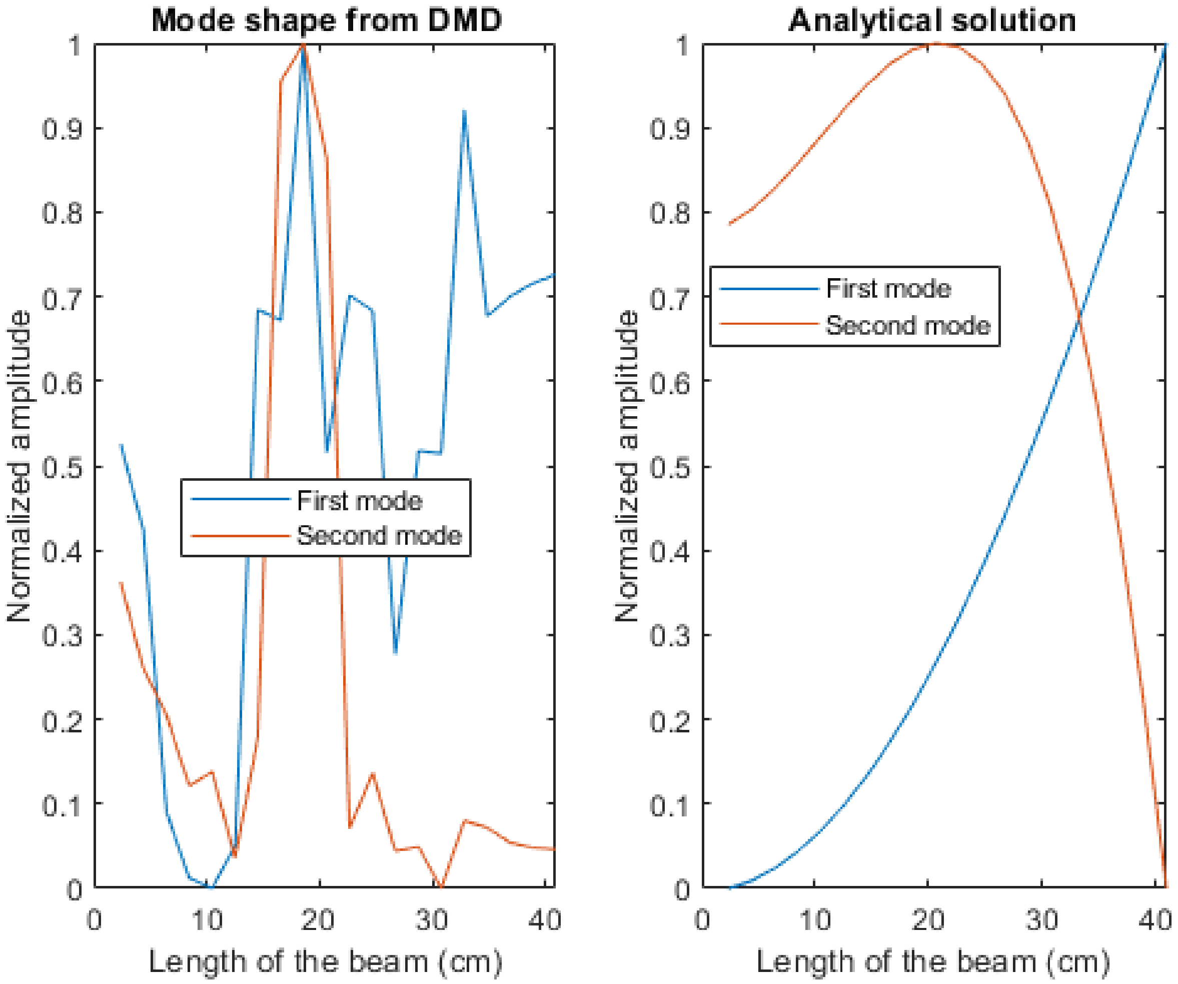


Figure B.9: DMD mode shape Laplacian Method. *The analytical solution corresponds with the FEM study. The DMD mode shape cannot be extracted.*

B.4 Subpixel Method

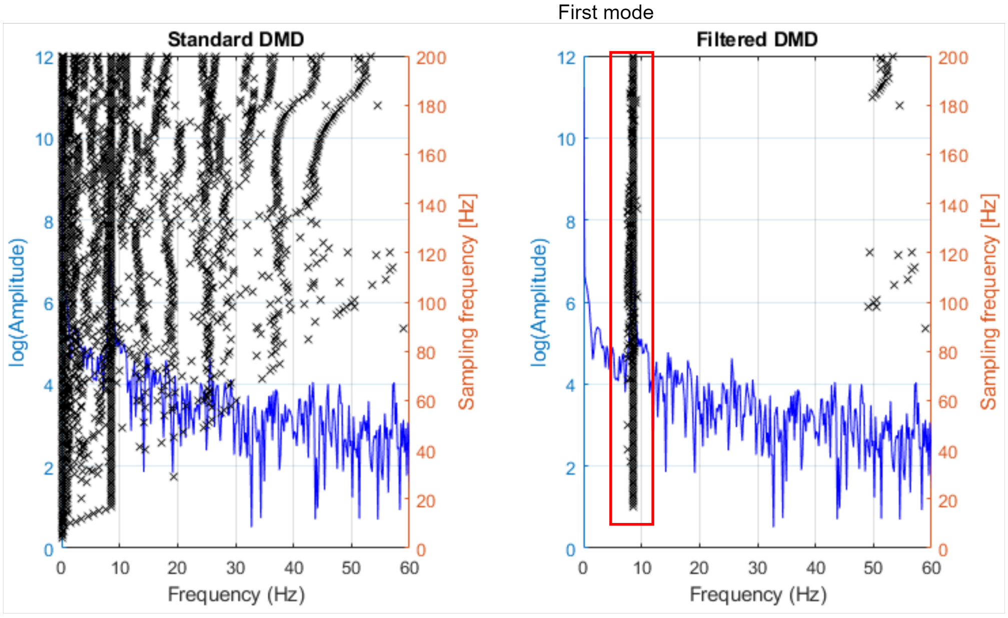


Figure B.10: DMD frequency Subpixel Method. *The first natural frequency can be identified.*

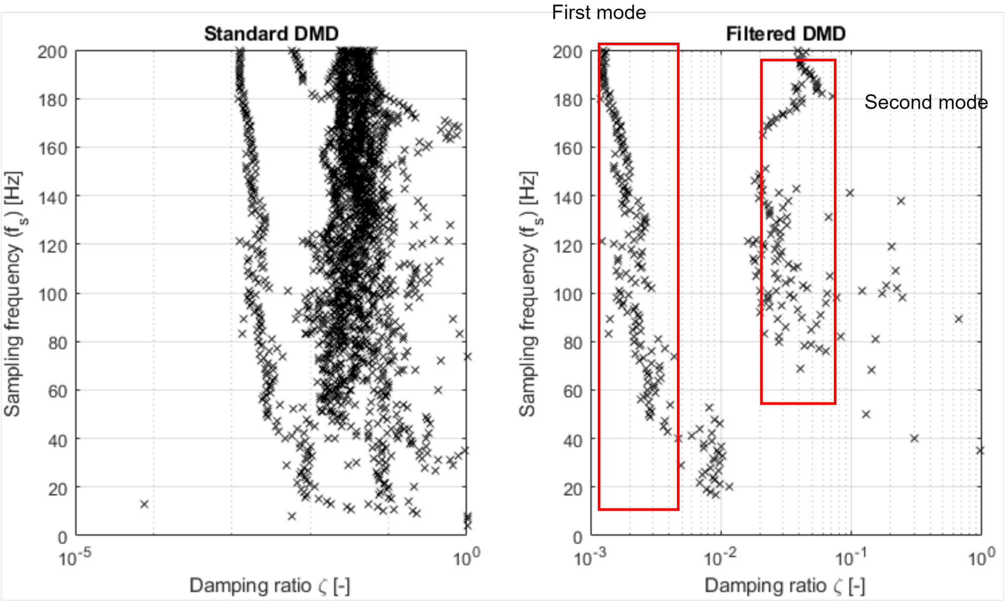


Figure B.11: DMD damping ratios Subpixel Method. *The first damping ratio is relatively stable. The second damping ratio cannot be extracted correctly.*

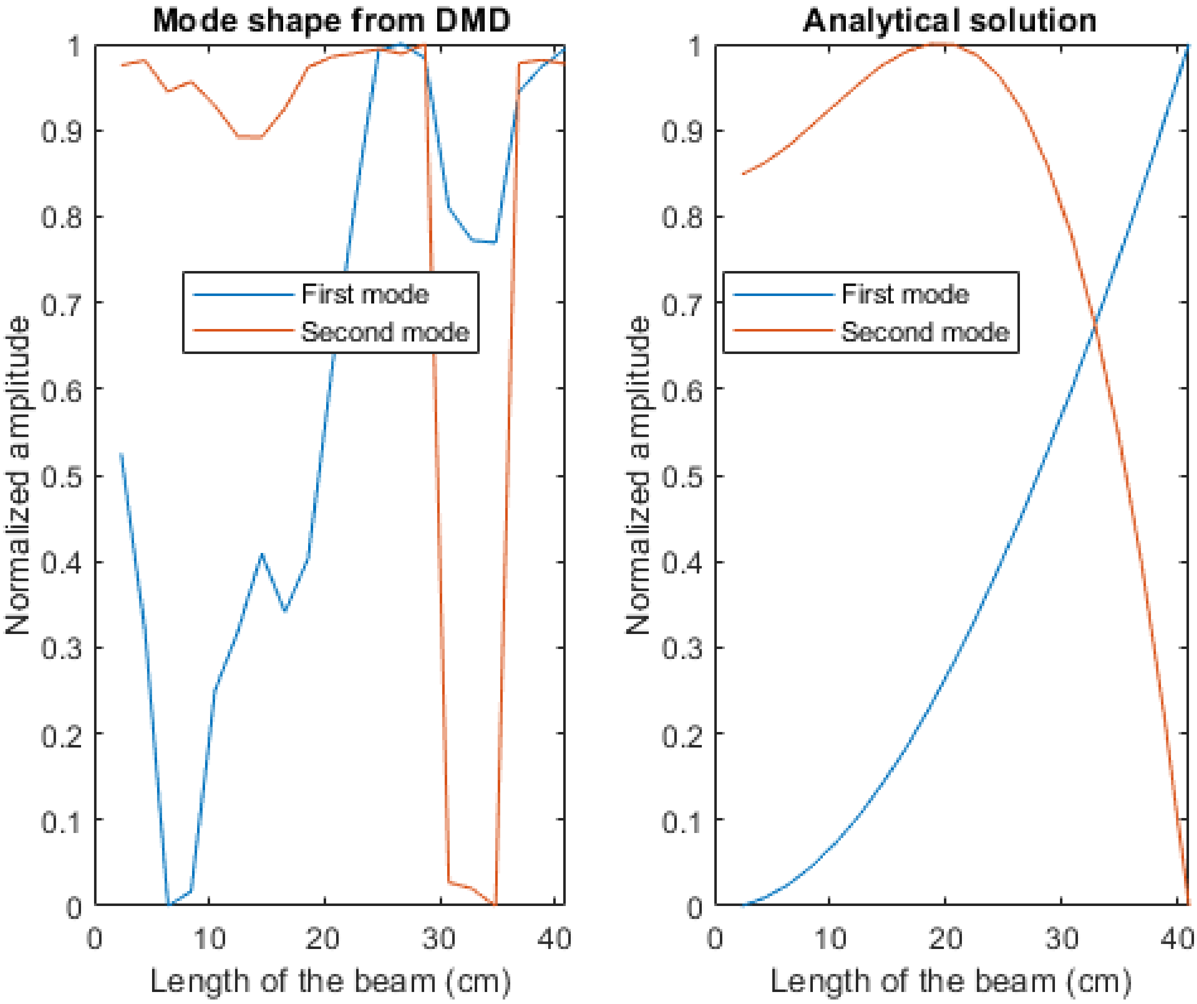


Figure B.12: DMD mode shape Subpixel Method. *The analytical solution corresponds to the FEM study. The DMD mode shape cannot be extracted accurately.*

B.5 Threshold Method

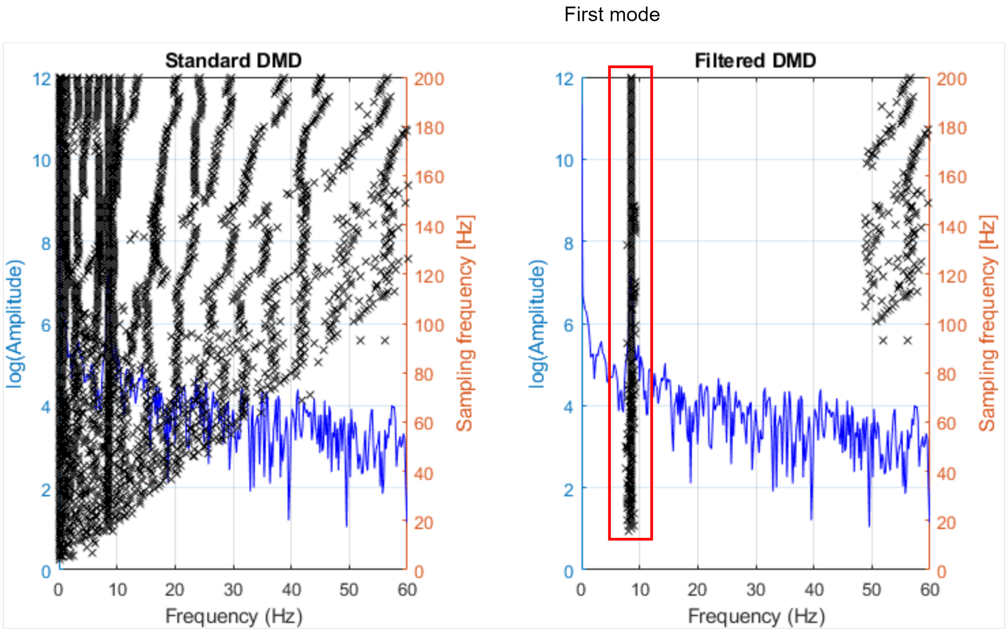


Figure B.13: DMD frequency Threshold Method. *The first frequency can be identified correctly.*

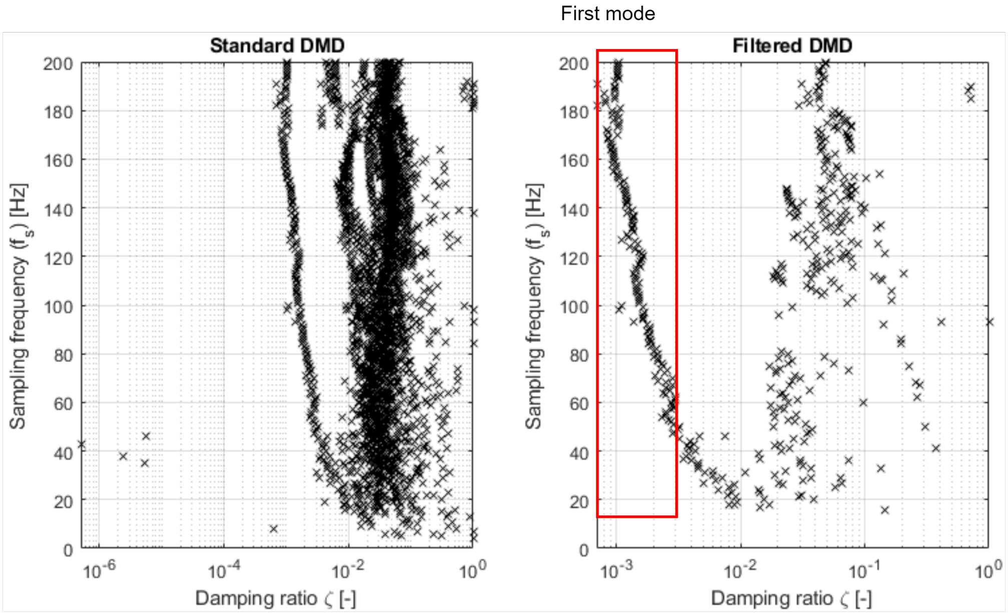


Figure B.14: DMD damping ratios Threshold Method. *The first damping ratio is relatively stable. The second damping ratio cannot be extracted accurately.*

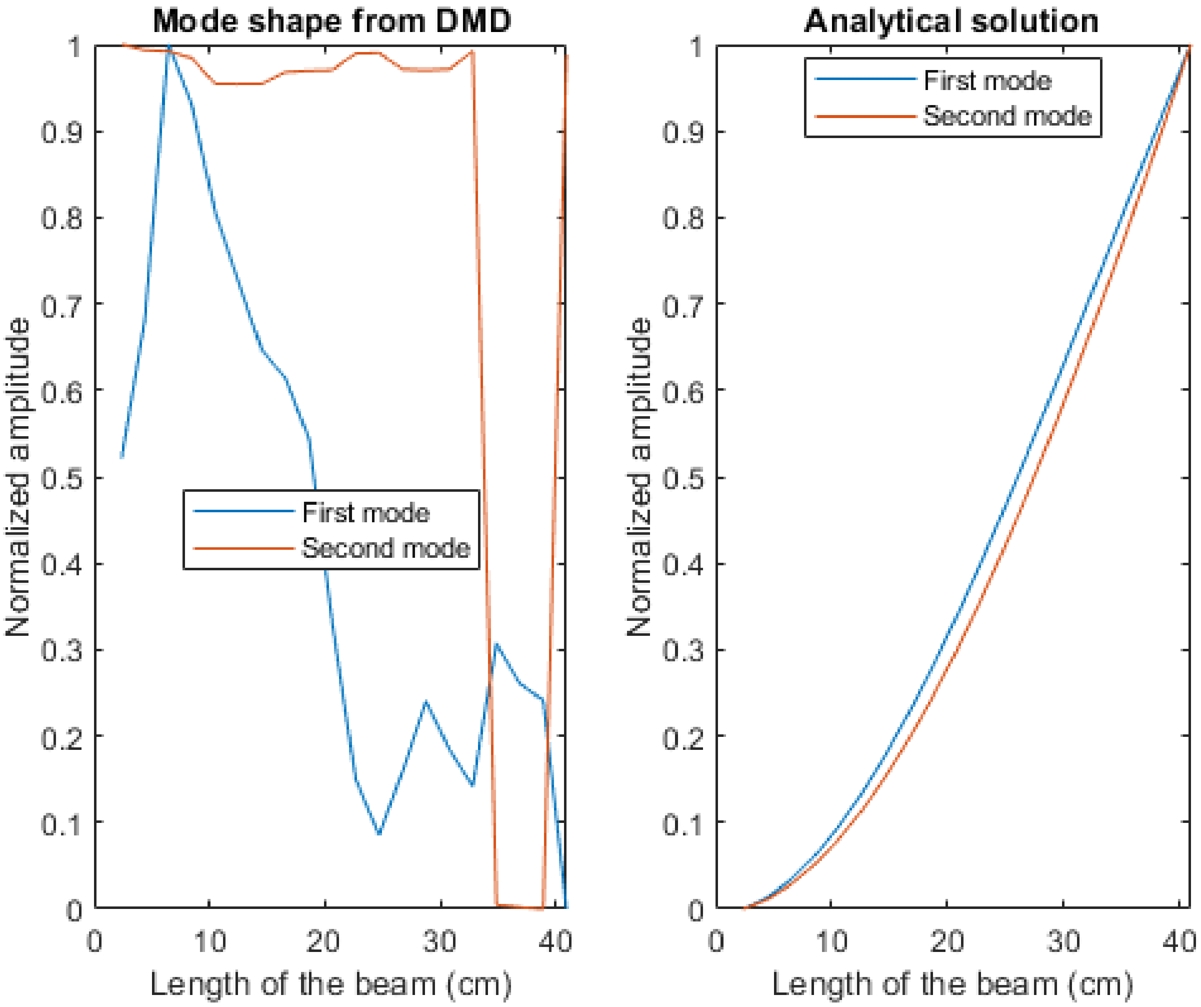


Figure B.15: DMD mode shape Threshold Method. *The analytical solution does not correspond to the FEM study. The DMD mode shape cannot be extracted successfully.*