

CRANFIELD UNIVERSITY

Thierry El Sayah

Phase-Based Motion Magnification in outdoor conditions: Robustness  
and Natural Frequency extraction.

School of Aerospace, Transportation and Manufacturing  
MSc by Research

MSc  
Academic Year: 2021 – 2022

Supervisor: Dr.Ivan Petrunin  
Associate Supervisor: Dr. Luca Zanotti Fragonara  
12/2021

CRANFIELD UNIVERSITY

School Of Aerospace  
MSc by Research

MSc

Academic Year 2021 - 2022

Thierry El Sayah

Phase-Based Motion Magnification in outdoor conditions: Robustness  
and Natural Frequency extraction

Supervisor: Dr.Ivan Petrunin  
Associate Supervisor: Dr. Luca Zanotti Fragonara  
12/2021

© Cranfield University 2021. All rights reserved. No part of this publication may be reproduced without the written permission of the copyright owner.

## **ABSTRACT**

In this study, Phase-Based Motion Magnification is combined with 2D Point Tracking to enhance data extraction under noisy outdoor conditions. The aim is to investigate whether the use of Phase Based Motion Magnification along with 2D PT can increase accuracy of frequencies extracted in robust conditions. The study presents three experiments: a fixed wing on a shaker machine, a light pole and a building. Frequencies are extracted in all three experiments and compared against previous research and finite element analysis. Finally, the study touches on ways to automate the suggested procedure.

Keywords:

2D Point Tracking, Convolutional Neural Networks, Denoising, Automation.

## **ACKNOWLEDGEMENTS**

The author would like to thank Dr.Ivan Petrunin, Dr.Luca Zanotti Fragonara, Dr.Marco Civera, Dr.Jafarali Parol and Dr. Gabriele Dessena for their valuable help during the study.

The author would also like to thank his family and friends who have always shown unconditional support, especially during this study.

# TABLE OF CONTENTS

|                                                              |    |
|--------------------------------------------------------------|----|
| ABSTRACT .....                                               | 3  |
| ACKNOWLEDGEMENTS .....                                       | 4  |
| TABLE OF CONTENTS .....                                      | 5  |
| LIST OF FIGURES.....                                         | 7  |
| LIST OF TABLES.....                                          | 9  |
| LIST OF EQUATIONS.....                                       | 10 |
| LIST OF ABBREVIATIONS.....                                   | 10 |
| 1 Introduction .....                                         | 11 |
| 1.1 Natural Frequency measurement .....                      | 13 |
| 1.2 Theoretical Background.....                              | 14 |
| 1.2.1 Phase-Based Motion Magnification.....                  | 14 |
| 1.2.2 2D Point Tracking.....                                 | 17 |
| 1.2.3 Convolutional Neural Network.....                      | 19 |
| 1.2.4 Denoising Convolutional Neural Network.....            | 20 |
| 1.3 Research Motivation .....                                | 21 |
| 1.4 Research Contribution .....                              | 21 |
| 2 State of art Review .....                                  | 21 |
| 3 Aim, Objectives and Methodology.....                       | 37 |
| 3.1 Research aim.....                                        | 37 |
| 3.2 List of objectives to achieve .....                      | 37 |
| 3.3 Methodology .....                                        | 38 |
| 3.3.1 Laboratory Experiment Procedure & Goal .....           | 39 |
| 3.3.2 Finite Element Analysis Procedure & Goal .....         | 41 |
| 3.3.3 Peak Detection.....                                    | 45 |
| 3.3.4 Apply different noise levels and compare results ..... | 49 |
| 3.3.5 Denoising using CNN.....                               | 51 |
| 4 Results and Discussion.....                                | 52 |
| 4.1 Laboratory results .....                                 | 52 |
| 4.2 FEA results .....                                        | 55 |
| 4.3 Light Pole results .....                                 | 55 |
| 4.4 Building Context & Results .....                         | 61 |
| 4.5 Denoising CNN results .....                              | 64 |
| 5 Automation of Procedure.....                               | 66 |
| 5.1 Challenges & Solutions in automation .....               | 66 |
| 5.2 Proposed automation.....                                 | 67 |
| 5.2.1 Motion Parallax .....                                  | 67 |
| 5.2.2 Alpha: Motion Amplification factor .....               | 68 |
| 5.2.3 Sigma: Blurring factor.....                            | 70 |
| 6 Conclusion, Potential applications & Future work .....     | 74 |
| 7 REFERENCES.....                                            | 77 |
| 8 APPENDIX .....                                             | 79 |
| 9 Matlab code.....                                           | 81 |
| 9.1 Signal to noise ratio .....                              | 81 |

|                                    |    |
|------------------------------------|----|
| 9.2 Add noise and denoise.....     | 81 |
| 9.3 PBMM & 2D PT.....              | 81 |
| 9.3.1 Point Tracking .....         | 83 |
| 9.3.2 Filtering.....               | 85 |
| 9.4 Train dCNN on custom data..... | 86 |
| 9.4.1 Test custom network .....    | 87 |
| 9.5 Blind identification .....     | 87 |
| 10 Publishing.....                 | 90 |

## LIST OF FIGURES

|                                                                                                                                |    |
|--------------------------------------------------------------------------------------------------------------------------------|----|
| Figure 1 Tacoma Narrows bridge twisting, allthatsinteresting .....                                                             | 11 |
| Figure 2 Tacoma Narrows bridge collapse, allthatsinteresting .....                                                             | 12 |
| Figure 3 PBMM procedure, <i>Wadhwa et al.</i> .....                                                                            | 14 |
| Figure 4: Frame N, known intensity at pixel (a) and known brightness variation around it .....                                 | 17 |
| Figure 5: Frame N + 1, known intensity at pixel (b), can deduce direction of movement through known brightness variation. .... | 18 |
| Figure 6 Convolution operation .....                                                                                           | 19 |
| Figure 7 Patch-based image processing <sup>12</sup> .....                                                                      | 20 |
| Figure 8 Instantaneous frequency plot .....                                                                                    | 22 |
| Figure 9 Suggested video processing procedure, Yang et al., 2017 .....                                                         | 24 |
| Figure 10 Frames from Motion Magnified Video (row 1) & corresponding mode shapes (row 2).....                                  | 25 |
| Figure 11 Comparison between Linear and Phase-Based approach .....                                                             | 26 |
| Figure 12 Overview of experimental procedure. Eitner et al., 2020.....                                                         | 27 |
| Figure 13 Summary of motion magnification steps, Ce Liu et al. ....                                                            | 28 |
| Figure 14 Eulerian video magnification steps.....                                                                              | 29 |
| Figure 15 RMSE vs Alpha at different noise levels. ....                                                                        | 29 |
| Figure 16 Memorial bridge and camera position .....                                                                            | 30 |
| Figure 17 Cantilever beam deflection shapes: Camera measurements vs accelerometers.....                                        | 31 |
| Figure 18 Ashim Khadka et al., experimental setup .....                                                                        | 33 |
| Figure 19 C. Z. Dong et al., experimental setup .....                                                                          | 34 |
| Figure 20 Vision based vs sensor measurement no vapor.....                                                                     | 34 |
| Figure 21 Vision based vs sensor measurement under vapor .....                                                                 | 35 |
| Figure 22 Doppler system for vibration monitoring.....                                                                         | 35 |
| Figure 23 Infrared thermography for vibration identification .....                                                             | 36 |
| Figure 24 Methodology Flowchart .....                                                                                          | 39 |
| Figure 25 Laboratory setup along with points monitored.....                                                                    | 39 |
| Figure 26 Different angle of lab setup .....                                                                                   | 40 |
| Figure 27 Light Pole studied.....                                                                                              | 41 |

|                                                                                                    |    |
|----------------------------------------------------------------------------------------------------|----|
| Figure 28 Dimensions of Light Pole .....                                                           | 42 |
| Figure 29 Light Pole FE model      Figure 30 Light pole mesh.....                                  | 44 |
| Figure 31 Trendline of data used for peak extraction .....                                         | 45 |
| Figure 32 Blind peak detection result .....                                                        | 46 |
| Figure 33 Frequency measurement : Light pole - video 1 .....                                       | 48 |
| Figure 34 Frequency measurement : Light pole - video 2 .....                                       | 48 |
| Figure 35 Displacement vs time of 2 points monitored: vision based approach .....                  | 52 |
| Figure 36 2D PT + PBMM results.....                                                                | 53 |
| Figure 37 Accelerometer results.....                                                               | 54 |
| Figure 38 Wing stabilization diagram.....                                                          | 54 |
| Figure 39 Image quality before PBMM processing .....                                               | 59 |
| Figure 40 Image quality after PBMM processing .....                                                | 59 |
| Figure 41 PBMM with $\alpha = 1$ .....                                                             | 60 |
| Figure 42 Satellite view of building studied.....                                                  | 62 |
| Figure 43 Building studied .....                                                                   | 62 |
| Figure 44 Motion Parallax.....                                                                     | 67 |
| Figure 45 Light pole width in pixels.....                                                          | 68 |
| Figure 46 Building width in pixels .....                                                           | 69 |
| Figure 47 Plot of Sigma vs SNR after PBMM: Light pole video .....                                  | 71 |
| Figure 48 Plot of Sigma vs SNR after PBMM: Building video .....                                    | 71 |
| Figure 49 Centered Fourier spectrum comparison of clear images (a-b) and blurred images (c-d)..... | 73 |
| Figure 50 Light Pole Mesh Properties .....                                                         | 80 |

## LIST OF TABLES

|                                                                                  |    |
|----------------------------------------------------------------------------------|----|
| Table 1 FFT and IFFT demonstration.....                                          | 16 |
| Table 2 Objective and corresponding methodology .....                            | 38 |
| Table 3 Wrought aluminum alloy properties .....                                  | 43 |
| Table 4 Aluminum alloy properties .....                                          | 43 |
| Table 5 Frequencies extracted from multiple videos using 2D PT. ....             | 47 |
| Table 6 FEA results.....                                                         | 55 |
| Table 7 Results comaprison between FEA and 2D PT .....                           | 55 |
| Table 8 Visual comparison of different Signal to Noise Ratio .....               | 56 |
| Table 9 2D PT results with and without PBMM, Light pole video 1. ....            | 57 |
| Table 10 2D PT results with and without PBMM, Light pole video 2 .....           | 57 |
| Table 11 PBMM before and after image quality comparison.....                     | 59 |
| Table 12 Number of points tracked vs Phase difference magnification factor ..... | 61 |
| Table 13 2D PT results with and without PBMM, Building video. ....               | 63 |
| Table 14 Results after using denoising CNN.....                                  | 64 |
| Table 15 SNR comparison .....                                                    | 65 |
| Table 16 Relation between Coverage and Amplification Factor .....                | 70 |
| Table 17 Sigma vs SNR: Light Pole .....                                          | 70 |
| Table 18 FM vs Sigma used.....                                                   | 74 |

## LIST OF EQUATIONS

|                                                               |    |
|---------------------------------------------------------------|----|
| Equation 1 Image reconstruction .....                         | 17 |
| Equation 2 2D Point Tracking Local Intensity Difference ..... | 18 |
| Equation 3 Maximum frequency detected .....                   | 49 |
| Equation 4 Noise calculation .....                            | 56 |
| Equation 5 SNR calculation .....                              | 56 |
| Equation 6 Image Quality measure (FM) .....                   | 72 |

## LIST OF ABBREVIATIONS

|       |                                                  |
|-------|--------------------------------------------------|
| PBMM  | Phase-Based Motion Magnification                 |
| 2D PT | 2D Point Tracking                                |
| SHM   | Structural Health Monitoring                     |
| FE    | Finite Element                                   |
| FEA   | Finite Element Analysis                          |
| FEMU  | Finite element model updating                    |
| CV    | Computer Vision                                  |
| FFT   | Fast Fourier transform                           |
| IFFT  | Inverse Fast Fourier transform                   |
| SHM   | Structural Health Monitoring                     |
| NN    | Neural Network                                   |
| CNN   | Convolutional Neural Network                     |
| dCNN  | Denoising Convolutional Neural Network           |
| BSS   | Blind Source Separation                          |
| FWL   | Frequency window length                          |
| alpha | Phase difference magnification factor            |
| sigma | Motion blurring factor in areas of low intensity |
| RMSE  | Root mean square error                           |
| NF    | Natural Frequency                                |
| FF    | Forced Frequency                                 |

# 1 Introduction

Structures around us, much like our bodies, require maintenance and health monitoring. The level of attention required increases with its age.

A Natural frequency is the rate at which an object vibrates when it is disturbed from its resting position. Different objects have different natural frequencies and can have one or more.

From the physics point of view, frequency is one of the characteristics of a wave, the number of times a certain point is recorded passing through a fixed reference in one second defines the frequency of that wave. Another characteristic of a wave is its amplitude, which refers to how high the peaks forming the wave are. Its measure from the center of the signal to the top of the peaks.

NF differs from a Forced frequency, the main difference being that a FF only occurs when applying a force at an object at a certain rate. Whenever the Forced frequency equals the Natural frequency of the object, the system will experience resonance, the amplitude of the resulting wave is equal to the sum of the amplitude of the FF and NF. In some cases resonance can cause disasters in structures, for example the collapse of the Tacoma Narrows bridge in Washington.



Figure 1 Tacoma Narrows bridge twisting, [allthatsinteresting](http://allthatsinteresting.com/tacoma-narrows-bridge)

The original bridge design was scrapped for a cheaper alternative, which had less resistance to bending and twisting. However the designers insisted it was structurally safe, but it wasn't. Even during the construction phase the deck would sway up and down significantly which earned it the name "Galloping Gertie".

Even after the bridge completion, it was still swaying. There were several attempts to fix the problem however none proved to be useful and on November 7, 1940 the bridge collapsed. In the investigation afterwards, many reasons for failure were discovered, a few of them being:

- The bridge had large width to depth ratio, 1 to 72, which meant it had low resistance to torsional twisting.
- Wind hitting the side of the bridge was separated which caused twisting. This twisting increased the wind separation which in turn created vortices that further increased the twisting.
- Cables at mid span slipped and led to two unequal segments holding the bridge, this changed the motion from vertical to torsional.

The change of motion from vertical to torsional, add to it that the wind hit the bridge at frequencies close to its natural frequencies made the structures' movement become self-generating. The wind was no longer in control, it was the bridge deck generating the forces.



Figure 2 Tacoma Narrows bridge collapse, [allthatsinteresting](https://www.allthatsinteresting.com/tacoma-narrows-bridge-collapse)

## 1.1 Natural Frequency measurement

Natural Frequency measurement is an important tool used for health monitoring in multiple fields, ranging from civil structures and infrastructures to mechanical systems. Frequency measurement has many benefits, one of them being damage detection, a structure that has sustained some kind of damage would display different frequencies than if it wasn't damaged. Another benefit is Finite Element model updating, one can compare results of frequency measurement of the built structure against results from the design phase of the structure and then perform model updating.

Typical methods of NF measurement include high-speed cameras, laser vibrometer and contact sensors. These devices provide highly accurate results, however; they are expensive, not straightforward to operate, some require ground preparation and provide discrete results (i.e. results from certain points on the structure) .

Computer Vision-based techniques solve these issues, they provide a continuous way of extracting results at a relatively low cost compared to previous techniques and require no ground preparation.

Some examples of CV based techniques include the Lukas-Kanade optical flow algorithm, Target Tracking and 2D point tracking. These techniques still suffer whenever they process low quality videos, in particular noisy videos. Phase-Based Motion Magnification is a video processing technique that has been proposed in recent years.

PBMM was firstly used to manipulate small motions in videos, it has been used in many industries ranging from healthcare to engineering. This method has had extensive testing indoors where variables like lighting, wind & stability can be fully controlled. However outdoor testing of this method still lacks. The main benefit of this method is that its highly insensitive to noise.

## 1.2 Theoretical Background

### 1.2.1 Phase-Based Motion Magnification

Phase-Based Motion Magnification, as the name implies, works on the phase of the image. The phase of the image is obtained by applying 2D Fourier transform on the image, which takes it from the time domain to the frequency domain. This process gives us two outputs, the phase and the magnitude. Each contain specific information relating to the original image. The phase contains structural information while the magnitude holds frequency amplitude information.

Given a certain video, PBMM will take the first frame as reference, compute its phase using 2D Fourier Transform and then calculate the phase difference between it and all remaining frames. After that, multiple copies of each phase are created at different scales and orientation using steerable pyramids, then the phases are band passed for the desired frequencies. The output phase is then magnified by the desired amplification factor  $\alpha$ . The video is then reconstructed using the modified phase difference which gives us the video with magnified motion.

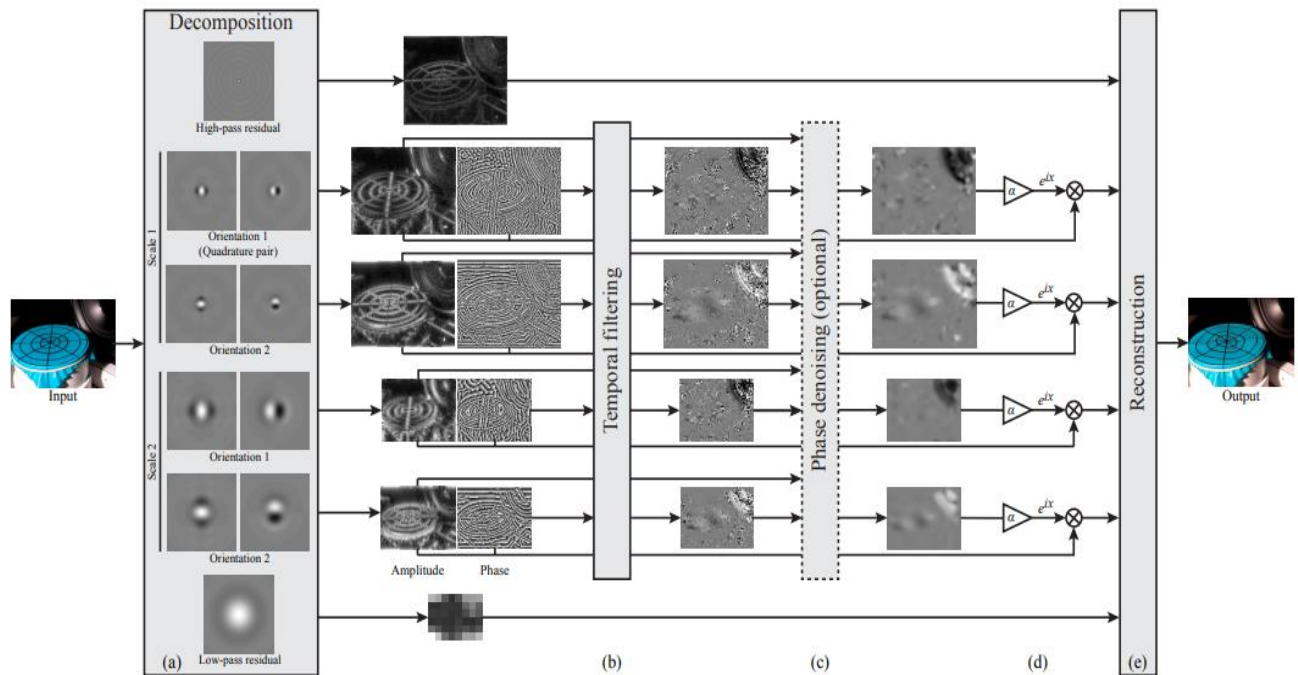


Figure 3 PBMM procedure, *Wadhwa et al.*

In the following article, “An Experimental Validation of Phase-Based Motion Magnification for Structures with Developing Cracks and Time-Varying

*Configurations” (Civera et al., 2021).* A very clear explanation of PBMM is provided in which they briefly describe the mathematics behind PBMM.

As you can see in table 1 below, we have two outputs from performing Fast Fourier Transform on the image of Burj Khalifa, one is for the magnitude and the other is for the phase. Reconstructing the images, by performing Inverse Fourier Transform, from phase and magnitude only provides us with the images in the far right column. The magnitude only part does not provide much meaning to the image, however in the phase only, one can clearly see the outline of the structure.

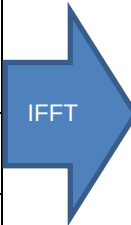
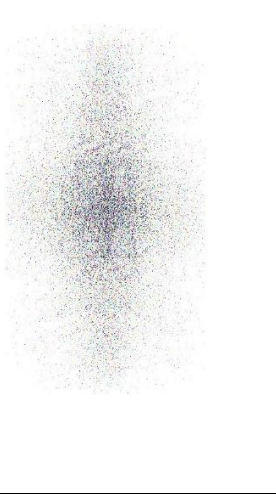
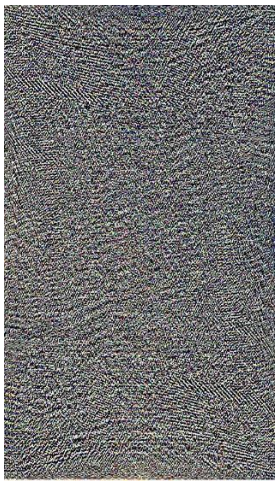
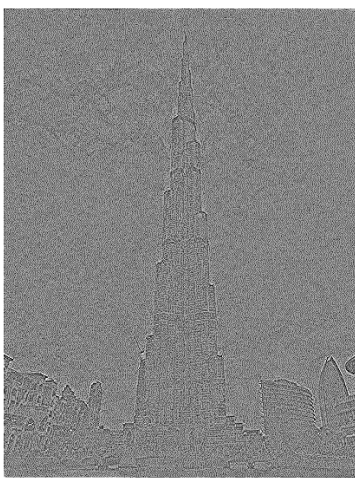
| Original Image                                                                     |                                                                                    | FFT output                                                                          |                                                                                      | Reconstruction                                                                        |
|------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
|  |  | Magnitude                                                                           |  | Magnitude only                                                                        |
|                                                                                    |                                                                                    |    |                                                                                      |    |
|                                                                                    |                                                                                    | Phase                                                                               |                                                                                      | Phase Only                                                                            |
|                                                                                    |                                                                                    |  |                                                                                      |  |

Table 1 FFT and IFFT demonstration

It is the phase of the image that PBMM operates on. To reconstruct the original image after modification, the following equation is applied:

### Equation 1 Image reconstruction

$$I_{Original} = I_{Magnitude} * e^{(1i * I_{Phase})} \quad 1$$

Where:

$I_{Original}$  = Original image to be reconstructed

$I_{Magnitude}$  = Magnitude output of FFT

$I_{Phase}$  = Phase output of FFT

### 1.2.2 2D Point Tracking

The 2D PT is based on Lucas-Kanade optical flow algorithm ([Lucas-Kanade in a Nutshell, Prof. Dr. Raul Rojas](#)), it works in the following way, knowing the pixel intensity of a certain pixel (x,y) in a frame F1, the increase in brightness per pixel in the x and y directions,  $I_x$  and  $I_y$  respectively. And finally the pixel intensity of pixel (x,y) in the succeeding frame, one can make a sensible assumption as to how the scene has shifted.

For example, if pixel(x,y) has an intensity of  $a = 100$  in frame 1 and an intensity of  $b = 200$  in frame 2, and knowing that going right in the x direction brightness increases by  $I_x$  per pixel and going down in the y direction brightness increases by  $I_y$  per pixel. One can safely assume that the scene has shifted to the left and up.

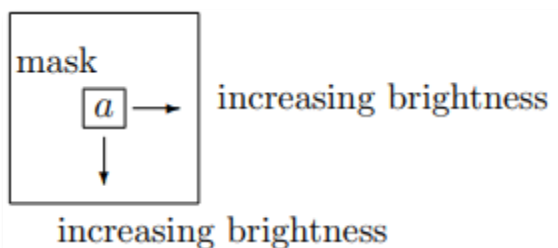


**Figure 4: Frame N, known intensity at pixel (a) and known brightness variation around it .**

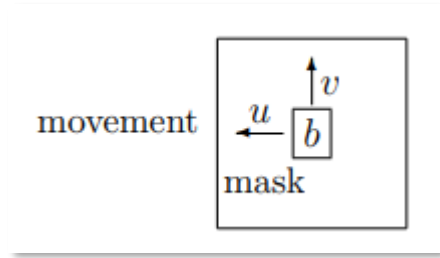


**Figure 5: Frame N + 1, known intensity at pixel (b), can deduce direction of movement through known brightness variation.**

Knowing the brightness increase per pixel, the following equation can be deduced:

**Equation 2 2D Point Tracking Local Intensity Difference**

$$I_x(x, y) * u + I_y(x, y) * v = b - a = I_t(x, y) \quad 2$$

Where:

u: Number of pixels moved in x direction

v: Number of pixels moved in y direction

$I_t$ : Total intensity difference

$I_x$ : Brightness per pixel in x direction

$I_y$ : Brightness per pixel in y direction

The above equation has 2 unknowns, u and v, one would need multiple equations in order to solve, hence the algorithm uses a neighbourhood of pixels (3\*3 for example) around the chosen pixel (x,y), this gives multiple equations and makes the system overdetermined and allows solving for the unknowns.

### 1.2.3 Convolutional Neural Network

[Convolutional neural networks](#) are a special type of neural networks that deal mainly with computer vision tasks like image classification and object recognition. A CNN is made up of the following three main types of layers:

1. Convolutional layer
2. Pooling layer
3. Fully connected layer

The convolutional layer takes the input, an image, and using a filter that will move across the image, a feature map will be created. This process is known as convolution. Each filter will extract a certain feature depending on the values it contains.

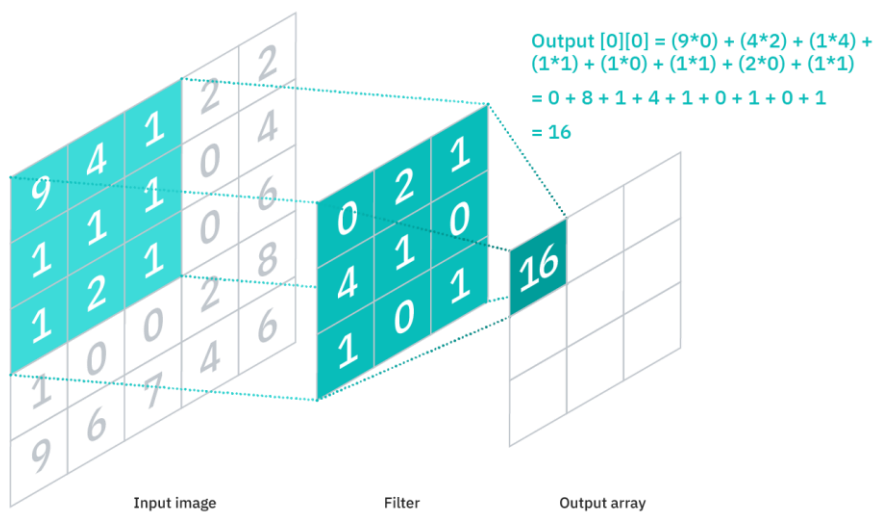


**Figure 6 Convolution operation**

The pooling layer performs down sampling or dimensionality reduction on the image. It does so using one of the two following ways:

1. Max pooling layer: In this layer, the filter kernel will sweep across the image, retaining the maximum value found within the kernel boundaries. This value will be used for output and the process continues on other parts of the image.
2. Average pooling layer: In this layer, the filter kernel will simply calculate the average value within the boundaries of the filter and use this value as output.

The values contained in each filter are updated after each iteration using backpropagation.

The output is flattened after the pooling layer and fed into the fully connected (FC) layer. The FC layer multiplies the input by a weight and adds a certain bias.

### 1.2.4 Denoising Convolutional Neural Network

Convolutional neural networks are a great asset for image processing and classification however they are computationally expensive, this cost increases when it comes to processing high resolution images.

High resolution images would require even more down sampling, only adding to the already expensive computation cost. Hence, patch based image processing was suggested, a CNN is trained on high resolution image patches for the purpose required.

This technique is used in a denoising convolutional neural network, the input image is processed in smaller patches, each patch is denoised on its own and then used to reconstruct the original image.

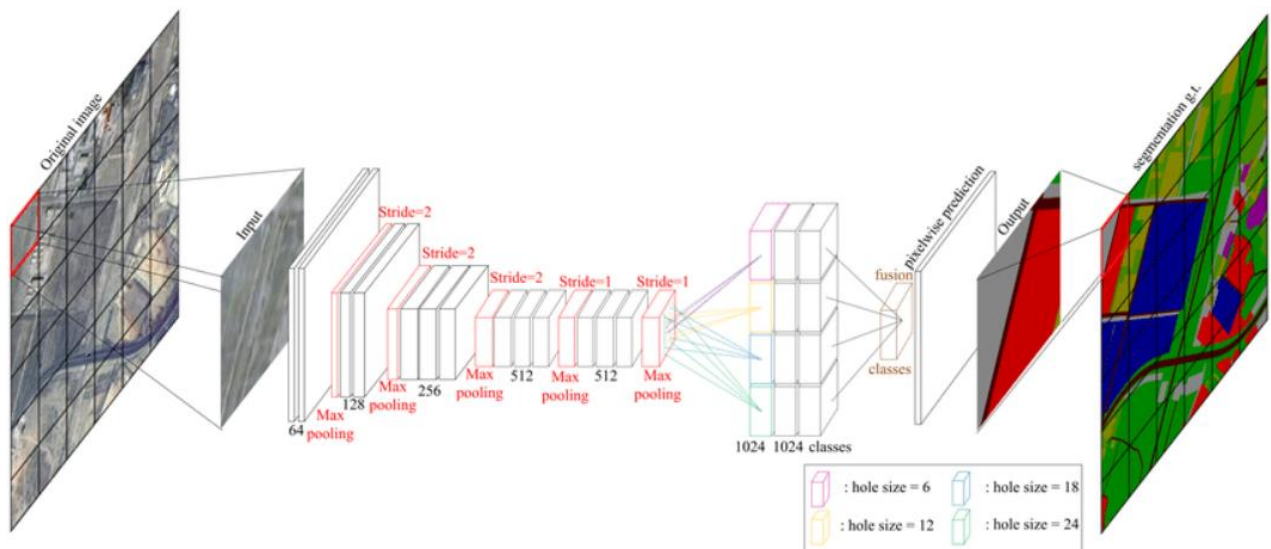


Figure 7 Patch-based image processing<sup>12</sup>

### **1.3 Research Motivation**

The motivation for this study is the will to create an alternative to expensive recording equipment and contact sensors used for measurements of all types. The two techniques used, 2D PT & PBMM, are both cutting edge vision based techniques whose interaction together is still to unravel.

There is a need to understand and enhance the interaction between 2D point tracking and Phase-Based Motion Magnification under different outdoor conditions while using commercial grade recording devices(mobile phones). Most measurements nowadays are made using expensive equipment which aren't accessible to everyone at all time.

The combination of the two methods above could prove to be useful and robust in outdoor conditions as it can be applied using relatively cheap devices. Hence, the motivation for this study is to learn and share the knowledge gained from understanding the interaction between PBMM & 2DPT, obtain an alternative to expensive equipment and to automate this procedure as automation would reduce the need for training on this technology.

### **1.4 Research Contribution**

This research contributes to 3 different areas, the first being the improved accuracy of the results of 2D PT under noisy conditions using PBMM. The second area of contribution is information obtained through testing about the usefulness & flexibility of PBMM in outdoor conditions. The final area of contribution relates to the automation of the procedures to be suggested later on in this study. By automation we are referring to automatic parameter selection depending on the situation at hand.

## **2 State of art Review**

Phase-Based Motion Magnification has been around for a while and many researchers have taken their time to explore its use cases, the following are some of them.

Firstly, *Civera et al. - 2021* validated PBMM for continuous Structural Health Monitoring, their aim was to identify the exact instance of damage occurrence to a structure by monitoring barely visible vibrations. For this aim, three different experiments were conducted, in the first experiment they monitored the vibration of a prototype wing spar fixed to a shaker machine, then in another run they added damage to the wing and recorded its behaviour again.

For the second experiment, they created a progressive increase in damage to a cantilever box beam, and vibration data was recorded using both contact sensors and camera recordings. For the third and final experiment the aim was to detect instantaneous changes in displacement and frequency, so a weight was added to the tip of a wing before shaking and then removed halfway through the experiment.

All three experiments were conducted in a laboratory with all variables like lighting and vibrations being controlled. PBMM was able to extract data as accurately as contact sensors. Also, extracted time histories, magnified by PBMM, were able to identify exact instant of structural changes.

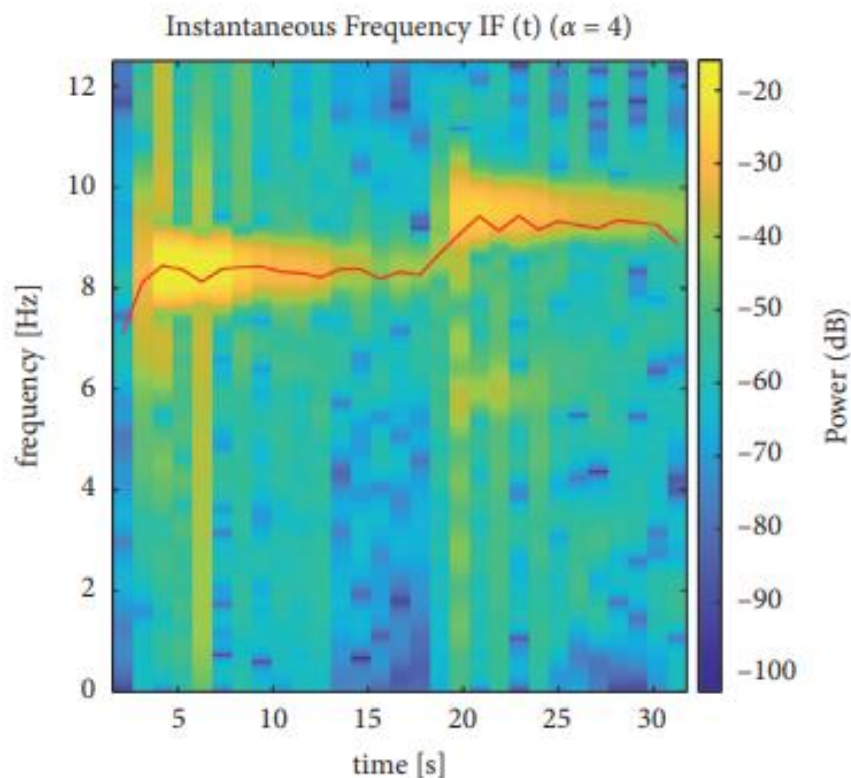


**Figure 8** Instantaneous frequency plot

The plot above shows the instantaneous frequency of the cantilever beam (experiment 3 in *Civera et al. - 2021*) , at around 17 seconds, the weight was removed and we can notice the change in instantaneous frequency from ~8.3 Hz to ~9.5 Hz.

Secondly, *Yang et al. – 2017* used PBMM for identification of natural frequency and mode shape extraction in output-only systems on a scaled building model in a laboratory experiment. PBMM was used in combination with Principal Component Analysis and Complexity Pursuit in order to achieve the above. Results were similar to accelerometer extracted data.

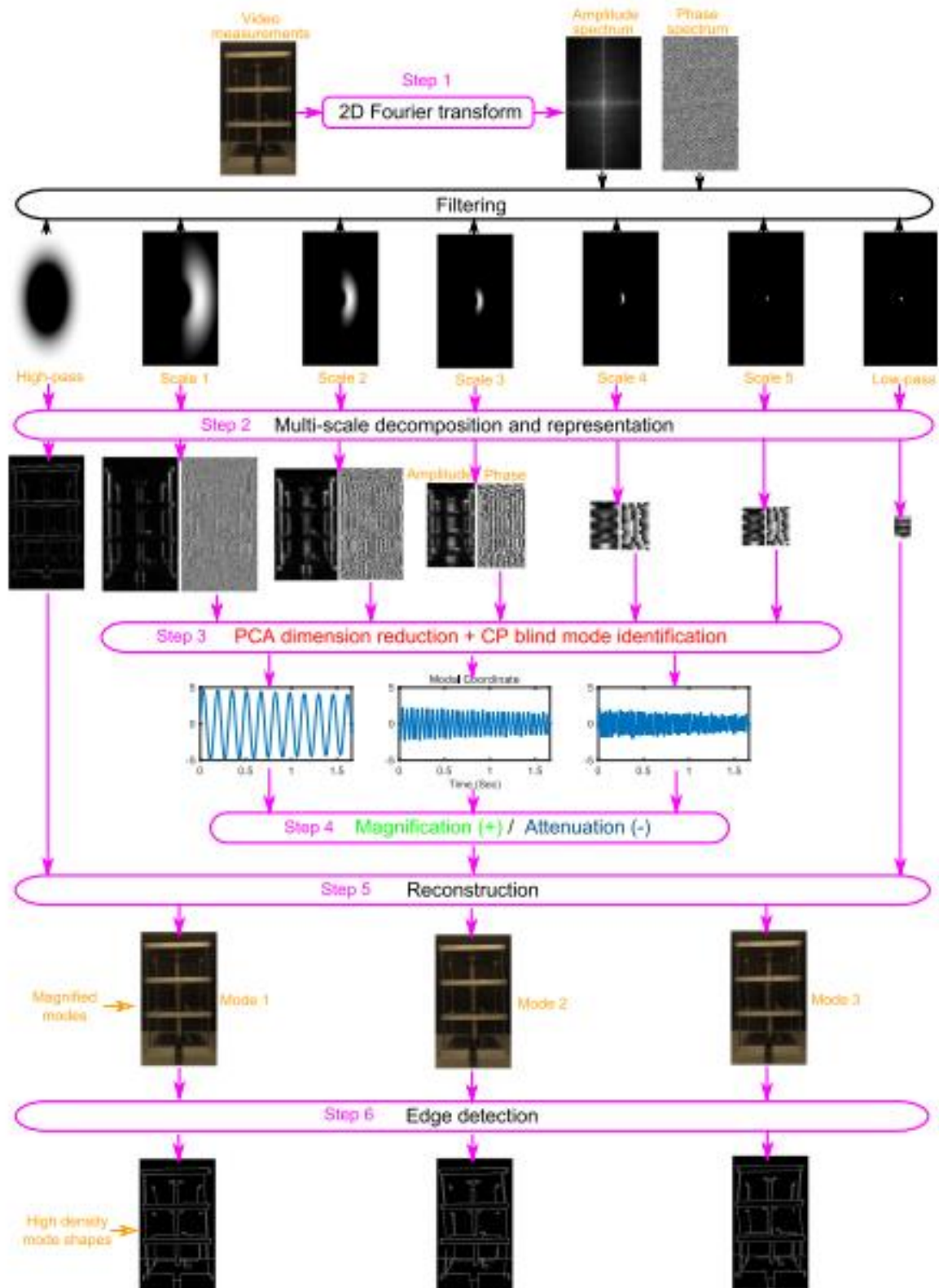


Figure 9 Suggested video processing procedure, Yang et al., 2017

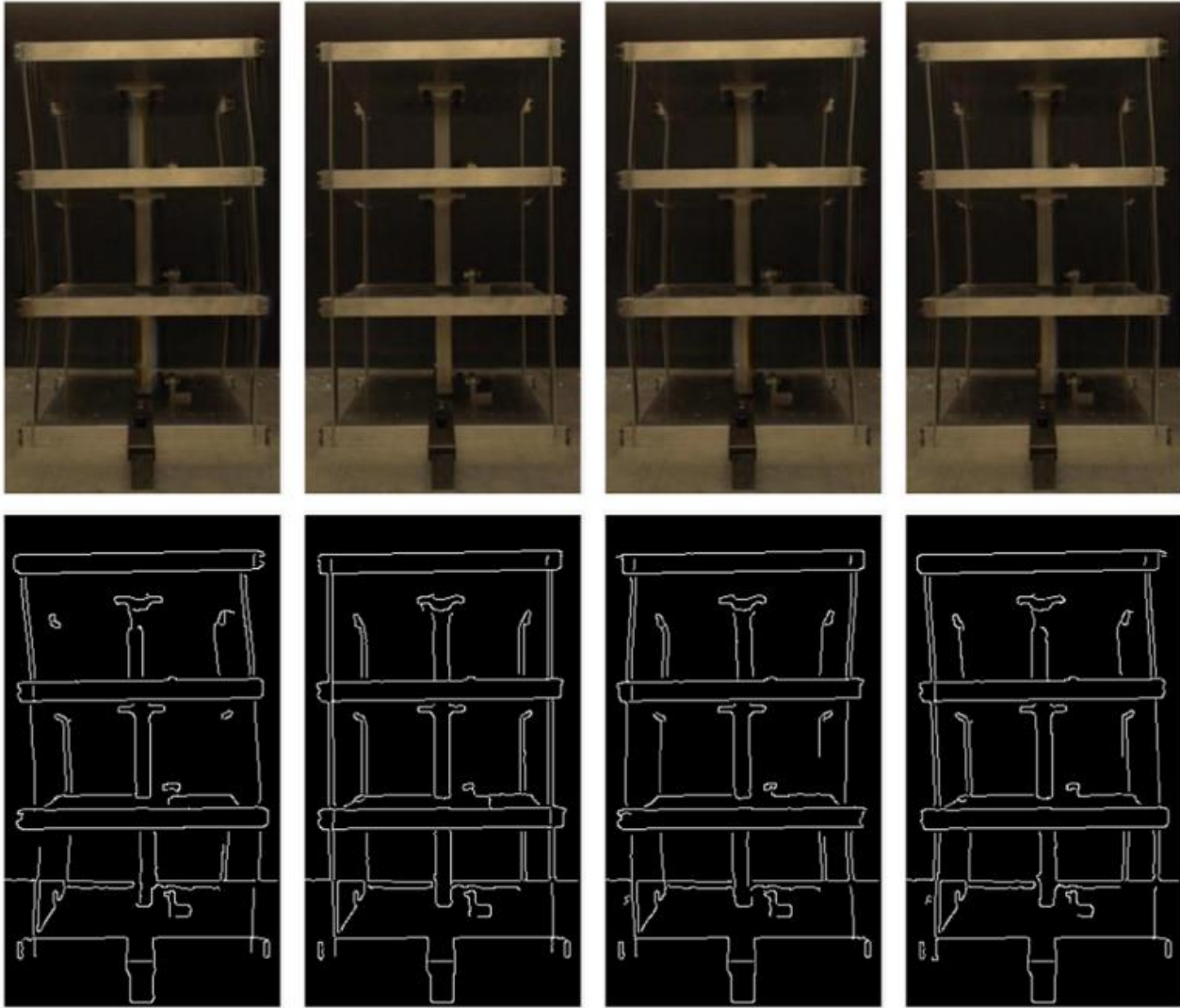


**Figure 10 Frames from Motion Magnified Video (row 1) & corresponding mode shapes (row 2)**

Through the procedure suggested in figure 9, Yang et al., 2017 were able to extract mode shapes of a model structure. This work is extremely interesting as it relied on blind source separation in order to blindly detect relevant frequencies.

In Figure 10, we can see the output of their processing in the second row. After motion magnifying the video around the relevant frequencies detected through BSS, edge detection was applied in order to extract mode shapes.

Thirdly, Wadhwa et al. – 2014 introduced PBMM and demonstrate the advantages it offers on different types of videos. However in this paper, no frequency measurements are done, only a demonstration of PBMM on videos recorded both inside and outside. They also touched on motion attenuation where setting  $\alpha$  to a negative value would zero out all phase changes over time.

The Eulerian approach for motion magnification performs better than the Lagrangian approach when it comes to moderate magnification and noisy inputs. PBMM further enhances the performance of the Eulerian approach by making it less sensitive to noise and support a larger amplification factor.

In the figure below, we notice a clear advantage in using phase based approach over the linear (Lagrangian) approach.

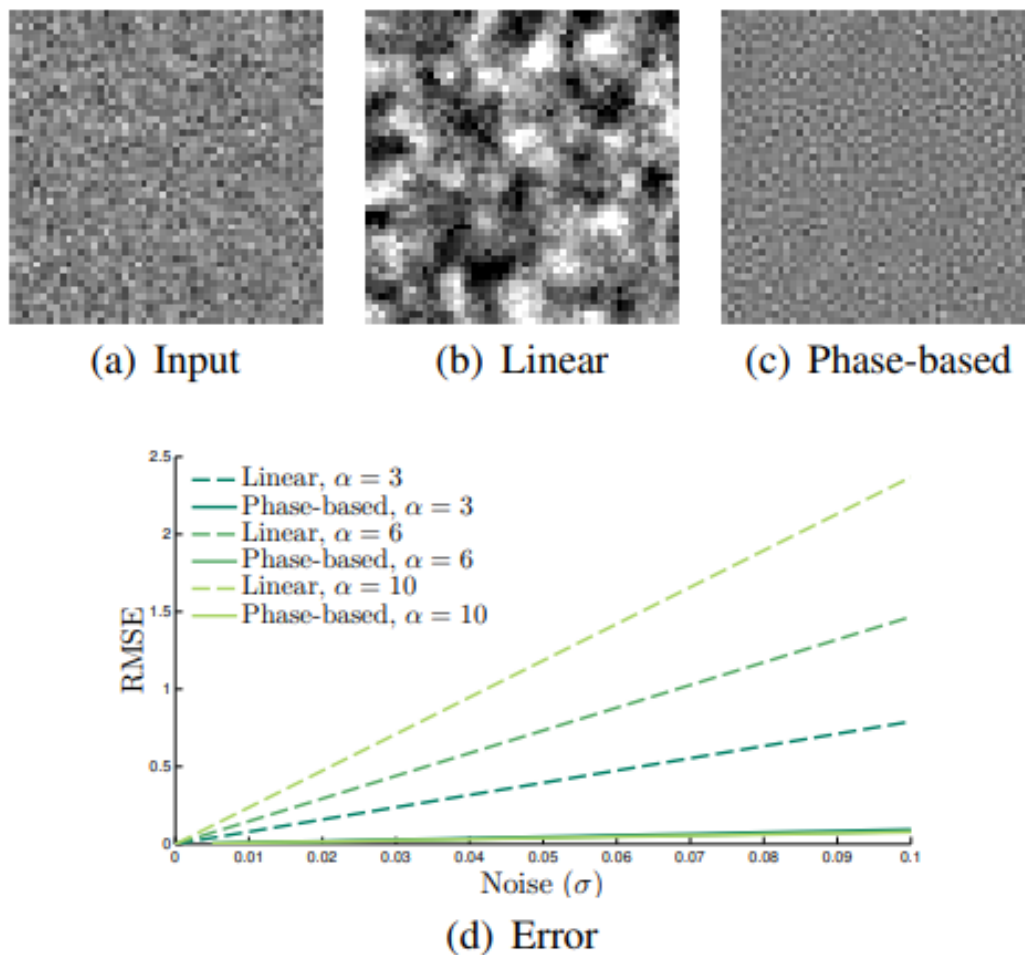


**Figure 11 Comparison between Linear and Phase-Based approach**

All studies mentioned above indicated the need for outdoor testing and a robustness check of Phase-Based Motion Magnification.

In the study by *Eitner et al., 2020*, 2D point tracking was used along with PBMM to extract frequencies and modes shapes of a vibrating nozzle. The experiment took place in a laboratory using a high speed camera.

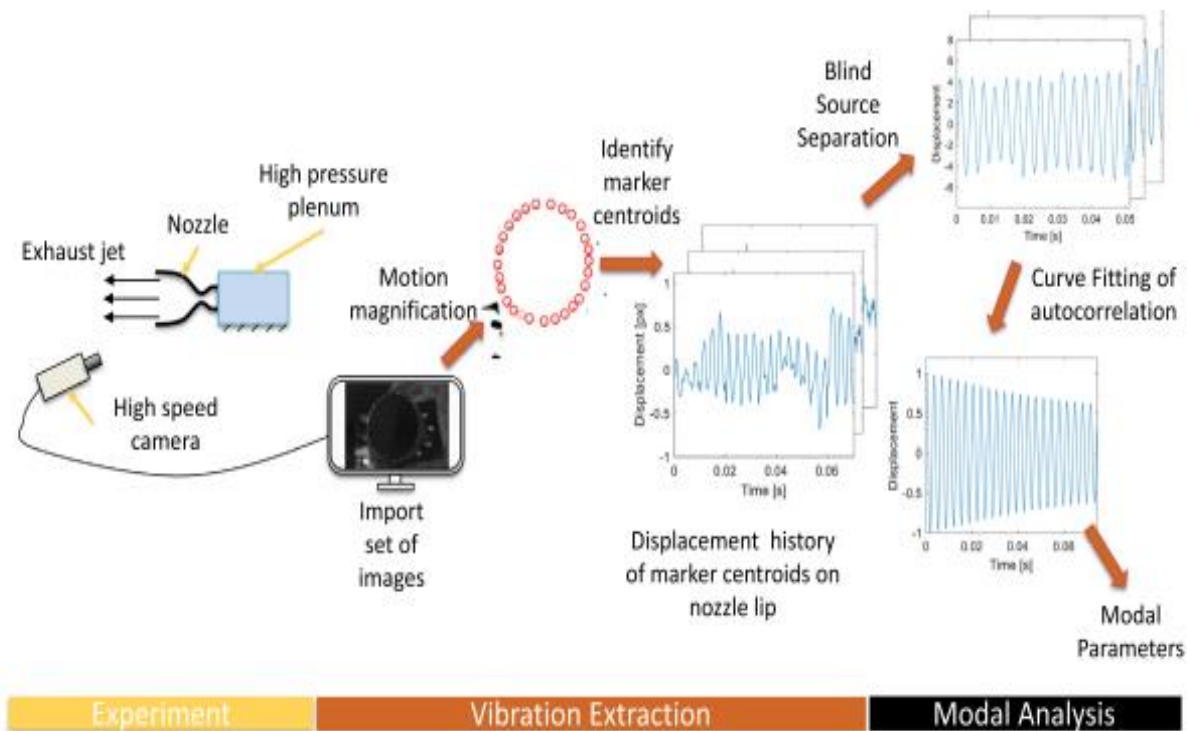


**Figure 12 Overview of experimental procedure. Eitner et al., 2020**

As they increased the magnification factor,  $\alpha$ , the results converged more and more to actual values, however the reason behind this behaviour was not discussed.

Ce Liu et al., in the paper titled *Motion Magnification*, presented a novel method for magnifying motion that is difficult to detect. It works by dividing the reference frame into groups, this segmentation happens based on colour, correlated motion and proximity. Then the user would select a certain group for magnification along with the magnification factor.

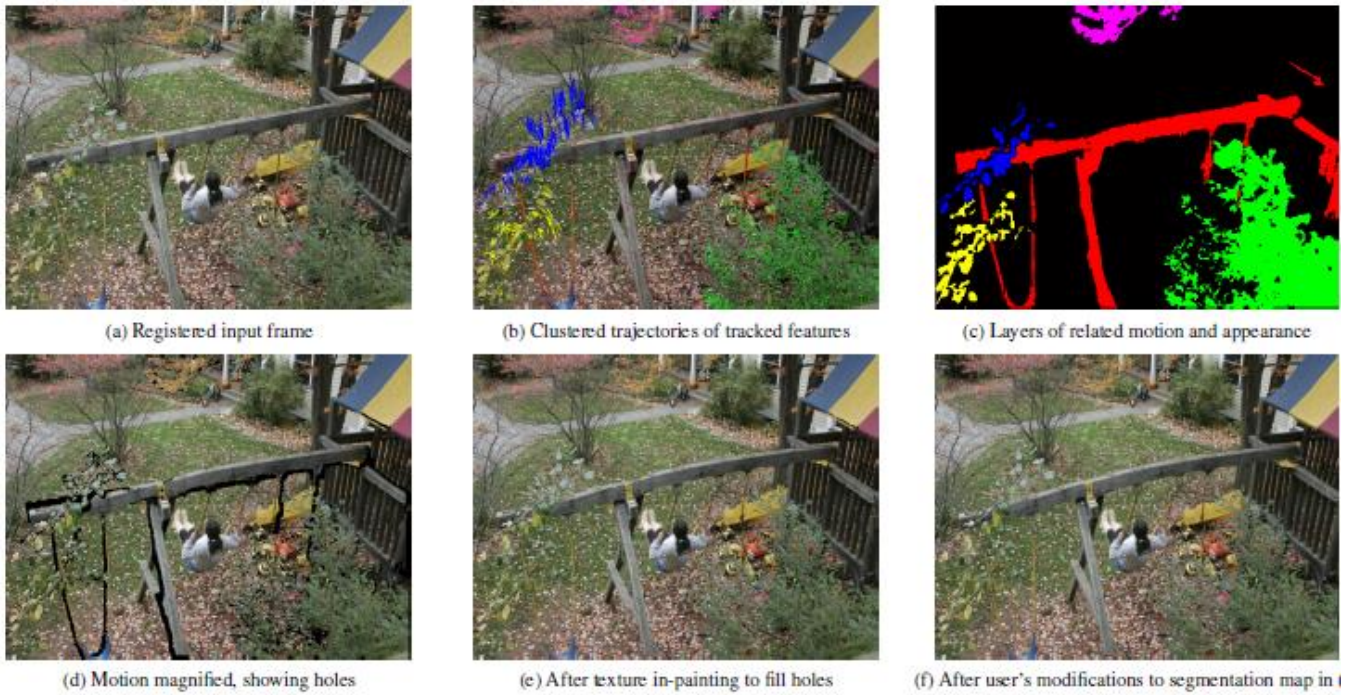


**Figure 13 Summary of motion magnification steps, Ce Liu et al.**

This is an interesting approach for magnifying small motions around us, however its performance when in noisy conditions is not mentioned, but one can assume that even with a certain level noise this method would still deliver reasonable results.

Another potential shortcoming of this method is that it has not been tested on buildings hence we don't know its ability to magnify small motions on a large scale. Also, it requires user intervention, from choosing the layers to magnify to setting the magnification factor.

In the paper titled *"Eulerian Video Magnification for Revealing Subtle Changes in the World"* by Hao-Yu Wu et al. They proposed a method named Eulerian Video Magnification to reveal unseen information like subtle motions and color amplification.

Their method works by decomposing the applying spatial decomposition onto the input video, then temporally filtering the desired frequencies and finally applying a magnification factor before reconstructing the video.

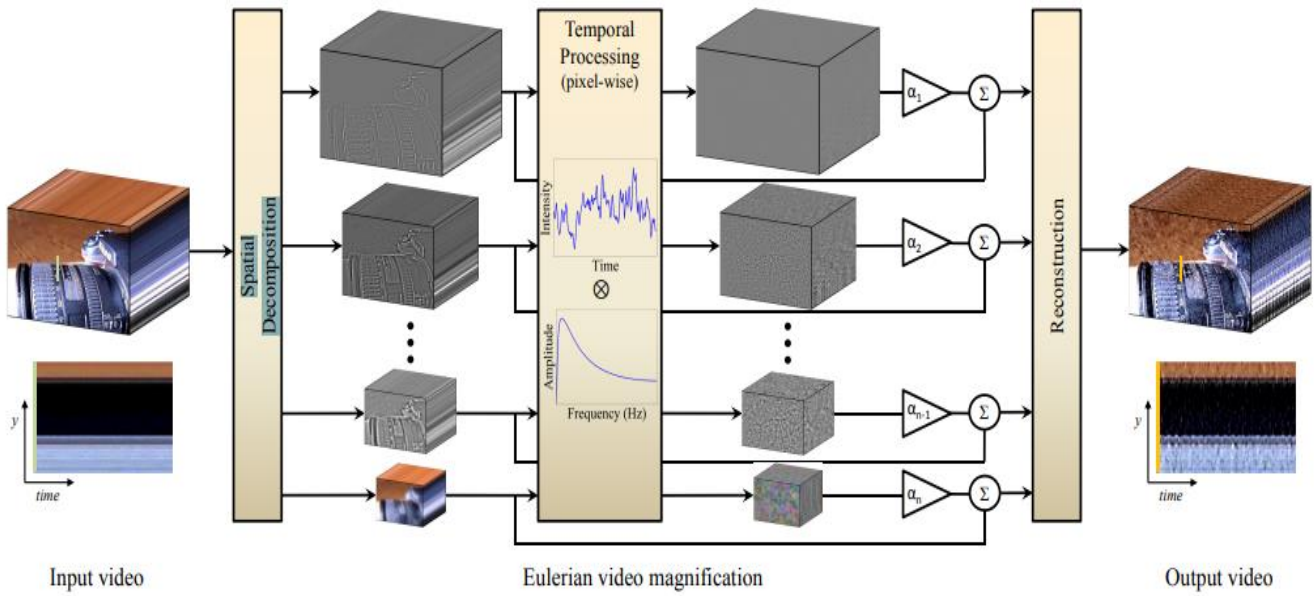


**Figure 14 Eulerian video magnification steps**

The limitations of this method are the following, firstly as the magnification factor alpha increases to high values, we risk increasing noise and creating artifacts in the reconstructed video.

Also, comparing this method to Lagrangian processing where motion is explicitly tracked, this method offers lower sensitivity to noise than the Lagrangian approach. This is due to the spatial filtering happening during spatial decomposition.

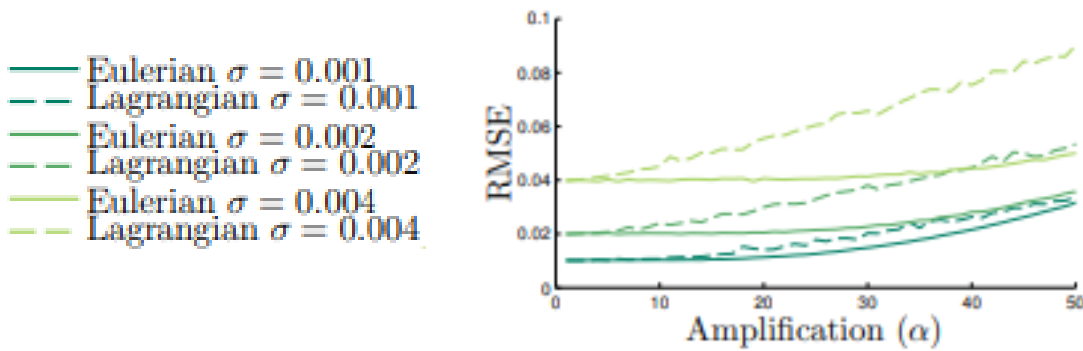


**Figure 15 RMSE vs Alpha at different noise levels.**

From the figure above, we notice that as the noise levels go up, RMSE for both the Lagrangian and Eulerian approach increase, however it is more stable and seems to converge for the Eulerian approach at higher amplification factors  $\alpha$ .

In the method suggested by Hao-Yu Wu et al., manual selection of the following parameters is needed, selection of amplification factor, selection of spatial frequency cut-off and selection of how  $\alpha$  is attenuated outside spatial wavelength, either zero or decrease linearly to zero.

Another study, *Justin G. Chen et al., 2018*, used phase-based motion magnification to identify natural frequencies of the Memorial Bridge in Portsmouth, New Hampshire. Videos were recorded then processed with PBMM, frequencies were extracted from displacement measurements and compared against sensors placed on the bridge and a finite element analysis.

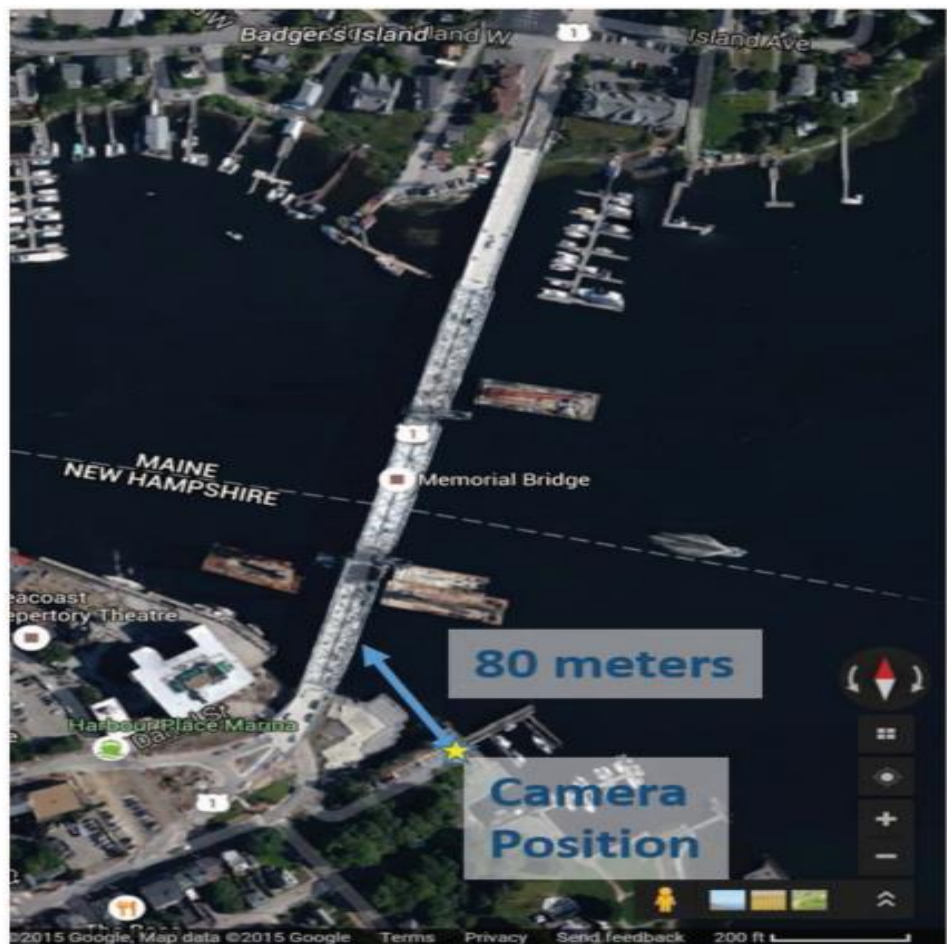


Figure 16 Memorial bridge and camera position

In their study, results from the vision based approach, sensors and FE model were in accord with each other, however the authors mentioned possible limitations of this vision based method, one of them being varying lighting conditions.

Hence why PBMM must be tested under different outdoor conditions.

Justin G. Chen et al., in their study titled “Modal identification of simple structures with high-speed video using motion magnification” utilized PBMM for mode shape detection. They performed laboratory experiments using high speed cameras on a cantilever beam and a pipe. Results obtained matched with ones from laser vibrometers and accelerometers.

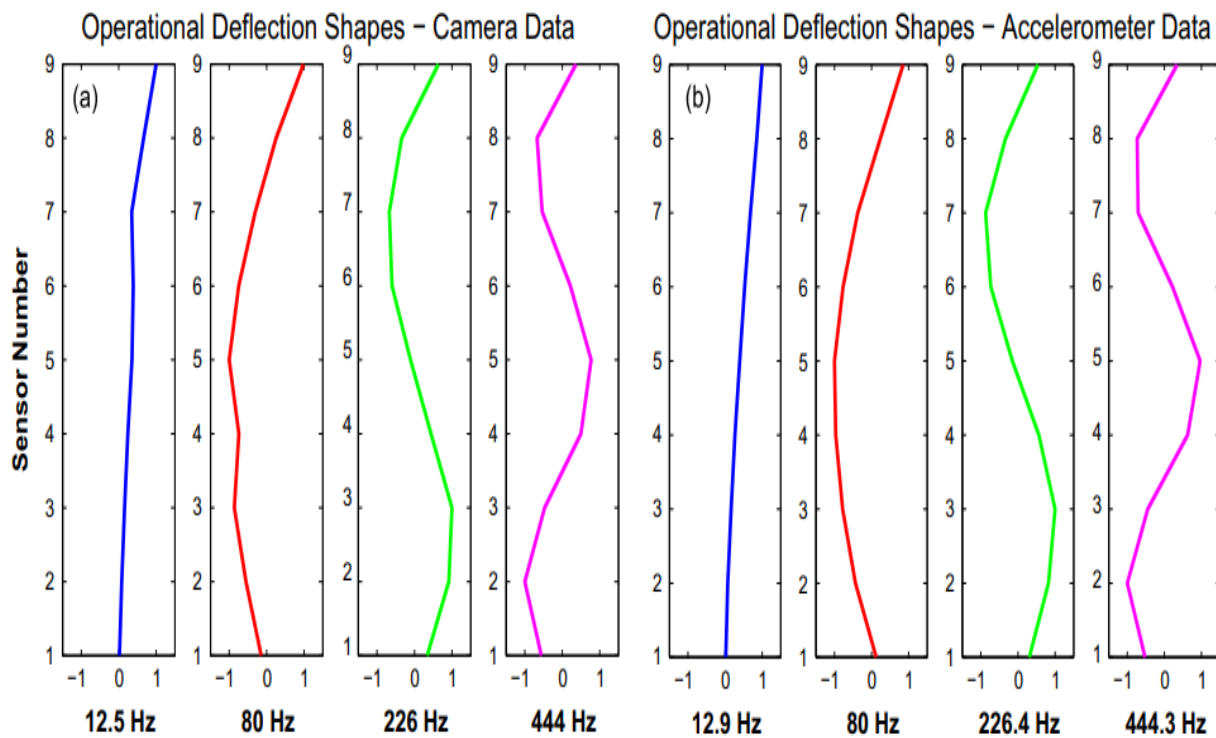


**Figure 17 Cantilever beam deflection shapes: Camera measurements vs accelerometers**

From the figure above it is clear that camera measurements are extremely close to accelerometer measurements which proves the eligibility of using PBMM for mode-shape extraction.

However, this paper also mentions the need for automatic mode shape extraction, outdoor testing in general, effects of camera movements and varying lighting conditions.

From what was presented above, we notice the need for robust testing of PBMM, ways of automating parameter selection and the lack of limit testing for 2D PT, all of which will be our aim for this study.

Other methods of non-destructive vibration monitoring were suggested, for example the study by Frank Neitzel et al., "VIBRATION MONITORING OF BRIDGES", 2011, they placed wireless accelerometers on the Komtur Bridge in Berlin, Germany. This bridge was chosen for its clearly perceptible vibrating behaviour.

The accelerometers were encased in a weatherproof box that can last up to 8 hours on battery. Multiple devices were placed at predetermined locations along both sides of the bridge.

The upside of this method is that it provides accurate results, however the downsides are the following, firstly we cannot obtain continuous results given that we can only place a certain number of sensors at discrete locations.

Secondly, it is both expensive and time consuming to setup and operate these sensors. Both of these downfalls are addressed in vision based vibration methods.

Ashim Khadka et al., in the study “Non-contact vibration monitoring of rotating wind turbines using a semi-autonomous UAV”, used unmanned aerial vehicles (UAV) for health monitoring of wind turbines

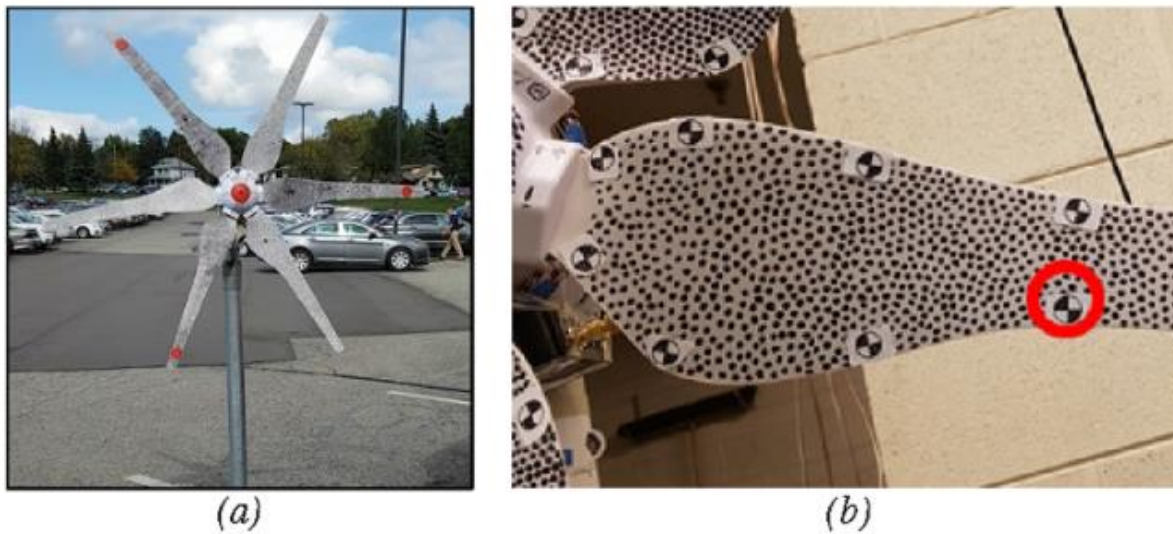


**Fig. 1.** (a) A six-blade wind turbine for photogrammetrical analysis, (b) Optical targets used for point tracking.

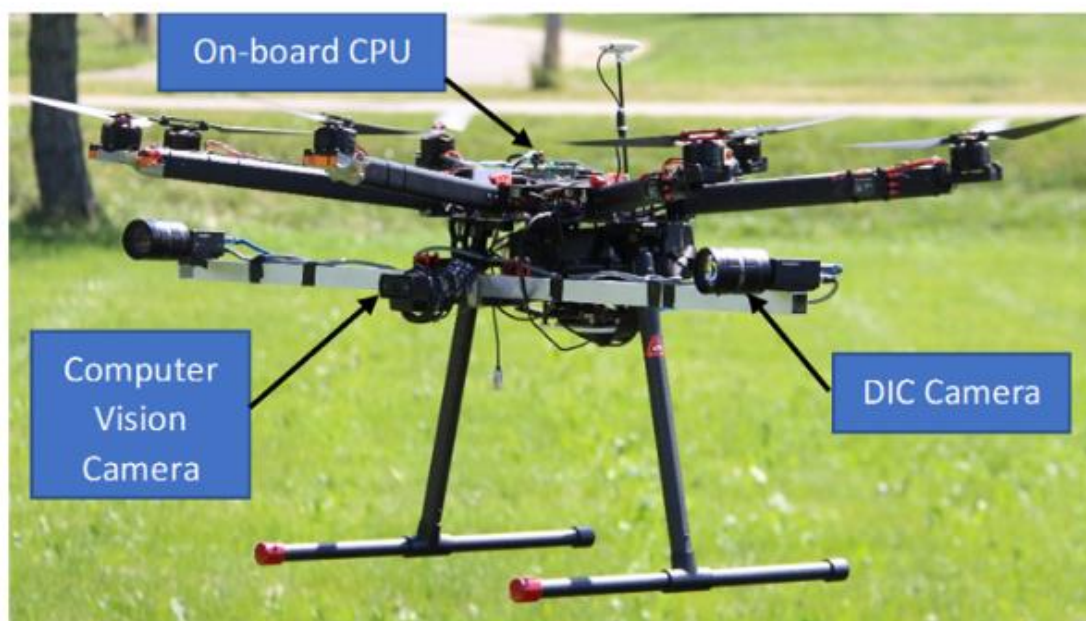


**Fig. 2.** Camera mounted on the drone for DIC measurements.

### Figure 18 Ashim Khadka et al., experimental setup

In the experiment they performed, although successful on a small scale, we notice an important point which is the use of markers on the wing of the turbine which might not be possible in real world applications. Other than that, the use of drones would increase the cost of monitoring and requires some specific knowledge.

In the study by C. Z. Dong et al., “Non-contact structural vibration monitoring under varying environmental conditions”, Multi point pattern matching was used for displacement measurement under varying levels of vapor.

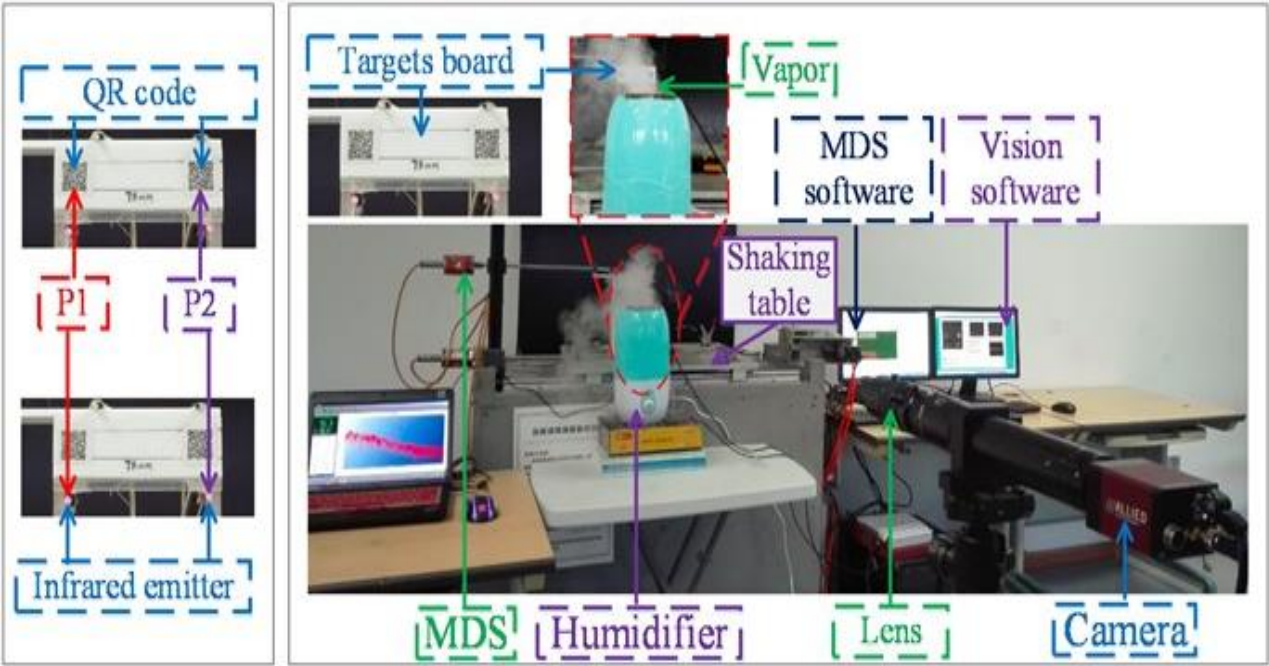


Figure 19 C. Z. Dong et al., experimental setup

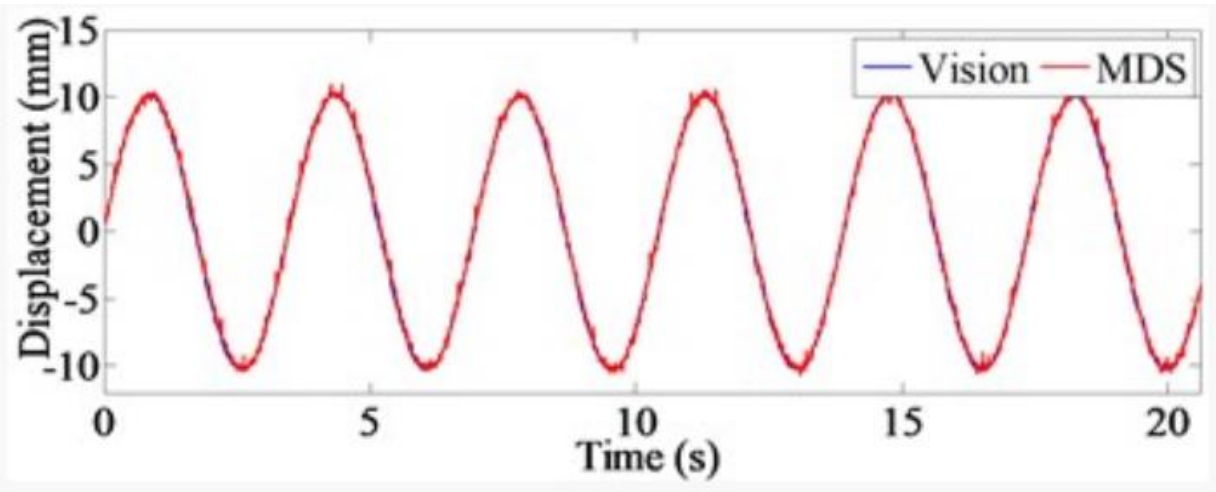


Figure 20 Vision based vs sensor measurement no vapor

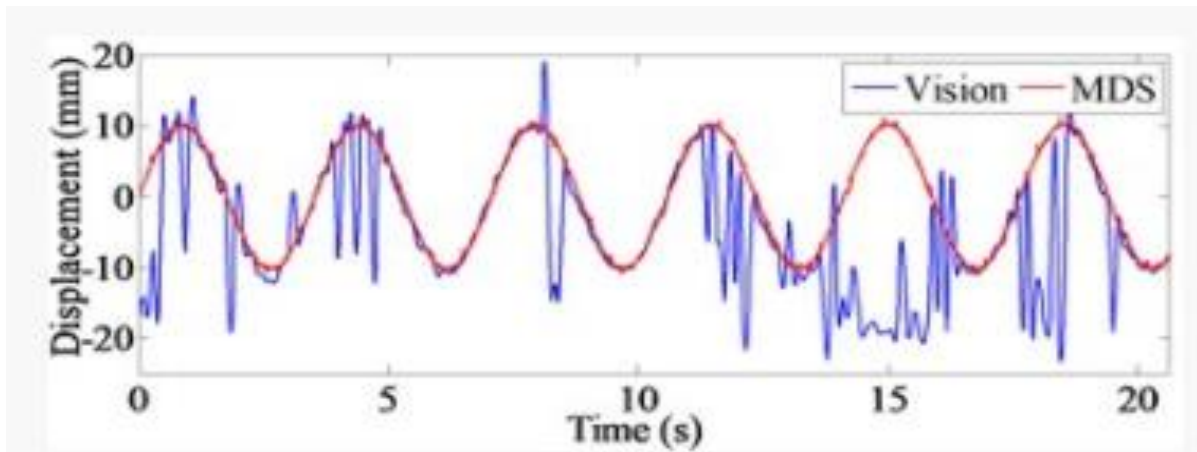


**Figure 21 Vision based vs sensor measurement under vapor**

We notice from the figure above that when vapor is present, the accuracy of the vision based approach drops drastically. However outdoor testing of this method is required.

Their study lines up with the direction this study is taking, which studying the effects of outdoor conditions on the accuracy of 2D PT and PBMM.

In the study by Raj Rakshit et al., doppler radar was used on objects to detect vibrational properties, mainly the vibration amplitude and frequency. The way this system works is by sending a signal  $T(t)$  towards the object studied, and then registering the received signal  $R(t)$  that has bounced off the object.

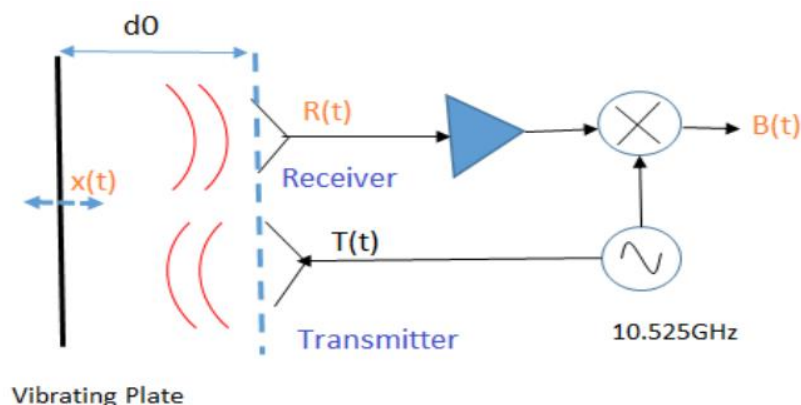


**Figure 22 Doppler system for vibration monitoring**

The received signal can then be studied and vibrational properties can be extracted.

The proposed method introduces robustness in the face of varying outdoor conditions especially lighting conditions as radar is not affected by it, also it can be applied using relatively low cost equipment.

In the study by Stephen M.Talai et al., 2019, infrared thermography was used to monitor the temperature change of a cantilever beam subject to vibration by a shaker machine, the temperature was monitored where the beam and a friction rod were touching.

the infrared camera measured temperatures over a period of time, from which we can extract frequencies by performing FFT.

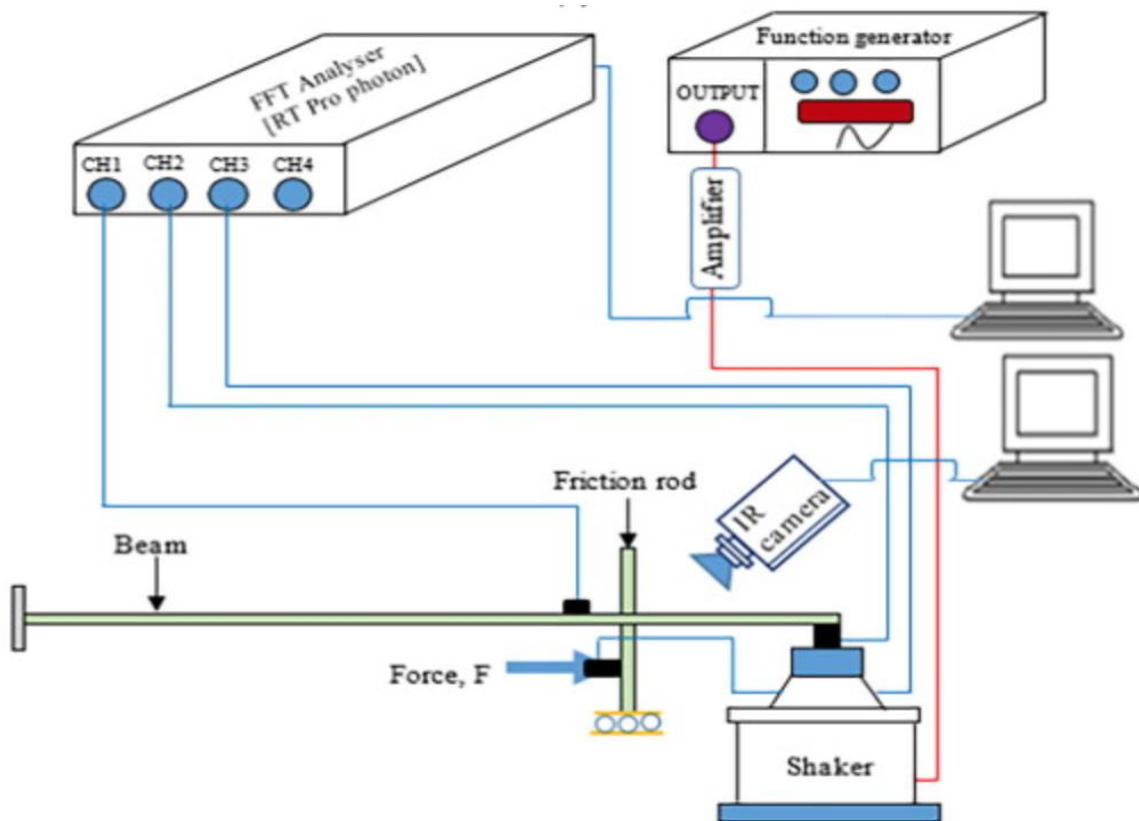


**Figure 23 Infrared thermography for vibration identification**

The proposed method is extremely interesting, especially for the mechanical and aerospace industries. However its application on large scale structures is yet to be tested, and might not be as easy as other contact free methods.

## **3 Aim, Objectives and Methodology**

### **3.1 Research aim**

In this study, the aim is to investigate and understand the interaction between 2D Point Tracking and Phase-Based Motion Magnification under different conditions, indoor but mainly outdoor, using commercial grade equipment. The reason for this is that professional equipment is not always available, is expensive and requires special attention in handling and usage. Also, the aim is to improve frequency extraction from noisy videos and to propose certain steps for the automation of procedure.

### **3.2 List of objectives to achieve**

1. Identifying potential area of improvements in vibration monitoring and frequency extraction.
2. Perform laboratory experiment in order to compare accuracy of 2D PT versus 2D PT combined with PBMM versus the accuracy of accelerometers, and understand the interaction between 2DPT and PBMM in indoor conditions before moving outdoor.
3. Obtain a baseline of frequencies for the light pole case by running modal analysis, and compare to the results obtained from the processing of videos using PBMM & 2D PT. This allows us to tune our vision based frequency extraction program.
4. Differentiate between actual frequencies and noise by trying different peak detection techniques and measuring & comparing their accuracy and practicality.
5. Extract & check accuracy of obtained results against previous research and finite element analysis, this step provides legitimacy to our results in case they match previous research & FEA.

6. Improve accuracy of results obtained by denoising data, then processing with 2DPT & PBMM and then comparing with our previously obtained results to check if the accuracy has improved.

### 3.3 Methodology

In order to accomplish the objectives stated previously, the following methodology is used.

| Objective                                                                                                                                                       | Methodology                                                                                                                                                                                                                                              |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Identifying potential improvements and gaps.                                                                                                                    | <ul style="list-style-type: none"> <li>Literature review.</li> <li>Understanding project requirements.</li> </ul>                                                                                                                                        |
| Measure accuracy of results extracted from PBMM & 2D PT against sensor measurements in lab experiment.                                                          | <ul style="list-style-type: none"> <li>Setup lab experiment</li> <li>Record measurements using accelerometers</li> <li>Extract results using PBMM &amp; 2DPT</li> </ul>                                                                                  |
| Finite element analysis of light pole, goal is to obtain credible frequency range to compare results obtained from the processing of videos using PBMM & 2D PT. | <ul style="list-style-type: none"> <li>Obtain dimensions of pole</li> <li>Vary material properties to get approximate range of frequencies.</li> <li>Record videos of light pole for later stage.</li> </ul>                                             |
| Explore different peak detection techniques, goal is to differentiate between noise and actual peaks of the system.                                             | <ul style="list-style-type: none"> <li>Plot frequencies obtained from PBMM and 2D PT</li> <li>Try different peak differentiation and detection techniques.</li> </ul>                                                                                    |
| Check accuracy of our results against FEA & previous research.                                                                                                  | <ul style="list-style-type: none"> <li>Extract frequencies from all recorded videos for all cases.</li> <li>Corrupt videos with different levels of noise and extract frequencies</li> <li>Compare against FEA results and previous research.</li> </ul> |
| Apply denoising techniques to enhance accuracy of extracted frequencies and compare with previous results.                                                      | <ul style="list-style-type: none"> <li>Search for different denoising techniques.</li> <li>Apply Denoising CNN on corrupted videos and then extract frequencies</li> </ul>                                                                               |

**Table 2 Objective and corresponding methodology**

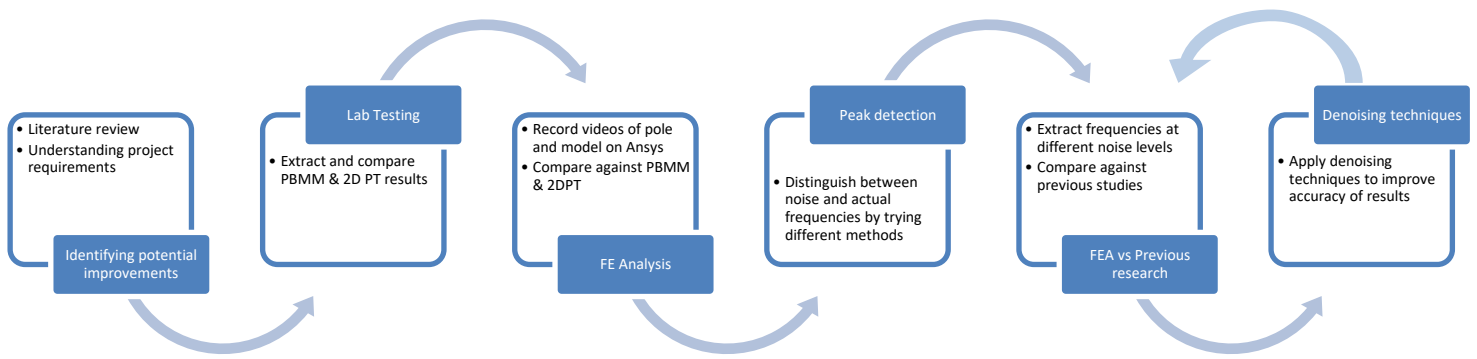


**Figure 24 Methodology Flowchart**

### 3.3.1 Laboratory Experiment Procedure & Goal

The setup for the laboratory experiment in this study is the following, a fixed wing on a shaker machine, the wing was fitted with accelerometers to record data. A red background wall in order to enhance contrast in the recorded video. A mobile phone was used to record all experiments in this study, a tripod was used to stabilize the camera.

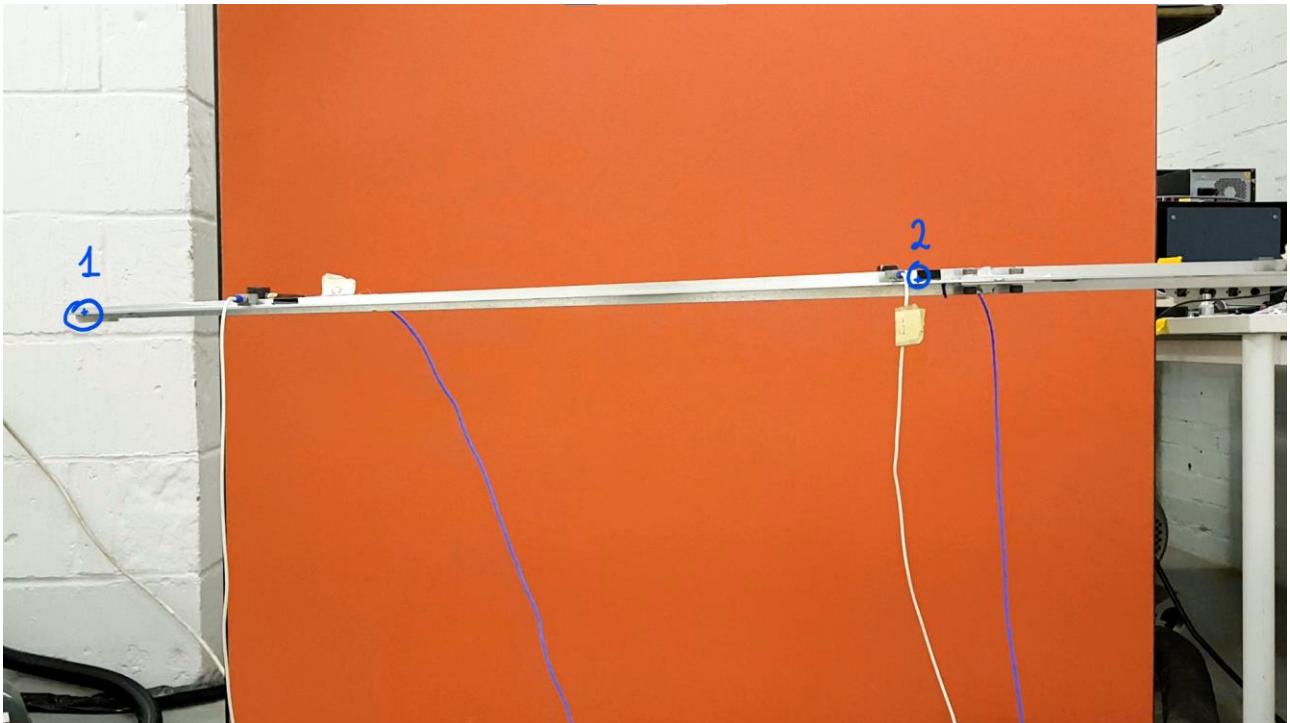


**Figure 25 Laboratory setup along with points monitored**



**Figure 26 Different angle of lab setup**

The goal of the laboratory experiment is to check the accuracy of 2D PT on its own, then combine it with PBMM and check against results from the accelerometers.

The wing was subject to random vibrations, multiple videos were recorded for accuracy of results.

### 3.3.2 Finite Element Analysis Procedure & Goal

The goal of this objective is to obtain a baseline of results to compare to the ones from 2D PT and PBMM. In this section we have moved from indoor testing to outdoor testing.

The experiment consists of recording videos of a light pole located on Cranfield University's campus. Then analysing the videos through the developed program in the previous section and finally compare results with FEA results.



**Figure 27 Light Pole studied**

In order to create the Finite Element Model of the light pole, the exact dimensions were obtained and modeled.

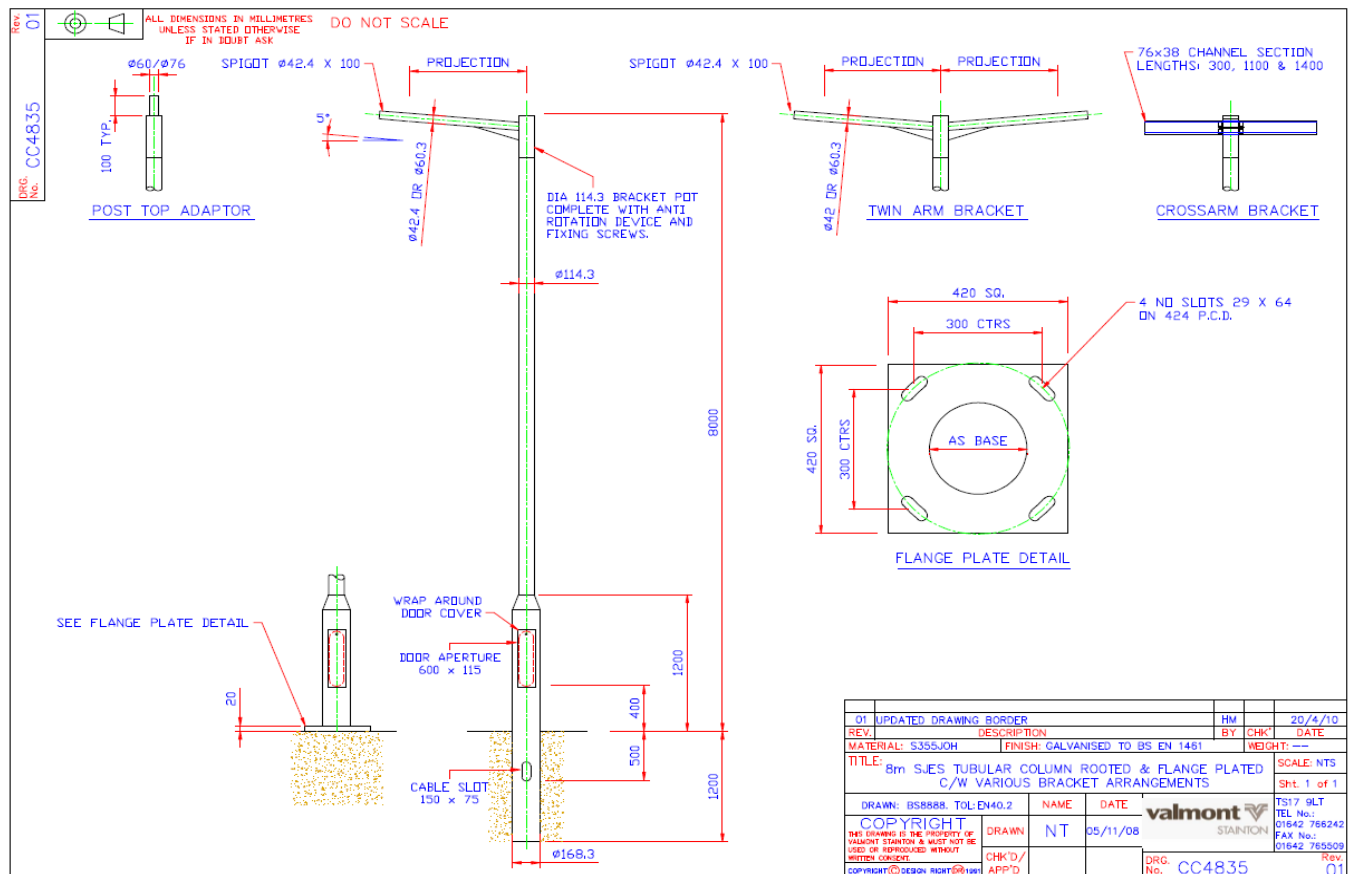


**Figure 28 Dimensions of Light Pole**

However, dimensions aren't the only parameter affecting our results, materials making up the pole are also of importance. To obtain the exact value for the frequency of the pole would be near impossible, hence why material properties were varied slightly to obtain a credible range of frequencies. The two materials used are the following:

The first (**Material 1**) is obtained from *Ansys Granta Materials Data for Simulation* library and has the following properties:

|                                    |                                   |
|------------------------------------|-----------------------------------|
| <b>Name</b>                        | Aluminum alloy, wrought, 6061, T6 |
| <b>Density (kg.m<sup>-3</sup>)</b> | 2713                              |
| <b>Young's Modulus (Pa)</b>        | 6.904 e <sup>10</sup>             |
| <b>Poisson's Ratio</b>             | 0.33                              |
| <b>Bulk Modulus (Pa)</b>           | 6.768 e <sup>10</sup>             |

|                           |                       |
|---------------------------|-----------------------|
| <b>Shear Modulus (Pa)</b> | 2.595 e <sup>10</sup> |
|---------------------------|-----------------------|

**Table 3 Wrought aluminum alloy properties**

The second material (**Material 2**) is obtained from the Ansys General Material Library and has the following properties:

|                                    |                       |
|------------------------------------|-----------------------|
| <b>Name</b>                        | Aluminum alloy        |
| <b>Density (kg.m<sup>-3</sup>)</b> | 2770                  |
| <b>Young's Modulus (Pa)</b>        | 7.10 e <sup>10</sup>  |
| <b>Poisson's Ratio</b>             | 0.33                  |
| <b>Bulk Modulus (Pa)</b>           | 6.960 e <sup>10</sup> |
| <b>Shear Modulus (Pa)</b>          | 2.669 e <sup>10</sup> |

**Table 4 Aluminum alloy properties**

The model of the light pole is not a complex model to create however attention must be paid to its detailed measurements. As mentioned before, material properties were varied a bit to obtain a credible range of frequencies.

A fine mesh with element size of 0.02m is needed to run the simulation as the structure contains sharp edges.

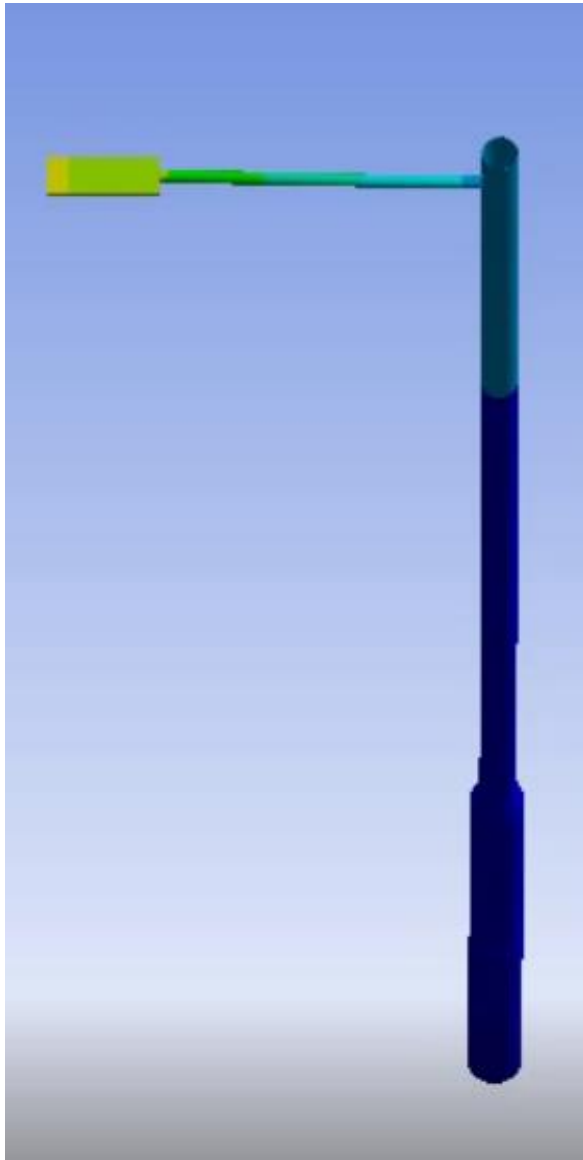


**Figure 29 Light Pole FE model**

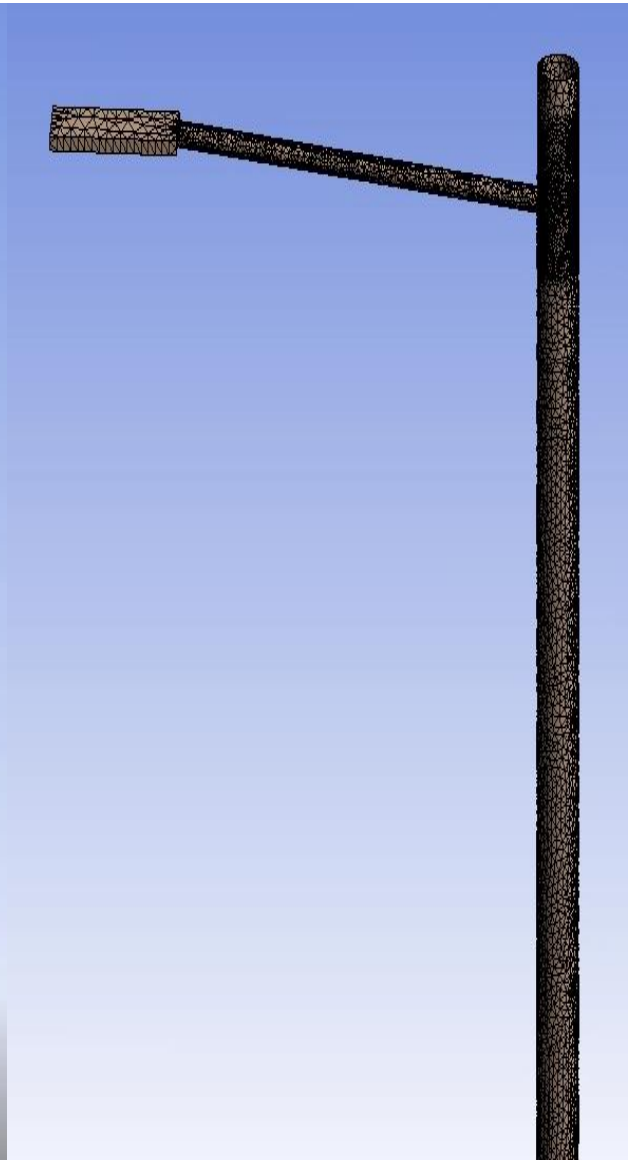


**Figure 30 Light pole mesh**

### 3.3.3 Peak Detection

The plots showing the frequencies of the light pole obtained from the combination of 2D PT and PBMM have shown to be quite noisy, which was anticipated. Hence a method was required in order to separate peaks relating to actual frequencies from peaks that are just noise.

The problem faced is not only related to differentiation of peaks but also to extraction. One of the methods suggested to try and extract frequencies is the following

1. Plot Trendline for the plot and extract its equation  $F(x)$ . The trendline needs to be of a relatively high order in order to be able to mimic the overall shape of the curve. In this case a 7<sup>th</sup> order polynomial was used.

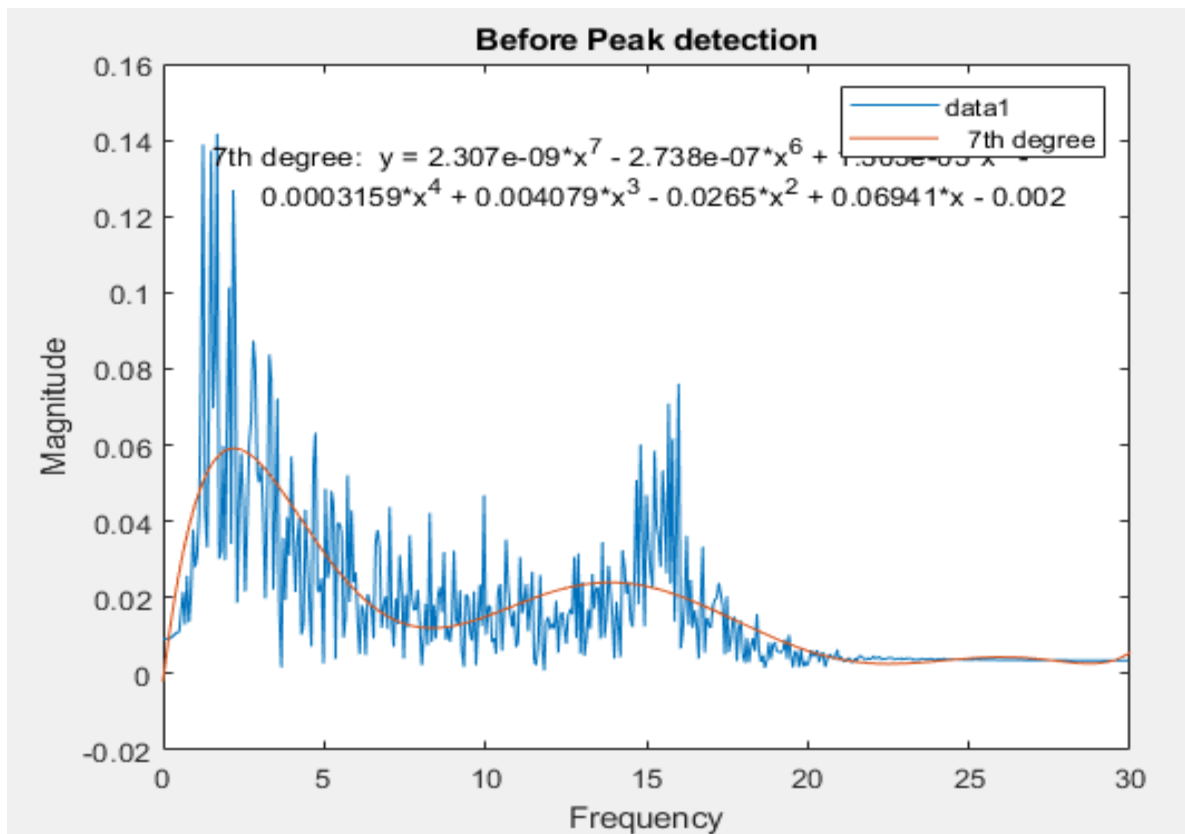


**Figure 31 Trendline of data used for peak extraction**

2. Set 1<sup>st</sup> derivative of  $F(x)$  equal to zero, this tells us where we have local minimums and maximums on the trendline.

3. Solving the 2<sup>nd</sup> derivative of  $F(x)$  for the values of  $x$  obtained in the previous step informs us whether we have a maximum or a minimum. Whenever we have a minimum at a given  $x$ , we can extract the maximum frequency in the range  $(x_1, x_2)$

4. Blind peak extraction result

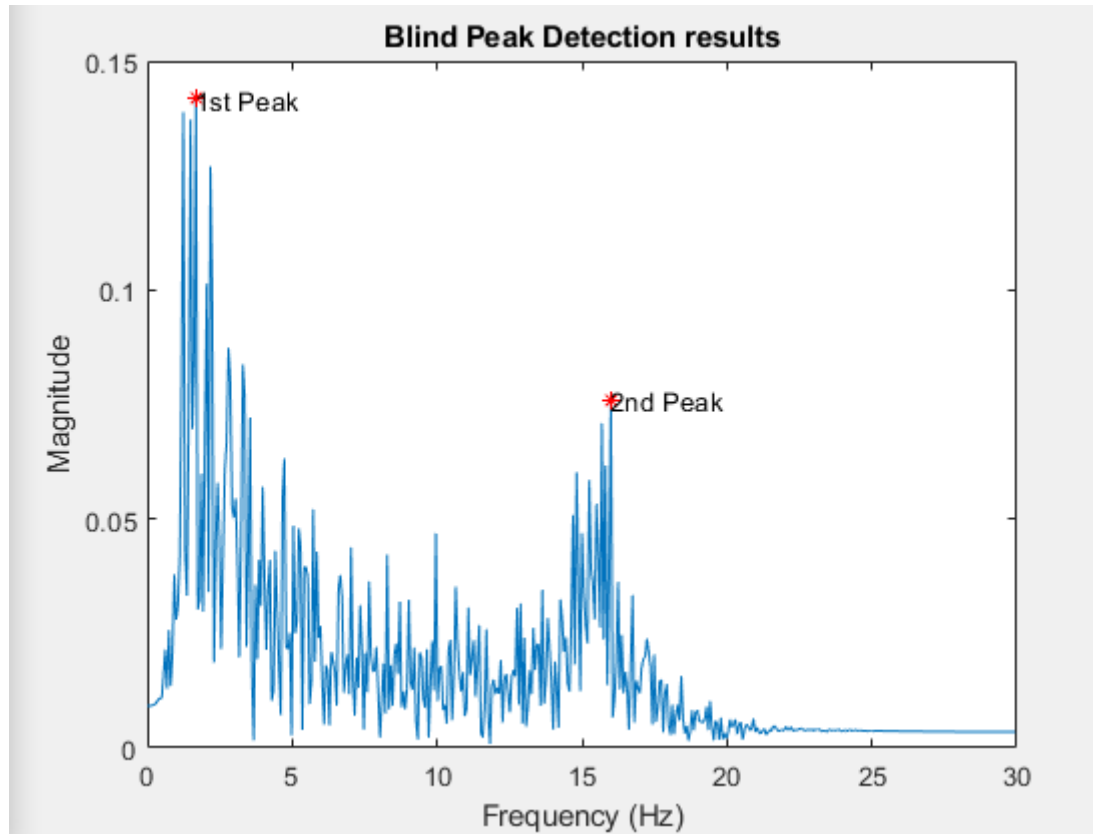


**Figure 32 Blind peak detection result**

The problem with the method suggested above are the following, firstly it cannot differentiate between actual peaks and noise. Secondly, whenever two maximum peaks are closely spaced, the trendline would not be able to provide accurate minimums and maximums.

The method selected is the following, multiple videos of the Light pole were recorded and each was analysed for frequencies detected which would provide with a reasonable range to work with.

**Table 5 Frequencies extracted from multiple videos using 2D PT.**

| Video | Frequency range (Hz) |
|-------|----------------------|
| 1     | 1.41→1.47            |
| 2     | 1.37→1.5             |
| 3     | 1.44→1.55            |
| 4     | 1.3→1.67             |
| 5     | 1.43→1.49            |

The value pairs seen in the table above (i.e. 1.41→1.47) represent the two highest consecutive peaks detected. The arrow is present to indicate that any value within this range can be the exact value and not only the two extremities.

A range of 1.43 till 1.49 Hz is used throughout this study for analysis and comparison of the light pole.

### **3.3.3.1 Plots showing frequencies detected**

The following two figures show frequency plots of 2 different videos of the light pole. In the first figure, the range of frequencies filtered is from 0 to 4 Hz. While in the second figure the range was unchanged ( $0 \rightarrow f_{\max}$ ), this was done to check whether other higher frequencies would appear. It is clear from both figures that the dominant frequency is near 1.4 Hz.

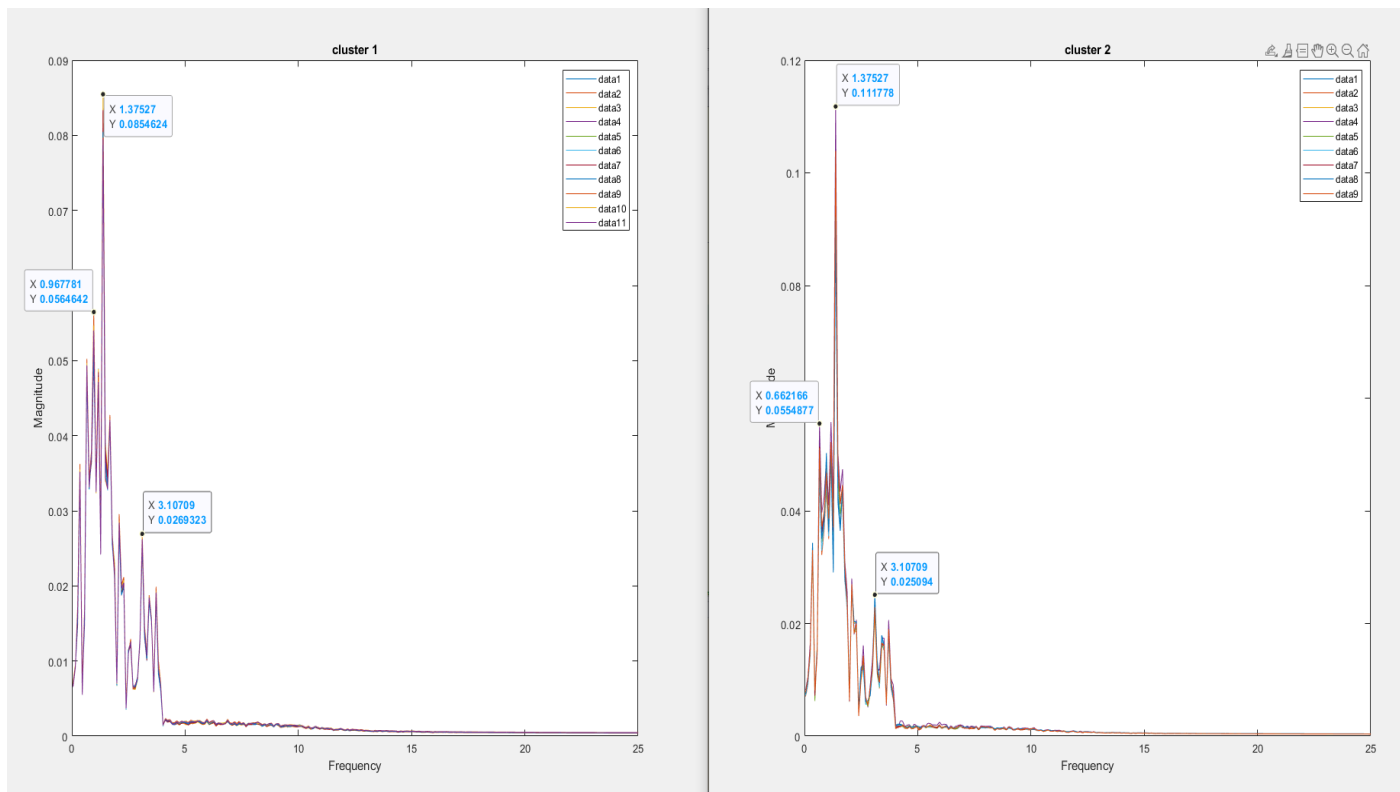


**Figure 33 Frequency measurement : Light pole - video 1**

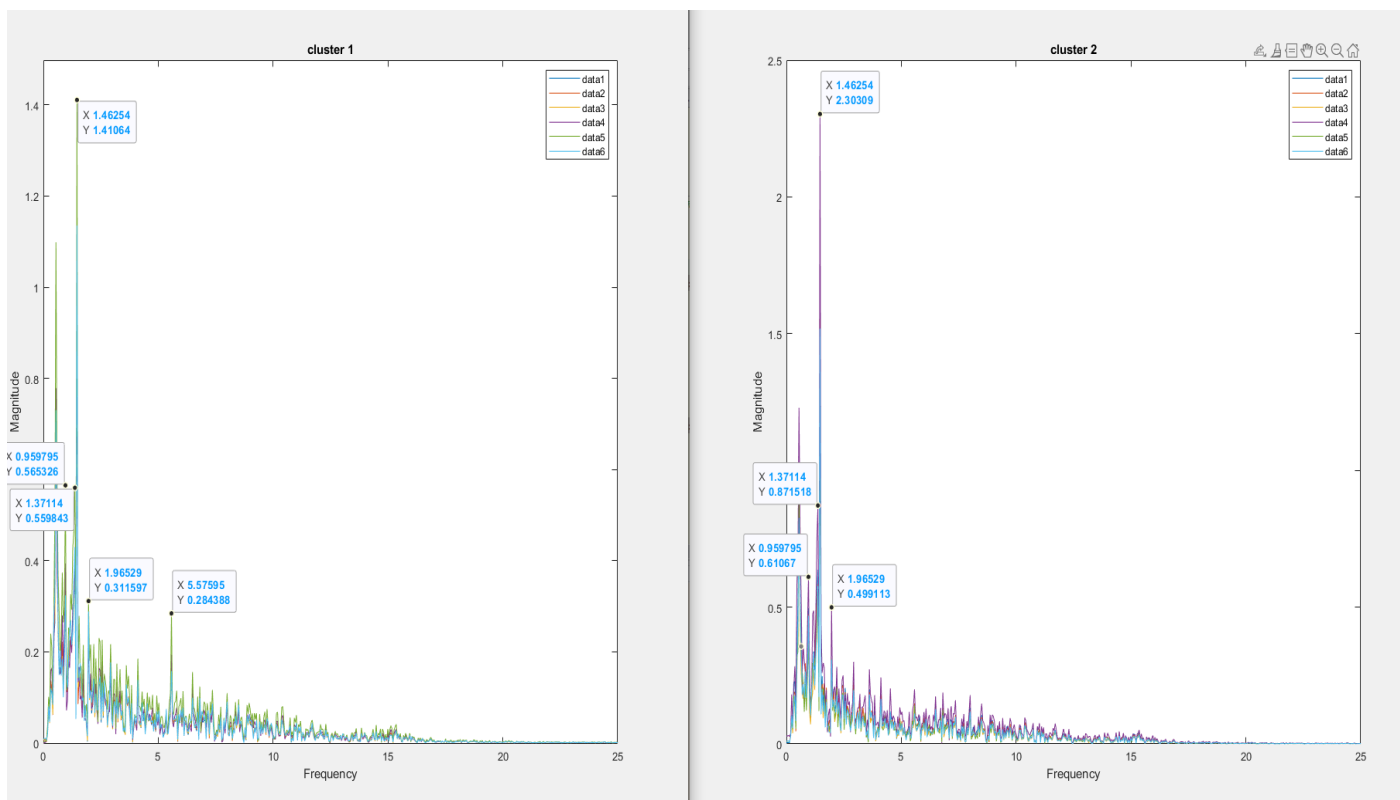


**Figure 34 Frequency measurement : Light pole - video 2**

In the previous plots, the maximum frequency shown depends on the frame rate of the recorded video, it is equal to half the number of frames per second. The frames per second of the recording device used is equal 60.

#### Equation 3 Maximum frequency detected

$$f_{max} = \frac{FPS}{2} \quad 3$$

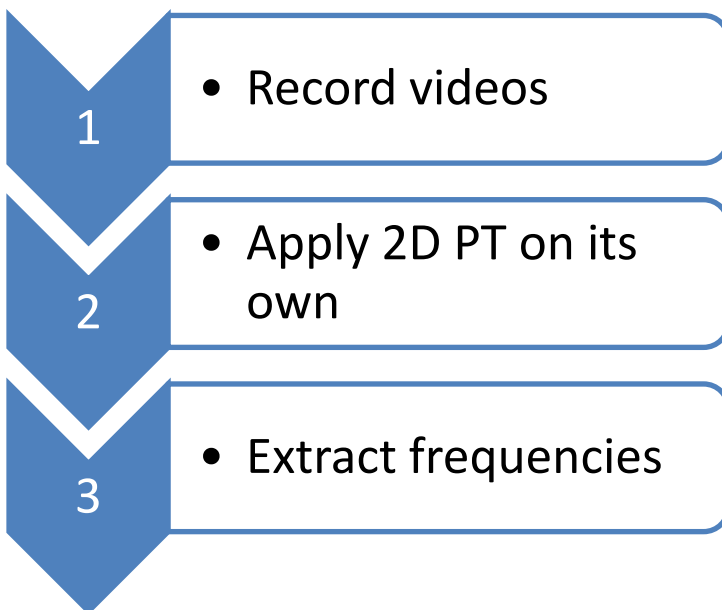
Where:

- $f_{max}$ : Maximum frequency detected.
- FPS: Frames per second of recording device.

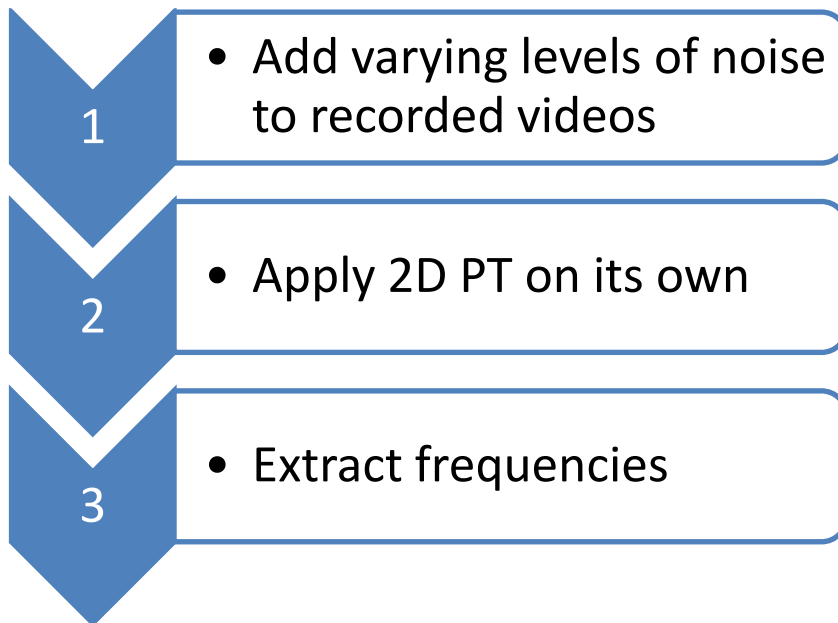
#### 3.3.4 Apply different noise levels and compare results

The procedure followed in this section is the following:

In the first run, the following steps are applied, the goal is to obtain a base of results to compare with future results.

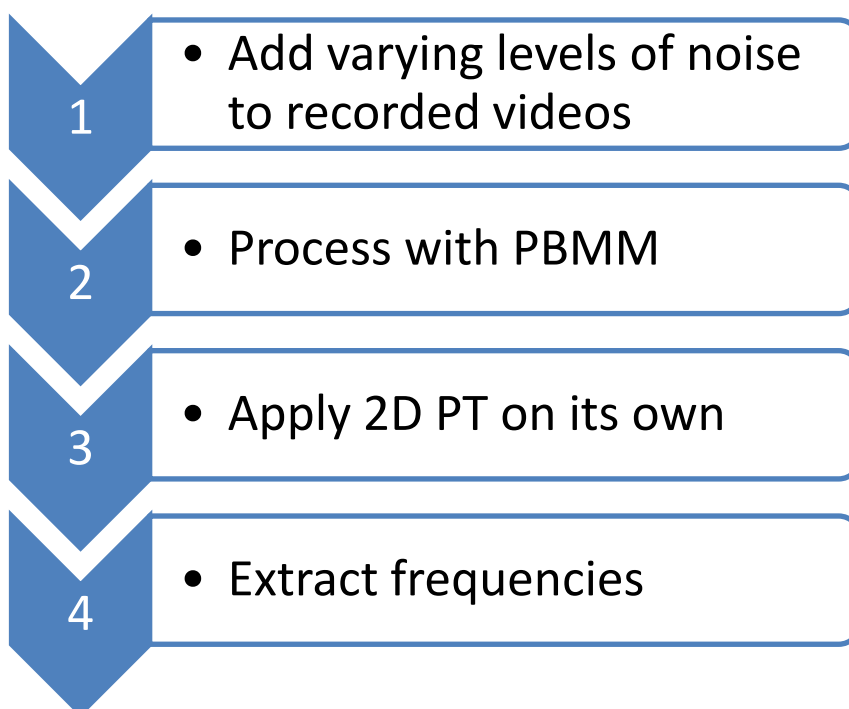


In the second run, different levels of noise were added to the recorded video and then frequencies extracted using 2D PT on its own.



The purpose of this run is to discover the robustness of 2D PT against noise.

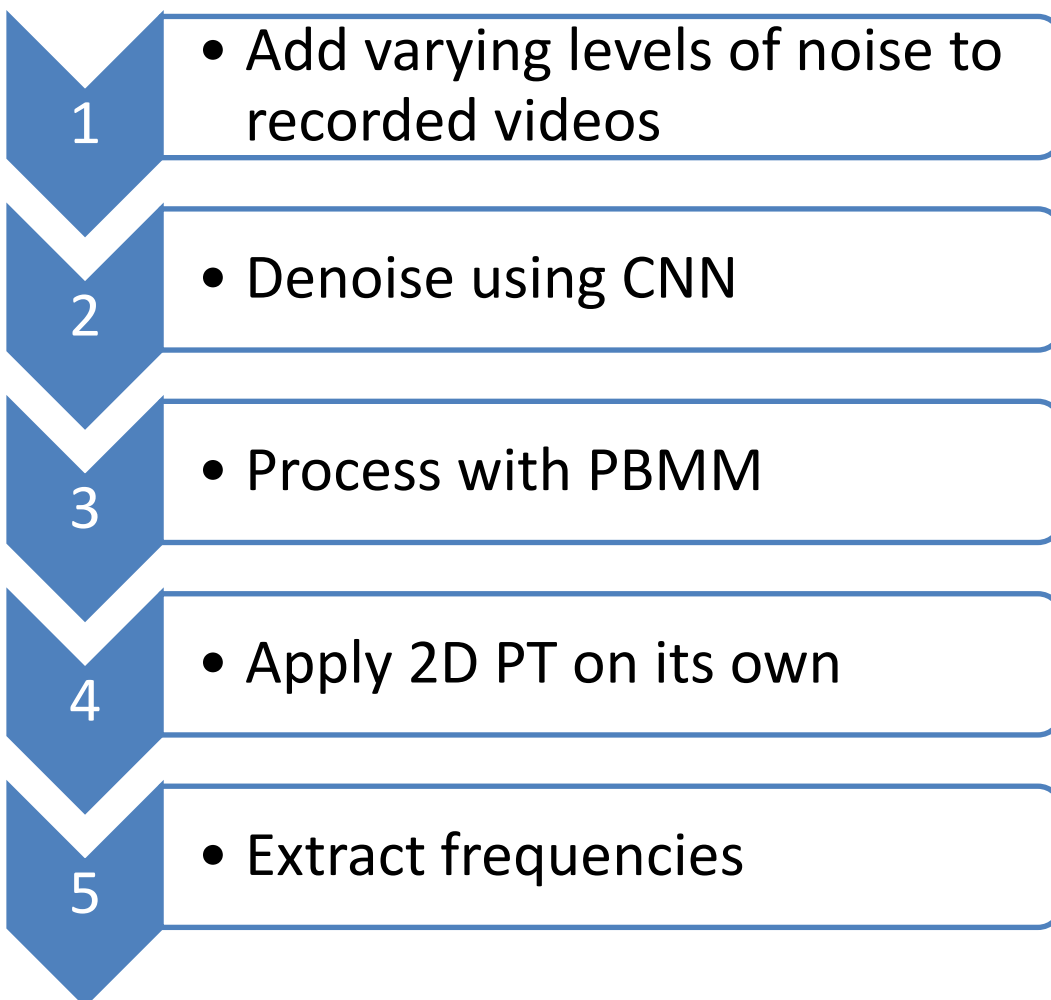
In the third run, the same videos that were corrupted with noise, will be processed by PBMM before applying 2D PT and extracting frequencies.



The purpose of this run is to attempt to enhance the quality of results extracted using 2D PT by applying PBMM beforehand.

### 3.3.5 Denoising using CNN

In an effort to get close to the originally obtained results, a denoising convolutional neural network was used in order to increase the SNR of the light pole videos before processing with PBMM and 2D PT.



The purpose of this section is to mimic realistic situations where better quality videos or equipment are not available.

## 4 Results and Discussion

### 4.1 Laboratory results

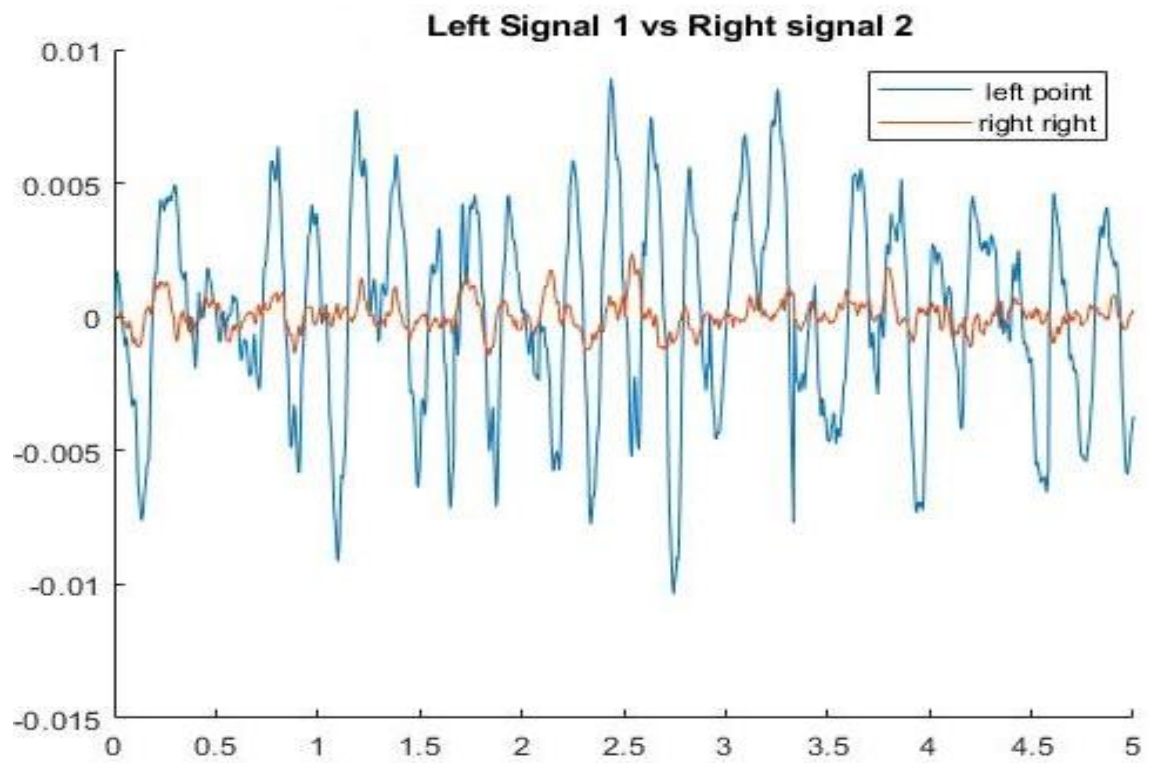


**Figure 35 Displacement vs time of 2 points monitored: vision based approach**

Taking a look at the displacement vs time plot above of the wing experiment, we notice that the left point monitored undergoes larger displacement than the right point. Which is logical considering it's near the tip of the wing.

| Frequency       | Accelerometer | 2D PT | 2D PT + PBMM |
|-----------------|---------------|-------|--------------|
| 1 <sup>st</sup> | 4.94          | 5     | 5.4          |
| 2 <sup>nd</sup> | 23.457        | 22    | 25.8         |
| 3 <sup>rd</sup> | 26.3          | 26    |              |

In the above table, we have a summary of the tests performed in the laboratory, we can notice the accuracy of 2D PT on its own in detecting the first frequency, we obtained a value of 5 Hz while the accelerometers gave us a value of 4.94 Hz, this equals a percent error of 1.1. However, when 2D PT was combined with PBMM, the percent error increased to 9, or a value of 5.4 Hz.

This shows that in a controlled environment where one can record pristine videos, 2D PT is enough on its own to provide accurate results.

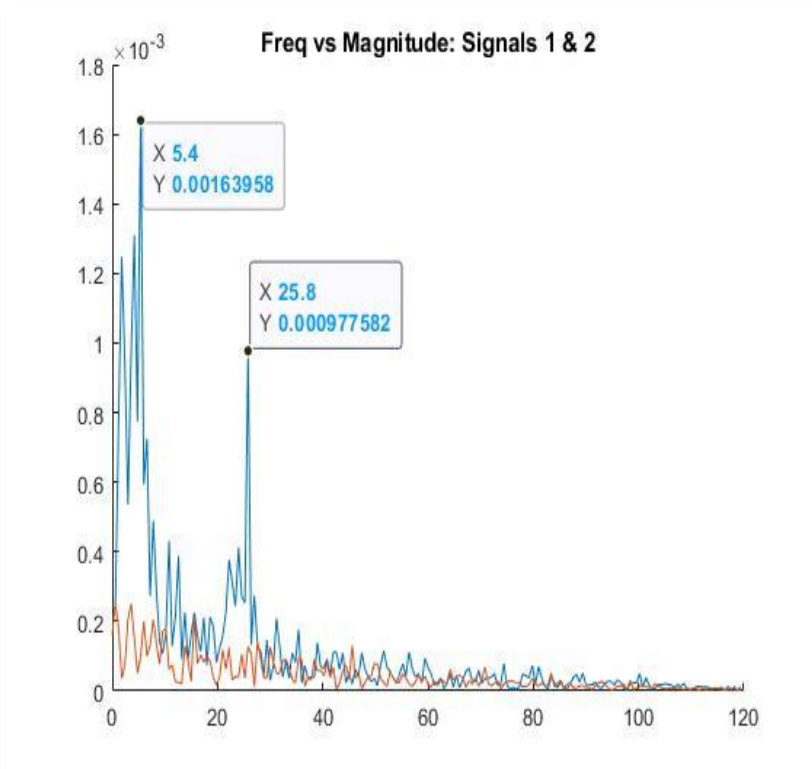


**Figure 36 2D PT + PBMM results**

| Frequency |   |
|-----------|---|
| 4.945     | 1 |
| 23.457    | 2 |
| 26.351    | 3 |

**Figure 37 Accelerometer results**

The graph below shows the stabilization diagram of the wing obtained from the accelerometer reading. One can notice the strong presence of the 1<sup>st</sup> frequency (~4.95 Hz) and the 3<sup>rd</sup> frequency (~26.3 Hz) but also the weak excitation of the 2<sup>nd</sup> frequency (~23.5 Hz). Closely spaced and weakly excited frequencies are a limitation for vision based frequency extraction, this explains the single clear value of 25.8 Hz obtained by combining 2D PT and PBMM.

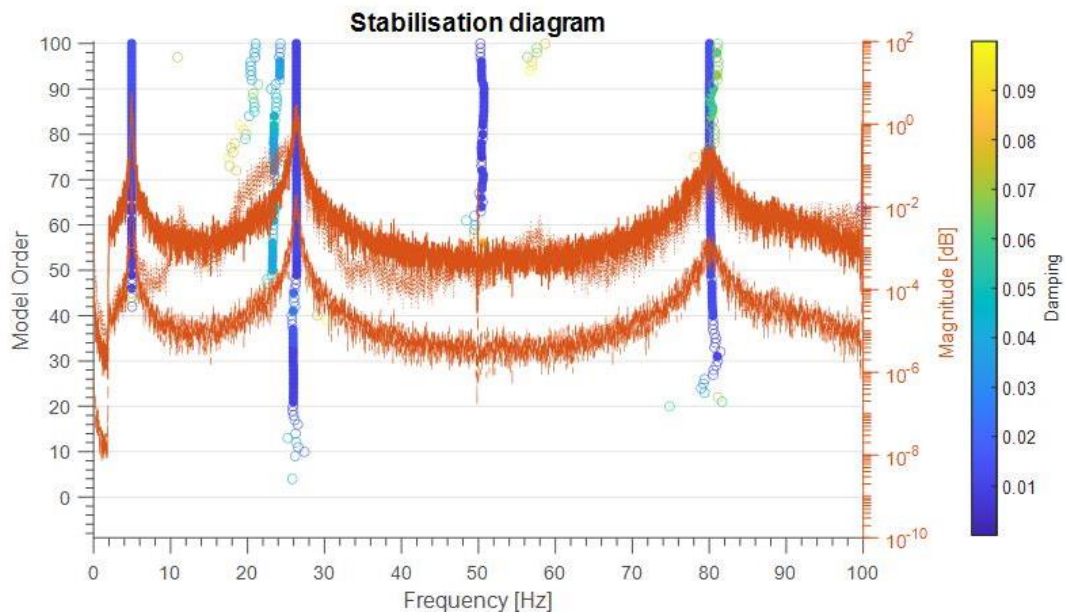


**Figure 38 Wing stabilization diagram**

## 4.2 FEA results

The following table presents the results of finite element analysis on the light pole

| Frequency            | Material 1 | Material 2 |
|----------------------|------------|------------|
| 1 <sup>st</sup> (Hz) | 1.30       | 1.37       |
| 2 <sup>nd</sup> (Hz) | 1.36       | 1.44       |

**Table 6 FEA results**

As mentioned before, two different materials were used to get an approximate range of frequencies as modelling the pole exactly is impossible.

This gives us a reasonable range of frequencies to compare to our vision based approach.

There is a slight difference between the first two natural frequencies of Material 1 and Material 2, this is expected since they are not the same, the results of the first material will be used for comparison throughout this study.

Both materials provided values close to the values extracted in section 3.3.3, however they are not the same and this could be for many reasons, the first is that wind was blowing when the videos were recorded which will definitely affect the accuracy of results obtained, and the second reason is the inability to match the exact materials and weight in finite element modelling, this also will give slightly varied results.

## 4.3 Light Pole results

|                |                 | FEA  | 2D PT       |
|----------------|-----------------|------|-------------|
| Frequency (Hz) | 1 <sup>st</sup> | 1.3  | 1.43 → 1.49 |
|                | 2 <sup>nd</sup> | 1.36 |             |

**Table 7 Results comparison between FEA and 2D PT**

|     |                                                                                   |                                                                                    |                                                                                     |
|-----|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
|     |  |  |  |
| SNR | 21.3                                                                              | 15.7                                                                               | 10.6                                                                                |

**Table 8 Visual comparison of different Signal to Noise Ratio**

The SNR values seen in the table above were calculated as follows:

1. Get noise by subtracting the clean image from the noisy.

4

$$noise = |I_{noisy} - I_{clean}|$$

#### Equation 4 Noise calculation

2. Calculate mean of clean image,  $M_{clean}$ .
3. Calculate mean of noise,  $M_{noise}$ .
4. Divide mean of clean image by mean of noise.

5

$$SNR = M_{clean} / M_{noise}$$

#### Equation 5 SNR calculation

The values were chosen to study the performance under different levels of noise.

| Light Pole extracted Frequency |              |                | Range Width (Hz) |
|--------------------------------|--------------|----------------|------------------|
| FEA (Hz)                       | 1.3 →1.36    |                | 0.06             |
| 2D PT (Hz)                     | 1.43 →1.49   |                | 0.06             |
| SNR \ 2D PT                    | Without PBMM | With PBMM (Hz) |                  |
| 21.3                           | N/A          | 1.3→1.4        | 0.1              |
| 15.7                           | N/A          | 1.23→1.5       | 0.27             |
| 10.6                           | N/A          | 1.23→1.9       | 0.67             |

Table 9 2D PT results with and without PBMM, Light pole video 1.

| Light Pole extracted Frequency |              |                | Range Width (Hz) |
|--------------------------------|--------------|----------------|------------------|
| FEA (Hz)                       | 1.3 →1.36    |                | 0.06             |
| 2D PT (Hz)                     | 1.37 →1.5    |                | 0.13             |
| SNR \ 2D PT                    | Without PBMM | With PBMM (Hz) |                  |
| 22                             | N/A          | 1.3→1.5        | 0.2              |
| 13.4                           | N/A          | 1.2→1.63       | 0.43             |
| 9.7                            | N/A          | 1.2→1.7        | 0.5              |

Table 10 2D PT results with and without PBMM, Light pole video 2

The arrows used in the previous tables represent the fact that the actual might be within the given range (1.3→1.5) and not limited to 1.3 and 1.5.

When noise was added to the recorded videos, 2D point tracking was no longer able to detect points to track on its own. However, when the videos were processed with Phase-Based Motion Magnification, 2D PT was then able to detect points to track.

The results of the two approaches are shown in the tables above, we notice that as the SNR goes down, meaning an increase in the level of noise, the frequency range detected increases from 0.06 with no noise to 0.5 with an SNR of 10.6. This indicates the increasing difficulty in detecting proper frequencies as the noise levels go up.

The reason why PBMM was able to improve to extraction of frequencies is the following, spatial filtering acts as a low pass filter eliminating noise from images. Also the fact that magnification of phase values does not magnify the noise.

| Before PBMM                                                                        | After PBMM                                                                          |
|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
|  |  |
| Figure 39 Image quality before PBMM processing                                     | Figure 40 Image quality after PBMM processing                                       |

Table 11 PBMM before and after image quality comparison

We notice an overall smoothing effect after processing with PBMM. The distortions seen in the previous page on the right are from the phase difference amplification factor.

These distortions can be eliminated as you can see in the image on the right by setting the amplification factor  $\alpha$  to 1, meaning no magnification happens.

Furthermore the Signal to Noise Ratio increases, the noise levels go down, after processing with PBMM:



Figure 41 PBMM with  $\alpha = 1$

|             | SNR 1 | SNR 2 | SNR 3 |
|-------------|-------|-------|-------|
| Before PBMM | 21.3  | 15.7  | 10.6  |
| After PBMM  | 56.62 | 44.46 | 22.68 |

This indicates the viability of using PBMM as a denoising technique. However further testing is required in this direction.

M. Eitner et al., 2021, suggested that at high frequencies (near 1400 Hz), recorded markers set on the studied object would have sub-pixel movement which would results in low SNR, and without any video processing 2D PT would not be able to track these markers. Hence why they used PBMM to enhance extraction at high frequencies.

Number of points tracked vs Phase difference magnification factor  
alpha and sigma = 5

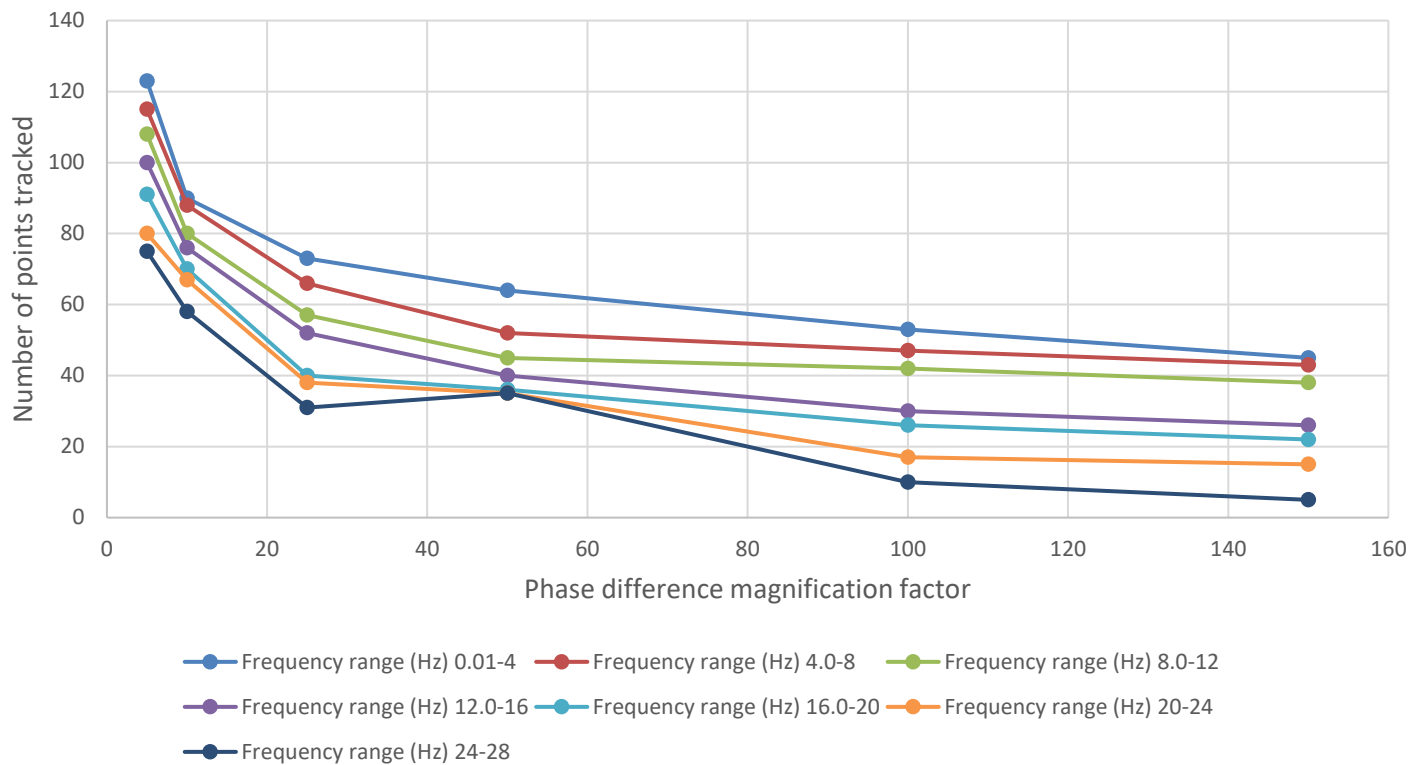


**Table 12 Number of points tracked vs Phase difference magnification factor**

The observations of M. Eitner et al., is confirmed by the graph above, we see a decrease in the number of points tracked by 2D PT, enhanced by PBMM, as the frequencies being studied increases. This is due to the decrease in SNR as the filtering frequencies go up.

## 4.4 Building Context & Results

For this case study, it was performed in Milton Keynes, close to Cranfield university, the target building is shown in Fig.27 along with the location where the camera was positioned, multiple videos at different resolutions and frame rates were recorded. The camera used is the same for all 3 experiments.

The reason why this case was chosen is to put PBMM & 2D PT under realistic conditions. Vibrations detected from the building could be caused by wind, loads and/or traffic nearby.



**Figure 42** Satellite view of building studied



**Figure 43** Building studied

| Building extracted Frequency  |                          |                       |
|-------------------------------|--------------------------|-----------------------|
| <b>Previous Research (Hz)</b> | 1.1                      |                       |
| <b>2D PT (Hz)</b>             | 1.07                     |                       |
| <b>SNR \ 2D PT</b>            | <b>Without PBMM (Hz)</b> | <b>With PBMM (Hz)</b> |
| <b>22</b>                     | 1.35                     | 0.91→1.24             |
| <b>13.4</b>                   | 1.35→1.67                | 0.91→1.13             |
| <b>9.7</b>                    | 0.91→1.45                | 1.13→1.45             |

**Table 13 2D PT results with and without PBMM, Building video.**

In the case of the building, when adding noise and in contrast to the light pole case, 2D PT was able to detect points on its own, however the values did not cover the actual value of 1.1 Hz. After processing with PBMM the values extracted improved slightly and shifted to cover the first frequency of 1.1 Hz.

The results however are still quite far from the actual value of 1.1 Hz, this variation can be due to the fact that the camera was positioned in a building that was itself subject to vibration (wind, load, traffic), combining this with PBMM might have affected the overall quality of the results.

## 4.5 Denoising CNN results

|                    |                           | Recorded Video | Same video with added noise | Same noisy video | Denoised video using CNN |
|--------------------|---------------------------|----------------|-----------------------------|------------------|--------------------------|
| Light pole Video 1 | Method used               | 2D PT          | 2D PT                       | PBMM + 2D PT     | CNN +<br>PBMM +<br>2DPT  |
|                    | Frequencies detected (Hz) | 1.43→1.49      | No Points Detected          | 1.23→1.9         | 1.3→1.5                  |
| Light pole Video 2 | Method used               | 2D PT          | 2D PT                       | PBMM + 2D PT     | CNN +<br>PBMM +<br>2DPT  |
|                    | Frequencies detected (Hz) | 1.37→1.5       | No Points Detected          | 1.2→1.7          | 1.35→1.57                |

**Table 14 Results after using denoising CNN**

Denoising the videos with dCNN and then processing with PBMM allowed the frequencies detected to converge back to the originally obtained results, but not exactly. When using 2D PT on its own the frequencies detected ranged from 1.43→1.49 Hz, however when noise was added to the same video, 2D PT was not able to track any points. So PBMM was introduced and aided 2D PT in the detection of frequencies. However the range was from 1.23 till 1.9 Hz, hence why a denoising convolutional neural network was implemented in an effort to try and recover originally obtained results, the frequencies detected after using a dCNN

ranged from 1.3 to 1.5 which is much better than 1.23→1.9 and closer to 1.43→1.49 Hz.

|                              | SNR 1 | SNR 2 | SNR 3 |
|------------------------------|-------|-------|-------|
| <b>Before PBMM</b>           | 21.3  | 15.7  | 10.6  |
| <b>After PBMM</b>            | 56.62 | 44.46 | 22.68 |
| <b>After dCNN &amp; PBMM</b> | 64.99 | 63.31 | 40.95 |

**Table 15 SNR comparison**

The table above shows Signal to Noise Ratios measurements at different stages of processing, we achieved the best SNR after processing with both dCNN and PBMM.

In this chapter, we looked at frequency extraction using both 2D PT and PBMM in indoor and outdoor conditions at different levels of noise. On one hand, during the lab experiment 2D PT performed well on its own and we witnessed a degradation in performance when combined with PBMM.

On the other hand, during outdoor testing, combining PBMM with 2DPT improved the accuracy of results extracted. In both cases, this is due to the spatial filtering, which acts as a low pass noise filter, in pristine videos, like in the laboratory test, it is the reason for quality downgrading, however in noisy videos, it helps improve the overall quality of the video.

## 5 Automation of Procedure

### 5.1 Challenges & Solutions in automation

In order to fully automate the procedure above, the following obstacles need to be overcome.

The first one is automatic object recognition or in other words informing the 2D PT where to place the bounding box and what size it is. This can be partially solved by setup of the recording equipment in a certain way, to centre the object of interest in the middle of the frame for example which can make it easier for the object detection algorithm to detect the proper object.

The second is automatic parameter selection, for the program to run properly, different parameters must be chosen initially, these parameters change from one case to another and currently are chosen manually.

These parameters include phase difference magnification factor  $\alpha$ , motion blurring factor in areas of low intensity,  $\sigma$  and frequency window length, let's call it FWL. Frequency window length refers to the length of the range of frequencies that PBMM will work with. PBMM will only amplify motion that falls within the frequency range specified. In order to do so, a blind source separation algorithm would need to be implemented in order to detect actual frequencies of the object studied and then use these frequencies for determining a valid range for PBMM to use.

The remaining two parameters,  $\alpha$  &  $\sigma$ , will be discussed in the following section.

## 5.2 Proposed automation

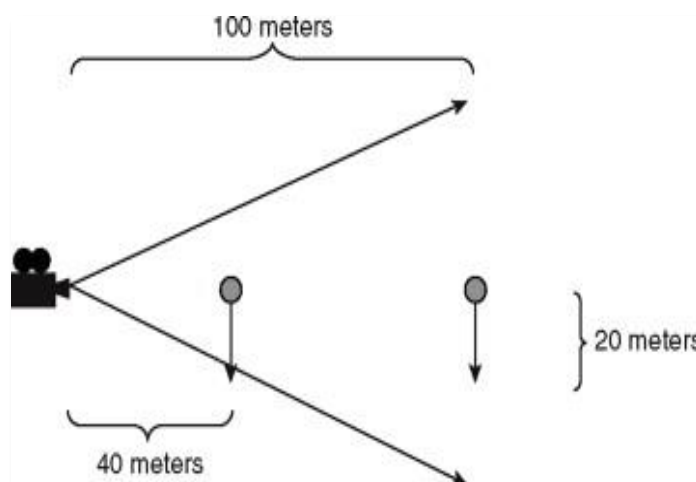
In this study, we will address the automatic parameter selection for the program to run, these parameters are the following:

1. Alpha: Phase difference amplification factor. It controls how much the motion is amplified.
2. Sigma: Motion smoothing factor. It controls how much the areas of low amplitude in the phases are blurred.

### 5.2.1 Motion Parallax

In order to select these parameters, let us first dive into how human motion perception is affected by distance. Let us consider two light pole, the first at a distance of 40 meters in front of us and the second at a distance of 100 meters.

We will assume that both poles are moving at the same speed, in the same direction and have the same materials and dimensions.



RON BRINKMANN, in *The Art and Science of Digital Compositing (Second Edition)*, 2008

**Figure 44 Motion Parallax**

Since both poles are moving at the same speed across the frame, the pole that is closer to the observer (camera / human eye) will appear to move a greater distance than the pole that is further away. This phenomenon is known as Motion Parallax

and the reason for this has to do with the distance the object travels compared to the amount of space it covers in the frame. The object that is closer will cover a larger portion of the frame.

Now we know that a moving object that occupies a large real estate of the frame will appear to travel a larger distance than an object travelling at the same speed that is further back.

### 5.2.2 Alpha: Motion Amplification factor

Relating this to our study, we know that the objects studied are moving, we need to determine how much of the frame they cover in order to determine the parameters. Objects that cover a large portion of the frame (close to the observer) would need lower phase difference amplification factor than objects that cover a smaller portion of the frame (further to the observer).

For the case of Light pole on Cranfield Campus, the main direction of movement was horizontal, or from left to right. Hence, in order to determine the amount of space it takes in the frame, the width of the pole was measured in the image and divided by the total width of the image in order to get a percentage.



**Figure 45 Light pole width in pixels**

The light pole width is equal to approximately 23 pixels while the image width is equal to 1080 pixels, this gives a percentage of 2.13%, an amplification factor of 10 was found to be suitable for this case.



**Figure 46 Building width in pixels**

In the case of the building, the width was equal to approximately 921 pixels while the image width was equal to 1920 pixels, this gives a percent coverage of 47.9%. The Amplification factor used in this case was equal to 4. As the object being studied covers a larger portion of the frame, we can afford to go down with the value of the amplification factor. Now the challenge is no longer parameter selection but automatic object recognition which we have discussed earlier.

| Coverage     | Amplification Factor |
|--------------|----------------------|
| <20 %        | 10 or more           |
| >20% - < 50% | Between 5 and 10     |
| >50%         | Between 2 and 5      |

**Table 16 Relation between Coverage and Amplification Factor**

### 5.2.3 Sigma: Blurring factor

As for the motion smoothing factor, sigma, a value of 2 was used for both cases, any value between 1 and 5 gave adequate results, however as sigma goes up the computation time goes up but more importantly the signal to noise ratio goes down, meaning an increase in noise levels.

| Sigma | SNR before PBMM | SNR after PBMM |
|-------|-----------------|----------------|
| 2     | 14.32           | 22.1           |
| 4     | 14.32           | 21.3           |
| 6     | 14.32           | 20.9           |

**Table 17 Sigma vs SNR: Light Pole**

After processing the same video multiple times, each time with a different motion smoothing factor. We notice the increase in signal to noise ratio, when comparing before and after processing with PBMM.

However, one should keep in mind that as sigma goes up, the noise levels postprocessing go up also. This is expected as the program is simply blurring the areas of no interest.

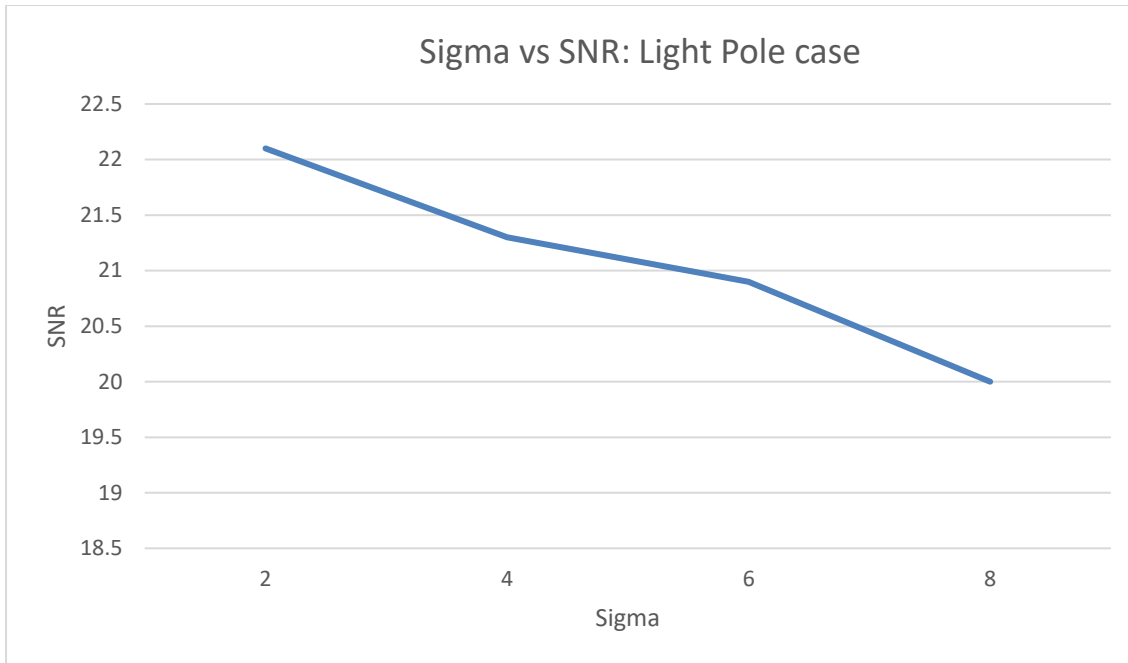


**Figure 47 Plot of Sigma vs SNR after PBMM: Light pole video**

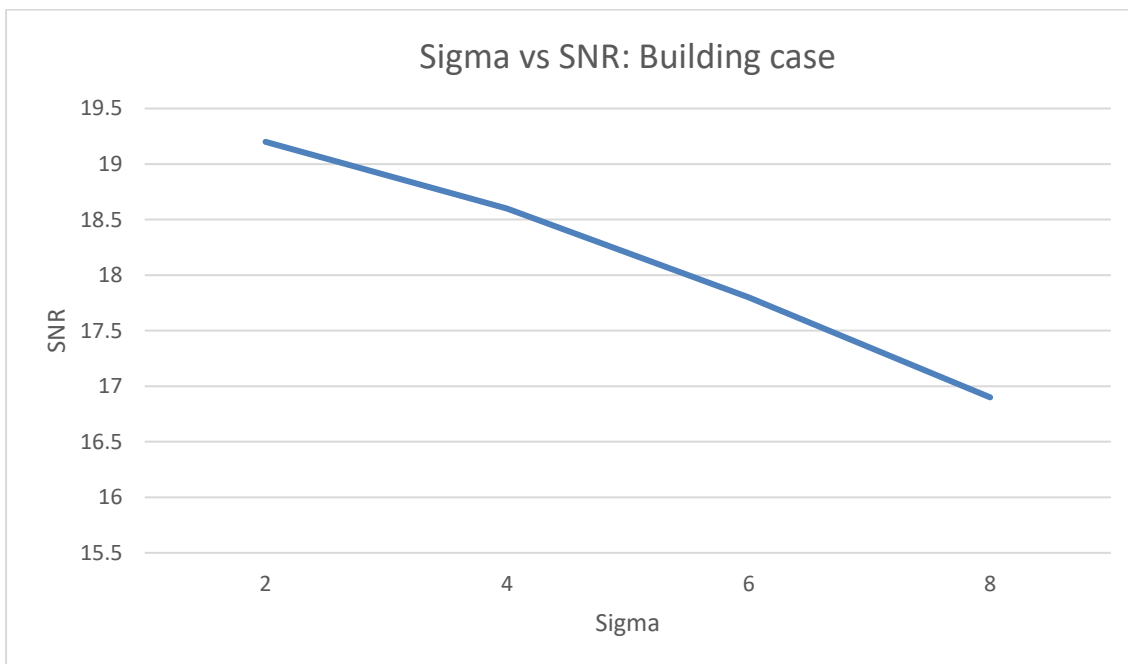


**Figure 48 Plot of Sigma vs SNR after PBMM: Building video**

In order to automate the selection of sigma, one would need a method to assess image quality knowing that there is no reference image to compare to. This method was suggested by *Kanjar De and V. Masilamani, 2013*, they proposed a method for image quality assessment with no reference, the technique works on the magnitude of the image in the frequency domain and it's the following:

Input: Image  $I$  of size  $M \times N$ .

Output: Image Quality measure (FM) where FM stands for Frequency Domain Image Blur Measure

Step 1: Compute F which is the Fourier Transform representation of image I

Step 2: Find Fc which is obtained by shifting the origin of F to centre.

Step 3: Calculate AF = abs (Fc) where AF is the absolute value of the centered Fourier transform of image I.

Step 4: Calculate M = max (AF) where M is the maximum value of the frequency component in F.

Step 5: Calculate TH = the total number of pixels in F whose pixel value > *thres*, where *thres* = M/1000.

Step 6: Calculate Image Quality measure (FM) from equation (1).

$$\text{Image Quality measure (FM)} = \frac{T_H}{M*N} \quad 6$$

#### Equation 6 Image Quality measure (FM)

As the image is sharper and less blurry, FM will increase. Notice in the figure below, the image that is sharper (a) has a richer Centered Fourier spectrum (b) than (d) which corresponds to a blurrier version (c) of (a).



**Figure 49 Centered Fourier spectrum comparison of clear images (a-b) and blurred images (c-d).**

Setting a global rule to follow for the automatic selection of sigma is not adequate as each case has an image size unique for it. Hence, for this study, an image size of 1920 \* 1080 will used.

| FM     | Sigma used |
|--------|------------|
| 0.0036 | 5          |
| 0.0034 | 3          |
| 0.0028 | 0          |

As the image gets blurrier, FM decreases, the need to blur out regions of low intensity becomes even less necessary, furthermore it is less useful and can make it more difficult for 2D PT to detect adequate points to track.

To conclude this chapter, in order to adequately choose a blurring factor, sigma, one must pay attention to the quality of the image he is working with. As the blurriness of the input image increases, sigma must decrease.

Secondly, for the amplification factor, alpha, it must be selected based on the proximity of the object studied to the camera. If the object is close to the camera, one can afford to decrease alpha, if it is further we can increase alpha.

## **6 Conclusion, Potential applications & Future work**

In conclusion, this study explored the possibility of using Phase-Based motion magnification as way to improve frequency extraction using 2D Point Tracking in outdoor noisy conditions, also a denoising convolutional neural network was used as a way to improve accuracy of extracted data.

First, the theoretical background of the methods used was written, then a state of art review to show existing research within this field. After that the objectives and methodologies were presented, followed by the results obtained and discussed. Finally, this research touched on the automation of the procedure suggested.

Phase-Based Motion Magnification proved to be useful in assisting 2D Point Tracking for frequency extraction in noisy conditions. Signal to noise ratios doubled on average when comparing before and after processing with PBMM. Furthermore, denoising the data before processing with PBMM gave better results when compared with PBMM on its own. The frequencies detected were closer to actual values and SNR values increased even more.

The procedure presented above could prove to be useful in conditions where better quality videos cannot be acquired or where time and proper equipment are a major constraint like checking safety of machinery and structures after major incidents.

Another use case is for model checking and updating during the construction phase of

the building, this can help make sure the building is behaving as expected and minimize the possibility of having delays.

Possibilities for future work are endless as vision based techniques are advancing by the day. The effect of contrast change on the quality of results should be studied on its own and in combination with different noise levels. Different types of neural network architectures can be tested for denoising or other purposes like image segmentation to isolate the object studied, or recurrent neural networks which are used for time series prediction and might prove to be useful in distinguishing between noise and actual peaks.

Another potential possibility for future work is the use of 2 or more cameras to record the object studied which might be useful in the following cases:

- Creating 3D models of the object studied from the recordings, these models could then be used for finite element analysis and assist in the blind identification of modal properties
- Improve accuracy of data extracted as we have 2 or more sources of data to compare against each other and against other sources.

Future work is not only focused on the image processing part but also on the automation part, efforts to make this program completely autonomous can be made, starting with automatic object identification to automatic parameter selection suggested above.

To summarize, this projects key findings are the following:

- PBMM when combined with 2D PT in indoor conditions for frequency extraction might provide slightly inaccurate results.
- Phase based motion magnification proved to be useful in aiding 2D PT for frequency extraction in noisy outdoor conditions.
- Denoising techniques (Denoising convolutional neural network for example) used before PBMM processing allowed more accurate results to be extracted.

This projects key contributions are the following, first, we showed that PBMM aided 2D PT in extracting more accurate results when used under noisy outdoor conditions.

Secondly, we tested the suitability of PBMM in outdoor scenarios (light pole and building case) and found it to be reliable in noisy conditions.

Thirdly, the use of relatively low cost cameras proved to provide satisfactory results, which would pave the way to eliminate the need for expensive equipment and the need for training.

Last but not least, techniques for automatic parameter selection were researched, created and suggested in this study.

The software used to make this project possible are Matlab to run PBMM, 2D PT etc. Ansys workbench for Finite element analysis.

## 7 REFERENCES

1. J. Morlier, P. Salom, and F. Bos, "New image processing tools for structural dynamic monitoring," *Key Engineering Materials*, vol. 347, pp. 239–244, 2007.
2. "Lucas-Kanade in a Nutshell" Prof. Dr. Raul Rojas
3. N. Wadhwa, J. G. Chen, J. B. Sellon et al., "Motion microscopy for visualizing and quantifying small motions," *Proceedings of the National Academy of Sciences*, vol. 114, no. 44, Article ID 11639, 2017.
4. M. Civera et al., "An Experimental Validation of Phase-Based Motion Magnification for Structures with Developing Cracks and Time-Varying Configurations", *Hindawi Shock and Vibration Volume 2021*, Article ID 5518163.
5. Hao-Yu Wu et al., "Eulerian Video Magnification for Revealing Subtle Changes in the World", *Quanta Research Cambridge, Inc.*
6. Yongchao Yang et al., "Blind identification of full-field vibration modes from video measurements with phase-based video motion magnification", *Volume 85*, 15 February 2017, Pages 567-590
7. Xiao-Yang Zhao et al., "A Convolutional Neural Networks Denoising Approach for Salt and Pepper Noise", *School of Computer and Information Technology, Liaoning Normal University, China.*
8. Zhu Mao et al., "Motion magnification for optical-based structural health monitoring", *Conference Paper · April 2020*
9. Neal Wadhwa et al., "Phase-Based Video Motion Processing", *MIT Computer Science and Artificial Intelligence Lab*
10. Eero P. Simoncelli et al., "Shiftable Multiscale Transforms", *IEEE TRANSACTIONS ON INFORMATION THEORY*, VOL 38, NO.2 MARCH 1992.
11. Neal Wadhwa et al., "Riesz Pyramids for Fast Phase-Based Video Magnification", [MIT Computer Science and Artificial Intelligence Laboratory](#)
12. Yan Liu et al., "Efficient Patch-Wise Semantic Segmentation for Large-Scale Remote Sensing Images", *School of Automation & Electrical Engineering, University of Science and Technology Beijing.*

13. Eitner M. et al., “Effect of broad-band phase-based motion magnification on modal parameter estimation”, *Mechanical Systems and Signal Processing*, Volume 146, 1 January 2021, 106995.
14. WU, H.-Y., RUBINSTEIN, M., SHIH, E., GUTTAG, J., DURAND, F., AND FREEMAN, W. 2012. Eulerian video magnification for revealing subtle changes in the world. *ACM Trans. Graph. (Proc. SIGGRAPH)* 31 (aug).
15. People.Csail.Mit, “Motion Microscopy for Visualizing and Quantifying Small Motions,” 2020, <http://people.csail.mit.edu/nwadhwa/motion-microscope/>.
16. Y.-J. Cha, J. Chen, and O. Buyukozturk, “Motion magnification based damage detection using high speed video,” in *Proceedings of the 10th International Workshop on Structural Health Monitoring (IWSHM)*, September 2015.
17. M. Civera, C. Surace, L. Zanotti Fragonara, and C. Surace, “An experimental study of the feasibility of phase-based video magnification for damage detection and localisation in operational deflection shapes,” *Strain*, vol. 56, Article ID e12336, 2020.
18. A. Sarrafi, Z. Mao, Wind turbine blade damage detection via 3-dimensional phase-based motion estimation, in: *The 11th International Workshop on Structural Health Monitoring*, 2017, <https://doi.org/10.12783/shm2017/14154>.
19. A. Sarrafi, Z. Mao, C. Niezrecki, P. Poozesh, Vibration-based damage detection in wind turbine blades using phase-based motion estimation and motion magnification, *J. Sound Vib.* 421 (2018) 300–318, <https://doi.org/10.1016/j.jsv.2018.01.050>.
20. J.G. Chen, N. Wadhwa, Y.-J. Cha, F. Durand, W.T. Freeman, O. Buyukozturk, Modal identification of simple structures with high-speed video using motion magnification, *J. Sound Vib.* 345 (2015) 58–71, <https://doi.org/10.1016/j.jsv.2015.01.024>.
21. L. Meirovitch, *Fundamentals of Vibration*, first ed., McGraw-Hill Higher Education, McGraw-Hill, 2001.
22. Kanjar DeV. And V.Masilaman, “Image Sharpness Measure for Blurred Images in Frequency Domain” , 2013.

23. Ce Liu et al., "Motion Magnification", Computer Science and Artificial Intelligence Lab. (CSAIL) Massachusetts Institute of Technology.
24. Justin G. Chen et al., "Camera-Based Vibration Measurement of the World War I Memorial Bridge in Portsmouth, New Hampshire", 2018.
25. Ashim Khadka et al., "Non-contact vibration monitoring of rotating wind turbines using a semi-autonomous UAV", 2020.
26. C. Z. Dong et al., "Non-contact structural vibration monitoring under varying environmental conditions", 2015.
27. Raj Rakshit et al., "On Characterization of Vibration Measurement Using Microwave Doppler Radar", 2018.
28. Stephen M. Talai et al., "Infrared thermography applied to the prediction of structural vibration behaviours", 2019.

## 8 APPENDIX

- Ce Liu et al., "Motion Magnification", Computer Science and Artificial Intelligence Lab MIT.
- Tae-Hyun Oh et al., "Learning-based Video Motion Magnification", MIT CSAIL.
- Neal Wadhwa et al., "Motion microscopy for visualizing and quantifying small motions", MIT CSAIL.
- Digital Image Processing, third edition, (Gonzalez & Woods)
- "[Collapse of the Tacoma Narrows Bridge](#)", November 2016 (Volume 25, Number 10)
- Billah, K. and Scanlan, R. "Resonance, Tacoma Narrows Bridge Failure, and Undergraduate Physics Textbooks," *American Journal of Physics* 59 (1991): 118–124.
- [Remove noise from color image using pretrained neural network](#)
- [Image super resolution using GAN](#)

| Details of "Mesh"                         |                        |
|-------------------------------------------|------------------------|
| <b>Display</b>                            |                        |
| Display Style                             | Use Geometry Setting   |
| <b>Defaults</b>                           |                        |
| Physics Preference                        | Mechanical             |
| Element Order                             | Program Controlled     |
| <input type="checkbox"/> Element Size     | Default                |
| <b>Sizing</b>                             |                        |
| Use Adaptive Sizing                       | Yes                    |
| Resolution                                | Default (2)            |
| Mesh Defeaturing                          | Yes                    |
| <input type="checkbox"/> Defeature Size   | Default                |
| Transition                                | Fast                   |
| Span Angle Center                         | Coarse                 |
| Initial Size Seed                         | Assembly               |
| Bounding Box Diagonal                     | 8.2092 m               |
| Average Surface Area                      | 0.32231 m <sup>2</sup> |
| Minimum Edge Length                       | 2.7978e-002 m          |
| <b>Quality</b>                            |                        |
| Check Mesh Quality                        | Yes, Errors            |
| Error Limits                              | Aggressive Mechanical  |
| <input type="checkbox"/> Target Quality   | Default (0.050000)     |
| Smoothing                                 | Medium                 |
| Mesh Metric                               | None                   |
| <b>Inflation</b>                          |                        |
| Use Automatic Inflation                   | None                   |
| Inflation Option                          | Smooth Transition      |
| <input type="checkbox"/> Transition Ratio | 0.272                  |
| <input type="checkbox"/> Maximum Layers   | 5                      |
| <input type="checkbox"/> Growth Rate      | 1.2                    |
| Inflation Algorithm                       | Pre                    |
| View Advanced Options                     | No                     |
| <b>Advanced</b>                           |                        |
| Number of CPUs for Parallel Pa...         | Program Controlled     |
| Straight Sided Elements                   | No                     |
| Rigid Body Behavior                       | Dimensionally Reduced  |
| Triangle Surface Mesher                   | Program Controlled     |
| Topology Checking                         | Yes                    |
| Pinch Tolerance                           | Please Define          |
| Generate Pinch on Refresh                 | No                     |
| <b>Statistics</b>                         |                        |
| <input type="checkbox"/> Nodes            | 3184                   |
| <input type="checkbox"/> Elements         | 1091                   |

Figure 50 Light Pole Mesh Properties

## 9 Matlab code

### 9.1 Signal to noise ratio

```
video=VideoReader('60fps_1080.mp4');

im = read(video,1);

vid=VideoReader('noisy_vid.avi');
img_noisy = read(vid,1);

im = imresize(im, [size(img_noisy,1) size(img_noisy,2)]);

noise = abs(img_noisy - im);

% Calculate the mean of each
signalMean = mean(im(:))

noiseOnlyMean = mean(noise(:))
% Get ratio of the means
snr1 = signalMean/noiseOnlyMean
```

### 9.2 Add noise and denoise

```
%% add noise and denoise
for k=1:length(Files)

    name =
    sprintf('D:/Msc/motion_magnification/output_images3/%d.png',k);
    img = imread(name);

    % Add noise
    noisyRGB = imnoise(img,'gaussian',0,0.025);

    noisy_dir =
    sprintf('D:/Msc/motion_magnification/noisy_images3/%d.png',k);
    imwrite(noisyRGB, noisy_dir);
    % Denoise

    nRGB = noisyRGB;

    [noisyR,noisyG,noisyB] = imsplit(nRGB);
    denoisedR = denoiseImage(noisyR,denoise_net);
    denoisedG = denoiseImage(noisyG,denoise_net);
    denoisedB = denoiseImage(noisyB,denoise_net);
    nRGB = cat(3,denoisedR,denoisedG,denoisedB);

    output_dir =
    sprintf('D:/Msc/motion_magnification/denoised_3_my_net/%d.png',k);
    imwrite(nRGB, output_dir);
end
```

### 9.3 PBMM & 2D PT

```
clearvars

resultsDir = 'D:\Msc\motion_magnification';
% mkdir(resultsDir);
```

```

% defaultPyrType = 'halfOctave'; % Half octave pyramid is default as
discussed in paper
scaleAndClipLargeVideos = true; % With this enabled, approximately 4GB of
memory is used

% Uncomment to use octave bandwidth pyramid: speeds up processing,
% but will produce slightly different results
defaultPyrType = 'octave';

% Uncomment to process full video sequences (uses about 16GB of memory)
%scaleAndClipLargeVideos = false;

% Crane
inFile = VideoReader('noisy_full_0.01.avi');
samplingRate = inFile.FrameRate;
windowLength = 4;

Spec_storage_Y = [];

tempSpec_Cluster_1 = [];
tempSpec_Cluster_2 = [];

for n = 0.01:windowLength: 4*((samplingRate/2) -1)

    loCutoff = n;      % Hz
    hiCutoff = loCutoff + windowLength ;      % Hz
    if n >=4
        break
    end
    if hiCutoff >= (samplingRate/2) -1
        hiCutoff = (samplingRate/2) -1;
    end

    fprintf('Frequency Range = %f to %f \n \n', loCutoff, hiCutoff);

    alpha = 10;
    sigma = 2;      % Pixels
    temporalFilter = @FIRWindowBP;
    pyrType = defaultPyrType;

    name = sprintf('noisy_full_0.01.avi-FIRWindowBP-band%0.2f-%0.2f-sr%i-
alpha%d-mp0-sigma%d-scale0.50-frames1-900-
octave.avi', loCutoff, hiCutoff, samplingRate, alpha, sigma);

    if isfile(name) == false
        if (scaleAndClipLargeVideos)
            phaseAmplify(inFile, alpha, loCutoff, hiCutoff, samplingRate,
resultsDir, 'sigma', sigma, 'pyrType', pyrType, 'temporalFilter',
temporalFilter, 'scaleVideo', 1/2);
        else
            phaseAmplify(inFile, alpha, loCutoff, hiCutoff, samplingRate,
resultsDir, 'sigma', sigma, 'pyrType', pyrType, 'temporalFilter',
temporalFilter, 'scaleVideo', 1);
        end
    end
end

```

```
    end
end
```

### 9.3.1 Point Tracking

```
videoReader = VideoReader(name);
videoPlayer = vision.VideoPlayer(); % 'Position',[100,100,680,520]

objectFrame = readFrame(videoReader);
% objectFrame = imresize(objectFrame,0.5);
% imshow(objectFrame);
objectRegion =[269,270,9,180];

objectImage =
insertShape(objectFrame,'Rectangle',objectRegion,'Color','red');
figure;
imshow(objectImage);

% title('Red box shows object region');

tracker = vision.PointTracker('MaxBidirectionalError',2); % 1 pixel
difference between forward and backward pass
% % Initialize the tracker.
% objectFrame = readFrame(videoReader);

points = detectMinEigenFeatures(im2gray(objectFrame),'ROI',objectRegion);
%

% num_of_points_to_track = 15;
%
% figure % Use this method to select new points
% imshow(objectFrame);
% coord=[];
% for i = 1:num_of_points_to_track
%     h =drawpoint;
%     coord =[coord; get(h,'Position')];
% end

% OR
% % Use already selected points

% coord =[];

initialize(tracker,points.Location,objectFrame);
setPoints(tracker,coord);
x_point = [];
y_point = [];

while hasFrame(videoReader)
    frame = readFrame(videoReader);
    [points,validity] = tracker(frame);
```

```

        y_point = [y_point, points(:,2)];
        x_point = [x_point, points(:,1)];
        out = insertMarker(frame,points(validity, :),'+');
        videoPlayer(out);
    end

    %% eliminate invalid points
    valid_points = find(validity == 1);

    temp_x = zeros(size(valid_points,1),size(x_point,2));
    temp_y = zeros(size(valid_points,1),size(x_point,2));

    for r = 1:size(valid_points)

        temp_x(r,:) = x_point(valid_points(r,1),:);
        temp_y(r,:) = y_point(valid_points(r,1),:);

    end

    x_point =temp_x;
    y_point = temp_y;

    final_frame_xyCoord =
    [x_point(:,size(x_point,2)),y_point(:,size(y_point,2))];

    num_of_clusters = 2;
    kidx = kmeans(final_frame_xyCoord,num_of_clusters);
    gscatter(x_point(:,size(x_point,2)),y_point(:,size(y_point,2)),kidx);
    set(gca, 'YDir','reverse')
    title('DBSCAN Using K-means')

    %1st cluster
    temp_x_cluster_1 = [];
    temp_y_cluster_1 = [];
    % 2nd cluster
    temp_x_cluster_2 = [];
    temp_y_cluster_2 = [];

    counter_1=1;
    counter_2=1;
    for num = 1:size(kidx,1)

        if kidx(num,1) == 1

            temp_x_cluster_1(counter_1,:) = x_point(num,:);
            counter_1 =counter_1+1;

        elseif kidx(num,1) == 2

            temp_x_cluster_2(counter_2,:) = x_point(num,:);
            counter_2 =counter_2+1;

        end
    end
end

```

### 9.3.2 Filtering

```
highpassCutoff = 1;
lowpassCutoff = 3;

fs = videoReader.FrameRate;
%lowpassCutoff =(fs/2)-1;
nyquistRate = fs/2;
order = 4;
[B_high, A_high] = butter(order, highpassCutoff/nyquistRate, 'high');
[B_low, A_low] = butter(order, lowpassCutoff/nyquistRate, 'low');

xf3= zeros(size(temp_x_cluster_1,1),1,size(temp_x_cluster_1,2));
yf3=zeros(size(temp_x_cluster_2,1),1,size(temp_x_cluster_2,2));

for r =1:size(x_point,2)
    xf3(:,1,r) = temp_x_cluster_1(:,r);
    yf3(:,1,r) = temp_x_cluster_2(:,r);

end

sig_arr = [];
sig_arr_y = [];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

points_to_filter = squeeze(xf3);
if size(xf3,1) == 1
    points_to_filter = temp_x_cluster_1(1,:);
end
spectrum_all_x = [];

    for g =1:size(points_to_filter,1)

        my = mean(points_to_filter(g,:));
        variancey = var(points_to_filter(g,:));
        point_being_filtered = (points_to_filter(g,:) - my)/variancey;

        high_filt = filter(B_high,A_high, point_being_filtered);
        point_being_filtered = filter(B_low,A_low, high_filt);

        x_fft=  fft(point_being_filtered);
        xdft_f = (1/length(point_being_filtered))*x_fft;
        freq_f = -(fs/2):(fs/length(point_being_filtered)):(fs/2)-
        (fs/length(point_being_filtered));
        spec = xdft_f;
        spectrum_f = fftshift(xdft_f);
        spectrum_f = spectrum_f(ceil(length(spectrum_f)/2) +
1:length(spectrum_f));
        if length([freq_f(freq_f>=0)])~= length(spectrum_f)
            freq_f = [0 freq_f(freq_f>=0)];
        else
            freq_f = freq_f(freq_f>=0);
        end

        spectrum_all_x = [spectrum_all_x;abs(spectrum_f)];
```

```

end

points_to_filter = squeeze(yf3);
if size(yf3,1) == 1
    points_to_filter = temp_x_cluster_2(1,:);
end
spectrum_all_y = [];

for g =1:size(points_to_filter,1)

    my = mean(points_to_filter(g,:));
    variancey = var(points_to_filter(g,:));
    point_being_filtered = (points_to_filter(g,:) - my)/variancey;

    high_filt = filter(B_high,A_high, point_being_filtered);
    point_being_filtered = filter(B_low,A_low, high_filt);

    x_fft= fft(point_being_filtered);
    xdft_f = (1/length(point_being_filtered))*x_fft;
    freq_f = -(fs/2):(fs/length(point_being_filtered)):(fs/2)-
(fs/length(point_being_filtered));
    spec = xdft_f;
    spectrum_f = fftshift(xdft_f);
    spectrum_f = spectrum_f(ceil(length(spectrum_f)/2) +
1:length(spectrum_f));
    if length([freq_f(freq_f>=0)])~= length(spectrum_f)
        freq_f = [0 freq_f(freq_f>=0)];
    else
        freq_f = freq_f(freq_f>=0);
    end

    spectrum_all_y = [spectrum_all_y;abs(spectrum_f)];

end

end

```

## 9.4 Train dCNN on custom data

```

imds = imageDatastore('D:/Msc/motion_magnification/new_images_1/');
dnimds = denoisingImageDatastore(imds, 'PatchSize', 50
, 'ChannelFormat', 'grayscale' );
layers = dnCNNLayers;

options = trainingOptions('adam', ...
    'MaxEpochs',5, ...
    'MiniBatchSize',90, ...
    'Plots','training-progress','InitialLearnRate',0.001);
% 'LearnRateSchedule','piecewise', ...
% 'LearnRateDropFactor',0.2, ...

```

```
%      'LearnRateDropPeriod',1, ...

net = trainNetwork(dnimds, layers, options);
```

### 9.4.1 Test custom network

```
i = imread('1.png');
imshow(i)

[noisyR, noisyG, noisyB] = imsplit(i);
net2 = denoisingNetwork('dncnn');
denoisedR = denoiseImage(noisyR, net2);
denoisedG = denoiseImage(noisyG, net2);
denoisedB = denoiseImage(noisyB, net2);
nRGB = cat(3, denoisedR, denoisedG, denoisedB);
figure()
imshow(nRGB)
```

## 9.5 Blind identification

```
order = 7;
for o = 3:order

    fprintf('order :%d \n', o)
    p = polyfit(freq(1, 1:length(multiply_array)), multiply_array, o);

%      p_prime= polyder(p);
%      answer = polyval(p_prime, 0.35);
%      p_second_prime = polyder(p_prime);

%% extract coefficients and create eq of line

syms x n
order_counter= o;
f=0;

for i = 1:o+1

    f= f + p(1,i)*x^order_counter;

    order_counter = order_counter-1;

end

%% 1st derivative
f_prime = diff(f);

%% find where we have min or max (x coord)
Sol = vpasolve(f_prime == 0, x);
```

```

%% 2nd derivative
f_second_prime = diff(f_prime);

%% check if x corresponds to min or max

% if 2nd derivative at x is negative --> we have a maximum
% if 2nd derivative at x is positive --> we have a minimum

syms x
c = coeffs(f_second_prime, x); % extract coeffs of 2nd derivative

x_where_we_have_minimum = [];

b = 1;

for j = 1:length(Sol)

    order_2nd_deriv = o - 2;
    order_2nd_deriv_counter = order_2nd_deriv;
    f2p = 0;
    for i = 1:order_2nd_deriv + 1

        f2p = f2p +
c(1,order_2nd_deriv_counter+1)*Sol(j,1)^order_2nd_deriv_counter;

        order_2nd_deriv_counter = order_2nd_deriv_counter-1;

    end
    if j == 1 && f2p > 0
        b = 0;
    end

    if f2p > 0
        x_where_we_have_minimum = [x_where_we_have_minimum,
Sol(j,1)];
    end
end

if b == 0
    x_where_we_have_minimum = [x_where_we_have_minimum];
else
    x_where_we_have_minimum = [0, x_where_we_have_minimum];
end

    corresponding_frequencies_TO_STORE = [];
%% find peaks
num_of_peaks_to_find = 2;
peaks = [];
corresponding_frequencies = [];

array_freq = [];
array_spectrum = [];
z = [];

```

```

        if num_of_peaks_to_find == length(x_where_we_have_minimum)

            x_where_we_have_minimum = [x_where_we_have_minimum,
freq(1,length(multiply_array))]]

        end

        for k=1:num_of_peaks_to_find
            array_freq=[];
            array_spectrum=[];
            z=find(freq > x_where_we_have_minimum(1,k) & freq <
x_where_we_have_minimum(1,k+1));

            if max(z) > length(multiply_array)

                z(1,find(z>=length(multiply_array))) =
length(multiply_array);

            end

            array_freq = freq(1,min(z):max(z));
            array_spectrum = multiply_array(1,min(z):max(z));
            [M,I] = max(array_spectrum);
            peaks = [peaks, M];
            corresponding_frequencies = [corresponding_frequencies,
array_freq(1,I)];
            % corresponding_frequencies_TO_STORE =
[corresponding_frequencies_TO_STORE; corresponding_frequencies];
            end

            % store all values of 'corresponding frequencies' for all orders of
eq, if same for
            % all, choose lowest order, if not ???

            str={};
            switch num_of_peaks_to_find
                case 1
                    str={'1st Peak'};
                case 2
                    str={'1st Peak' '2nd Peak'};
                case 3
                    str={'1st Peak' '2nd Peak' '3rd Peak'};
                case 4
                    str={'1st Peak' '2nd Peak' '3rd Peak' '4th Peak'};
            end

            figure()
            plot(freq(1,1:length(multiply_array)),multiply_array)

            hold on
            plot(corresponding_frequencies,peaks,'r*')
            text(corresponding_frequencies,peaks,str)
            hold off
            xlabel('Frequency (Hz)');
            ylabel('Magnitude');
            title ('Blind Peak Detection results');
        end
    end

```

## **10 Publishing**

Publishing to a journal will be considered once the final satisfactory outcome of this MSc is received. Once received and in coordination with my supervisor, plans for publication will be drawn and executed.