NATIONAL RESEARCH UNIVERSITY
HIGHER SCHOOL OF ECONOMICS

Faculty of Computer Science
Bachelor's Programme "Data Science and Business Analytics"

Software Team Project Report on the Topic:

Computer-vision-based Non-contact Vibration Diagnostics

(interim, the first stage)

Submitted by the Students:

group #БПАД222, 2rd year of study	Bokhyan Roman Beniaminovich
group #БПАД222, 2rd year of study	Sukhoparov Timophey Maksimovich
group #БПАД222, 2rd year of study	Argirov Georgy Atanasovich

Approved by the Project Supervisor:

Pervishko Anastasiia Aleksandrovna
Assistant Professor
Project Center for Next Generation Wireless and IoT, Skoltech

Moscow 2024

Contents

Annotation

This project aims to introduce and optimize alternative approaches to vibration analysis. Using computer vision based algorithms, the proposed method, which will be implemented, provides non-contact vibration diagnostics. By capturing and matching filtered event-frames of an object, the system will provide a more accurate and error-free representation of vibration patterns.

Аннотация

Целью данного проекта является внедрение и оптимизация альтернативных подходов к анализу вибраций. Используя алгоритмы, основанные на компьютерном зрении, предлагаемый метод обеспечит бесконтактную вибродиагностику. Захватывая, фильтруя и сопоставляя кадры объекта, предложенный подход обеспечит получение боллее точных результатов

Keywords

Condition monitoring, Event-based(neuromorphic) camera, High-speed camera, Micro-vibration, Vibration diagnostics, Fourier Transformation, Image processing

1 Introduction

On every modern production line it is crucial to control every mechanism in order to ensure the robustness of the system and the smooth operation of production. Constant machine monitoring allows companies to prevent errors and failures before they even happen. There are numerous ways of constant monitoring and checking the stability of some mechanisms. However, using vibration analysis stands out in predictive maintenance, it is being used in many spheres [16], as it makes early detection of potential faults possible.

The next challenge is conducting vibrations measurements and ensuring their reliability and truthfulness. Most of the methods which were introduced before included attachment of physical sensors. These approaches are indeed quite effective, however they have a serious drawback - due to direct placement on an object the data is getting distorted by the movement of a sensor itself, moreover the area of study is limited by the sensor placement, therefore, a unique approach should be developed for each mechanism [8], there were also some attempts of making such methods universal [4]. To this date these problems remain unsolved and many companies spend considerable part of their budget on the failures that remained undetected due to the limitations of traditional physical sensor-based vibration measurement methods.

For this reason the totally new methods had to be introduced and the main goal of these methods is to get rid of the impact of the external environment on ongoing measurements and experiments. One of the methods that may be considered contact-less is using laser vibrometer [6], even though these method is a considerable improvement compared to physical sensors, using laser vibrometers for vibration measurements brings its own challenges. The accuracy of measurements is influenced by factors such as surface reflectivity. Maintaining the correct distance and focus between the vibrometer and the target surface is crucial, and variations in these parameters can result in errors. One of the other methods is infrared thermography [15], however the problem behind it is obvious, in condition with critical temperatures it becomes unreliable at all. So the final variant that comes to help and becomes more and more popular in all industries is vision-based methods. These methods for non-contact vibration measurements utilize high-speed cameras or event-based camers and computer vision techniques to capture and analyze the movement of specific points on the surface of an object. This approach offers several advantages, including the ability to measure deformations with high resolution and in a non-intrusive manner.

Our project aims to implement and optimize one of such vision-based approaches that will allow us to measure vibration levels using videos recorded on high-speed or neuromorphic cameras. As part of our study we decided to use both high-speed camera with combination of methods to transform high-speed camera captured frames into event frames and the event-based (neuromorphic) cameras. As the result we will get to different data inputs one event-frames produced by neuromorphic cameras and other event-frames assembled by methods that going to be introduced later. Below you can see the structure of this idea, which includes the planned implementation of the algorithm and the development of the application, as well as distribution of responsibilities. Details of the implementation ideas can be found in the sections "Plan of future works" and "Division of implementation by participants" where each part will be described in as much detail as possible by the student who will implement that part – link to the parts.

Algorithm:

1 Retrieving data from either transformed frames of the high speed camera or even-frames created by event-based camera and to enhance the final results and ease the execution of other steps of algorithm: clearing, normalizing and denoising the obtained frames.

2 Frames correlation and template matching. The part is aimed at finding the region of the source frame that matches the given image-template as closely as possible, using the normalized mutual correlation function to estimate the degree of correlation without scaling or rotating the template (this is a planned implementation of template matching and may eventually use another more optimized version if appropriate)

3 Visualization. The plan is to develop a user-friendly application in which all interaction and results will take place.

2 Overview and comparative analysis

When diving into the world of constant monitoring and specifically vision-based micro vibration measurements, it becomes relatively quickly clear that there is no simple user-friendly app or system, which will make use of either event-based cameras or high-speed cameras to effectively and relatively fast, measure the vibration of the object of study.

The main focus of our project is scoping through all the available knowledge and approaches in vision-based vibration diagnostics that already exist. Simply said, the main advantage of our project is using several methods and combining them, to get more advanced result. For example, combination of denoising [1] and several methods of event-frames filtering will result in a better event-frame quality which is crucial in vibration detection. Our approach will include image processing before event-frames are used further in our process. Moreover, by combining several approaches such as edge tracking, template matching, optical flow tracking, will allow user get the best data possible [5, 14, 9]. Existing papers proposing various vibration measurement and image manipulation techniques have ubiquitously used the Fourier transform. In particular, one of the main articles on this topic is [14]. It also describes the usual Fourier transform, but one of our ideas would also be to try to optimise the Fourier transform.

In addition, while discussing the methods of data processing and analytics is important, our projects aims to take it further. After scoping the market it became obvious that there are no solutions which are using described methods, algorithms and moreover they include convenient world-wide accessible free to use app. For average customer that will use constant monitoring techniques it is crucial for approach to include:

- Easy to use app
- Various methods
- Acceptable budget
- Convenient data visualization

After we conducted the analysis of described approaches and methods of implementations, our project will take into account the shortcomings of current methods and it will be able to become an excellent assistant in industry and in those projects where it is necessary to engage in constant monitoring, in particular vibration diagnostics.

3 Plan of the upcoming work

3.1 Event-frame generation and filtering (by Georgy Argirov):

3.1.1 Generalization

The following part of this project is dedicated to an in-depth exploration of methods aimed at capturing high-quality event-frames in the most efficient way possible. The primary objective is to research the most innovative approaches that not only enhance the quality of event-frames but also optimize the speed of their generation. This research and further implementation will elevate the precision and effectiveness of our non-contact vision-based method for analyzing object vibrations.

The main challenges of that part that are planned to be resolved are the following:

- Exploring both methods: using high-speed cameras, using event-based cameras. Analyzing in what conditions which method will be more effective and what will influence our decisions.

- In depth analyze of the process of extracting video frames from the video sample received from the high-speed camera. Experimenting and investigating how the frame rate of the high-speed camera impacts the capturing of vibrations.

- Considering different and developing techniques of the transformation of regular frames into event-frames. Explore various computer vision techniques like background subtraction, edge detection, and optical flow algorithms.

- Examining and implementing the most effective way of normalizing the obtained event-frames. As well as evaluating the problem of noise and irregularities in the event frames. Introducing different event-filtering techniques and adaptive filter creating algorithms to deal with that problem.

- Develop and analyze quantitative metrics for evaluating the performance of the entire process. Provide the numerical way of evaluating the efficiency of received frames. Based on the results from the developed metrics adjust and optimize the design of the solution.

- (Possible additional task) Explore and discuss the possibility of the algorithm scalability in terms of real-time processing.

3.1.2 Extracting video frames and frame rate impact

As the first method of receiving event-frames that we are going to discuss includes recording video of the object using the high-speed cameras it is important to take into account how frame-rate impacts the results and precision. And indeed frame-rate has a huge impact on results, and it may be obvious to say that the higher the frame rate the better the pixel movement detection is, however it is not that obvious. According to the Nyquist-Shannon sampling theorem, the sampling rate must be at least twice the highest frequency present in the signal to avoid aliasing, so frame rate for example of 2000 fps should ensure correct analysis of recorded images of measurement object in frequency range : 0 up to 1000 Hz [7]. Moreover the insufficient frame-rate will lead to considerable errors which was proved already [17]. However when using high-frame rate video samples it sometimes impacts such methods as edge-detection as images become more noise-infected with increase of fps [10].

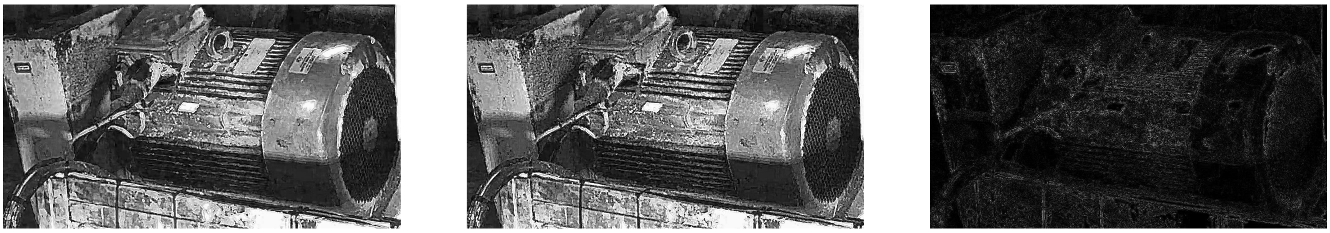
Getting back to frame extraction, based on our frame rate conclusion, and using built-in methods, specifically "VideoCapture" class from OpenCV [12] allows us to adjust frame rate to our preferences mentioned above, when transforming video fragment into series of images.

3.1.3 Event-frame generation

The next step of our algorithm, is analysing the received frames and creating event-frames based on them. The most reliable and obvious method to receive event-frame is by finding the absolute difference between two "neighbor" images (which were transformed to gray-scale format beforehand), in that case we can make user of another OpenCV [12] method, ".absdiff": The function cv::absdiff calculates: Absolute difference between two arrays when they have the same size and type:

$$dst(I) = saturate(|src1(I) - src2(I)|)$$

By processing every pair of two images we will receive the set of ndarrays with basically the changes in every two images, that will be our "unpolished" event-frames.



(a) Previous image.

(b) Following image.

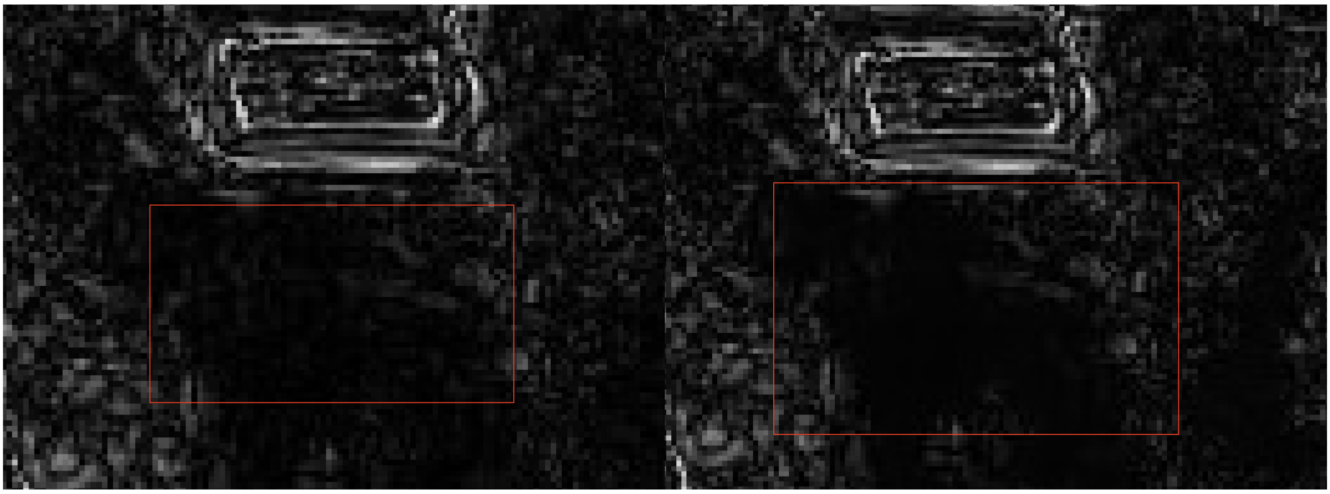
(c) Raw event image.

Figure 3.1: The process of generation of the raw event frame.

3.1.4 Event-frame filtering and clearing

The most crucial and important step of this part of the project is how one can filter the raw event-image and leave only valuable information on it, as well as clear all error and noise from it. Various techniques exists and as one of achievements of ours project will be a successful implementation of one of the methods and optimizing it in terms of time that is needed to process one image.

Let's start with how we can get rid of noise which will impact our vibration diagnostics badly. One of the most effective methods that may be used in this project and was implemented as a start with plans to test it in future work, is the Non-Local Means Denoising [1]. The principle of the first denoising methods was quite simple: replacing the color of a pixel with an average of the colors of nearby pixels, but no one can guarantee that the nearby pixels will be similar, so we need to search somewhere else. The idea of this method is computing the average color of the most resembling pixels. The resemblance is evaluated by comparing a bigger area around each pixel that is being investigated, this new filter is called non-local means.



(a) Before (b) After

Figure 3.2: The process of denoising using NL-mean-denoising approach.

After enhancing the quality of the resulting frames we need to understand how we want to get in the focus of our attention the main object to study it's vibrations. To do it we need to throw everything else away from our event-frames and to do this we need to design some kind of a event-filter [14], who's "main role" will be to adapt for every frame and depict an object of studies on it. The core idea here is that we limit below the absolute differences in intensities of one pixel in two frames with some coefficient c and throw everything else away: $|I(i,j)_n - I(i,j)_{n+1}| > c$

One of the plans is to design the following event filter: $F_t = \sum_{n=0}^{k} b_n * F_{t-n} \sum_{m=1}^{k} a_m * F_{t-m}$, which will adapt to every frame (for example by implementing genetic algorithm, which will in its place adjust parameters b_n and a_m accordingly). However it is not the final approach that will be taken into account. The methods that were also mentioned above in the "generalization", such as edge detection [5] will be also considered, used and implemented.

As a result we will get a well-structured combination of already implemented denoising methods with a thorough design of a suggested event-filter, which will allow us to maximize the quality and efficiency of event-data we are receiving from both high-speed and event-based cameras.

3.1.5 Metrics to evaluate the quality of event-frames

Evaluating the quality of event frames is crucial to ensure that the detected events accurately represent occurrences in the data. Several metrics can be used to assess the performance of event frame detection methods, however the choice of these metrics will be influenced by final parts and methods which will be used in the final design of the event-frames generator.

3.2 Pattern matching (by Roman Bokhyan):

3.2.1 Generalization

This describes a potential implementation and the final version may differ from what is described here. The goal of this part of the project is to find the relative shift between frames, so that we can then use the result to derive vibration levels and visualize this data. Pattern matching can be implemented in a variety of ways, but we plan to focus on the phase correlation [14] method of pattern matching, which uses the frequency domain to accurately determine the shifts between images. This method is characterized by its robustness to illumination changes and its ability to provide accurate subpixel motion detection. By utilizing the unique properties of phase correlation, we aim to enhance the ability to effectively monitor and analyze vibrations. The following will describe implementation ideas, method descriptions, and at the end a rough outline of how to do this using the python language and its libraries.

3.2.2 How correlation applies

Correlation plays an important role in data analysis and interpretation by assessing the degree of association between different variables. In the context of video analysis, this approach is used to identify similarities between successive video frames, allowing the detection of object movement or changes in appearance. In the industrial world, correlation allows tracking object movements and deformations, which is important for object monitoring [2].

3.2.3 Phase Correlation for Template Matching - PCTM

Phase correlation is a technique used to determine the relative shift between two images or frames. Its advantage over other methods is its robustness to illumination, making it an ideal way to analyze video recorded on a high-speed camera. This method uses the Fourier transform [3] to translate the image into the frequency domain (it is important to note that Fourier translates regardless of the position of the object in the frame). The phase correlation itself calculates the cross power spectrum of two frames [14], which allows you to accurately determine the shifts between them, including if they are not just pixils but subpixils.

3.2.4 Fourier Transformation

Before describing the Fourier application in our case there will be a brief explanation of what a Fourier transform is. *The Fourier transform* [3] is a representation of an image as a sum of complex exponentials of varying magnitudes, frequencies, and phases. The Fourier transform plays a critical role in a broad range of image processing applications, including enhancement, analysis, restoration, and compression.Two formulas, Fourier transform and inverse Fourier transform, will be described below.

The discrete Fourier transform (DFT) of an image f of size $M \times\ N$ is an image F of same size defined as:

$$F(u,v) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)e^{-j\,2\pi\left(\frac{um}{M}+\frac{vn}{N}\right)} \tag{1}$$

The inverse discrete Fourier transform computes the original image from its Fourier transform:

$$f(m,n) = \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{+j\,2\pi\left(\frac{um}{M}+\frac{vn}{N}\right)} \tag{2}$$

For convenience, the Fourier series is denoted by $\mathcal{F}$ for the Fourier transform and $\mathcal{F}^{-1}$ for the inverse Fourier transform, respectively.

Formula notations

- $F(u,v)$ is the Fourier Transform of the image at frequencies u and v.
- $f(m,n)$ is the intensity of the pixel at coordinates (m,n).
- M and N are the number of rows and columns in the image, respectively.

- m and n are indices over the spatial domain of the image.
- u and v are indices over the frequency domain.
- The exponential term $e^{-j2\pi\left(\frac{um}{M}+\frac{vn}{N}\right)}$ is the kernel of the Fourier transform, incorporating both the real and imaginary parts of the Fourier coefficients.

A description of how the DFT and inverse Fourier transform work. Note that, *the spatial domain* represents the location of pixels or points in the image space, and *the frequency domain* represents the change in signal intensity in that space, determining how fast the pixel values change.

DFT

- The input is an image (in spatial domain) (e.g. in our case it will be an event-frame) where each pixel has a different intensity.
- Centering. Centering is often used to move low frequencies to the center of the spectrum. This could be done by multiplying the pixel intensity by $(-1)^{x+y}$.
- Direct Fourier applications.Using the 2D Fourier property that considers frequencies in two spatial axes we can transform each pixel of the image into a complex number that corresponds to a point in the frequency domain.
- Result. From the previous point we have obtained a two-dimensional transformation of the frame in the frequency domain (there each point represents a certain frequency in the original frame). Such a frame will contain information about amplitude and phase of frequency components. This is what we plan to use for vibration extraction.

Inverse Fourier

- As input we receive a frame in the frequency domain.
- We transform a frequency domain frame into a spatial domain frame by inverse Fourier transform.
- If the frame was centered before DFT, then reverse centering (inversion) is performed.
- As a result, we obtain the original frame in the spatial domain.

3.2.5 Cross-power spectrum and normalization

The cross power spectrum is calculated by multiplying the Fourier transform of one frame by the complex conjugate Fourier transform of the other frame. [14] This is done in order to extract common frequencies between two frames without considering moments that are not common to these frames.To accurately determine the shift between frames it is necessary to normalize the cross power spectrum, this is done because normalization ensures that the value of the spectrum does not affect the phase information.

$$R(u,v) = \frac{G_a(u,v)G_b^*(u,v)}{|G_a(u,v)G_b^*(u,v)|} \tag{3}$$

where $G_a(u,v)$ and $G_b(u,v)$ are the $\mathcal{F}$ of the two images being compared, and $G_b^*(u,v)$ is the complex conjugate of $G_b(u,v)$. This process ensures that the result focuses solely on the phase difference between the images, which is used to calculate the shift.

3.2.6 Idea of implementation

A possible example of an implementation of what has been described above will be described below.

Libraries from python to use

- NumPy [11]: A fundamental Python computing package used for arrays and complex math operations.
- OpenCV [12]: A huge open-source library for computer vision, machine learning, and image processing.
- scipy.fft [13]: A library to perfom Fourier transforms

Implementation Steps

- Centering if needed.
- Applying ffft2 from scipy library to our frames to compute 2D DFT.
- Computing Normalized-Cross-Power Spectrum.
- Applying $\mathcal{F}^{-1}$ to find the peak wich corresponds to the shift. Using ifft2 from scipy.
- Remove centering (if used).

- Calculating shifts for each pair of frames.

3.2.7 Conclusion

In this part of the project, the planned idea of implementing Pattern matching was described. Time was given to the theoretical part, where it was explained how it will work, what mathematical methods should be used, what they mean and their advantages. Also the practical part was shown, which libraries from python can be used. This approach is currently the most efficient way to implement PCTM. In the future it is planned to implement this part and to understand if and how it can be optimized.

3.3 Vibration extaction and visualization in a custom app (by Timofey Sukhoparov):

3.3.1 Product

It is evident that a user-friendly application is necessary to visualize the entire work and examine vibrations in a more concrete manner. To achieve this, a desktop application was developed with simplicity and practicality in mind, enabling individuals less versed in theory to analyze mechanical systems and assess their efficiency. In basic terms, this marks the realization of the project for widespread and active use.

The main screen of the application features a section for uploading the required video, captured on a high-speed or neuromorphic camera. The video is uploaded through a button in the panel menu that opens the device's files, allowing the selection of the necessary video file. Thanks to the user-friendly interface, users can interact with the uploaded video, such as selecting a specific frame using the position slider or using the play, stop, and delete buttons.

Once the video is uploaded, users can initiate the analysis process. After the resulting pixel displacement will need to be converted in order to determine the relationship between pixel displacement and physical displacement. This conversion will require knowledge of the camera properties, such as focal length and sensor size, as well as any scaling factors used in the image acquisition.

Once the transformation has been performed, the data will be analyzed to obtain vibrations. By tracking the pixel displacement between consecutive frames of video, we can quantify the amount and direction of movement of each pixel, which will allow us to determine the frequency of the oscillating contour and the most oscillating region of the object. For the calculation we need two important quantities: time and number of oscillations, with their help we will calculate the frequency and period of oscillations. Further, the obtained data can be visualized with the help of graphs or diagrams to clearly see the changes.

After completing the analysis, an additional tab appears at the bottom of the application window, displaying the output for further actions on the analyzed mechanical system. This tab allows users to view a video with visually displayed micro-vibrations, facilitating an examination of their changes over time.

A convenient function is planned for analyzing specific parts of the mechanical system. Users can navigate to an additional section, select the part of the video of interest, and apply the algorithm. This feature offers a more specific view, allowing users to discard less interesting parts.

Importantly, all extracted results can be saved for use outside the application, aiding in further analysis.

3.3.2 Development

The development process involved choosing the most popular Python module for creating GUI applications, PyQt5, and the cross-platform interface development environment QtDesigner.

QtDesigner provided a clear interface design environment for visualizing the structure before writing code. A main window was created with a video frame and control buttons, along with additional sections to display results.

Code writing commenced, where all the main visual elements were defined. During the process, QMediaPlayer was integrated to enable control of video playback, including handling media player states, position changes, video duration, and error handling. Methods were implemented for user actions such as playing, pausing, or deleting a video.

Listing 1: Video Playback Controls in PyQt

```python
def play_video(self):
    if self.media_player.state() == QMediaPlayer.PlayingState:
        self.media_player.pause()
        self.btn_play.setText("Play")
    else:
        self.media_player.play()
        self.btn_play.setText("Pause")

def stop_video(self):
```

```
    self.media_player.stop()
    self.btn_play.setText("Play")

def delete_video(self):
    self.media_player.setMedia(QMediaContent())
    self.label_status.clear()
    self.position_slider.setVisible(False)

def setPosition(self, position):
    self.media_player.setPosition(position)

def mediaStateChanged(self, state):
    if self.media_player.state() == QMediaPlayer.PlayingState:
        self.btn_play.setIcon(self.style().standardIcon(
                QStyle.SP_MediaPause))
    else:
        self.btn_play.setIcon(self.style().standardIcon(
                QStyle.SP_MediaPlay))



def positionChanged(self, position):
    self.position_slider.setValue(position)

def durationChanged(self, duration):
    self.position_slider.setRange(0, duration)
```

To load video files, a QFileDialog dialogue box was implemented to allow users to select files on the device. Uploaded video files were processed by the QMediaPlayer.

Listing 2: File Open and Video Load in PyQt

```
def open_file_dialog(self):
    options = QFileDialog.Options()
    file_name, _ = QFileDialog.getOpenFileName(self, "Open␣Video␣File",
    "", "Video␣Files␣(*.mp4␣*.avi␣*.mkv␣*.mov)", options=options)
```

```
    if file_name:
        self.load_video(file_name)

def load_video(self, file_path):
    media_content = QMediaContent(QUrl.fromLocalFile(file_path))
    self.media_player.setMedia(media_content)
    self.label_status.setText(f"Loaded: {file_path}")
    self.position_slider.setVisible(True)
```

Error handling was provided for additional transparency and stable use of the application.

The result is a user-friendly Qt application, serving as a convenient tool for analyzing micro-vibrations and simplifying the path from theory to practice for interested individuals.

References

[1] Jean-Michel Morel Antoni Buades Bartomeu Coll. *Non-Local Means Denoising*. URL: https://www.ipol.im/pub/art/2011/bcm_nlm/.

[2] *Digital image correlation and tracking*. URL: https://en.wikipedia.org/wiki/Digital_image_correlation_and_tracking.

[3] *Fourier Transform*. URL: https://homepages.inf.ed.ac.uk/rbf/HIPR2/fourier.htm.

[4] Heung Soo Kim Haichao An Byeng D. Youn. *A methodology for sensor number and placement optimization for vibration-based damage detection of composite structures under model uncertainty*. URL: https://www.sciencedirect.com/science/article/pii/S0263822321013027.

[5] Aisha Javed; Hyeongill Lee; Byeongil Kim; Youkyung Han. *Vibration measurement of a rotating cylindrical structure using subpixel-based edge detection and edge tracking*. URL: https://www.sciencedirect.com/science/article/pii/S0888327021007858.

[6] Salvatore Salamone Korkut Kaynardag Chi Yang. *A rail defect detection system based on laser Doppler vibrometer measurements*. URL: https://www.sciencedirect.com/science/article/pii/S0963869523000737.

[7] Piotr KOWALSKI. *Measurements of Vibration Using a High-Speed Camera – Comparative Tests*. URL: https://vibsys.put.poznan.pl/_journal/2020-31-2/articles/vibsys_2020211.pdf.

[8] Matthias Baeßler Lukas Eichner Ronald Schneider. *Optimal vibration sensor placement for jacket support structures of offshore wind turbines based on value of information analysis*. URL: https://www.sciencedirect.com/science/article/pii/S0029801823017912.

[9] Qibing Yu; Aijun Yin; Quan Zhang; Shiyang Ma. *Optical flow tracking method for vibration identification of out-of-plane vision*. URL: https://www.researchgate.net/publication/318073386_Optical_flow_tracking_method_for_vibration_identification_of_out-of-plane_vision.

[10] HEMANTH YANAMADALA PRADEEP KUMAR MURALIDHARAN. *Comparative study of vision camera-based vibration analysis with the Laser vibrometer method*. URL: https://www.diva-portal.org/smash/get/diva2:1608598/FULLTEXT01.pdf.

[11] *NumPy*. URL: https:/numpy.org.

[12] *OpenCV is a open-source library of programming functions mainly for real-time computer vision*. URL: https://opencv.org/.

[13] *SciPy*. URL: https://docs.scipy.org/doc/scipy/tutorial/fft.html.

[14] Woong-jae Na; Kyung Ho Sun; Byeong Chan Jeon; Jaeyun Lee; Yun-ho Shin. *Event-based micro vibration measurement using phase correlation template matching with event filter*. URL: https://www.sciencedirect.com/science/article/pii/S0263224123004311.

[15] P. Stephan Heyns Stephen M. Talai a Dawood A. Desai a. *Infrared thermography applied to the prediction of structural vibration behaviour*. URL: https://www.sciencedirect.com/science/article/pii/S1110016819300419.

[16] D. Wach W. Aleite. *Supervision and Fault Diagnosis for the Safe Operation of Nuclear Power Plants*. URL: https://www.sciencedirect.com/science/article/pii/S1474667017511209.

[17] Chia-Hung Wu Yih-Nen Jeng. *Frequency Identification of Vibration Signals Using Video Camera Image Data*. URL: https://www.mdpi.com/1424-8220/12/10/13871.