Farneback Optical Flow Application for Automotive Onboard Vibration Monitoring

1st Leonardo Pezenatto da Silva
Pollux, Part of Accenture
Florianópolis, Brazil
leonardo.p.silva@accenture.com
leonardo.pezenatto@posgrad.ufsc.br

2st Luiz Fernando Martins Pastuch
Stellantis N.V.
Florianópolis, Brazil
luiz.pastuch@stellantis.com
luiz.fernando.pastuch@posgrad.ufsc.br

***Abstract*—This research explores a vision-based vibration monitoring algorithm that may be used to detect and measure vibrations on a vehicle with an onboard camera. To achieve this, we explored Optical Flow algorithms techniques such as Lucas-Kanade, Horn-Schunck and Farneback to analyze pixel motion between frames and relate it to vibrations. For testing and validation, CARLA simulator was used to acquire the front camera image and IMU data in scenarios where the vehicle drives over road bumps and potholes. The visual vibration estimation was compared with the z-axis acceleration data of an IMU and the results shows that it is possible to use optical flow methods to detect vibration on the vehicle.**

***Index Terms*—optical flow, vehicle vibration, camera, computer vision**

I. Introduction

Traditional vibration detection methods used to rely on sensors such as accelerometers, but these can be expensive and challenging for automakers to install and maintain in their customer vehicles. Using a camera with optical flow techniques can reduce reliance on physical sensors like Inertial Measurement Unit (IMU), as it enables the calculation of velocity and acceleration using only the camera. This simplifies the system, reducing costs and maintenance efforts. Therefore, this paper proposes a vision-based vibration estimator that uses an onboard camera to detect and measure vibrations on a vehicle. By leveraging existing visual data, given the increased rate of adoption of advanced driver assistance systems (ADAS) and autonomous driving (AD) systems, this method offers a cost-effective and scalable solution to vibration analysis in real-time.

Several techniques are available to estimate vibration in a sequence of frames, with Optical Flow algorithms being among the most prominent. These algorithms, including the Lucas-Kanade, Horn-Schunck, and Farneback methods, track the movement of pixels between consecutive frames to estimate the flow of motion within an image [1] [2]. These techniques provide tools for measuring scene dynamics, each with its advantages and disadvantages, enabling accurate motion estimation for applications such as object tracking and vibration analysis.

To test and validate the proposed vibration estimator, the CARLA simulator was used to provide an environment to simulate various road conditions, including bumps and potholes. Data was collected from both the vehicle's frontal camera and its IMU, allowing for a direct comparison between the visual estimator and traditional sensor-based methods.

II. Optical Flow Methods

Optical flow refers to the pattern of apparent motion of objects, surfaces, edges or entire image pixels between two frames, caused by the relative motion between an observer and the scene [3].

In the context of vibration detection for vehicles, optical flow algorithms are popular in this field due to their high accuracy and computational efficiency. They are employed to analyze pixel motion between successive frames captured by an onboard camera [1] [2] [3] and by measuring these pixel displacements, it is possible to infer vibrations and relate them to the movement of the vehicle on the road [4].

Our interest with this method was to capturing detailed pixel-level vibrations (e.g. small oscillations caused by road texture, high amplitudes caused by bumps and changes in the frequency-domain), and optical flow provides a dense representation of the entire field of view for that, that may be used in a wide range of road condition and predictive maintenance applications.

Between the existent algorithms, this section focuses on the three most used: Lucas-Kanade, Horn-Schunck, and Farneback.

A. Lucas-Kanade Method

The Lucas-Kanade method is a widely adopted algorithm to estimate the movement of key features between consecutive images of a scene. It operates under the assumption that the images are captured with a minimal time interval, ensuring little displacement of objects, which makes it effective for slow-moving subjects. The algorithm works best with textured objects that exhibit smooth variations in gray shades and does not utilize color information or search for exact pixel matches. It utilizes spatial and temporal gradients of image intensity, setting up equations that relate pixel intensity changes to motion. A single pixel often lacks sufficient information to determine motion reliably, but by considering surrounding pixels, the algorithm gathers more contextual data, which mitigates the effects of noise and improves gradient calculations. [5].

However, the algorithm has disadvantages, such as poor performance in regions with little texture or weak intensity gradients, which can lead to ill-conditioned matrices. Furthermore, its reliance on small time increments limits its ability to capture rapid movements accurately, and the presence of noise or outliers can affect the solution's reliability.

B. Horn-Schunck Method

Unlike the Lucas-Kanade method, which focuses on local motion estimation through small neighborhoods tracking prominent points like corners, Horn-Schunck provides a global approach that incorporates smoothness constraints across the entire image to enhance robustness against noise and improve estimation accuracy, but also under the assumption of brightness constancy and the image gradients computation [6].

In brief, both methods share similar approaches, but Horn-Schunck employs a global analysis, providing greater robustness to noise and excelling in fast and complex motions. However, this advantage comes with an increased computational complexity.

C. Farneback Method

The Farneback algorithm is a dense optical flow estimation technique that computes flow by modeling pixel neighborhoods using polynomial expansions. Unlike the Lucas-Kanade and Horn-Schunck methods, which rely on derivatives, Farneback generates dense flow by approximating local neighborhoods of an image as quadratic polynomials [7]. This enables the algorithm to estimate motion between frames in a more detailed and nuanced manner ensuring smooth and continuous flow estimation even in complex motions. Experiments results from [8] [9] [10] shows that Farneback algorithm outperforms in terms of execution time and capability.

For vibration detection, Farneback is especially advantageous when a dense and accurate flow field is needed to capture the minute displacements caused by road-induced vibrations. By tracking pixel movements across the entire image, the algorithm provides high-resolution flow data, which can be analyzed to detect even slight changes in the vehicle's position due to vibrations.

III. Translating Optical Flow to Vibration Data

Once the optical flow is computed, the pixel displacements must be translated into meaningful vibration data. This involves analyzing the motion patterns over time to identify recurring shifts in the image that correspond to the vehicle's oscillatory motion as it travels over uneven terrain or due to mechanical component issues (e.g., car suspension). The key steps in this process include:

1) Pixel Motion Conversion to Z-Axis Pixel Acceleration: Once the optical flow algorithm is applied and the displacement of the pixels is acquired, a conversion from displacement to acceleration must be performed. The algorithm below describes a straightforward method to achieve this.
2) Time Domain Analysis: By examining the pixel z-axis acceleration as it changes over time, focusing on amplitude and duration, lower accelerations amplitudes may indicate rough surfaces, while higher accelerations amplitudes could suggest larger bump, road undulations or potholes.
3) Frequency Domain Analysis: By examining the frequency domain using techniques like the Fast Fourier Transform (FFT). This helps identify dominant frequencies and their amplitudes revealing the primary frequencies of the vehicle's vibrations.
4) Correlation with IMU Data: To validate the resultant z-axis acceleration from the optical flow analysis, the output can be compared to traditional inertial measurement unit (IMU) data. The z-axis acceleration data from the IMU can serve as a ground truth for the vehicle's vertical vibrations, and a strong correlation with the optical flow data would confirm the effectiveness of the visual estimator.

IV. Visual Vehicle Vibration Estimator

A script was developed to process video data and estimate the vertical (z-axis) acceleration of a vehicle using optical flow analysis. The method involves defining a grid of points across the image to track motion vectors at regular intervals within the video frames. This grid provides a structured means of sampling the motion across the entire field of view.

For each frame, the script calculates the optical flow between consecutive frames using the Farneback algorithm. In this context, we focus on the vertical (y-axis) component of the motion, which corresponds to the vehicle's z-axis, or its upward and downward movement.

At each point in the grid, the vertical component of the optical flow vector is extracted and averaged across the entire frame. This averaged displacement provides a measure of the vertical motion between frames. To estimate the z-axis acceleration of the vehicle, the velocity is calculated by taking the first derivative of the vertical displacement, and the vertical acceleration is subsequently obtained by computing the second derivative of velocity (expressed in pixels/s^2).

To further analyze the vibration characteristics, a Fast Fourier Transform (FFT) is performed on the acceleration data. The FFT allows for the identification of the dominant frequency components within the signal, revealing the primary frequencies of the vehicle's vibrations. The most significant frequency is identified by locating the peak in the magnitude spectrum, which corresponds to the most significant vibration frequency.

Algorithm 1 describes the steps implemented and the Fig. 1 illustrates the grid formation used in the optical flow analysis for a single video frame.

V. Experiment and Results

To replicate the challenges faced in real-world driving scenarios and provide a validation platform for our algorithm, we introduced a test scenario using a customized map in

Algorithm 1 Optical Flow Vertical Acceleration Estimation

Input: Video frames $\mathbf{I} = [I_1, I_2, \ldots, I_n]$; Grid resolution
for $k = 1$ **to** $n - 1$ **do**
  Step 1: Optical Flow Calculation
  Compute the optical flow between frames I_k and I_{k+1} using the Farneback algorithm
  Extract the motion vectors $\mathbf{V}$ from the optical flow result
  for each point (i, j) in the grid **do**
    Step 2: Vertical Component Extraction
    Calculate the vertical component $v_{y_{i,j}}$ of the motion vector $\mathbf{V}$
  end for
  Step 3: Averaging Displacement
  Compute the average vertical displacement $\overline{D_k}$ across the grid
  Append $\overline{D_k}$ to the array $\mathbf{D}$
end for
Calculate the time interval: $\Delta t \leftarrow \frac{1}{\text{fps}}$
Step 4: Velocity Calculation
for $i = 1$ **to** $n - 1$ **do**
  $v_i \leftarrow \frac{d_{i+1} - d_i}{\Delta t}$ {Calculate velocity from displacement}
end for
Step 5: Acceleration Calculation
for $i = 1$ **to** $n - 2$ **do**
  $a_i \leftarrow \frac{v_{i+1} - v_i}{\Delta t}$ {Calculate acceleration from velocity}
  Append a_i to the array $\mathbf{A}_z$
end for
Step 6: Frequency Analysis
Perform Fast Fourier Transform (FFT) on the acceleration data $\mathbf{A}_z$
Identify the dominant frequency by locating the peak in the magnitude spectrum
Output: Vertical acceleration data $\mathbf{A}_z$; Dominant vibration frequency =0

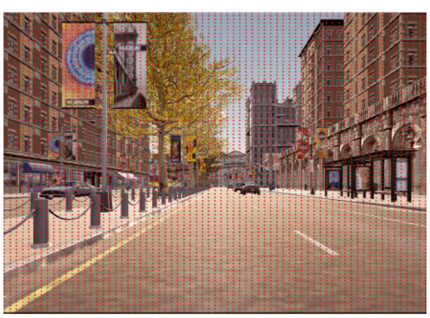

Fig. 1. Grid applied to a frame captured in CARLA.

CARLA simulator. CARLA is built upon on Unreal Engine 4 and it is open-source, allowing the user to clone its source repository, modify its assets and generate a new simulator package.

A map was created with features such as speed bumps and potholes (Fig. 2), that are intentionally placed within the virtual environment to induce significant vertical accelerations in the vehicle to acquire vibration data to evaluate the vehicle's response to uneven road surfaces.

The simulator uses Nvidia PhysX to model vehicle physics in the virtual environment. Specifically, the PhysX Vehicle SDX [11] depicts the vehicle as a system composed of chassis/wheel/tires represented as rigid bodies connected by a set of springs. This representation allows to accurately simulate vehicle dynamics for a wide range of automobile segments (sport cars, sedans, SUVs, wagons, trucks, etc.) using parameters like vehicle mass, center of mass, moment of inertia, wheel dimensions, suspension properties (natural frequency, damping, stiffness), engine, steering and braking characteristics to estimate forces and accelerations applied to the vehicle and tires.

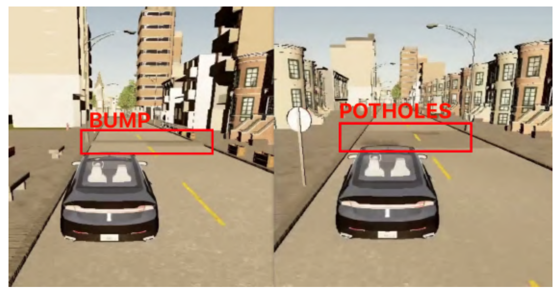

Fig. 2. Custom map in CARLA simulator.

The table below shows some parameters of the ego vehicle used in the experiment, namely the vehicle model, sensor poses and suspension properties. Positions are expressed relative to vehicle origin.

Simulation Parameters	
CARLA Version	0.9.15
Simulation Frequency	10 Hz
Vehicle Type	lincoln.mkz_2017
IMU Position	x=0, y=0, z=0 [m]
Camera Position	x=2.5, y=0, z=1.0 [m]
Mass	1920 kg
Center of mass	x=0.1, y=0, z=-0.2 [m]
Suspension Force Offset	0.0 N
Suspension Max Raise	7.5 cm
Suspension Max Drop	7.5 cm
Suspension Natural Frequency	9.5 Hz
Suspension Damping Ratio	1.0

TABLE I
TABLE OF SIMULATION PARAMETERS

The vibration data from the simulation was compared to data obtained from an IMU in the virtual vehicle and is shown in Figures 3 and 4. By analyzing the correlation between the simulated vibration data and the IMU data, we could assess the performance of the IMU under different driving conditions.

Figure 3, a time-domain analysis, shows that the algorithm is capable to capture the oscillation when the vehicle goes over the pothole and speed bump. The graph also shows a different curve and peaks behavior, that may be attributed to the fact that the IMU and the camera were not aligned

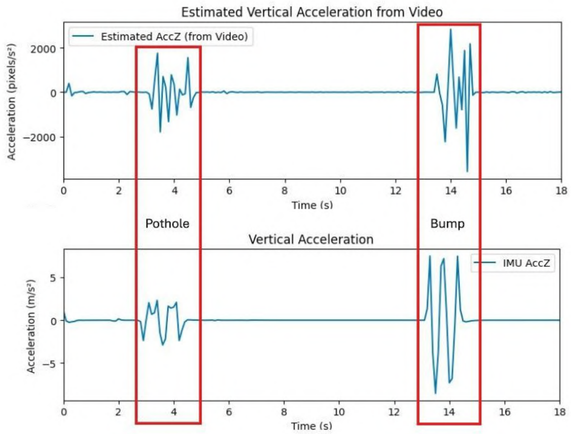

Fig. 3. Estimated Vertical Acceleration

in the same physical location (Table I). Moreover, the result shows a delay of approximately 0.2 seconds caused by the data acquisition frequency. Since images are captured at a frequency of 10 Hz, each image takes 0.1 seconds to acquire. The algorithm requires at least two images to generate the first estimate, creating an initial delay of 0.2 seconds. Additionally, the algorithm's processing time is added to this delay, further increasing the total response time. This highlights the need for further experiments to thoroughly examine not only the sensor placement but also other potential factors that may contribute to these differences and identify if refinements are necessary for improved results.

Using FFT on the complete data, we can identify the most significant frequency (or peak frequency) and calculate the mean value of all peaks obtained throughout the entire simulation. (Fig. 4).

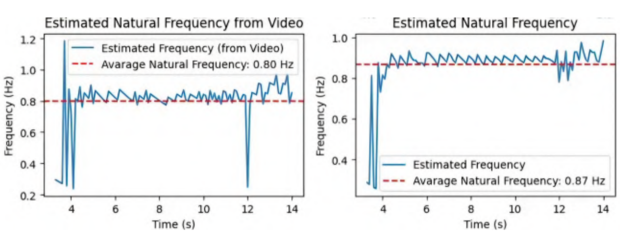

Fig. 4. Mean value of all peaks from video (left) and from IMU (right).

The natural frequency of a vehicle's vibration can be identified as the peak in the frequency spectrum obtained from the FFT analysis. This peak corresponds to the dominant frequency at which the system (in this case, the vehicle's chassis or suspension) tends to oscillate naturally, especially under steady-state conditions or without external forces. The results show that, in the frequency domain, the data from IMU and camera using the optical flow algorithm are also very similar, 0.87 Hz and 0.80 Hz, respectively.

VI. Conclusion and Outlook

In this paper, we identified the possible methods used to acquire motion from camera video data, implemented Farneback optical flow algorithm and executed it in a virtual environment for evaluation.

The obtained results have shown a strong correlation between the optical flow methods and IMU data, confirming that it is feasible to detect vehicle vibrations using visual inputs alone. This vision-based approach offers a promising basis for future vehicle monitoring systems, especially in autonomous driving applications where minimizing hardware complexity is critical.

For future work, the conversion of the optical flow data from pixels/s² to m/s² must be evaluated. Additional experiments are needed to thoroughly examine sensor placement influence and other potential factors that may contribute to the differences detected in the data. Furthermore, machine learning algorithms can be trained with the obtained data for road condition monitoring, anomaly detection, and predictive maintenance systems.

Acknowledgment

This work was partially supported by FUNDEP Rota 2030/Linha VI grant 29271.02.01/2022.01-00 and 29271.03.01/2023.04-00.

References

[1] Q. Yu, A. Yin, Q. Zhang, and S. Ma, "Optical flow tracking method for vibration identification of out-of-plane vision," Journal of Vibroengineering, Vol. 19, No. 4, pp. 2363–2374, Jun. 2017, https://doi.org/10.21595/jve.2017.17771

[2] Nie G-Y, Bodda SS, Sandhu HK, Han K, Gupta A. "Computer-Vision-Based Vibration Tracking Using a Digital Camera: A Sparse-Optical-Flow-Based Target Tracking Method". Sensors. 2022; 22(18):6869. https://doi.org/10.3390/s22186869

[3] Xiu C, Weng Y, Shi W. "Vision and Vibration Data Fusion-Based Structural Dynamic Displacement Measurement with Test Validation". Sensors. 2023; 23(9):4547. https://doi.org/10.3390/s23094547

[4] P. M. Harikrishnan and V. P. Gopi, "Vehicle Vibration Signal Processing for Road Surface Monitoring" in IEEE Sensors Journal, vol. 17, no. 16, pp. 5192-5197, 15 Aug.15, 2017, doi: 10.1109/JSEN.2017.2719865

[5] R. Rojas, "Lucas-Kanade Optical Flow Tutorial," Freie Universität Berlin, 2017. Available: http://www.inf.fu-berlin.de/inst/ag-ki/rojas_home/documents/tutorials/Lucas-Kanade2.pdf.

[6] J. T. J. Chao, P. W. F. Liu, and C. K. K. Lee, "Optical Flow Estimation: A

[7] Comprehensive Review," Sensors, vol. 22, no. 13, pp. 5017, 2022, doi: 10.3390/s22135017. G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," Computer Vision Laboratory, Linköping University, SE-581 83 Linköping, Sweden, [Online]. Available: http://www.isy.liu.se/cvl/

[8] N. A. Nemade and V. V. Gohokar, "Comparative performance analysis of optical flow algorithms for anomaly detection," [Online]. Available: https://elsevier-ssrn-document-store-prod.s3.amazonaws.com/19/07/14/ssrn_id3419775_code3382686.pdf.

[9] J. de Boer and M. Kalksma, "Choosing between optical flow algorithms for UAV position change measurement," [Online]. Available: https://www.cvds-nhlstenden.com/wp-content/uploads/SC2015_opticalflow.pdf.

[10] R. Radhakrishnan, D. Sharma, and V. Murthy, "A review on particle image velocimetry and optical flow methods in riverine environment," in Proceedings of the 1st International Conference on Recent Trends in Engineering Technology, Pune, India, Feb. 2017.

[11] NVIDIA, "Vehicles," NVIDIA Omniverse PhysX Documentation, v5.1.0. [Online]. Available: https://nvidia-omniverse.github.io/PhysX/physx/5.1.0/docs/Vehicles.html. [Accessed: 13-Oct-2024].