Smart Structures and Systems, *Vol. 35, No. 4 (2025) 195-208*
https://doi.org/10.12989/sss.2025.35.4.195

Non-stationary vision sensing for time-frequency analysis in vehicle-bridge interaction system

Jae Hun Lee [1], Sang Bin Lee [2], Jae Hun Lee [3] and Robin Eunju Kim*[2]

[1] *Department of Civil and Environmental Engineering, Hanyang University, Seoul, Republic of Korea*
[2] *Department of Architecture and Architectural Engineering, Seoul National University, 1, Gwanak-gu, Seoul 08826, Republic of Korea*
[3] *Infrastructure Bridge Engineering Division, Hyundai Engineering & Construction, 75, Yulgok-ro, Jongno-gu, Seoul 03058, Republic of Korea*

(*Received February 27, 2025, Revised April 24, 2025, Accepted April 28, 2025*)

Abstract. Global monitoring of structures is vital for assessing their structural integrity, especially with the impact of moving vehicles on railroad bridges. This necessitates simultaneous monitoring of both systems to understand interaction dynamics comprehensively. In vibration-based Structural Health Monitoring fields, demands for directly obtaining displacement responses increase, leading to non-contact sensing adoption. Computer Vision (CV)-based methods, using feature tracking techniques for displacement measurements, have become practical alternatives. The proposed approach utilizes Poor Feature Points, offering a global view and overcoming spatial resolution limitations. Addressing challenges related to camera ego-motion in large-scale monitoring, strategies for re-assigning regions of interest based on feature quality are introduced, and camera ego-motion is compensated by calibrating feature points. The You Only Look Once algorithm is used for vehicle wheel detection, localizing contact points to examine Vehicle-Bridge Interaction dynamics. A laboratory-scale experiment validation confirms the feasibility of global monitoring with vision sensors, especially in interpreting VBI dynamics.

Keywords: KLT (Kanade Lucas Tomasi) algorithm; MST (Modified S-Transform); poor-feature points; vehicle track bridge interaction dynamics; yolo

1. Introduction

In the field of structural health monitoring (SHM), measuring the dynamic responses of structures is essential for assessing their integrity. Many algorithms and sensors have been developed to implement vibration-based SHM on in-service bridges (Spencer *et al*. 2025). While obtaining dynamic displacement of a structure is preferred due to their direct correlation to the structural parameters (Martini *et al*. 2022), accelerations have been accepted widely in real practice so far: Unlike traditional displacement sensors, such as linear variable differential transformers, which require a stationary reference point, accelerometers could be attached directly on the structure allowing easy and cost-effective monitoring system (Spencer *et al*. 2025). Accordingly, several researchers have implemented full-scale monitoring systems on in-service bridges using a set of wired and wireless accelerometers to show the potential of using acceleration responses for estimating dynamic displacement (Park *et al*. 2013, Kim *et al*. 2014, Cho *et al*. 2015). Although multiple acceleration responses allow for determining structural parameters such as natural frequencies, a direct measurement of displacement is often necessary for examining the serviceability of a structure.

The demands on flexible displacement measurements have driven research in non-contact sensing systems. The non-contact sensors for SHM rely on laser sensors, optic sensors, radar sensors, etc. Among those, Computer Vision (CV)-based methods have shown potential as practical alternatives for dynamic displacement measurements. CV-based displacement measurements are typically enabled by tracking features: Primary feature tracking techniques are (i) Area-based template matching and (ii) Feature-based template matching (Feng and Feng 2018). The area-based approach utilizes the correlation of the pixels between the frames. Kim *et al*. (2013) adopted the normalized correlation coefficient method to estimate cable tension force from the dynamic displacement measurements. In addition, Dworakowski *et al*. (2016) implemented the zero-normalized cross-correlation coefficient, which is known to be robust in noisy environments, to measure in-plane displacement for damage detection. More literature on this technique can be found in Dong and Catbas (2021). On the other hand, the Feature-based approach detects extracted image features that remain locally invariant and thereby known to be more robust in scale changes or rotations, etc. (Feng and Feng 2018). Outlier detection algorithms are typically followed to remove false matching. Yoon *et al*. (2016) used the Kanade-Lucas-Tomasi (KLT) algorithm and the Maximum likelihood sample consensus to perform system identification of a laboratory-scale six-story shear building. Also, this approach is widely used in outdoor applications as well (Jana and Nagarajaiah 2021, Aliansyah *et al*. 2021, Tan *et al*. 2023), due to its advantage of robustness (Lucas and Kanade 1981, Tomasi and Kanade 1991, Shi 1994, Kalal *et al*. 2010).

*Corresponding author, Ph.D., Assistant Professor,
E-mail: robinekim@snu.ac.kr

ISSN: 1738-1584 (Print), 1738-1991 (Online)

Owing to precise tracking accuracies with a stationary camera, techniques that compensate the camera movement, the camera ego-motion, due to environmental effects (Hwangbo *et al*. 2009, Jana and Nagarajaiah 2021) or due to vision systems mounted on Unmanned Aerial Vehicle platforms (Yoon *et al*. 2018, Ribeiro *et al*. 2021, Özcan and Özcan 2021) have been developed. In such circumstances, the use of the conventional KLT algorithm becomes improper, because the fundamental assumption of the algorithm is small temporal change across the frame, which is prone to be violated by the large and fast rotation and movement arising from camera ego-motion. To overcome this feature, several camera motion subtraction techniques are proposed. Two widely accepted approaches in SHM include (i) using inertial measurements from IMU sensors (Ribeiro *et al*. 2021) and (ii) using stationary background objects (Jana and Nagarajaiah 2021, Yoon *et al*. 2018). The first approach fuses inertial sensors and vision sensors to compensate for the camera movement (Hwangbo *et al*. 2009). Ribeiro *et al*. (2021) used a set of accelerometers and gyroscopes to numerically eliminate the displacements and the rotations of the UAV and validated indoor and outdoor usages. The second approach compensates the camera motion assuming that some features of a structure (e.g., boundaries), or the background remain stationary as the target structure and the vision sensor displace. Using artificial markers, Hoskere *et al*. (2019) demonstrated that the global responses of a pedestrian bridge under ambient vibration can be achieved. However, as asserted by Dong and Catbas (2021), deciding an appropriate number of feature points for tracking is still an open challenge making long-distance monitoring difficult. Thus, the aforementioned full-scale applications mostly focused on examining the displacement of a specific range of the structure under ambient vibration.

Nonetheless, examining the impact of moving vehicles on the bridge may provide meaningful indicators of the serviceability of the structure (Kim *et al*. 2015, Tian and Zhang 2020). The interaction between the vehicle and the bridge results in the evolution of dynamic characteristics of the structure and the vehicle (Kim *et al*. 2016, Cantero *et al*. 2016, Zhan *et al*. 2020, Lee *et al*. 2022). So far, only a limited number of studies have conducted vision-based approaches for examining the influence of moving forces. Jian *et al*. (2019) used a stationary webcam and applied YOLO V3 to recognize the positions of multiple vehicles. Zhao *et al*. (2021) proposed a deep learning-based framework to perform real-position-based calibration of the influence line. In their work, while the non-constant velocity of the vehicle was well compensated by a stationary dual camera, the response of the bridge was collected from vision sensors. Using artificial markers on a bridge and a stationary camera, Aliansyah *et al*. (2021) measured displacement under a freight train. A separate vision sensor was implemented to count the vehicle and correlate it with the bridge responses from shared timestamps. Then, Martini *et al*. (2022) conducted indoor testing with artificial targets and synchronized three stationary cameras; two cameras captured mid-pan and three-quarter span displacement, and the third camera covered the entire bridge span to track the position of the vehicle. Estimated displacement influence lines are used for bridge model updating. To fully examine the coupled behavior of moving vehicles, simultaneously measuring bridge dynamics in a global manner and vehicle location is essential. However, due to trade-offs between spatial resolution and field of view (Dong and Catbas 2021, Zhu *et al*. 2021), limited studies have developed non-stationary camera systems for VBI monitoring applications.

Thus, in this study, we propose a framework that simultaneously detects dynamic responses of a bridge at a global level and vehicle wheel locations using a non-stationary camera. The proposed approach aimed to utilize Poor Feature Points (PFPs), where the features are measured at the single-pixel level, such that the distance from a vision sensor to the bridge can be further to include a global view. Strategies for re-assigning Region of Interest (ROI) based on the quality of the tracking features are proposed. Then the camera-ego motion is compensated by calibrating the feature points. Regarding the vehicle wheel location, the wheels are detected using the You Only Look Once (YOLO) algorithm. The wheels are detected instead of the vehicle body, to better localize the contact points for examining the VBI dynamics. Detected bridge dynamics and vehicle locations are then used to perform the time-frequency analysis (TFA). The proposed algorithms are validated from a laboratory-scale experiment; the results obtained from computer vision (CV) algorithms are compared with data from contact-based sensors. The results demonstrated the effectiveness of the proposed method in enabling global monitoring of civil infrastructure with vision sensors, particularly in the context of interpreting VBI dynamics.

2. Problem description: VBI monitoring with non-stationary vision sensors

Vision technologies have shown successful applications in monitoring dynamic responses of the structures. However, to utilize a non-stationary vision sensor for monitoring VBI systems, the existing object detection and tracking algorithms may face some challenges, which can be summarized as: 1) The limitations of the KLT algorithm for tracking bridge response globally and 2) vehicle position tracking difficulties. In the subsequent subsection, each challenge is discussed in detail.

2.1 Applying conventional KLT algorithms for VBI monitoring <Single feature tracking>

Vision sensors employ camera and image analysis techniques in various applications. Typically, the KLT algorithm is employed in tracking the features of an object. The KLT algorithm has various strengths in SHM applications for image-based dynamic response detection and structure displacement measurement. The algorithm detects sparse feature points and uses those to estimate the object's motion. This method offers advantages for mobile cameras, including reduced reliance on high-resolution images, robustness against lighting changes, and fast computational speed for real-time monitoring of large

structures (Soh *et al*. 2014, Feng and Feng 2018).

However, few applications have been made for measuring the global response of a structure with traditional KLT. To obtain the global response, the distance from the vision sensor to the structure increases, to obtain full scale of the structure within the Field of View (FoV). Long distance may result in the occurrence of Poor Feature Points (PFPs) where feature objects are measured at the level of a single pixel, i.e., single feature for each tracking region. However, the performance of the KLT algorithm for PFP tracking is affected by the size of the window. Shi (1994) reported that a smaller window size may detect more detailed motion, but it is more sensitive to noise and outliers. Conversely, a larger window size can estimate larger motions, but it results in an increased smoothing effect, which can cause motion to be smoother and detailed information to be lost. As a result, conventional KLT algorithms may not be sufficient for detecting multiple PFPs and capturing detailed displacements at a distance.

In addition, in the case of a nonstationary vision sensor, the distance between objects and the camera continuously changes, causing variations in object size. This can lead to ambiguity in captured images and blurred boundaries between the background and objects. Camera ego-motion can also decrease the accuracy of the tracking results and reduce consistency between frames, making the tracking results unstable. Moreover, detection size and rotation transformations are sensitive to the accuracy of absolute non-stationary vision sensors. Therefore, careful selection of tracking window size and detecting object rotation transformations is needed for utilizing KLT algorithms for VBI monitoring.

2.2 Vehicle positioning tracking application

In the case of a railroad bridge, where the vehicle mass is comparable to that of the bridge, the natural frequencies of the bridge vary as the vehicle moves along the bridge. Yang *et al*. (2013) solved the single degree of freedom (SDOF) mass interacting with the SDOF bridge problem to show analytical solutions for both vehicle and bridge. Especially, the initial frequency of vehicle (ω_{v0}) is larger than that of the bridge (ω_{b0}) time-varying frequency variation of vehicle (ω_v) and bridge (ω_b) become

$$\omega_v(vt) = \frac{\omega_{v0}^2}{2} + \frac{\omega_{b0}^2}{2} + \frac{m_v\omega_{v0}^2}{m_b}\Phi^2(vt) + \sqrt{\left(\frac{\omega_{v0}^2}{2} + \frac{\omega_{b0}^2}{2} + \frac{m_s\omega_{v0}^2}{m_b}\Phi^2(vt)\right)^2 - \omega_{v0}^2\omega_{b0}^2} \quad (1)$$

$$\omega_b(vt) = \frac{\omega_{v0}^2}{2} + \frac{\omega_{b0}^2}{2} + \frac{m_v\omega_{v0}^2}{m_b}\Phi^2(vt) - \sqrt{\left(\frac{\omega_{v0}^2}{2} + \frac{\omega_{b0}^2}{2} + \frac{m_s\omega_{v0}^2}{m_b}\Phi^2(vt)\right)^2 - \omega_{v0}^2\omega_{b0}^2} \quad (2)$$

where m_b and m_v are the bridge and vehicle's mass, respectively, Φ is the mode shape used for describing the bridge, and v is the speed of the vehicle. As can be seen, frequency variations are not only controlled by the vehicle and bridge initial frequencies, but also by the location of the vehicle (i.e., $L_v = vt$). Thus, to observe frequency variation characteristics in the VBI system, precise tracking of the vehicle's position is of necessity along with the bridge dynamic response measurements.

However, unlike bridges, where the selected feature remains the same once the target is chosen, the features to track for vehicles are rather unpredictable. As a new vehicle enters the bridge, new feature points need to be identified and detected depending on the vehicle type. Moreover, vehicle operation on the road is characterized by a dynamic and complex environment. Vehicles may exhibit sudden accelerations, decelerations, and other maneuvers. These movements can impact the estimation of the vehicle's speed and position, requiring real-time processing and analysis capabilities from the vision sensor. Thus, the tracking vehicle strategies should be set differently from that for the bridge, while simultaneous tracking must be offered to ensure integrated monitoring of the two systems.

3. Methodology

In this section, an algorithm that estimates the time-varying frequency of the VBI system is developed using images captured by a non-stationary single-vision sensor. The proposed strategy is summarized in Fig. 1, which involves the following processes: 1) Bridge response tracking from PFPs, 2) Vehicle detection and localization, and 3) Time-varying frequency estimation using the Reassigned Modified S-Transform (RMST). In the subsequent sections, the description of the selected hardware is discussed first, followed by each process of the proposed framework.

3.1 Hardware selection

In this study, the iPhone 13 Pro is chosen as a vision sensor, and its specifications are summarized in Fig. 2. The selected device is capable of capturing images with a maximum resolution of 3,840\times2,160 pixels (4K) at a sampling rate of 60 Hz. The aperture value can be adjusted between f/1.5 and f/2.8, enabling image capture in various lighting conditions. The smart High Dynamic Range (HDR) feature enhances object recognition and finding feature points, and the Auto Image Stabilization feature corrects images captured while in motion.

3.2 Bridge response tracking from PFPs

This subsection develops tracking strategies to estimate of dynamic responses of the bridge in a global manner. The conventional KLT algorithm is updated to successfully track PFPs and then camera ego-motion is compensated. In this work, the tracking process composed of 5 sequential steps is developed:

Step 1. Set ROI and Detect Feature Points: To estimate bridge-vehicle interaction, the Range of Interest (ROI) is determined for both the feature points on the bridge surface and the fixed reference points on the bridge

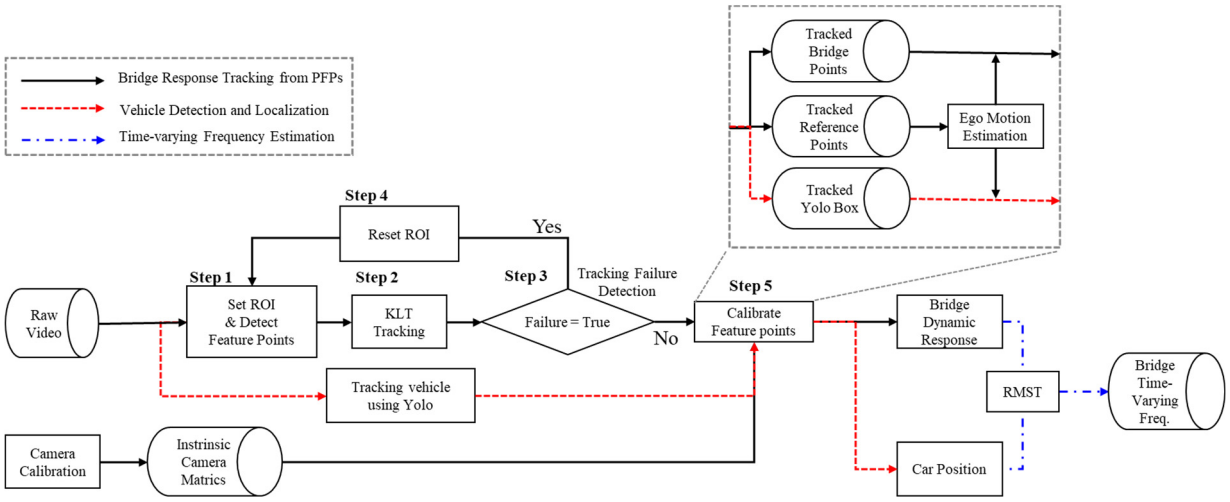


Fig. 1 Framework for time-varying frequency estimation using an ego-motion vision sensor

Resolution: 3,840 × 2,160 pixels (4K)
Sampling rate: 24, 25, 30 or 60 Hz
Aperture: (Telephoto)f/2.8, (Wide)f/1.5, (Ultra Wide)f/1.8
Other Features:
Smart High Dynamic Range (HDR)
Auto Image stabilization
Night mode portraits enabled by LiDAR Scanner

Fig. 2 iPhone 13 Pro technical specification

supports in the initial frame. In this study, the corner detection method proposed by Harris and Stephens (1988) was first employed, as given by

$$\mathbb{x} = (\boldsymbol{x}, \boldsymbol{y}) \tag{3}$$

$$\boldsymbol{E}_{\mathbb{x}} = (d\mathbb{x})\boldsymbol{M}(d\mathbb{x})^{\boldsymbol{T}} \tag{4}$$

where $\mathbb{x}$ is the 2-dimensional point of features in the image by $\boldsymbol{x}$ and $\boldsymbol{y}$. The weighted sum of the squared difference, $\boldsymbol{E}_{\mathbb{x}}$, represents the corner response function, and is affected by a small translational shift, $d\mathbb{x}$. The point at which this value reaches its maximum corresponds to the final corner point, $\mathbb{x}$. In this context, the value $\boldsymbol{M}$ denotes the intensity gradient matrix used for measuring corner features.

In this work, to represent the PFPs, the tracking of surrounding feature points is minimized by specifying the size of the ROI to 5\times5 (Shi 1994). This assumption indicates that the PFPs on the full-scale structure do not exceed 2 pixels. In addition, using minimized ROI can reduce the computational cost and achieve robustness by focusing only on the calculations for individual points without performing image-wide corrections during the subsequent camera calibration and ego-motion estimation steps. Therefore, the overall procedure can become less sensitive to variations in lighting, size, and shape of the feature points caused by the camera's ego motion.

Step 2. KLT Tracking: This step employs the conventional KLT algorithm to PFPs set from Step 1. The authors in Shi (1994) reported the correlation between image motion and window size for tracking suitable feature points. The authors explained the relationship between these two factors in the context of tracking feature points, which can be summarized as

$$\int_{\boldsymbol{W_i}} (\boldsymbol{h} - \boldsymbol{g} \cdot \boldsymbol{d})^2 \boldsymbol{w}\, \boldsymbol{dA} = 0 \tag{5}$$

The $\boldsymbol{W_i}$ indicates the entire window on the image coordinate and $\boldsymbol{h}, \boldsymbol{g}, \boldsymbol{d}$ are the disparity vector, the image gradient, and the displacement vector, respectively. Also, $\boldsymbol{A}$ and $\boldsymbol{w}$ represents the linear spatial transformation and the weighting function, respectively. A valid tracking condition is when integration of the equation can determine vector $\boldsymbol{d}$ that minimizes $\boldsymbol{\omega}$. In this case, $\boldsymbol{d}$ must be smaller than the window size, and failure to meet this criterion in tracking is caused by one of the following: 1) Displacement vectors exceeding the window size correspond to areas that the

algorithm does not consider. As a result, the motion in those areas is disregarded, leading to information loss. 2) Movements beyond the window size can be mistakenly estimated as displacements by the algorithm. This can result in tracking errors, where the obtained results differ from the actual motion of the feature point. Therefore, the failure incident must be examined as will be followed in the subsequent step.

Step 3. Tracking Failure Detection This step involves the examination of tracked PFPs to determine the cases when the conventional KLT algorithm fails. Herein, vision.PointTracker in Matlab® (MATLAB R2023b) is employed. As one of the inputs to the function, the number of pyramid levels is set as 1 to be less sensitive to noise in dealing with PFPs. Then, the function is set to output Error score and validity from Forward-backward error. First, the Error score is calculated using the Sum of Squared Differences (SSD) as (Lucas and Kanade 1981)

$$\boldsymbol{Error\ score} = \mathbf{1} - \boldsymbol{SSD}(\boldsymbol{p}) \,/\, \sum_{\mathbb{x}\in \boldsymbol{W_i}} \mathbf{255^2} \tag{6}$$

where

$$\boldsymbol{SSD}(\boldsymbol{p}) = \sum_{\mathbb{x}\in \boldsymbol{W_i}} \left[\boldsymbol{I}\left(\boldsymbol{W_i}(\mathbb{x};\boldsymbol{p})\right) - \boldsymbol{J}(\mathbb{x})\right]^2 \tag{7}$$

Eq. (7) checks the SSD of intensities between two consecutive frames ($\boldsymbol{I}$ and $\boldsymbol{J}$) within the window $\boldsymbol{W_i}$. Then $\boldsymbol{Error\ score}$ in Eq. (6) is obtained by normalizing SSD by the squared sum of the maximum intensity value, which is 255. Subsequently, the normalized value is subtracted from 1 to obtain the $\boldsymbol{Error\ score}$ for comparing frames. Thus, when $\boldsymbol{Error\ score}$ is near 1, tracked features are similar to that from previous frame. A threshold, $\boldsymbol{TH_{SSD}}$, can be set to indicate failure.

Then, Maximum Bidirectional Error (MBE) is calculated from the Forward-Backward (FB) error detection, proposed by Kalal *et al*. (2010). In this method, failures in tracking are detected from the Euclidean distance between the forward and backward feature point trajectories $\left(\boldsymbol{T_f^k}, \boldsymbol{T_b^k}\right)$

$$\boldsymbol{FB}\left(\boldsymbol{T_f^k}|\boldsymbol{S}\right) = \boldsymbol{distance}\left(\boldsymbol{T_f^k}, \boldsymbol{T_b^k}\right) = ||\mathbb{x}_t - \hat{\mathbb{x}}_t|| \tag{8}$$

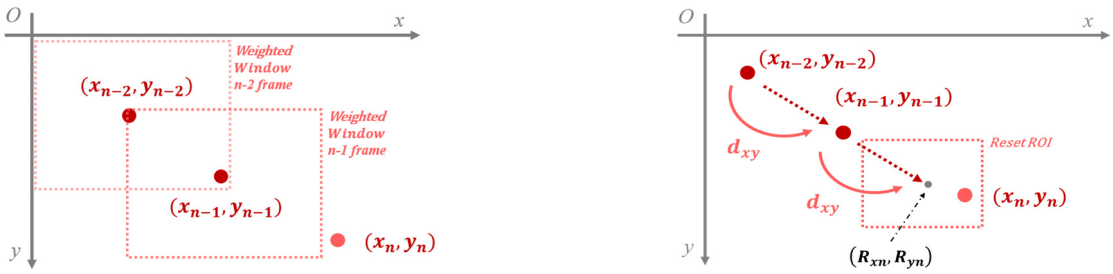
The variables $\mathbb{x}_t$ and $\hat{\mathbb{x}}_t$ represent the forward and backward feature points at time *t*, respectively. One can set a threshold, $\boldsymbol{TH_{MBE}}$, to report an error exceeding specified pixel value as Boolean. In the proposed method, any condition violating the threshold limits is determined as a tracking failure.

Step 4. Reset ROI: In the case a failure is detected from Step 3, ROI is reset. To illustrate the problem, Fig. 3(a) shows the case when the single feature at n^{th} frame, $\mathbb{x}_n$, exceeds $(n-1)^{th}$ frame. In this case, as shown in Fig. 3(b), the position of the new ROI center points $[R_{xn}\ R_{yn}]$ can be determined as follows

$$\mathbb{x}_{n-1} = \mathbb{x}_{n-2} + \boldsymbol{d_{n-2}} \tag{9}$$

$$[\boldsymbol{R_{xn}}\ \boldsymbol{R_{yn}}] = \mathbb{x}_{n-1} + \boldsymbol{d_{n-2}} \tag{10}$$

Here, $\boldsymbol{d_{n-2}}$ is displacement vector between feature points from $(n-2)^{th}$ frame, $\mathbb{x}_{n-2}$ to $(n-1)^{th}$ frame,



(a) (b)

Fig. 3 (a) the feature point moved outside the weighted window case; (b) ROI reset and feature point re-tracking process

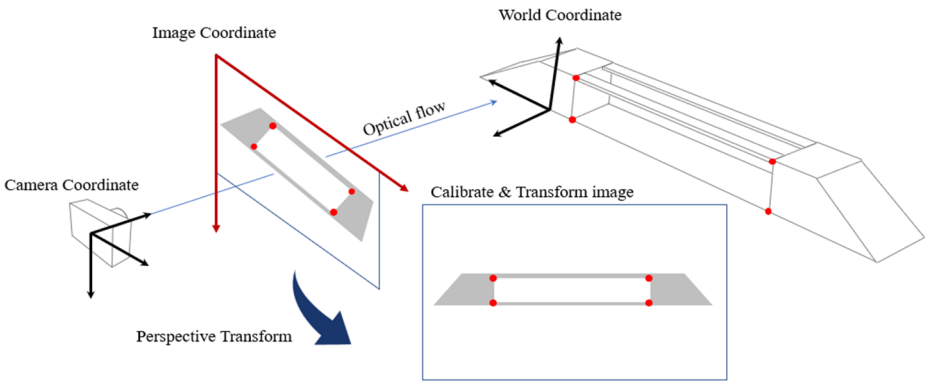


Fig. 4 The process of calibrating and transforming images into world coordinates

$\mathbb{x}_{n-1}$. The $[R_{xn}\ R_{yn}]$ is determined by assuming the gradient is the same distance and direction, $\boldsymbol{d_{n-2}}$, frame $\mathbb{x}_{n-1}$. Then, the new ROI can be set as 5×5 pixels having center point as $[R_{xn}\ R_{yn}]$. Harris-Stephens algorithm (Harris and Stephens 1988) is applied to detect corners within the new ROI and set corner points as new features. The proposed approach aims to prevent the occurrence of feature points falling outside the weighted window due to the sudden ego motion of the vision sensor.

Step 5. Calibrate Feature Points: Using the detected feature points, camera ego-motion is compensated to obtain the absolute dynamic response of the bridge. In this step, along with camera ego-motion compensation, lens distortion is simultaneously corrected using camera intrinsic parameters.

In this step, the lens distortion of an image is corrected using the camera intrinsic matrix (K)

$$K = \begin{bmatrix} F_x & 0 & 0 \\ g & F_y & 0 \\ o_x & o_y & 1 \end{bmatrix} \tag{11}$$

where, $[o_x\ o_y]$ represents the optical center, and $[F_x\ F_y]$ denotes the focal length with scale factors included, and g represents the skewness coefficient. Because the coefficients are unique for each lens, and thus can be identified from preliminary tests using images of a known geometry at multiple viewpoints.

In the current n^{th} frame, 2D image coordinates $[x_n\ y_n]$ are converted into 3D world coordinates $[X_n\ Y_n\ Z_n]$ in the motion of a 6 DOF ego-motion camera

$$s_n[x_n\ y_n\ 1] = [X_n\ Y_n\ Z_n\ 1]\begin{bmatrix} R_n \\ r_n \end{bmatrix} K \tag{12}$$

$$D_n = \begin{bmatrix} R_n \\ r_n \end{bmatrix} K \tag{13}$$

Here, the n^{th} frame of the captured image is represented by a 3×3 rotation matrix (R_n), an arbitrary scale factor (s_n) and a 1×3 translation vector (r_n). Once the distortion is corrected and converted into 3D world coordinate coordinates, the relative motion of structure to the camera ego-motion is compensated. Here, each converted feature point in a frame is geometrically fit to fixed points. For example, knowing that the features at boundary conditions are relatively fixed, those can be used as fixed points, where those features are always transformed into a fixed coordinate. Other feature points can then be mapped into the coordinates of the fixed points (Goshtasby 1988, Yoon *et al*. 2016). This process is iterated for every frame to obtain absolute displacement of the features.

3.3 Vehicle detection and localization

Along with bridge tracking, the vehicle is detected and localized separately within the proposed framework. Herein, You Only Look Once (YOLO) algorithm, YOLO v5 (Jocher *et al*. 2021), is utilized to execute wheel recognition and tracking. Tracking wheels is preferred over tracking vehicle bodies, because wheels are relatively simpler than vehicles in terms of number and types of feature points: less number of images can be used in training and the model can be applied to various types of vehicles regardless of their shape of the car body. As outputs of the YOLO model, predicted objects are noted with coordinates of bounding boxes. However, due to the orientation of the camera, not perpendicular to the vehicle, YOLO provides, the error between the center of the bounding box and the actual center in world coordinates may exist (Li and Yoon 2023). To accurately localize the vehicle position, a method to correct the center of the detected wheel by YOLO is proposed in this process.

To illustrate the problem, Fig. 5(a) shows an example bounding box (B) result from YOLO, when the wheel is detected with a camera angle. As an output from YOLOv5, the estimated center of the detected bounding box is exported with $[C_{B_{x,0}}\ \ C_{B_{y,0}}]$, and the width and height of the box is B_b and H_b, respectively. Utilizing these outputs, the coordinates of the bounding boxes can be expressed as

$$B = \begin{bmatrix} C_{B_{x,0}} + \frac{B_b}{2}, C_{B_{y,0}} - \frac{H_b}{2} \\ C_{B_{x,0}} - \frac{B_b}{2}, C_{B_{y,0}} - \frac{H_b}{2} \\ C_{B_{x,0}} - \frac{B_b}{2}, C_{B_{y,0}} + \frac{H_b}{2} \\ C_{B_{x,0}} + \frac{B_b}{2}, C_{B_{y,0}} + \frac{H_b}{2} \end{bmatrix} = \begin{bmatrix} a_B \\ b_B \\ c_B \\ d_B \end{bmatrix} \tag{14}$$

Applying Camera distortion and camera-ego motion compensation strategies, step 5 in Section 3.2, on B, one can obtain absolute displacement B' in World coordinates as in Fig. 5(b). The ellipse-shaped wheel from Fig. 5(a) is adjusted to show a circle-shaped wheel in Fig. 5(b), while the shape of the bounding box transformed from rectangular to a tangential quadrilateral. Now the bounding box has been transformed with coordinates of four vertices being $[a'_B\ \ b'_B\ \ c'_B\ \ d'_B]^{\mathrm{T}}$. In this work, the center points $[C_{W_x}\ \ C_{W_y}]$ in the transformed coordinates are used to localize the centers of each wheel.

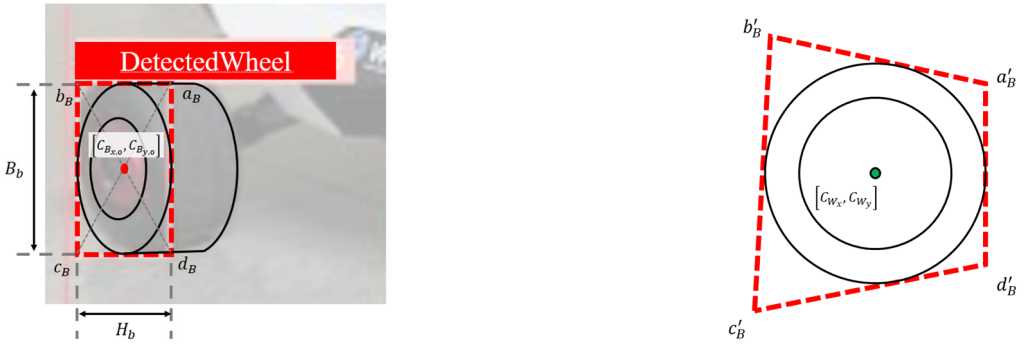
3.4 Time-varying frequency estimation using the MST

Time-frequency analysis (TFA) is employed in this section to the tracked bridge response and localized vehicle to interpret the temporal frequency variation. Herein, the Modified S-Transform (MST) method is applied, with its efficiency for assessing the frequency variations of the non-stationary signals. MST is based on the S-transform (ST) introduced by Stockwell (1996), which is defined as

$$ST = \int_{-\infty}^{+\infty} u(t) W_g(t-\tau) e^{-i2\pi f t} dt \tag{15}$$

$$W_g(t-\tau) = \frac{|f|}{\sqrt{2\pi}} e^{-\frac{(t-\tau)^2 f^2}{2}} \tag{16}$$

where $u(t)$ and W_g denote time-variant input signals and Gaussian window function, respectively. ST can be also represented in continuous wavelet transform (CWT) by



(a) (b)

Fig. 5 Detected bounding box correction for wheel center localization (a) illustration of bounding box; (b) corrected bounding box geometry

defining the basis wavelet as the combination of the Gaussian window function and the sinusoidal window function. Here, the Gaussian window function includes a time variable, while the sinusoidal function does not; thus, the Gaussian window function is affected by the variable time step, making the total window function adaptable for both frequency and time axis. Such a factor makes the ST more optimized for local frequency variation detections compared to the other TFA methods. Detailed derivations can be found in Zhang *et al*. (2021). Also, mathematically ST is different from CWT because ST cannot satisfy the admissibility condition represented. Furthermore, by manipulating the Gaussian window function, ST can be more optimized in time and frequency axis in spectrograms; which is defined as MST

$$MST = \int_{-\infty}^{+\infty} u(t) W_{g_{modified}}(t-\tau) e^{-(\sqrt{-1})2\pi f t} dt \quad (17)$$

Modified Gaussian window function $W_{g_{modified}}$ is achieved by manipulating the standard deviation scale function $\sigma(f)$ inside, which is defined as

$$\sigma(f) = \frac{1}{f} \quad (18)$$

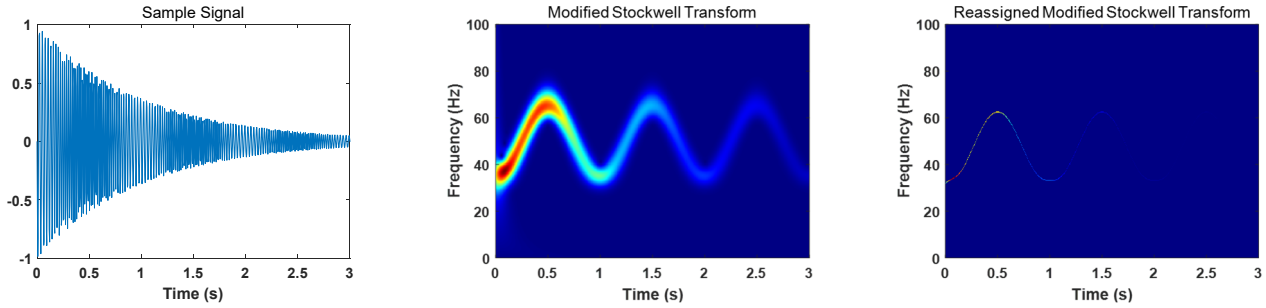
Various approaches for obtaining standard deviation scale functions are suggested (Yuan *et al*. 2022). In this paper, a function suggested by Yuan *et al*. (2022) is adopted as shown in Eq. (19).

$$\sigma(f) = \frac{1}{\alpha(f+\beta)^{\gamma}} \quad (19)$$

where α, β and γ each denote independent parameters that can be achieved through the optimization process suggested by Yuan *et al*. (2022). Further, an advanced version of MST, reassigned MST (RMST), was proposed by Zhang *et al*. (2021) using the reassignment technique for the obtained MST results. Features of RMST are 1) decomposing the MST result to frequency and time axis components, and 2) reallocating to each center of gravity. Through this process, RMST achieves a sharp and concentrated spectrogram. The following equation by Zhang *et al*. (2021) describes a definition of RMST

$$RMST(t', f') = \iint |MST|^2 \delta(t'-\hat{t}) \delta(f'-\hat{f}) dt df \quad (20)$$

where t' and f' are the default spectrogram coordinates; where $\hat{t}$ and $\hat{f}$ represent the reallocated coordinates over the time and frequency axis. Thus, due to the Dirac delta function (δ), Eq. (20) selectively retains only the MST values positioned at the center of gravity for each axis, while discarding the values below or over it to zero. This process effectively serves as a frequency extraction tool. A simple signal in Eq. (21) is used to illustrate the comparison between MST and RMST as shown in Fig. 6. As can be seen, a thinner spectrogram is achieved by RMST and MST. Thus, in this paper, RMST is adopted to decompose frequency and time components from tracked dynamic



(a) (b) (c)

Fig. 6 Time-frequency resolution comparison between MST and RMST; (a) Sample signal; (b) MST result; (c) RMST result

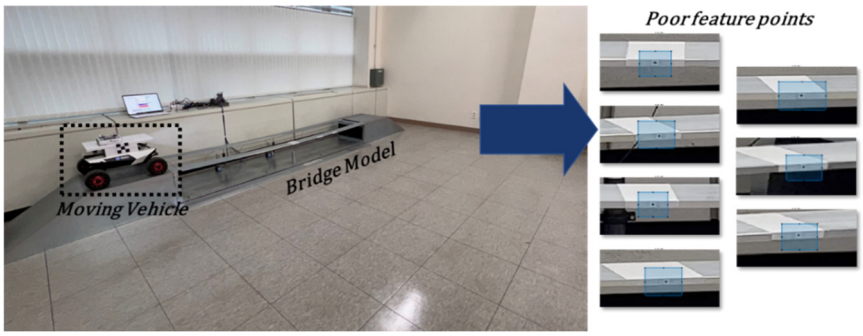
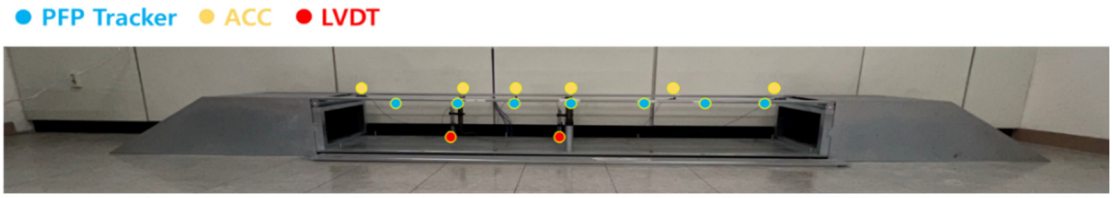


Fig. 7 Experimental setup



Location (cm)	5	37.5	75	112.5	150	187.5	225	262.5	295
Vision		●	●	●	●	●	●	●	
ACC	●		●	●	●		●		●
LVDT			●		●				

Fig. 8 Installed sensors and locations

displacement response of bridge from non-stationary vision sensors.

$$u(t)_{sample} = e^{-t} sin\,(5\pi(100t + sin(2\pi t)) \qquad (21)$$

4. Experimental validation

This section performs laboratory-scale indoor experiments to validate the proposed framework. To validate the performance of the proposed non-stationary vision sensors-based tracking algorithm, a set of wired sensors are installed on the bridge as reference signals.

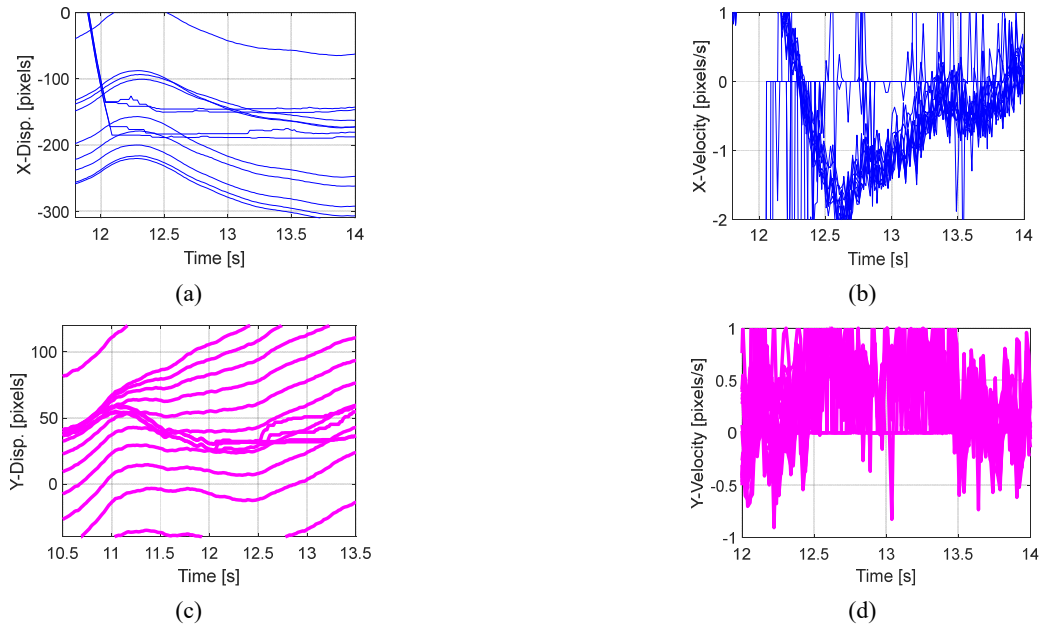


Fig. 9 Conventional KLT tracking error due to PFPs; (a) X-directional displacement; (b) X-directional velocity; (c) Y-directional displacement; (d) Y-directional velocity

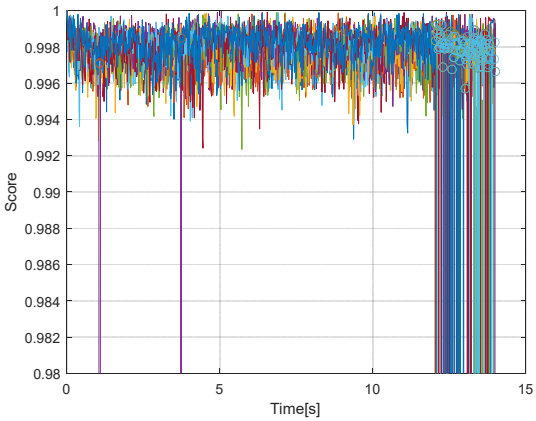


Fig. 10 Error score for tracked points

4.1 Test setup

Fig. 7 shows the experimental setup used in this study, which includes a bridge, Remote Controlled (RC) vehicle, and wired sensors. The bridge is designed and manufactured as two simply supported beams, each beam is 3 m long (the span length), and the width and thickness are 30 cm and 12 mm, respectively. To represent the PFPs on the bridge model, seven circular point markers with a radius of 3 mm were attached at every one of eight locations (i.e., 1/8, 2/8, 3/8, 4/8, 5/8, 6/8, and 7/8 of the span). An RC vehicle of about 23 kg is prepared and is capable of moving at a constant speed set by the user (Wego Robotics 2021). Note that a checkerboard was attached at the center distance between the front and rear wheels. This checkerboard allows localizing the center point, which will be used for examining the 'Vehicle detection and localization' process described in Section 3.3.

As the vehicle moves back and forth along the bridge at 1 m/s, an iPhone in Fig. 2 was handheld at a distance of approximately 3 m from the bridge. During video capture at 60 fps, camera ego-motion was randomly generated by the author as walked around the bridge. At the same time, six PCB accelerometers (353B03, PCB Piezotronics 2021) with a measurable range of ±500 g and resolution of 10 mV/g and two strain-type LVDTs (CDP50, 2021) with a measurable range of 50 mm and a sensitivity of 200 $\times 10^{-6}$ strain/mm are used for validation purpose. Both types of sensors are connected to a data acquisition system (NI cDAQ-9147) and sampled at 2,000 Hz. The locations of sensors are summarized in Fig. 8.

4.2 Bridge response tracking performance validation

Figs. 9(a)-(d) show tracked bridge responses achieved by applying the conventional KLT algorithm (i.e., without applying the proposed framework). As can be seen, features are lost during the tracking, returning a meaningless value or lastly tracked value. Such incidences can be noted with lower error scores from Eq. (5), as shown in Fig. 10. Any point with a score lower than 0.98 or having FB error in Eq. (6), herein defined as larger than 1 pixel, are invalid tracking, requiring a Reset ROI step. Fig. 11 shows the results obtained by applying steps 1 to 4 in the 'Bridge responses tracking from PFPs' process. As can be seen, the tracking remained well until the final frame of the video, capturing PFPs. The results demonstrate the performance of the proposed method for tracking PFP to yield relative displacements of the bridge.

Then, the 'Ego-motion compensation' step is employed to obtain absolute displacement. To quantitatively assess the performance of the bridge response tracking, the results are compared with wired LVDTs as summarized in Table 2. Note that the downward displacement is noted in a positive direction. The root mean squared error (RMSE) of the proposed algorithm referencing the LVDT dataset was about 2.46 mm. Thus, the results confirmed that the proposed algorithm reliably captures the dynamic displacement of the bridge with PFPs in a global manner.

4.3 Vehicle detection and localization performance

To track vehicles with YOLO v5, an image database is constructed for the vehicle used in this study. While the wheels are only tracked, the images were labeled with wheel and car body. An example of an image with the label is shown in Fig. 13. In total, 237 images were collected, and their accuracy was improved by data preprocessing and augmentation. The preprocessing option used herein includes grayscale, resize, and isolate objects. The augmentation processes include crop, shear, blur, noise, and rotation at the image level, and flip at the bounding box level. With defined configurations, a total of 5,080 images were utilized, and the detailed augmentation parameters are provided in Table 1.

The YOLO model used in this study was the middle-sized segmentation model, from an open-source deep learning framework (Github 2022), which employs 22 parameters. The model has been chosen to demonstrate

(a) (b)

Fig. 11 Relative displacements of feature points after applying the proposed framework; (a) lateral displacement; (b) vertical displacement

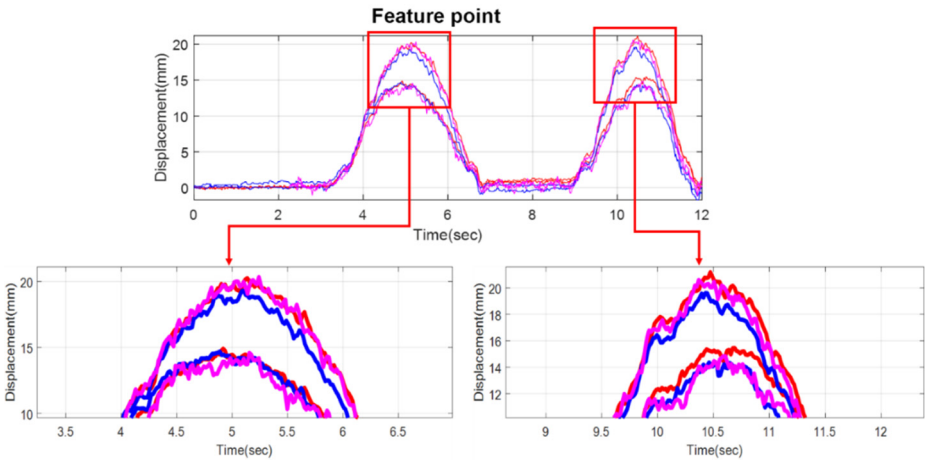


Fig. 12 Comparison of LVDT and Vision feature points

Fig. 13 Example of labeled vehicle

satisfactory speed and training time. Table 2 compares the performance of different YOLOv5 models showing that the middle-sized segmentation model is specifically advantageous with optimized sampling rate capabilities. With its efficient architecture, the model enables the processing of high-resolution sensor data at a faster rate, facilitating real-time monitoring and analysis of structural conditions. This feature of a fast sampling rate allows for more comprehensive and accurate detection of anomalies or defects in structures, enhancing the effectiveness.

Now, the training was conducted with a batch size of 16, an image size of 416 × 416, and a total of 150 epochs. The training on the custom dataset yielded promising results, as shown in Fig. 14. The results well converge for each index, demonstrating the successful creation of a YOLO custom dataset model with appropriate performance.

Tracked wheel points from YOLO are then transformed into absolute displacement by employing the camera's intrinsic parameters and frame-specific parameters. Then the vehicle position is calculated from the center of the two tracked wheels, as shown in Fig. 15(a). The vehicle took about 3.6 seconds to completely cross the bridge in one direction, stayed outside the bridge for 2 seconds, and then returned backward in about 3 seconds. To validate the result, Fig. 15(b) compares the position obtained by tracking the checkerboard attached to the center of the vehicle using the conventional KLT algorithm. The maximum difference between the two approaches is about 2.68%. Although two algorithms yielded agreeing results, both algorithms offer approximated values, requiring an exact localization tool for further validation. Further, the velocity of the vehicle is found by differentiating the estimated position and plotted in Fig. 15(c). The average velocity is about 1 m/s in the first crossing, while the second crossing was slightly accelerated. The tracked vehicle position and bridge responses are used for time-frequency analysis in the subsequent section.

4.4 Time-varying frequency analysis

In this section, time-varying frequency is estimated using the vehicle position obtained using YOLO. Herein a numerical model that describes vehicle bridge interaction is utilized. The model parameters used for the model are described in Lee *et al*. (2022), whose initial first and second natural frequencies are 4.7 Hz and 19.1 Hz, respectively. Tracked vehicle positions from YOLO are plugged into the numerical model to numerically estimate frequency variation as shown in Fig. 16. In the plot, static time-varying frequency is also plotted assuming the constant speed of 1 m/s in the corresponding result. Both results observe frequency variations up to 3.2 Hz for the 1st mode and up to 22.6 Hz for the second mode. Especially, when the vehicle speed was nearly constant, a more precise

Table 1 Database augmentation parameters

Crop	Rotation	Shear	Random Gaussian Blur	Random noise
Zoom 0~20%	$-15° \sim +15°$	Horizontal $\pm 15°$ Vertical $\pm 15°$	Up to 10 pixels	Up to 5% of pixels

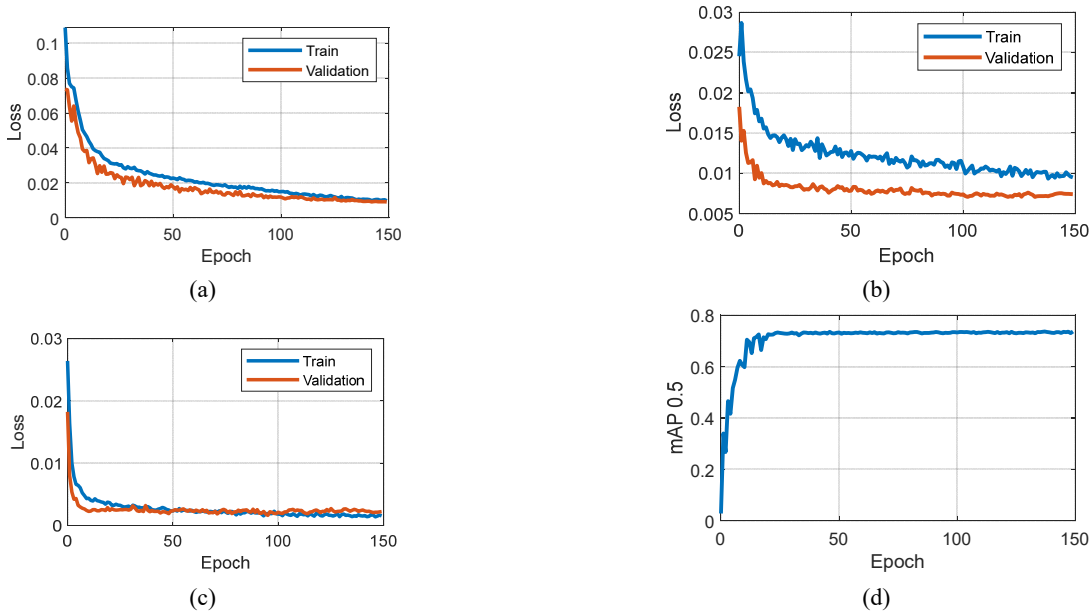


Fig. 14 The variation of evaluation index (a) box loss; (b) object loss; (c) classification loss; (d) mAP 0.5

Table 2 Performance comparison of algorithms

Model	Size (pixels)	mAPbox 50-95	mAPmask 50-95	Train time 300 epochs A100 (hours)	Speed ONNX CPU (ms)	Speed TRT A100 (ms)	Params (M)	FLOPs @640(B)
YOLOv5n	640	27.6	23.4	80:17:00	62.7	1.2	2	7.1
YOLOv5s	640	37.6	31.7	88:16:00	173.3	1.4	7.6	26.4
YOLOv5m	**640**	**45**	**37.1**	**108:36:00**	**427**	**2.2**	**22**	**70.8**
YOLOv5l	640	49	39.9	66:43 (2x)	857.4	2.9	47.9	147.7
YOLOv5x	640	50.7	41.4	62:56 (3x)	1579.2	4.5	88.8	265.7

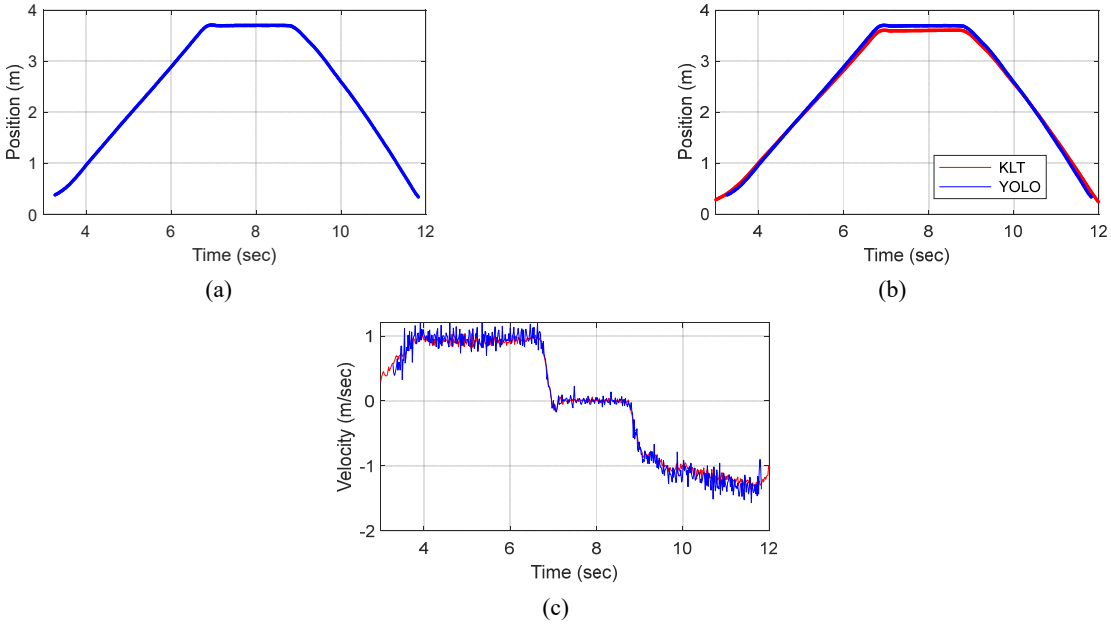


Fig. 15 Vehicle position and velocity: (a) Tracked position from YOLO; (b) Vehicle positions tracked by YOLO in comparison with KLT; (c) Calculated velocity from YOLO in comparison with KLT

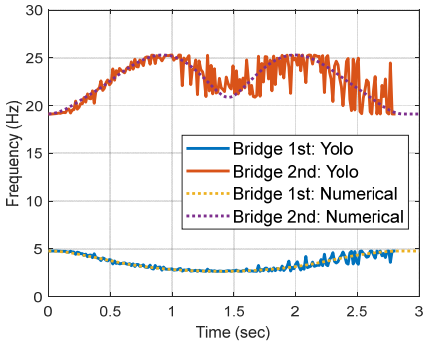


Fig. 16 Frequency variation substituting to the numerical model

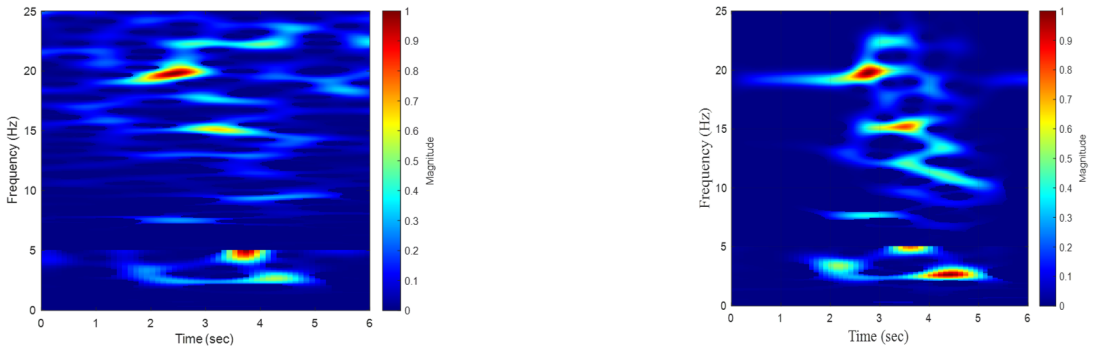
estimation with less oscillation was achieved. Nonetheless, the results confirm that the time-varying frequency reference of the bridge can be obtained using the estimated YOLO-based vehicle positions.

Next, using the tracked bridge dynamic displacement response, the RMST spectrogram is plotted in Fig. 17(a). To better validate the result, the RMST spectrogram using data from an accelerometer is plotted in Fig. 17(b). Here, displacement and acceleration measured at a 3/8L location are used. Although each plot represents different physical quantities, i.e., displacement and acceleration, both RMSTs captured similar frequency variations; the first mode exhibited variations within the 3-5 Hz range and the second mode showed variations in 19-23 Hz zones. Also, the frequency energies show high concentrated around 20 Hz in 2-3 seconds and near 5 Hz in 3-4 seconds. Considering that the acceleration responses are achieved from a sensor in contact with the bridge, the results confirm that the proposed tracking algorithm can efficiently capture the dynamic characteristics, which can be used for time-frequency analysis.

The presented results show the potential of simultaneously tracking bridge dynamics and vehicle location, which is critical for examining VBI dynamics. However, to further enable real-time VBI monitoring the limitation of the proposed algorithm must be considered the following: 1) enhancing resolution: the spectrogram achieved from the proposed algorithm must be enhanced to have finer precision. Also, strategies to interpret the MST spectrogram must be proposed. 2) Synchronization: to facilitate global response monitoring, a strategy to compensate for rolling shutter effect to achieve synchronization of PFPs is needed. 3) Larger scale validation: The proposed algorithms need to be validated from real-world scenarios, addressing challenges associated with environmental factors and out-of-phase responses. 4) Vehicle dynamic detection: To comprehensively examine VBI, the dynamic responses of the vehicle and the impact of mass must be incorporated.

5. Conclusions

Advances in computer vision (CV) technologies have attracted great attention for non-contact sensor-based structural health monitoring applications. Various CV algorithms have demonstrated that vision sensors are flexible and efficient in achieving dynamic displacement measurements. However, to ensure robust and precise tracking, early algorithms often require the stationary position of the sensors. To utilize non-stationary cameras, camera ego-motions are compensated by additional IMU sensors or using stationary background information. Due to such requirements, global monitoring of large-scale civil infrastructure, especially railroad bridges where the impact of moving vehicles is substantial, remained challenging. Thus, in this paper, a framework that 1) tracks the dynamic bridge responses and 2) localizes the moving vehicle has been proposed. In bridge tracking, the algorithm aimed to utilize a nonstationary camera along with global distance measurement, resulting in the tracking of nonstationary poor feature points (PFPs, i.e., features are measured at the single-pixel level). In vehicle localization, wheel centers are detected from the You Only Look Once (YOLO) algorithm as bound boxes. Camera-ego motion and camera distortions are compensated and then coordinates are transformed to better represent the wheel centers. Developed algorithms are validated over laboratory-scale experiments. Measured bridge displacements from the proposed approach are compared with LVDT displacement, yielding a root mean square error (RMSE) value of 2.46 mm. The vehicle localization algorithm is also validated by tracking a checkerboard attached at the center of the vehicle, showing a 2.68% error in those two approaches. Then, a time-



(a) (b)

Fig. 17 Spectrogram cmparison: (a) Displacement from vision data; (b) Accelerometer data

frequency analysis is performed to further demonstrate the efficacy of the proposed algorithms in examining the dynamic characteristics due to vehicle bridge interaction. When compared with contact-based accelerometers the proposed algorithm captured similar frequency deviation. The results demonstrated the potential of monitoring large-scale infrastructure in a global manner, which suffers from nonstationary poor feature points. Moreover, by localizing vehicle wheels simultaneously with the bridge responses, the proposed algorithm can further be utilized in examining vehicle bridge interaction dynamics in real time.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2023-00217983 and No. RS-2024-00454760).

References

Aliansyah, Z., Shimasaki, K., Senoo, T., Ishii, I. and Umemoto, S. (2021), "Single-camera-based bridge structural displacement measurement with traffic counting", *Sensors*, **2**(13), p. 4517. https://doi.org/10.3390/s21134517

Cantero, D., Ülker-Kaustell, M. and Karoumi, R. (2016), "Time–frequency analysis of railway bridge response in forced vibration", *Mech. Syst. Signal Process.*, **76**, 518-530. https://doi.org/10.1016/j.ymssp.2016.01.016

Cho, S., Yun, C.B. and Sim, S.H. (2015), "Displacement estimation of bridge structures using data fusion of acceleration and strain measurement incorporating finite element model", *Smart Struct. Syst., Int. J.*, **15**(3), 645-663. http://dx.doi.org/10.12989/sss.2015.15.3.645

Dong, C.Z. and Catbas, F.N. (2021), "A review of computer vision–based structural health monitoring at local and global levels", *Struct. Health Monitor.*, **20**(2), 692-743. https://doi.org/10.1177/1475921720935585

Dworakowski, Z., Kohut, P., Gallina, A., Holak, K. and Uhl, T. (2016), "Vision-based algorithms for damage detection and localization in structural health monitoring", *Struct. Control Health Monitor.*, **23**(1), 35-50. https://doi.org/10.1002/stc.1755

Feng, D. and Feng, M.Q. (2018), "Computer vision for SHM of civil infrastructure: From dynamic response measurement to damage detection – A review", *Eng. Struct.*, **156**, 105-117. https://doi.org/10.1016/j.engstruct.2017.11.018

Github (2022), Ultralytics YOLO Vision. https://github.com/ultralytics/yolov5/releases

Goshtasby, A. (1988), "Image registration by local approximation methods", *Image Vision Comput.*, **6**(4), 255-261. https://doi.org/10.1016/0262-8856(88)90016-9

Harris, C. and Stephens, M. (1988), "A combined corner and edge detector", In: *Alvey Vision Conference*, Manchester, UK.

Hoskere, V., Park, J.W., Yoon, H. and Spencer Jr, B.F. (2019), "Vision-based modal survey of civil infrastructure using unmanned aerial vehicles", *J. Struct. Eng.*, **145**(7), p. 04019062. https://doi.org/10.1061/(ASCE)ST.1943-541X.0002321

Hwangbo, M., Kim, J.S. and Kanade, T. (2009), "Inertial-aided KLT feature tracking for a moving camera", In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, USA, October.

Jana, D. and Nagarajaiah, S. (2021), "Computer vision-based real-time cable tension estimation in Dubrovnik cable-stayed bridge using moving handheld video camera", *Struct. Control Health Monitor.*, **28**(5), p. e2713. https://doi.org/10.1002/stc.2713

Jian, X., Xia, Y., Lozano-Galant, J.A. and Sun, L. (2019), "Traffic sensing methodology combining influence line theory and computer vision techniques for girder bridges", *J. Sens.*, **2019**(1), p. 3409525. https://doi.org/10.1155/2019/3409525

Jocher, G., Stoken, A., Borovec, J., Changyu, L., Hogan, A., Chaurasia, A., Diaconu, L., Ingham, F., Colmagro, A., Ye, H. and Poznanski, J. (2021), "ultralytics/yolov5: v4. 0-nn. SiLU () activations, Weights & Biases logging, PyTorch Hub integration", Zenodo. https://ui.adsabs.harvard.edu/link_gateway/2021zndo...4418161J/doi:10.5281/zenodo.4418161

Kalal, Z., Mikolajczyk, K. and Matas, J. (2010), "Forward-backward error: Automatic detection of tracking failures", In: *2010 20th International Conference on Pattern Recognition*, Istanbul, Turkey, August.

Kim, S.-W., Jeon, B.-G., Kim, N.-S. and Park, J.-C. (2013), "Vision-based monitoring system for evaluating cable tensile forces on a cable-stayed bridge", *Struct. Health Monitor.*, **12**(5-6), 440-456. https://doi.org/10.1177/1475921713500513

Kim, J., Kim, K. and Sohn, H. (2014), "Autonomous dynamic displacement estimation from data fusion of acceleration and intermittent displacement measurements", *Mech. Syst. Signal Process.*, **42**(1), 194-205. https://doi.org/10.1016/j.ymssp.2013.09.014

Kim, R.E., Moreu, F. and Spencer, B.F. (2015), "System identification of an in-service railroad bridge using wireless smart sensors", *Smart Struct. Syst., Int. J.*, **15**(3), 683-698. https://doi.org/10.12989/sss.2015.15.3.683

Kim, R.E., Moreu, F. and Spencer Jr, B.F. (2016), "Hybrid model for railroad bridge dynamics", *J. Struct. Eng.*, **142**(10), p. 04016066. https://doi.org/10.1061/(ASCE)ST.1943-541X.0001530

Lee, J., Lee, Y.J. and Kim, R.E. (2022), "Identification of system frequency variations in vehicle-bridge interaction systems", *J. Comput. Struct. Eng. Inst. Korea*, **35**(1), 23-28. https://doi.org/10.7734/COSEIK.2022.35.1.23

Li, S. and Yoon, H.S. (2023), "Vehicle localization in 3D world coordinates using single camera at traffic intersection", *Sensors*, **23**(7), p. 3661. https://doi.org/10.3390/s23073661

Lucas, B.D. and Kanade, T. (1981), "An iterative image registration technique with an application to stereo vision", In: *JCAI'81: 7th International Joint Conference on Artificial Intelligence*, Volume 2, pp. 674-679, Vancouver, BC, Canada, August.

Martini, A., Tronci, E.M., Feng, M.Q. and Leung, R.Y. (2022), "A computer vision-based method for bridge model updating using displacement influence lines", *Eng. Struct.*, **259**, 114-129. https://doi.org/10.1016/j.engstruct.2022.114129

MATLAB (2023), "Computer Vision Toolbox (R2023b), The MathWorks Inc., Natick, MA, USA.

Özcan, O. and Özcan, O. (2021), "Automated UAV based multi-hazard assessment system for bridges crossing seasonal rivers", *Smart Struct. Syst., Int. J.*, **27**(1), 35-52. https://doi.org/10.12989/sss.2021.27.1.035

Park, J.W., Sim, S.H. and Jung, H.J. (2013), "Displacement estimation using multimetric data fusion", *IEEE/ASME Trans. Mechatron.*, **18**(6), 1675-1682. https://doi.org/10.1109/TMECH.2013.2275187

Ribeiro, D., Santos, R., Cabral, R., Saramago, G., Montenegro, P., Carvalho, H., Correia, J. and Calçada, R. (2021), "Non-contact structural displacement measurement using Unmanned Aerial Vehicles and video-based systems", *Mech. Syst. Signal Process.*, **160**, p. 107869. https://doi.org/10.1016/j.ymssp.2021.107869

Shi, J. (1994), "Good features to track", In: *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*,

Seattle, WA, USA, June.
Soh, Y., Qadir, M., Mehmood, A., Hae, Y., Ashraf, H. and Kim, I. (2014), "A feature area-based image registration", *Int. J. Comput. Theory Eng.*, **6**(5), 407-411. https://doi.org/10.7763/IJCTE.2014.V6.899
Spencer Jr, B.F., Sim S.H., Kim, R.E. and Yoon, H. (2025), "Advances in artificial intelligence for structural health monitoring: A comprehensive review", KSCE J. Civ. Eng., 29(3), 100203. https://doi.org/10.1016/j.kscej.2025.100203.
Stockwell, R.G., Mansinha, L. and Lowe, R.P. (1996), "Localization of the complex spectrum: the S transform", *IEEE Trans. Signal Process.*, **44**(4), 998-1001. https://doi.org/10.1109/78.492555
Tan, D., Ding, Z., Li, J. and Hao, H. (2023), "Target-free vision-based approach for vibration measurement and damage identification of truss bridges", *Smart Struct. Syst., Int. J.*, **31**(4), 421-436. https://doi.org/10.12989/sss.2023.31.4.421
Tian, Y. and Zhang, J. (2020), "Structural flexibility identification via moving-vehicle-induced time-varying modal parameters", *J. Sound Vib.*, **474**, p. 115264. https://doi.org/10.1016/j.jsv.2020.115264
Tomasi, C. and Kanade, T. (1991), "Detection and tracking of point", *Int. J. Comput. Vision*, **9**(137-154), 3.
Yang, Y.B., Cheng, M.C. and Chang, K.C. (2013), "Frequency variation in vehicle–bridge interaction systems", *Int. J. Struct. Stab. Dyn.*, **13**(02), p. 1350019. https://doi.org/10.1142/S0219455413500193
Yoon, H., Elanwar, H., Choi, H., Golparvar-Fard, M. and Spencer Jr, B.F. (2016), "Target-free approach for vision-based structural system identification using consumer-grade cameras", *Struct. Control Health Monitor.*, **23**(12), 1405-1416. https://doi.org/10.1002/stc.1850
Yoon, H., Shin, J. and Spencer Jr, B.F. (2018), "Structural displacement measurement using an unmanned aerial system", *Comput.-Aided Civ. Infrastruct. Eng.*, **33**(3), 183-192. https://doi.org/10.1111/mice.12338
Yuan, P.P., Zhang, J., Feng, J.Q., Wang, H.H., Ren, W.X. and Wang, C. (2022), "An improved time-frequency analysis method for structural instantaneous frequency identification based on generalized S-transform and synchroextracting transform", *Eng. Struct.*, **252**, p. 113657. https://doi.org/10.1016/j.engstruct.2021.113657
Zhan, Y., Au, F.T.K. and Yang, D. (2020), "Extraction of bridge information based on the double-pass double-vehicle technique", *Smart Struct. Syst., Int. J.*, **25**(6), 679-691. http://dx.doi.org/10.12989/sss.2020.25.6.679
Zhang, J., Yang, D., Ren, W.X. and Yuan, Y. (2021), "Time-varying characteristics analysis of vehicle-bridge interaction system based on modified S-transform reassignment technique", *Mech. Syst. Signal Process.*, **160**, p. 107807. https://doi.org/10.1016/j.ymssp.2021.107807
Zhao, D., He, W., Deng, L., Wu, Y., Xie, H. and Dai, J. (2021), "Trajectory tracking and load monitoring for moving vehicles on bridge based on axle position and dual camera vision", *Remote Sens.*, **13**(23), p. 4868. https://doi.org/10.3390/rs13234868
Zhu, J., Lu, Z. and Zhang, C. (2021), "A marker-free method for structural dynamic displacement measurement based on optical flow", *Struct. Infrastruct. Eng.*, **18**(1), 84-96. https://doi.org/10.1080/15732479.2020.1835999