Received: 15 August 2023 | Accepted: 8 February 2024

DOI: 10.1111/mice.13177

RESEARCH ARTICLE

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

# Real-time displacement measurement for long-span bridges using a compact vision-based system with speed-optimized template matching

**Miaomin Wang**[1,2] | **Fuyou Xu**[1] | **Ki-Young Koo**[2] | **Pinqing Wang**[1]

[1]School of Infrastructure Engineering, Dalian University of Technology, Dalian, China

[2]Vibration Engineering Section, College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter, UK

**Correspondence**
Fuyou Xu, School of Infrastructure Engineering, Dalian University of Technology, Dalian 116024, China.
Email: fuyouxu@hotmail.com

Ki-Young Koo, Vibration Engineering Section, Faculty of Environment, Science and Economy, University of Exeter, Exeter, EX4 4QF, UK.
Email: k.y.koo@exeter.ac.uk

**Funding information**
National Science Fund for Distinguished Young Scholars, Grant/Award Number: 52125805; Engineering and Physical Sciences Research Council, Grant/Award Number: EP/W005816/1

**Abstract**

This paper introduces a new accelerating algorithm, efficient match slimmer (EMS), specifically designed to lighten computational loads of sophisticated template matching algorithms, enabling these algorithms to be effectively run on single-board computers. Utilizing EMS in conjunction with a robust template matching algorithm, we have developed Raspberry Vision—a compact, cost-effective, and real-time vision-based system. Its compactness and portability facilitate a practical measurement strategy that not only minimizes the camera-to-target distance but also simplifies the camera calibration process in bridge displacement monitoring, thereby enhancing measurement accuracy. The performance of the system is estimated on two operational suspension bridges. The results demonstrate that Raspberry Vision, equipped with the measurement strategy, can significantly improve the measurement accuracy in the long-span bridge test and is also suitable for cross-sea bridge measurements.

## 1 | INTRODUCTION

Structural health monitoring (SHM) has become a very important area in bridge engineering, playing a key role in keeping structures safe and lasting longer (X. Pan et al., 2023). At its core, SHM involves the continuous observation of structures using various sensing technologies to detect and assess changes in the structures, thereby ensuring structural integrity and functionality (Javadinasab Hormozabad et al., 2021). Usually, the information collected from bridges includes vibration response (Perez-Ramirez et al., 2019; Pezeshki, Adeli, et al., 2023; Pezeshki, Pavlou, et al., 2023), deformation (Park et al., 2015), and so forth.

Bridge displacement is a critical index for SHM (Brownjohn et al., 2017) and can be measured by displacement transducers, such as linear variable differential transformers (Boothby et al., 1998), global positioning systems (GPS; Le & Nishio, 2019; Msaewe et al., 2021; Nakamura, 2000), potentiometric displacement sensors (Guo et al., 2015), dial gauges (Bidez et al., 1986), and optic fiber sensors (Hampshire & Adeli, 2000). Additionally, displacement can be indirectly estimated through methods such as strain (Oh et al., 2017), double integration of acceleration data

(Bajwa et al., 2020; Bunce et al., 2023; K. Feng et al., 2023; F. Liu et al., 2021), a combination of accelerometers and strain gauges, or a millimeter wave radar (Ma et al., 2023). Recently, the integration of data fusion techniques in displacement measurement has garnered increasing attention. These techniques involve combining data from different sources, such as GPS and acceleration data (Shen et al., 2023; Xu et al., 2017; Yang et al., 2021), or acceleration and strain (Zhu et al., 2020), to enhance measurement accuracy and reliability. However, installing these contact-type sensors can be labor-intensive and, in some cases, impractical in many bridge applications (Wang, Xu, et al., 2022).

In contrast to contact sensors, non-contact displacement sensors such as microwave interferometric radar (H. Zhang et al., 2023; Zhao et al., 2020), laser sensor (Kim & Jung, 2022; H. Zhang et al., 2023), and total station (Pehlivan, 2022) enable remote measurement of bridge displacement without physical access to the structure. While these contactless sensors offer significant advantages in terms of safety and convenience, they are often accompanied by high costs, which can limit their widespread use.

Vision-based displacement systems represent another contactless approach and are capable of remotely capturing structural displacement. These systems not only eliminate the need for physical contact but also provide a highly accurate solution for measuring bridge displacement (Ma et al., 2022; Wang, Xu, et al., 2022). These systems typically use a camera to record video footage of targets on the bridge, which is then analyzed by image processing software to calculate bridge displacement (Luo et al., 2018; Xu & Brownjohn, 2018). Examples of such systems include single-lens reflex camera-based systems (Ge et al., 2023; Hoag et al., 2017; Narazaki et al., 2021), camcorder-based systems (Shao et al., 2023), unmanned aerial vehicle systems (Hoskere et al., 2019; Weng et al., 2021; Yoon et al., 2018), and action camera-based systems (Xu et al., 2018; Lydon et al., 2019).

Image signals of cameras are susceptible to a variety of environmental factors, prompting the development of diverse image processing algorithms to enhance their reliability. Techniques such as template matching, feature point matching, sparse optical flow, and motion magnification have been advanced for this purpose. Among these, template matching algorithms are particularly noted for their robustness against lighting variations and occlusions. When combined with subpixel refinement techniques, these algorithms are capable of delivering highly accurate measurement results. However, a significant limitation of template matching algorithms lies in their exhaustive computational requirements. This aspect becomes particularly challenging in real-time measurement systems, especially when operating on computers with limited computational resources (Shajihan et al., 2022; Yu et al., 2023). Additionally, more sophisticated and robust versions of template matching algorithms tend to be computationally more complex, exacerbating the challenge of achieving rapid computations.

Existing real-time vision-based systems for bridge displacement measurement typically rely on high-performance computing hardware to achieve operational speed. These systems often incorporate various acceleration strategies, such as defining a local region for matching (Luo & Feng, 2018) and employing multi-thread processing (Shuai et al., 2018), to enhance efficiency. While many such systems have been successfully implemented in full-scale bridge monitoring (Brownjohn et al., 2017; Hu et al., 2023; Jeong & Jo, 2022; X. Pan et al., 2023; Tian & Pan, 2016), their widespread adoption is impeded by the high costs and complexity associated with high-performance hardware and intricate installation processes (C. Liu et al., 2016; Su et al., 2019).

In an effort to mitigate these cost barriers, our previous work introduced a low-cost, real-time monitoring system based on the Raspberry Pi platform (Wang et al., 2023). This system used a basic template matching algorithm, zero-mean normalized cross-correlation coefficient (ZNCC), to capture structural displacement. However, the ZNCC algorithm's limited robustness against dramatic illumination changes and heavy occlusions poses significant challenges for long-term monitoring in diverse environmental conditions.

To address this limitation, this paper proposes a new accelerating algorithm, efficient match slimmer (EMS), designed to enhance the performance of a more robust template matching algorithm, gradient matching via voting (GMV; Wang, Ao, et al., 2022a), for real-time operation on Raspberry Pi. EMS aims to decrease computational loads by implementing a balanced risk-return criterion technique and defining a localized matching region. The goal is to enable sophisticated template matching algorithms to operate in real time on single-board computers (SBCs), reducing the costs of real-time vision-based systems.

In addition, applications of the existing real-time vision systems in certain scenarios remain challenging:

1. Long-span bridges (Fukuda et al., 2013; Ye et al., 2013; J. Zhang et al., 2022): The considerable object–camera distance (the distance between a test point [e.g., midspan] and the camera's location [e.g., riverbank]) can lead to reduced measurement accuracy due to the camera's extensive field of view (FOV) and optical turbulence.
2. Cross-sea bridges (He et al., 2022; Jung et al., 2019; Zhao & Yu, 2020): Identifying a stable ground area for installing the system hardware may prove difficult, presenting a significant challenge for real-time vision-based displacement measurement in such cases.

To address the two limitations, a practical measurement strategy is proposed in this study. In this strategy, the vision-based system is mounted on a bridge tower or pier, effectively reducing the object–camera distance and simplifying the camera calibration process when measuring the girder displacement as compared to scenarios where the system is positioned on the riverbank. Notably, a stable ground is not required, facilitating the possibility of displacement measurements in cross-sea bridge applications. To facilitate this strategy, we have developed a more compact and efficient system, called Raspberry Vision. This system represents an evolution of the previous model, achieved through a comprehensive redesign of the hardware package and the integration of a speed-enhanced GMV (SGMV) algorithm facilitated by EMS.

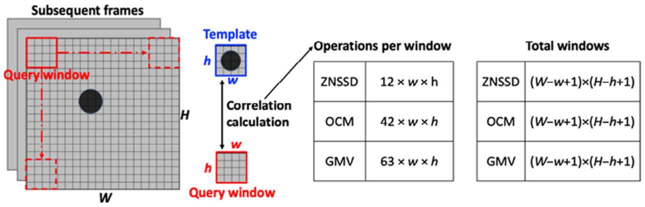
This article is structured as follows: Section 2 introduces EMS for accelerating template matching algorithms. Section 3 provides an overview of the hardware and software components of Raspberry Vision. Section 4 documents a laboratory experiment conducted to assess the system's performance and explore the impact of different factors in SGMV on its computation time and accuracy. In Section 5, a field test on a long-span suspension bridge is carried out, demonstrating the advantages of the proposed measurement strategy over traditional methodologies. Section 6 presents a displacement measurement of a cross-sea bridge, illustrating the effectiveness of Raspberry Vision in such applications. Finally, Section 7 summarizes the conclusions drawn from this study.

## 2 | EMS FOR COMPUTATIONAL OPTIMIZATION OF TEMPLATE MATCHING ALGORITHMS

### 2.1 | Computational complexity analysis of widely used template matching algorithms

Template matching for object tracking involves selecting a region of interest (ROI) in the initial video frame as a template (with a size of w × h), and then searching for the most similar region in subsequent frames. This process involves comparing the template with a query window of the same dimensions, initially positioned at the frame's upper left corner and moving pixel-by-pixel as shown in Figure 1. The window that exhibits the highest correlation with the template is identified as the matched window, which is considered as the target's location in each frame (a size of $W \times H$).

The correlation $C_q$ between the template and a query window is calculated using a general formula $C_q = \frac{\sum_{n=1}^{w\times h} f(A_{i,j},B_{i,j})}{w\times h}$, where $f(A_{i,j}, B_{i,j})$ is a function that estimates the correlation between pixel features $A_{i,j}$ and $B_{i,j}$ in the template and a window, respectively. Different algorithms employ various methods for calculating this correlation in each query window, leading to differences in computational complexity. Additionally, more advanced and robust template matching algorithms tend to be computationally more complex. For example, the zero-mean normalized sum of squared differences (ZNSSD; B. Pan et al., 2010; Tian & Pan, 2016) involves multiple steps. It requires $w \times h$ operations for calculating the average, an equal number of subtractions to adjust the pixel intensities by removing the average, and $3 \times w \times h$ operations for standard deviation computations. These steps are performed for both the template and the corresponding query window, culminating in $5 \times w \times h$ operations for each. Subsequently, $w \times h$ subtractions are carried out to determine the differences between them. Finally, $w \times h$ squarings are necessary to calculate the ZNSSD. As a result, the total number of operations required in the ZNSSD algorithm amounts to $12 \times w \times h$.



Operations per window

| | |
|---|---|
| ZNSSD | 12 × w × h |
| OCM | 42 × w × h |
| GMV | 63 × w × h |

Total windows

| | |
|---|---|
| ZNSSD | (W−w+1)×(H−h+1) |
| OCM | (W−w+1)×(H−h+1) |
| GMV | (W−w+1)×(H−h+1) |

FIGURE 1 Template matching algorithms and their computational complex. GMV, gradient matching via voting; OCM, orientation coding matching; ZNSSD, zero-mean normalized sum of squared differences.

In contrast to traditional template matching algorithms, GMV estimates the correlation based on gradient information between the template and its corresponding query window. A key distinction of GMV is its extraction of the target's edge as the template, which consequently involves fewer pixels in the correlation calculation. The process begins with GMV applying a 3 × 3 Sobel operator to compute the gradient vectors for each pixel in both the template and the query window, necessitating $34 \times w \times h$ operations. Following this, it calculates the correlation coefficients for both direction and magnitude, requiring $13 \times w \times h$ and $15 \times w \times h$ operations, respectively. Finally, GMV conducts a voting process, where the correlation at each pixel is determined as either 1 or 0, requiring an additional $1 \times w \times h$ operations. Cumulatively, this results in a total of $63 \times w \times h$ operations for the correlation calculation between the template and a corresponding query window.

Similarly, the orientation coding matching (OCM) algorithm (D. Feng & Feng, 2016; Fukuda et al., 2013), another prevalent method, computes the gradient orientation angle at each pixel when estimating the correlation between

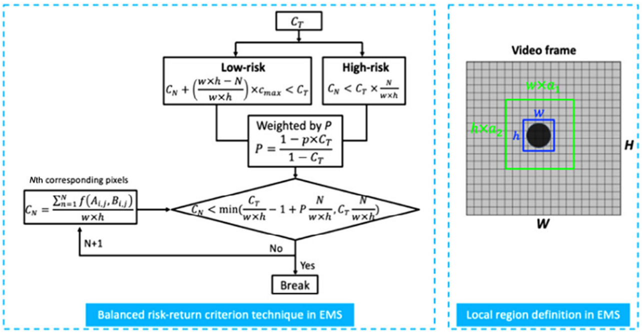

$C_T$
Low-risk
$C_N + \left(\frac{w\times h - N}{w\times h}\right)\times c_{max} < C_T$
High-risk
$C_N < C_T \times \frac{N}{w\times h}$
Weighted by $P$
$P = \frac{1 - p\times C_T}{1 - C_T}$
Nth corresponding pixels
$C_N = \frac{\sum_{n=1}^{N} f(A_{i,j}, B_{i,j})}{w\times h}$
$C_N < \min(\frac{C_T}{w\times h} - 1 + P\frac{N}{w\times h}, C_T\frac{N}{w\times h})$
N+1
No
Yes
Break
Balanced risk-return criterion technique in EMS
Video frame
$w\times a_1$
$w$
$h\times a_2$
$h$
$H$
$W$
Local region definition in EMS

FIGURE 2 Overview of efficient match slimmer (EMS).

the template and the corresponding query window. This operation-intensive process requires a total of 42 × $w \times h$ operations, including multiplications, additions, divisions, and square roots, among others.

The need to calculate correlation in all query windows within a video frame further increases the computational complexity, with the number of windows being $(W—w + 1) \times (H—h + 1)$.

## 2.2 | EMS for accelerating template matching algorithms

The exhaustive computations inherent in traditional template matching algorithms result in a high volume of operations, significantly slowing down image processing speeds, particularly in vision-based displacement measurement applications. This presents substantial challenges for real-time measurement, especially in systems based on SBCs, where computational resources are limited. To address this issue, this paper introduces EMS, an efficient approach designed to reduce the operational load in correlation calculations for template matching algorithms. The overview of EMS is shown in Figure 2. EMS optimizes correlation computation in each query window by implementing a balanced risk-return criterion technique and reduces the number of windows by defining a local matching region.

### 2.2.1 | A balanced risk-return criterion technique in EMS for correlation computation optimization

The correlation between the matched window and the template is indicative of the moved target's completeness in comparison with its initial state. In vision-based structural displacement measurements, the target typically undergoes translational movement and maintains its integrity, resulting in a high correlation for the matched window. Conversely, most query windows, especially those far from the moved target, exhibit a significantly lower correlation. Consequently, performing correlation computations at all pixels within these low-correlation windows is inefficient.

To address this, EMS employs a threshold combined with a balanced risk-return criterion technique to selectively reduce the number of operations needed in the correlation computation as shown in Figure 2. This threshold, denoted as $C_T$, represents the anticipated minimum completeness of the target during measurement. $C_T$ should fall in the range from 0 to 1, and its value is defined by users.

*A low-risk criterion*

The correlation between the template and a query window does not require complete evaluation when the matching process based on the threshold $C_T$ that a potential match must reach. Assume that $C_N$ is the normalized sum of the $f(A_{i,j}, B_{i,j})$ of all preceding pixels when $C_q$ is calculated up to the $N^{\text{th}}$ $f(A_{i,j}, B_{i,j})$:

$$C_N = \frac{\sum_{n=1}^{N} f\left(A_{i,j}, B_{i,j}\right)}{w \times h} \tag{1}$$

If, when calculating $C_q$ of a query window, it becomes evident that even with all remaining pixels within the window having a max $f(A_{i,j}, B_{i,j})$ marked as $c_{\max}$, $C_q$ still cannot exceed $C_T$, then the calculation can be discontinued. There is no need to continue a calculation that will not meet the threshold. This can be formulated:

$$C_N + \frac{(w \times h - N) \times c_{\max}}{w \times h} < C_T \tag{2}$$

Since $c_{\max}$ is often 1 in most matching template algorithms, Formula (2) can be rearranged as

$$C_N < C_T - 1 + N/(w \times h) \tag{3}$$

Formula (3) represents a low-risk criterion for stopping the correlation calculation at a given threshold $C_T$.

*A high-risk criterion*

Another criterion introduces a more stringent requirement: Each incremental addition must exceed a threshold value, $C_T$. If this condition is not met, the process of calculating correlation is halted immediately. This stopping criterion is mathematically expressed as

$$C_N < C_T \times N/(w \times h) \tag{4}$$

While this approach ensures efficiency, it also presents a notable limitation in terms of matching accuracy. This is particularly evident when segments of the moved target that are yet to be completed are assessed first. In such cases, the cumulative score may not reach the required threshold, leading to potential errors in correctly identifying the target in subsequent frames.

*A weight factor*

To ensure a very low probability of incorrectly localizing the moved target, the low-risk and high-risk criteria are utilized as follows:

$$C_N < \min\left(\frac{C_T}{w \times h} - 1 + \frac{N}{w \times h}, C_T \frac{N}{w \times h}\right) \tag{5}$$

Typically, the low-risk criterion is dominant in Equation (5), primarily governing the stopping condition. To further accelerate the correlation computation, a weight factor $p$ in the range [0, 1] is introduced to balance the weights between the two criteria. This can be formulated as

$$C_N < \min\left(\frac{C_T}{w \times h} - 1 + P\frac{N}{w \times h}, C_T \frac{N}{w \times h}\right) \tag{6}$$

$$P = \frac{1 - p \times C_T}{1 - C_T} \tag{7}$$

When $p = 0$, $P$ attains its maximum for a given for a given $C_T$, causing the high-risk criterion to predominantly control the stopping condition. Conversely, when $p = 1$, $P$ becomes 1 regardless of $C_T$ making Equation (6) equivalent to Equation (5), where the low-risk criterion is the primary determinant. Typically, the safe factor can be set to values as high as 0.9 without incorrectly localizing the moved target. The impact of the value of $p$ on matching accuracy will be further discussed in Section 3.

Furthermore, to prevent the generation of incorrect measurement results caused by a low threshold $C_T$, EMS is designed to record the value of $C_N$ each time the stopping criterion is triggered during the correlation calculation for every query window. If the highest recorded $C_N$ is less than the threshold $C_T$, the displacement result for that particular frame will not be outputted. This precaution helps to ensure the reliability of the measurement outcomes.

### 2.2.2 | Local region definition in EMS for reduction of the number of query windows

In traditional template matching algorithms applied to video frames with large dimensions, such as 1920 × 1080, the correlation calculation for all query windows within subsequent frames results in a substantial operational load. However, in full-scale bridge measurements, the camera's FOV is typically quite extensive (on the scale of meters or more), while the structural displacement is relatively minor (at the level of centimeters or millimeters). Consequently, the target moves only within a small area in the image. Recognizing this, EMS strategically defines a local region around the initial target's location. This approach significantly reduces the computational demand by decreasing the number of query windows that need processing, making it a more efficient solution for large-scale structural monitoring.

In the initial frame, the coordinate of ROI containing the target is ($w$, $h$). The dimension of the local region, ($W_{lcoal}$, $H_{local}$), is defined as

$$W_{local} = w \times a_1 \tag{8}$$

$$H_{local} = h \times a_2 \tag{9}$$

where $a_1$ and $a_2$ are amplification factors for width and height, respectively. The values of these factors can be determined by users based on the structural vibration characteristics.

In bridge displacement measurement, users typically have prior knowledge of the maximum physical displacement as outlined by relevant technical standards. Additionally, the physical dimensions of the measuring target and its corresponding size in the image plane can generally be estimated. Based on this information, it is possible to approximate a range for the target's displacement within the image plane. Consequently, this enables a rough estimation of the values for $a_1$ and $a_2$. To further ensure measurement reliability, it is prudent to set these values at 1.2 times their initial estimates. This approach balances computational efficiency with measurement accuracy.

## 2.3 | Demonstration of computational efficiency improvement using EMS

This section evaluates the computational efficiency improvements facilitated by EMS in three widely used template matching algorithms, thereby demonstrating EMS's effectiveness. An image of 1920 × 1080 pixels featuring an annulus with an external diameter of 50 pixels and an internal diameter of 20 pixels was created for this test. A 50 × 50 pixel ROI containing the annulus served as the template. The objective was to match this template within the image using three sophisticated algorithms: OCM, ZNSSD, and GMV. These processes were executed on a Raspberry Pi 4B with 8 GB RAM.

**TABLE 1** Comparison of computational times for various algorithms with and without EMS (unit: ms).

| Case | ZNSSD | OCM | GMV |
|---|---|---|---|
| 1 | 243,902 | 336,623 | 72,748 |
| 2 | 344.608 | 447.905 | 98.344 |
| 3 | 179.234 | 276.531 | 57.463 |

Abbreviations: EMS, efficient match slimmer; GMV, gradient matching via voting; OCM, orientation coding matching; ZNSSD, zero-mean normalized sum of squared differences.

Three distinct cases were established for the test.

Case 1: no implementation of an accelerating algorithm.

Case 2: only implementation of a local region definition, setting both amplification factors $a_1$ and $a_2$ at 2, which is the widely used method in the community.

Case 3: implementation of EMS, using the same amplification factors as in Case 2, but with $p = .9$. For OCM and ZNSSD, the threshold $C_T$ was set to 0.3 to assess the discrepancy between the template and query windows during the matching process. In contrast, the threshold $C_T$ was set to 0.7 for GMV to calculate the correlation.

Each case was executed 20 times to ensure reliability, and the average computational times were recorded and are presented in Table 1.

The test results clearly demonstrate the efficiency of EMS in three template matching algorithms. Without EMS, processing times for localizing a template in a 1920 × 1080 pixels image were excessively long, making real-time processing on a Raspberry Pi platform impractical. However, a significant reduction in computational time was observed when using a local region to decrease the number of query windows for all template matching algorithms. Despite this improvement, achieving the desired sampling rate for real-time measurement with only the local region definition remains a challenge. As indicated in Table 1, the implementation of EMS led to further reductions in computational time: a 48% reduction for ZNSSD, a 38% reduction for OCM, and a 42% reduction for GMV, compared to the reductions achieved by solely defining a local region. This enhanced efficiency renders these template matching algorithms more suitable for SBC-based real-time measurement. Among the three algorithms, GMV achieved the matching with the least time. This is because GMV extracts the target's edge as the template, which consequently involves fewer pixels in the correlation calculation. The influence of EMS parameter values on matching accuracy remains an area for further exploration as will be discussed in Section 3.

## 3 | A COMPACT REAL-TIME VISION-BASED DISPLACEMENT SYSTEM

### 3.1 | Hardware

The components of the previous system are combined and mounted on a tripod, as illustrated in Figure 3a, making it portable and easy to be placed on the ground (e.g., riverbank) for bridge displacement measurement. However, the installation location requirements may result in significant object–camera distance in long-span bridge applications, leading to reduced measurement accuracy. Additionally, it can be impractical to find a suitable area for the system during cross-sea bridge tests.

Raspberry Vision consolidates the components within an aluminum case, designed not only for easy mounting on structural surfaces but also for enhanced sealing and protection. This feature effectively guards against potential damage or interference from external elements like rain and dust. The design details are showcased in Figure 3b. A square hole in the front wall of the case, covered by high-transparency glass, ensures the camera maintains a clear view of the targets. An acrylic sheet serves as a roof, guaranteeing the output on the monitor is easily visible. The components are secured to the bottom plate, which is separate from the wall, preventing movements from wind-induced vibrations.

A Raspberry Pi 4B is used in this study for image processing. It has a 1.5 GHz quad-core 64-bit ARM Cortex-A72 central processing unit (CPU) and 8GB memory, operating on Raspbian Buster system. For video frame capture, we use a Raspberry Pi High Quality (HQ) Camera. This camera utilizes a Sony IMX477 sensor, which has a pixel size of $1.55 \times 1.55\,\mu$m and a native resolution of 4056 × 3040 pixels. However, for actual measurements, the resolution is often reduced to 1280 × 720 pixels. In this configuration, the camera's image processing involves supersampling, where the original horizontal resolution of 4056 pixels is effectively downscaled to 1280 pixels. Correspondingly, to maintain the aspect ratio, a proportionate number of pixels in the vertical direction is not utilized in the captured images.

Image sequences featuring targets on the bridge are captured by the Raspberry Pi HQ camera and subsequently transmitted to the Raspberry Pi computer for image processing via a camera serial interface cable. The captured images and measurement data are displayed on the monitor for easy monitoring. All devices are powered by a 57,000 mAh portable power bank, capable of operating for over

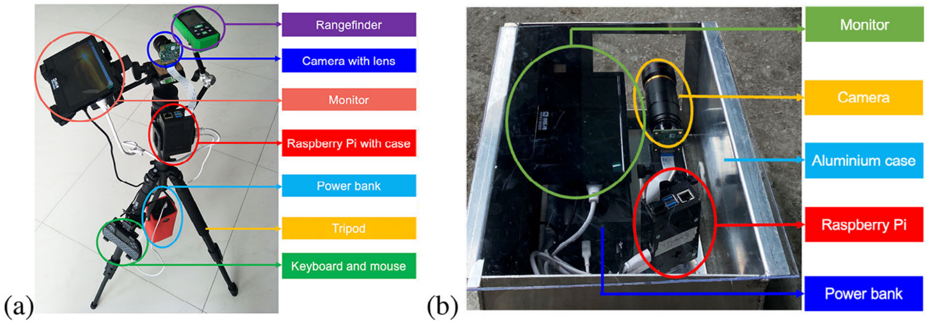


FIGURE 3 The hardware of (a) the previously developed system and (b) the currently developed system.

40 h. The system is controlled using a wireless mouse and keyboard.

## 3.2 | Software

### 3.2.1 | Introduction of GMV

The authors previously introduced GMV to facilitate accurate measurements in challenging conditions, such as when moved targets in video frames are incompletely recorded due to dramatic changes in illumination. Traditional template matching algorithms typically do not assign weights to pixel-level similarities when evaluating the similarity (or correlation) between the template and a query window in video frames. In contrast, GMV employs a voting scheme that weights the similarities of all pixels. This approach ensures that GMV can robustly track targets, even when they undergo significant feature losses. The efficacy of the GMV algorithm has been verified through a laboratory experiment and two field tests.

The basic workflow of GMV is structured as follows: Initially, an ROI with a size of $w \times h$ containing the measuring target is selected in the first frame of the video. From this ROI, the edge points of the target are extracted to form the template, with each point in the template acting as a voter. In the subsequent step, these voters either vote for or abstain from voting for corresponding pixels in the first query window. This process determines the pixel similarity (PS) scores at the corresponding pixels. The region similarity (RS) score for the first query window is then calculated by averaging all the PS scores within this window, with the resulting value expected to fall within the range of [0, 1]. This voting procedure is repeated for all windows within the subsequent frame, and the position of the query window with the highest RS score is identified as the new location of the moved target. More details about GMV can be found in Wang, Ao, et al. (2022a).

However, using GMV can become time-consuming when dealing with a significant number of voters or large frame sizes. For instance, the computational time exceeded 3 s per frame when processing 980 voters in 110 × 1160 resolution video frames (Wang, Ao, et al., 2022a). These computations were done on a Lenovo T540P laptop, equipped with an Intel i7-4700MQ CPU. Considering these factors, the processing time on a Raspberry Pi would be even longer. This poses a significant challenge for real-time measurements.

### 3.2.2 | SGMV facilitated by EMS

GMV is accelerated by using EMS to ensure its real-time functionality on the Raspberry Pi 4B. In this context, we define the PS score as $S_{PS}$ and RS score as $S_{RS}$. Let $S_{RS_T}$ be the threshold for the RS score. Assume that $S_{RS_N}$ is the sum of the $S_{PS}$ of all preceding pixels when $S_{RS}$ is calculated up to the $N$th $S_{PS}$:

$$S_{RS_N} = \frac{\sum_{n=1}^{N} S_{PS_n}}{w \times h} \tag{10}$$

The balanced risk-return criterion technique in EMS for GMV is expressed as follows:

$$S_{RS_N} < \min\left(\frac{S_{RS_T}}{w \times h} - 1 + P\frac{N}{w \times h}, S_{RS_T}\frac{N}{w \times h}\right) \tag{11}$$

If, during the calculation of $S_{RS}$ for a query window, $S_{RS_N}$ meet the condition set in Equation (11), the calculation for that window can be discontinued.

Additionally, EMS defines a local region to reduce the number of query windows as outlined in Equations (8) and (9). Given that the primary displacement in bridges typically occurs vertically, the corresponding amplification factors for this direction can be set as $a_1 = 1.2$ and $a_2 = 2$.

**TABLE 2** Comparison of the previous system, Raspberry Vision, and a commercial system.

| Aspects | | The previous system | Raspberry Vision | Commercial system (Brownjohn et al., 2017) |
|---|---|---|---|---|
| Hardware | Computing unit | Raspberry Pi 4B | Raspberry Pi 4B | High-performance PC |
| | Imaging unit | Raspberry Pi HQ camera | Raspberry Pi HQ camera | Industrial camera |
| | Mounting | Tripod | Aluminum case | Tripod only for camera |
| Image processing algorithm | | ZNCC | SGMV | Unknown |
| Maximum sampling rate | | 30 Hz | 30 Hz | 50 Hz |
| System installation location | | Ground | Bridge tower or pier | Ground |
| Camera calibration | Method | Scale factor | Scale factor | Unknown |
| | Distance measurement | By a laser rangefinder (maximum 100 m) | Based on design drawings | By a laser rangefinder |
| | Tilt angle | Often significant and it is difficult to be measured | Rarely significant | – |
| Cost | | $480 | $400 | More than $10,000 |

Abbreviation: ZNCC, zero-mean normalized cross-correlation coefficient; SGMV, speed-enhanced GMV.

### 3.2.3 | Physical displacement calculation

After SGMV successfully localizes the moved target in the subsequent frames, the relative displacement on the image plane is determined by comparing the target's current position to its initial position. Considering that Raspberry Vision is specifically designed for long-span bridge measurements, the camera is equipped with a long-focus industry lens that exhibits minimal distortion (Fryer et al., 1994). Given the requirement for real-time processing, the frames captured during measurement are not corrected for distortion. Instead, we employ a straightforward camera calibration method involving a scale factor *SF* to translate pixel displacement into physical displacement, which can be formulated as follows (D. Feng et al., 2015):

$$SF = (e \times r) \times \frac{D}{f\cos^2\theta} \tag{12}$$

where $e$ is the physical dimension of a pixel on the camera sensor (1.55 $\mu$m); $r$ denotes the super-sampling rate, defined as the ratio of the camera's original horizontal resolution of 4056 pixels to its adjusted resolution used in Raspberry Vision, typically set at 1280 pixels; $f$ is the lens focal length; $\theta$ is the tilt angle between the camera's optical axis and the target; and $D$ is the distance between the target and the camera.

## 3.3 | A practical measurement strategy

Typical real-time vision-based systems generally comprise a high-performance computer, an industrial-grade camera, a tripod, and other essential peripherals. These systems are often required to be installed on stable ground, such as a riverbank, to ensure effective operation. However, this traditional setup presents three main drawbacks when measuring the displacement of long-span bridges (girders):

1. The distance between the camera and the measuring target, often encompassing the main span, side span, and approach, results in a substantial distance, which can decrease measurement accuracy.
2. Accurately measuring this long distance can be challenging, leading to potential errors in the measurement results.
3. A significant tilt angle $\theta$ often arises due to the height difference between the camera and the target on the bridge girder, complicating the application of Equation (12). Accurately determining this angle in field applications can be difficult.

To overcome the aforementioned challenges, this paper proposes a practical measurement strategy. In our approach, Raspberry Vision is mounted on the lower crossbeam or column of a bridge tower or on a bridge pier. This positioning offers three key advantages over traditional setups:

1. It significantly reduces the distance between the camera and the measuring target, as it excludes the side span and approach, thereby enhancing measurement accuracy.
2. This shorter distance can be easily and precisely determined using the bridge's design drawings.
3. The minimal height difference between the camera's position on the bridge tower or pier and the

target on the bridge girder can typically be disregarded. This simplification greatly eases the camera calibration process.

However, this measurement strategy is limited by the distribution range of the targets. Adhering to this approach, the optical axis of the camera is aligned parallel to the longitudinal direction of the bridge. Consequently, only the targets within the camera's depth of field can be captured, which inherently limits the measurement range.

### 3.4 | Comparative analysis: Previous system, Raspberry Vision, and a commercial system

Raspberry Vision has been developed to overcome certain limitations observed in the previous system, particularly in the context of field applications like long-span and cross-sea bridge measurements. A detailed comparison between these two systems is presented in Table 2. Notably, Raspberry Vision incorporates a more robust image processing algorithm and offers a camera calibration process that is both more convenient and practical for field applications. Furthermore, in conjunction with the proposed measurement strategy, it significantly reduces the distance between the camera and the measuring target. An additional advantage of Raspberry Vision is its cost-effectiveness: By eliminating the need for a laser rangefinder, it is more economical than the previous system. For broader context, Table 2 also includes a comparison with a commercial system, Dynamic Monitoring Station produced by Imetrum Ltd. (Brownjohn et al., 2017), which particularly highlights differences in cost.

## 4 | LABORATORY EXPERIMENT FOR SGMV PERFORMANCE ESTIMATION

In this section, a laboratory experiment was carried out to test the performance of SGMV and study how the values of parameters in SGMV influence its computational time and measurement accuracy.

The experimental setup is shown in Figure 4. A black ring pattern, featuring an outer diameter of 200 pixels and an inner diameter of 100 pixels, was generated on a 24-inch liquid crystal display (LCD) monitor using an LCD-based motion simulation technique (LMST; Wang, Bownjohn, et al.,2022). The pattern moved downward from its starting point by 10 positions, pausing at each position for a duration of 1 s. The distance between two consecutive positions was 5 pixels. After reaching the 10th position, the pattern retraced its path in reverse order. The entire exercise duration was 20 s. The pixel pitch of the monitor is 0.277 mm/pixel. Consequently, the true physical relative displacement between the pattern at two consecutive positions amounts to 1.385 mm, which is calculated by multiplying the pixel pitch (0.277 mm/pixel) with the pixel displacement (5 pixels). This true physical displacement of the pattern is used to evaluate the measurement accuracy of Raspberry Vision.

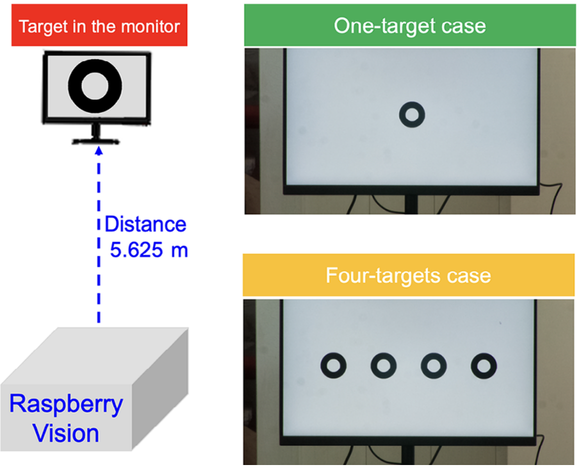


FIGURE 4 The setup of laboratory experiment.

The Raspberry Vision was positioned 5.625 m away from the LCD monitor, with its optical axis oriented perpendicularly to the monitor. The camera, equipped with a 50-mm focal length lens, was configured with a resolution of 1280 × 640 pixels and a sampling rate of 30 Hz.

We conducted camera calibration prior to the experiment to determine the scale factor. Given that the camera's optical axis was aligned perpendicularly to the monitor, the tilt angle $\theta$ was determined to be 0. The supersampling rate was calculated by dividing the original horizontal resolution of 4056 pixels by the adjusted resolution of 1280 pixels, yielding a rate of 3.169. Employing the above parameters in Equation (7), the scale factor was subsequently calculated to be $SF_{lab} = 0.553$ mm/pixel.

In employing this camera calibration method, potential sources of error include the following aspects. First, the distance between the camera and the target was measured using a laser rangefinder, which has an accuracy of approximately 2 mm. This introduces a minor margin of error in the distance measurement. Second, adjusting the camera's optical axis to be perfectly perpendicular to the target is challenging, and slight deviations might occur. However, it is important to note that these factors are unlikely to significantly impact the overall measurement accuracy.

The experiment consists of two separate tests: the first one assesses the factors that influence the SGMV computational time, while the second one examines its measurement accuracy.

**TABLE 3** SGMV computational time with different $a_1$ and $a_2$ (unit: ms).

| $a_1$<br>$a_2$ | 1.2 | 1.5 | 2 |
|---|---|---|---|
| 1.5 | 32.8 | 57.3 | 108.7 |
| 2 | 49.9 | 100.5 | 181.9 |
| 3 | 87.7 | 186 | 346.4 |
| 4 | 123.3 | 262.1 | 476.9 |

## 4.1 | The effect of the factors on SGMV computational time

EMS, as implemented in SGMV, combines a balanced risk-return criterion technique for optimizing $S_{RS}$ calculations, a local region definition to cut down the number of query windows, and multi-threaded parallel processing for handling multiple targets. The first test was performed to examine how the values of parameters in EMS influence SGMV's computational time.

The Raspberry Vision system was activated to start the measurement. In the initial frame, an ROI containing the target was selected, sized at 160 × 160 pixels. From this ROI, 468 edge points were extracted to serve as a template, with each point (or pixel) within the template acting as a voter.

### 4.1.1 | Sub-test 1: The dimension of the local region

After selecting the ROI, we determine the dimensions of the local region using parameters $a_1$ and $a_2$. An exploration into how the values of these parameters impact SGMV's computational time was conducted, with the results listed in Table 3. Notably, in this instance, the threshold $S_{RS_T}$ was set at 0, and the weight factor $p$ at 1.

When the entire image, sized at 1280 × 640 pixels, is utilized for template matching, the SGMV computational time extends to 2900.7 ms, failing to satisfy real-time processing prerequisites. In contrast, parameters $a_1$ and $a_2$ are employed to define a local region, consequently reducing the number of query windows and facilitating an increase in the SGMV computational speed. Data in Table 3 illustrate that smaller values of $a_1$ and $a_2$ are associated with a faster computational speed for SGMV. Specifically, with $a_1$ and $a_2$ set to 1.2 and 2, respectively, the computational time for SGMV is significantly reduced to a mere 49.9 ms.

In all cases above, Raspberry Vision produced the same measurement result, indicating the moved target fell into the defined local region during the measurement. To achieve the quickest computational time, $a_1$ and $a_2$ can be set to 1.2 and 2, respectively.

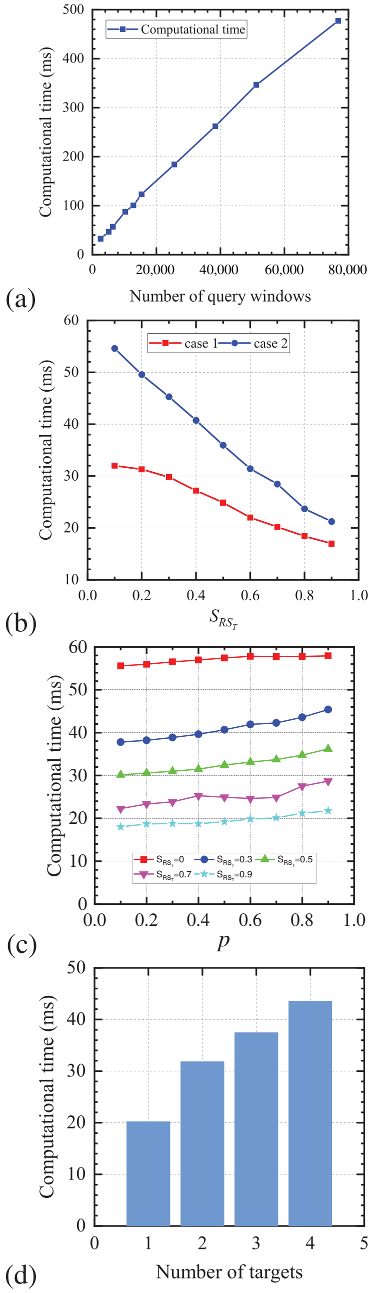


**FIGURE 5** Effect of influence factors on gradient matching via voting computational time: (a) the number of query windows, (b) $S_{RS_T}$, (c) $p$, and (d) the number of targets.

The number of query windows can be determined by the formula as follows: $(a_1 - 1) \times (a_2 - 1) \times 160^2$. Figure 5a depicts the correlation between SGMV computational time and the number of query windows, revealing a roughly inverse linear relationship. By selecting reasonable values for $a_1$ and $a_2$, while ensuring that the moved target remains within the local range, the computational speed of SGMV can be effectively enhanced.

### 4.1.2 | Sub-test 2: The value of $S_{RS_T}$

The value of $S_{RS_T}$ signifies the anticipated minimum completeness of the target during measurement. In this sub-test, we consider two cases: In the first case, $a_1$ is set to 1.2 and $a_2$ to 1.5, while in the second case, $a_1$ is set to 1.2 and $a_2$ to 2. $S_{RS_T}$ values vary from 0.1 to 0.9, incrementing in steps of 0.1. In both cases, $p$ was set as 1, indicating no weight on the high-risk criterion. The correlation between SGMV computational time and the value of $S_{RS_T}$ is shown in Figure 5b.

When a high value is assigned to $S_{RS_T}$, it is not necessary to calculate $S_{PS}$ for all pixels within the majority of query windows, thereby boosting SGMV's computational speed. With $S_{RS_T}$ set to 0.9, the computational time drops to 17.0 ms in Case 1 and 21.3 ms in Case 2, marking a reduction of 48% and 57%, respectively, when compared to scenarios where $S_{RS_T}$ is not incorporated into the calculation. This represents a more than 170-fold increase in speed, compared to GMV. However, when target completeness is low, assigning a high value to $S_{RS_T}$ might lead SGMV to mislocate the target. Hence, it is crucial to consider both computational speed and calculation correctness when setting this threshold.

### 4.1.3 | Sub-test 3

The weight factor $p$ is strategically implemented in SGMV to balance the high-risk and low-risk criteria, aiming to optimize the algorithm's computational speed. It is important to note that the value of $p$ influences not only the computational time of SGMV but also its accuracy. The sub-test conducted focused specifically on assessing computational time.

The impact of different values of $p$ on computational time, under varying $S_{RS_T}$ settings, was examined and is depicted in Figure 5c. For a given $S_{RS_T}$, there appears to be a roughly linear relationship between the computational time of SGMV and the value of $p$. A lower $p$-value increases the emphasis on the high-risk criterion, leading to reduced computational time. However, this approach might introduce potential measurement errors, particularly when the moved target is not fully recorded. This aspect, concerning the trade-off between speed and accuracy, will be further explored in Section 4.2.

### 4.1.4 | Sub-test 4: The number of targets

In practical bridge measurement applications, Raspberry Vision frequently engages in multi-target tasks, and the number of targets can influence the computational speed of SGMV. For this sub-test, four black ring targets were generated in the monitor using LMST. Raspberry Vision measured these targets simultaneously, with varying numbers of targets in different scenarios.

Figure 5d demonstrates the correlation between the number of targets and the SGMV computational time. As SGMV shifts from processing one target to two, the computational time rises from 20.2 to 31.8 ms, an increase of 11.6 ms. However, with further increments in target numbers, the corresponding increase in the computational time becomes less pronounced. Specifically, the computational time increases by merely 5.6 ms when processing three targets, compared to two, and by 6.1 ms when processing four targets, compared to three.

This diminishing increase in computational time can be attributed to SGMV's adaptive processing strategy. When dealing with a single target, SGMV employs multithreaded parallel calculations for $S_{RS}$ calculation. However, as the number of targets increases, SGMV adapts by shifting to single-core processing for each target, simultaneously allocating the remaining cores to the additional targets. This efficient resource allocation can improve the processing speed of SGMV in multi-target tasks.

The results demonstrate that the proposed speed-up strategy can significantly enhance the computational speed of SGMV, enabling it to operate in real time on a Raspberry Pi single-board computer. User assignment is required for some parameters, with their values having potential implications on not just the computational speed but also the accuracy of the calculations. This necessitates a comprehensive evaluation when considering these parameters.

## 4.2 | The effect of the factors on SGMV measurement accuracy

The threshold value $S_{RS_T}$ and the weight factor $p$ are crucial parameters that can significantly affect the computational time of SGMV. However, these factors may also contribute to incorrect localization of the moved target, especially in scenarios where the target is substantially incomplete due to dramatic illumination changes or occlusion. Such incorrectness can lead to measurement errors. To evaluate the impact of these factors on the accuracy of SGMV measurements under various conditions, this test was designed with five distinct cases.

In the first three cases, targets with varying degrees of completeness were used: 100% completeness in Case 1, 80% in Case 2, and 60% in Case 3 as illustrated in Figure 6a–c. Case 4 was designed to simulate illumination changes, wherein a fully complete target (100%) is set against a darkening background as depicted in Figure 6d. Finally, Case 5

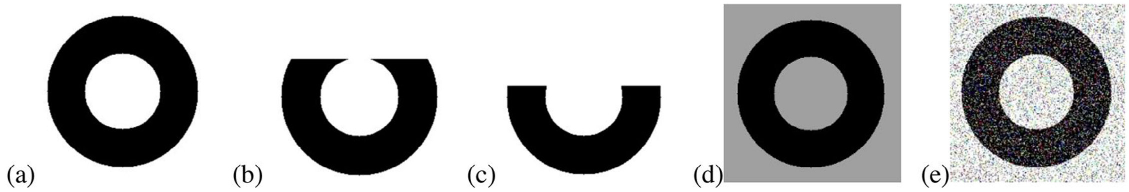


FIGURE 6 The target in different cases.

involved the addition of 30% salt and pepper noise to the simulated images displayed on the monitor as shown in Figure 6e.

In each of these cases, an ROI encompassing the target was selected in the initial frame. This ROI was a size of 160 × 160 pixels. From this specified area, a total of 468 edge points were extracted to create a template. The amplification factors were set with $a_1$ at 1.2 and $a_2$ at 2.

### 4.2.1 | Sub-test 1: The value of $S_{RS_T}$

In this sub-test, the threshold value $S_{RS_T}$ was varied from 0.1 to 0.9, incrementing in steps of 0.1. In all cases, $p$ was set to 1 to ignore its effect on the stopping criteria. The pattern displayed on the monitor was measured by Raspberry Vision at these different $S_{RS_T}$ values, and the root-mean-square error (RMSE) of the measured displacement was calculated by comparing it with the true displacement.

Figure 7a presents the RMSE of the measurement results in all five cases. In Case 1, Raspberry Vision consistently produced stable and accurate displacement results across all $S_{RS_T}$ values, with an RMSE of less than 0.036 mm. In Case 2, accurate displacement measurements were obtained when $S_{RS_T}$ was set below 0.7, also with an RMSE of 0.036 mm. However, when $S_{RS_T}$ exceeded 0.8, Raspberry Vision began to produce incorrect matches, leading to larger measurement errors. This increase in error is attributable to the target's 80% completeness in this case. Given that a local region was defined as twice the height of the ROI, the maximum error did not exceed 80 mm. In Case 3, where the target's completeness is 60%, setting $S_{RS_T}$ higher than 0.6 resulted in substantial measurement errors. Under the varying illumination conditions of Case 4 and the noise conditions of Case 5, the measurement errors slightly increased with larger $S_{RS_T}$ values. The RMSE in Case 4 was less than 0.06 mm, while in Case 5, it was less than 0.07 mm. Based on these findings, an $S_{RS_T}$ value of around 0.5 appears to be optimal to ensure reliable measurement accuracy across various conditions. Additionally, when $S_{RS_T}$ is set at 0.5, Raspberry Vision achieves a computational speed that is conducive to real-time bridge displacement measurement.

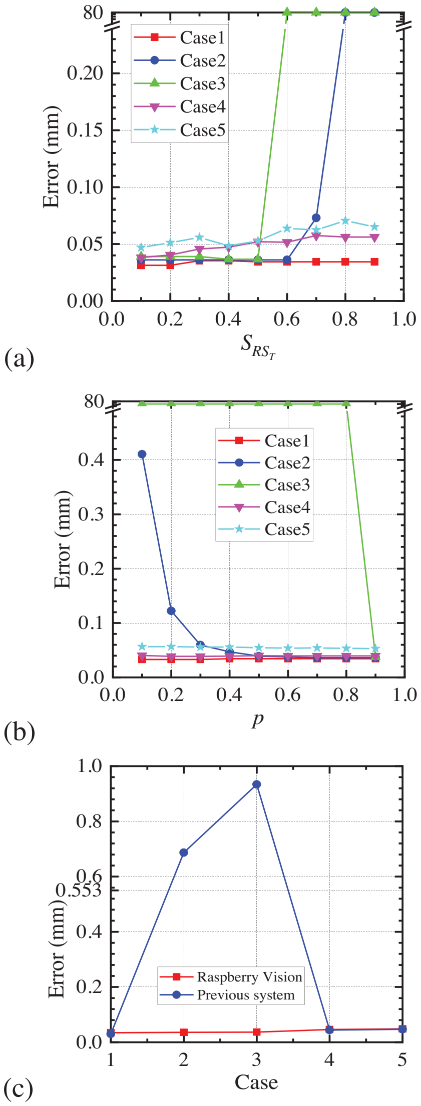


FIGURE 7 Root-mean-square error of measurement result: (a) at different values of $S_{RS_T}$, (b) at different values of $p$, and (c) between the previous system and Raspberry Vision.

### 4.2.2 | Sub-test 2: The value of $p$

In this sub-test, we examined the impact of the weight factor $p$ on measurement accuracy. The value of $p$ was varied from 0.1 to 0.9, in increments of 0.1. In all cases, the threshold $S_{RS_T}$ was set to 0.5, a value at which Raspberry Vision consistently produced accurate results in previous tests.

Figure 7b illustrates the RMSE of the measurement results across the five cases. In Case 1, Raspberry Vision achieved stable and accurate displacement measurements for all values of $p$, with an RMSE of less than 0.034 mm. However, when the target was incomplete, the value of $p$ began to influence measurement accuracy. In Case 2, for example, a $p$-value smaller than 0.2 led to a significant increase in measurement error, attributable to the amplified weight of the high-risk criterion in these scenarios. In Case 3, where the target's completeness was only 60% and $S_{RS_T}$ was set at 0.5, even a slight increase in the weight of the high-risk criterion could result in incorrect localization, causing large errors in displacement results when $p$ was set below 0.9. In the cases involving varying illumination (Case 4) and noise (Case 5), the value of $p$ did not significantly impact measurement accuracy.

Considering the results obtained in Sections 4.1 and 4.2, setting $S_{RS_T}$ at 0.5, $p$ at 0.9, $a_1$ at 1.2, and $a_2$ at 2 appears to offer an optimal balance between computational speed and measurement accuracy. Under these settings, Raspberry Vision is capable of delivering real-time measurement data at approximately 27.6 Hz (a response time of 36.2 ms), while maintaining an RMSE of less than 0.07 mm.

### 4.3 | Measurement accuracy comparison of the previous system and Raspberry Vision

In this section, we conducted a test to compare the measurement accuracy between the previous system and Raspberry Vision. Both systems were positioned with the same object–camera distance, and each employed a Lanczos interpolation algorithm to enhance integral-pixel displacement to a subpixel level. The test encompassed five cases, mirroring those in Section 4.2. For Raspberry Vision, the settings were as follows: $S_{RS_T}$ at 0.5, $p$ at 0.9, and the amplification factors $a_1$ and $a_2$ at 1.2 and 2, respectively. Figure 7c displays the RMSE of the measurement results from both systems. Notably, the figure specifically highlights the scale of 0.0553 mm, which is the value of $SF_{lab}$; any data exceeding this threshold are considered indicative of incorrect localization during the template matching process.

In Case 1, both Raspberry Vision and the previous system achieved accurate displacement results, with RMSEs of 0.034 and 0.03 mm, respectively. The minor discrepancy in accuracy is likely due to the different methods used for correlation calculation, which can affect the correlation scores for subpixel interpolation. However, in scenarios where the target was incomplete (Cases 2 and 3), the previous system failed to accurately localize the moved target, leading to larger measurement errors (RMSEs of 0.687 mm in Case 2 and 0.934 mm in Case 3, respectively). In contrast, Raspberry Vision maintained accurate and stable measurement results in these cases, with RMSEs of 0.036 mm for Case 2 and 0.037 mm for Case 3, respectively. In cases involving varying illumination (Case 4) and noise (Case 5), both systems produced accurate results. These test results indicate that while the previous system is capable of achieving satisfactory results under conditions of illumination changes and noise, it lacks robustness in scenarios with incomplete targets. On the other hand, Raspberry Vision consistently delivers accurate and stable displacement measurements across various adverse conditions, demonstrating its superiority over the previous system.

## 5 | FIELD TEST ON A LONG-SPAN SUSPENSION BRIDGE

The object–camera distance becomes significant when measuring the displacement of a long-span bridge because stable ground, such as a riverbank, is often far from the bridge's test points. This can result in decreased measurement accuracy and reliability. However, the high portability of Raspberry Vision offers a solution to this challenge; the system can be mounted directly on the bridge tower, significantly reducing the object–camera distance.

In this section, the midspan displacement of a long-span suspension bridge, Beida Suspension Bridge, is measured using Raspberry Vision, employing the proposed measurement strategy. The advantages of this strategy for long-span bridge displacement measurements are demonstrated.

### 5.1 | Test setup

The Beida Suspension Bridge is a steel truss suspension bridge located in Dalian City, China, connecting Laohutan Ocean Park and Mount Bird's Nest Park, as illustrated in Figure 8a. The main span measures 132 m, with 48 m on each side span and a width of 12 m. The bridge tower, constructed from carbon steel plate, stands at a height of 35 m, and the distance between the towers is 133 m.

The bridge tower is treated as a stationary structure in this study. To measure the vertical and horizontal displacement (HD) of the bridge girder, we installed the camera and computer components of Raspberry Vision on the surface of the tower on the Mount Bird's Nest Park side as shown in Figure 8b. These components were mounted on the horizontal surface of an L-shaped steel plate, which was magnetically affixed to the tower's surface. The camera was equipped with a 120-mm lens. The remaining elements of the system, including the monitor and power bank, were stationed on the sidewalk of the bridge deck.

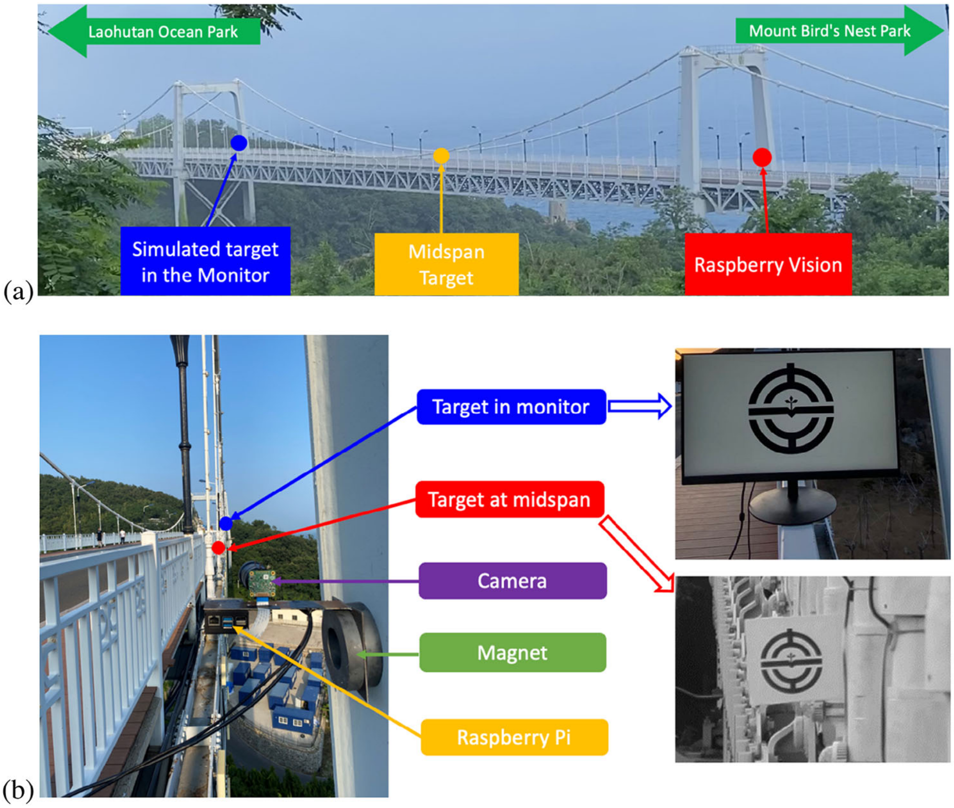


FIGURE 8 Test setup of Beida Suspension Bridge displacement measurement: (a) Beida Suspension Bridge and (b) test setup.

A wireless mouse and keyboard were employed to control the system.

A simplified logo pattern of the Dalian University of Technology (DUT) was used as a target. This pattern was adhered to the surface of an aluminum plate and installed on the midspan railing, parallel to the cross-section of the girder, as shown in Figure 8b. To evaluate the measurement accuracy of Raspberry Vision, the same pattern was simulated on a 24-inch LCD monitor using the LMST technology. The movement of the simulated DUT logo pattern in the monitor is the same as that of the laboratory experiment in Section 3. The LCD monitor was positioned atop the railing near the bridge tower on the Laohutan Ocean Park side as shown in Figure 8b. This approach for measurement accuracy estimation has been applied in the literature (Wang et al., 2023).

The settings of Raspberry Vision were as follows: $S_{RS_T}$ at 0.5, $p$ at 1, and the amplification factors $a_1$ and $a_2$ at 1.2 and 2, respectively.

## 5.2 | Measurement accuracy estimation of the simulated target in the monitor

Before conducting the experiment, camera calibration was carried out to establish the scale factor. The camera was set to a resolution of 1280 × 640 pixels and operated at a sampling rate of 30 Hz. To ensure accuracy in measurements, the camera's optical axis was carefully adjusted to be perpendicular to the monitor, thereby setting the tilt angle, $\theta$, to 0. The distance from the camera to the target during this setup was 133 m. Based on these parameters, the scale factor, $SF_{beida1}$, was calculated and determined to be 5.444 mm/pixel.

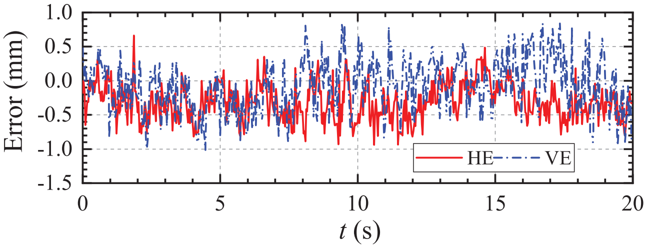


FIGURE 9 Error plot of Raspberry Vision in measuring the simulated target in the Beida Suspension Bridge test.

With no vehicles on the bridge, the monitor's position can be considered a fixed point. The displacement of the simulated target in the monitor was measured by Raspberry Vision, with the results shown in Figure 9. In this figure, *HE* represents the error in the horizontal direction, and *VE* signifies the error in the vertical direction.

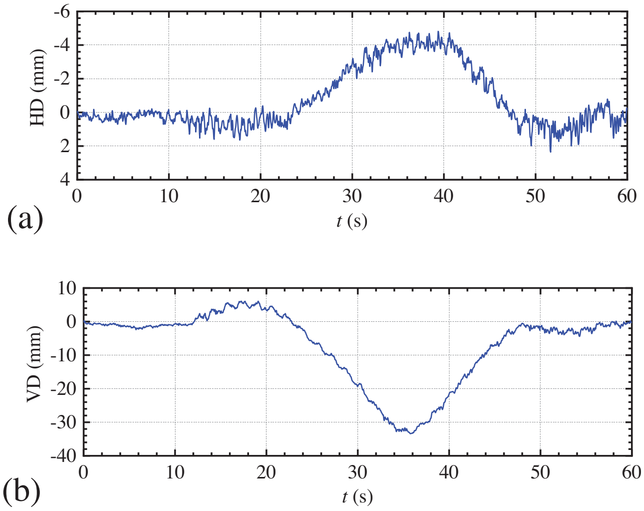


**FIGURE 10** Beida Suspension Bridge midspan's 2D displacement signals outputted from Raspberry Vision: (a) horizontal displacement (HD) and (b) vertical displacement (VD).

The measurement results reveal that most data points have an error within 1 mm, with the maximum lateral error being 0.941 mm and the maximum vertical error being 1.017 mm. RMSE of the measurement results is calculated, resulting in 0.411 mm in the horizontal direction and 0.400 mm in the vertical direction. RMSE for the entire dataset is 0.405 mm, which corresponds to 1/13 pixel in the image plane. The measurement accuracy in this case is less than that achieved in the laboratory experiment in Section 3.

## 5.3 | Measurement result of the DUT target at midspan

With camera parameters held constant, the DUT target at midspan was measured over a period of 1 min. The distance between the midspan target and the camera (the object–camera distance) was 66.5 m. The $SF_{beida2}$ is calculated to be 2.722 mm/pixel.

The HD and vertical displacement (VD) signals outputted are presented in Figure 10. During this period, a heavy truck (approximately 30 tons) traversed the bridge. The heavy truck entered the bridge at approximately 11.9 s from the Mount Bird's Nest Park side and drove toward Laohutan Ocean Park. This action amplified the VD of the main girder at the midspan and heightened horizontal vibration. Between 11.9 and 22.7 s, the truck was driving on the Mount Bird's Nest Park side span. During this period, the load was primarily concentrated on the side span, resulting in an upward deformation of the main span at the midspan. As the truck moved across the main span between 22.7 and 48.0 s, considerable deformation in both vertical and horizontal directions occurred at the midspan.

From the measurement result of the simulated target in the monitor, it can be inferred that the RMSE of the Raspberry Vision data when measuring the target at midspan should be less than 0.405 mm. This is due to the following rationale: Apart from the measurement distance, all other factors such as the measurement environment, the measured target, and camera parameters remained consistent in both tests. When measuring the midspan target, the object–camera distance was shorter, implying a higher measurement accuracy in theory.

The quality of the captured displacement signal can be estimated based on the simulated target measurement. RMSE of the midspan target result is expected to be approximately 0.411 × (66.5/133) = 0.206 mm in the horizontal direction and 0.405 × (66.5/133) = 0.203 mm in the vertical direction. The displacement range of the midspan target was 7.17 mm in the horizontal direction and 39.47 mm in the vertical direction. Consequently, the normalized RMSE (NRMSE) can be calculated as follows: 0.206/7.17 × 100% = 2.9% in the horizontal direction and 0.203/39.47 × 100% = 0.5% in the vertical direction. These calculations suggest a high-quality displacement signal in the midspan target measurement.

## 5.4 | Advantages of Raspberry Vision with the new measurement strategy

In a prior test conducted in January 2022 (Wang et al., 2023), the previous system—featuring all components integrated onto a tripod—was positioned on a ground surface rather than being mounted on the bridge tower. In both this and the previous test, the monitors that displayed the simulated targets were consistently placed at the same location, and the midspan targets were also held in a constant position. In the earlier test, the distance from the camera to the simulated target was 181.1 m, while in the current test, the object–camera distance was reduced by 26.6% to 133 m. Similarly, the distance from the camera to the midspan target was 115.2 m in the prior test, but in the current test, the object–camera distance was reduced by 42.3% to 66.5 m.

The average error of the measured displacement in this test is calculated by averaging the horizontal and vertical errors. Figure 11a illustrates the errors of the simulated target measurements for both the previous and current tests. In the figure, the errors from the previous test are computed by averaging the measurement results from two separate scenarios. RMSE of the displacement output from the prior test was 0.675 mm, while the corresponding value in the current study is 0.405 mm—demonstrating a

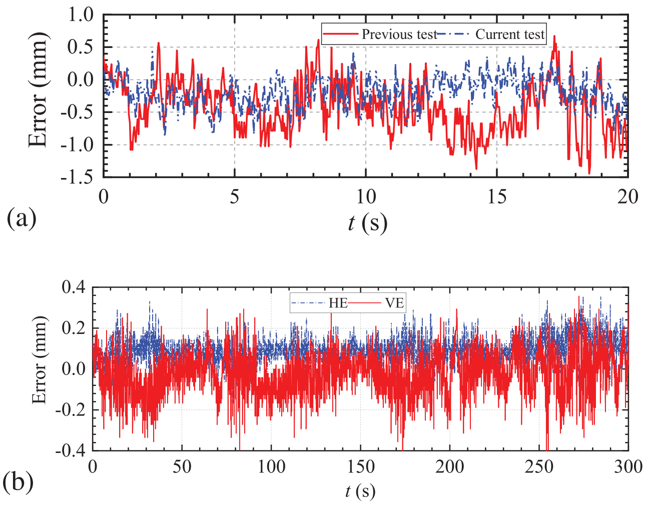


**FIGURE 11** (a) The errors of the simulated target measurements in both the previous and current tests and (b) differences between the artificial and natural targets' measurement results in the test.

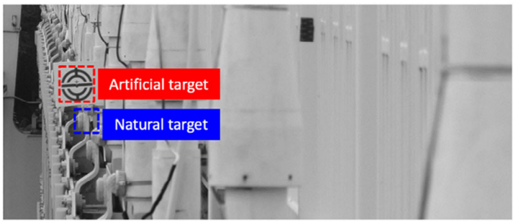


**FIGURE 12** Artificial and natural targets.

reduction of 40.0%. This signifies a significant improvement in measurement accuracy due to the application of Raspberry Vision combined with the new measurement strategy.

As for the midspan target measurements across both tests, an enhancement in measurement accuracy is anticipated in the current test, relative to the previous one. This expectation is based on the substantial reduction in object–camera distance in the current setup.

### 5.5 | Comparison between measuring different feature targets

The stability of Raspberry Vision's measurement results when tracking artificial and natural targets at the same cross-section of the bridge is assessed. As the camera was mounted on the tower surface with its optical axis parallel to the girder, selecting a suitable natural texture at midspan for measurement presents a challenge. Consequently, artificial and natural targets at the 1/4 cross-section were measured as shown in Figure 12. The difference between the two target measurements is then calculated. The differences in both vertical and horizontal directions are shown in Figure 11b, with an RMSE of 0.109 mm. These findings suggest that Raspberry Vision retains similar measurement accuracy when handling targets with different features.

### 5.6 | Displacement measurement under varying lighting conditions

Lighting conditions vary significantly over a 24-h period, posing a challenge for long-term displacement monitoring using vision-based systems. To showcase the potential of Raspberry Vision in such applications, the system was used to monitor bridge displacement for an hour during the evening. At the beginning of the measurement, the light was weak, leading to a dark frame captured by Raspberry Vision, as shown in Figure 13a. Approximately 5 min later, the decorative lights on the bridge were switched on, and the intensity and color of these lights continued to fluctuate as shown in Figure 13b,c.

Accurately tracking the target under such conditions might present a challenge when using traditional algorithms. Fortunately, SGMV in Raspberry Vision demonstrates robustness against lighting changes, ensuring the acquisition of reliable displacement signals, as shown in Figure 14.

Throughout the measurement period, no heavy vehicles passed the bridge nor were there strong winds. Consequently, the bridge's displacement response was much smaller than during instances involving a heavy truck. The HD signal appeared noisy. Most VD data were less than $SF_{beida2}$ (2.722 mm), implying that the movement of the target in the captured frames was at the subpixel level. This result illustrates that Raspberry Vision can detect subpixel displacement under significantly varying lighting conditions, thereby demonstrating the efficacy of the developed system in such challenging environments.

In order to assess the reliability of displacement signals captured by Raspberry Vision under varying lighting conditions, the power spectral density of these signals is calculated as shown in Figure 15. The analysis revealed four mode frequencies in the VD signal. Conversely, the HD did not exhibit any distinct peak, suggesting a low signal-to-noise ratio.

In a previous modal testing of this bridge conducted in January 2022 (Wang et al., 2023), four modes were identified. A comparison of the results from these two tests reveals differences listed in Table 4, all of which are less than 5%.

The discrepancies can be mainly attributed to the source of bridge vibration: A heavy truck was used to excite the

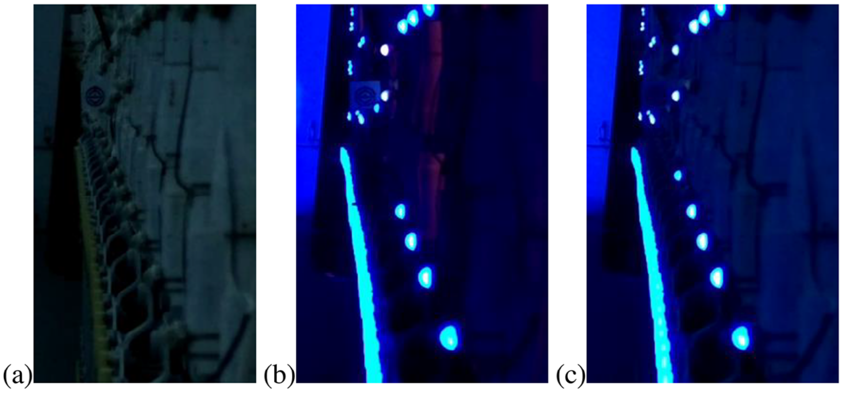


**FIGURE 13** Beida Suspension Bridge midspan displacement measurement under significantly varying lighting conditions at: (a) 4 min, (b) 6.1 min, and (c) 13.5 min.

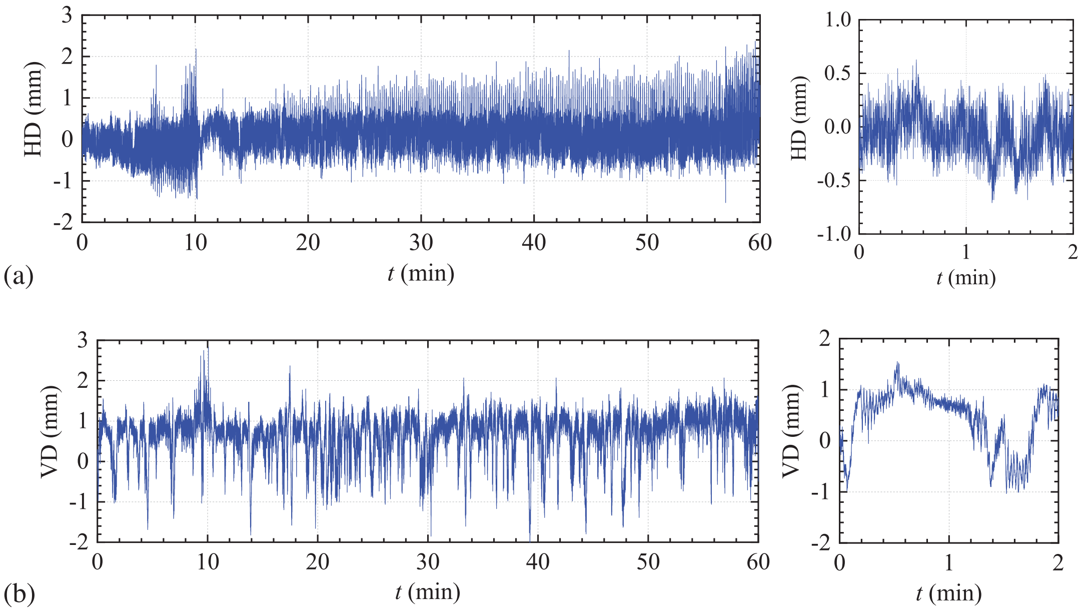


**FIGURE 14** Beida Suspension Bridge midspan displacement measured by Raspberry Vision in significantly varying lighting conditions: (a) HD and (b) VD.

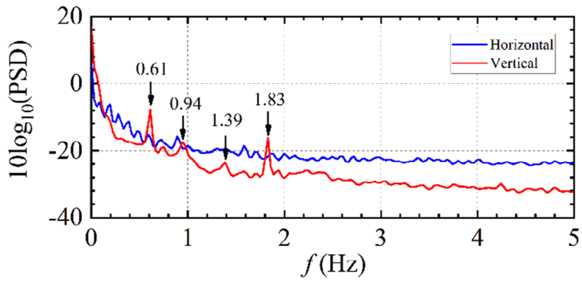


**FIGURE 15** The power spectral density (PSD) of Beida Suspension Bridge midspan target in significantly varying lighting conditions.

**TABLE 4** Comparison between the identified mode frequencies of the previous and current tests (unit: Hz).

| Mode | Previous test | Current test | Difference (%) |
|---|---|---|---|
| 1 | 0.64 | 0.61 | 4.7% |
| 2 | 0.90 | 0.94 | 4.4% |
| 3 | 1.42 | 1.39 | 2.1% |
| 4 | 1.80 | 1.83 | 1.7% |

bridge's vibration in the earlier test, while the current experiment measured the ambient vibration of the bridge. Moreover, it is important to note that the current test had a duration of 1 h, in contrast to the previous test, which lasted for 14 min. Identifying natural frequencies based on longer-duration ambient vibration signals is generally more accurate and reliable. Despite significantly varying lighting conditions during measurement, the incorporation of Raspberry Vision with the proposed measurement strategy effectively captured the ambient vibration of the bridge. This demonstrates the robustness of this approach under diverse lighting conditions.

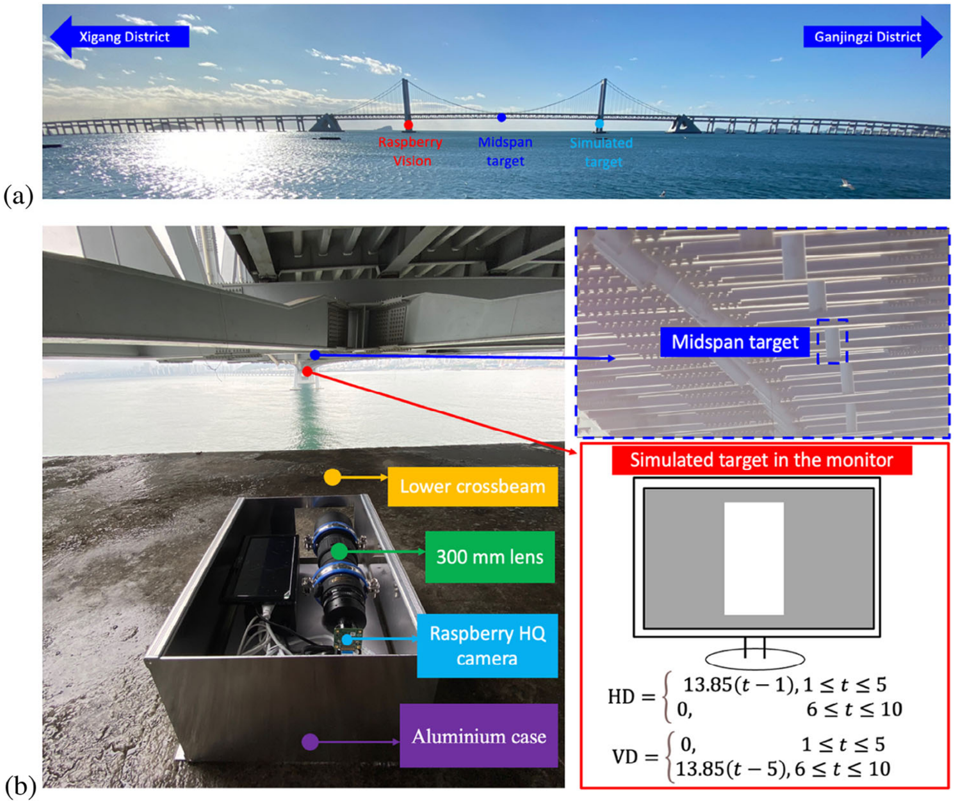

(a)
Xigang District
Ganjingzi District
Raspberry Vision
Midspan target
Simulated target
(b)
Midspan target
Simulated target in the monitor
Lower crossbeam
300 mm lens
Raspberry HQ camera
Aluminium case
$\mathrm{HD} = \begin{cases} 13.85(t-1), & 1 \le t \le 5 \\ 0, & 6 \le t \le 10 \end{cases}$
$\mathrm{VD} = \begin{cases} 0, & 1 \le t \le 5 \\ 13.85(t-5), & 6 \le t \le 10 \end{cases}$

FIGURE 16 Test setup of Xinghaiwan Cross-sea Bridge measurement: (a) The Xinghaiwan Cross-sea Bridge and (b) Raspberry Vision on the lower crossbeam of the bridge tower.

## 6 | FIELD TEST ON A CROSS-SEA BRIDGE

In the context of cross-sea bridge applications, finding a stable ground area for system hardware installation can be challenging. This poses a difficulty for real-time displacement measurements using vision-based systems. To address this, Raspberry Vision was employed in this study to monitor the displacement of a cross-sea bridge, the Xinghaiwan Cross-sea Bridge in Dalian City, thereby demonstrating its effectiveness in such applications.

### 6.1 | Test setup

The Xinghaiwan Cross-sea Bridge is a cross-sea channel connecting Ganjingzi District and Xigang District, with a total length of 6 km. The main bridge is a double-layer steel truss suspension bridge with a main span of 460 m and side spans of 180 m as shown in Figure 16a. The bridge tower is a reinforced concrete structure with a height of 114.31 m.

The lower crossbeam of the bridge tower can be considered as a stationary area during the measurement. Raspberry Vision was set up on this crossbeam to measure the midspan displacement as shown in Figure 16b. The camera was equipped with a 300-mm Nikon lens, and its resolution was configured to 1280 × 640 pixels. The optical axis of the camera was carefully adjusted to be perpendicular to the target, resulting in the tilt angle $\theta$ being 0.

Due to the impossibility of installing artificial targets at the bottom of the girder, we selected a white drainage pipe at the midspan as the measurement target as illustrated in Figure 16b. The distance between the camera and the target was 277 m, yielding $SF_{xinghaiwan1}$ of 3.718 mm/pixel.

The settings of Raspberry Vision were as follows: $S_{RS_T}$ at 0.5, $p$ at 1, and the amplification factors $a_1$ and $a_2$ at 1.2 and 2, respectively.

### 6.2 | Measurement accuracy estimation of the simulated target in the monitor

The midspan target was a white drainage pipe, approximately 160 mm in length and 82 mm in diameter. In the captured frames, its shape appeared as a white rectangle. Using the LMST technique, we simulated a white rectangle on a gray background in the monitor. The physical dimensions of the rectangle were 157.89 mm in height and 83.10 mm in width, closely mirroring the actual measured target size, as shown in Figure 16b. The simulated

**TABLE 5** Comparison of measurement accuracy in three tests.

| Tests | Object–camera distance | Lens | Scale factor | RMSE |
|---|---|---|---|---|
| Laboratory | 5.625 m | 50 mm | 0.553 mm/pixel | 0.034 mm (1/16 pixel) |
| Beida Bridge | 133 m | 120 mm | 5.444 mm/pixel | 0.405 mm (1/13 pixel) |
| Xinghaiwan Bridge | 454 m | 300 mm | 7.433 mm/pixel | 1.390 mm (1/5 pixel) |

Abbreviation: RMSE, root-mean-square error.

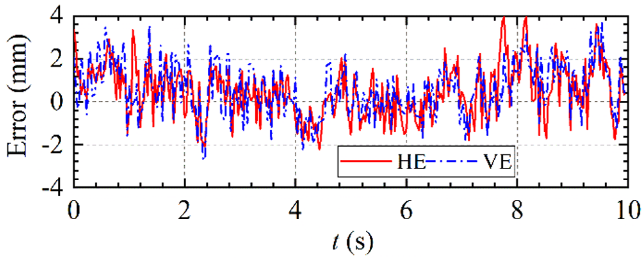


**FIGURE 17** The measurement errors of the simulated target in the Xinghaiwan Cross-sea Bridge test.

rectangle experienced 10 coordinates, and its HD and VD were computed using the formula provided in Figure 16b.

The monitor was placed on the lower crossbeam of another bridge tower, at an object–camera distance of 454 m. The scaling factor is calculated as $SF_{xinghaiwan2} = 7.433$. The camera's optical axis was set perpendicular to the monitor. Raspberry Vision was used to measure the simulated target in real time, and the measurement errors are illustrated in Figure 17.

In the figure, the maximum error identified is 3.97 and 3.71 mm in the horizontal and vertical directions, respectively. RMSE of the measurement results is computed to be 1.375 mm in the horizontal direction and 1.404 mm in the vertical direction, equivalent to 1/5 pixel in both directions. This represents a decline in measurement accuracy in comparison to the results from the Beida Suspension Bridge test. Several factors contribute to this decrease: The longer object–camera distance combined with high-humidity air can lead to significant non-uniformities in the air, resulting in minor distortions of the target in the captured frames. This phenomenon aligns with findings from the experiments conducted by Luo et al. (2020).

## 6.3 | Discussion of the influence factors on measurement accuracy

The measurement accuracy of Raspberry Vision was evaluated through a laboratory experiment, the Beida Bridge test, and the Xinghaiwan Cross-sea Bridge test. Table 5 presents key details such as the object–camera distance, the focal length of the camera lens, scale factors, and the corresponding RMSE for each of these tests. For the laboratory experiment, the measurement result of Case 1 in Section 4.2.1 is selected for comparison here. The results indicate a decrease in measurement accuracy with an increase in object–camera distance. This decline can be attributed to several factors. Primarily, the longer distance to the object, especially in conditions of high humidity, can cause significant non-uniformities in the air. This results in minor distortions of the target in the captured frames. Such a phenomenon is consistent with the findings reported in the experiments conducted by Luo et al.

## 6.4 | Measurement result of the midspan target

The measurement was conducted on the morning of June 15, 2022. Using Raspberry Vision, the white drainage pipe at the midspan was monitored starting at 09:29:36 a.m. and continued for a duration of 100 min. The HD and VD of the midspan are shown in Figure 18.

From the measurement data, a gradual deflection over time at the midspan of the bridge can be observed. This phenomenon can be attributed to a key factor: The monitoring commenced in the morning, and as the day progressed, the temperature consistently increased. This rise in temperature led to the thermal expansion of the steel girder ridge. Given the girder's pre-camber—wherein the midspan elevation is higher than both ends—this thermal expansion resulted in a downward displacement at the midspan. The result shows that the obtained displacement signals are consistent with the material properties and mechanical laws governing the bridge, thereby demonstrating the effectiveness of Raspberry Vision in the cross-sea bridge measurements.

The quality of the captured midspan displacement signal can be assessed. RMSE of the midspan target result is anticipated to be approximately $1.375 \times (277/454) = 0.688$ mm in the horizontal direction and $1.404 \times (277/454) = 0.702$ mm in the vertical direction. The range of the HD signal was determined to be 4.69 mm, while the vertical signal measured 13.63 mm. As a result, NRMSE of the midspan target displacement signals can be calculated as follows: $(0.688/4.69) \times 100\% = 14.7\%$ in the horizontal direction and $(0.702/13.63) \times 100\% = 5.2\%$ in the vertical direction. Although the recorded HD exhibited a

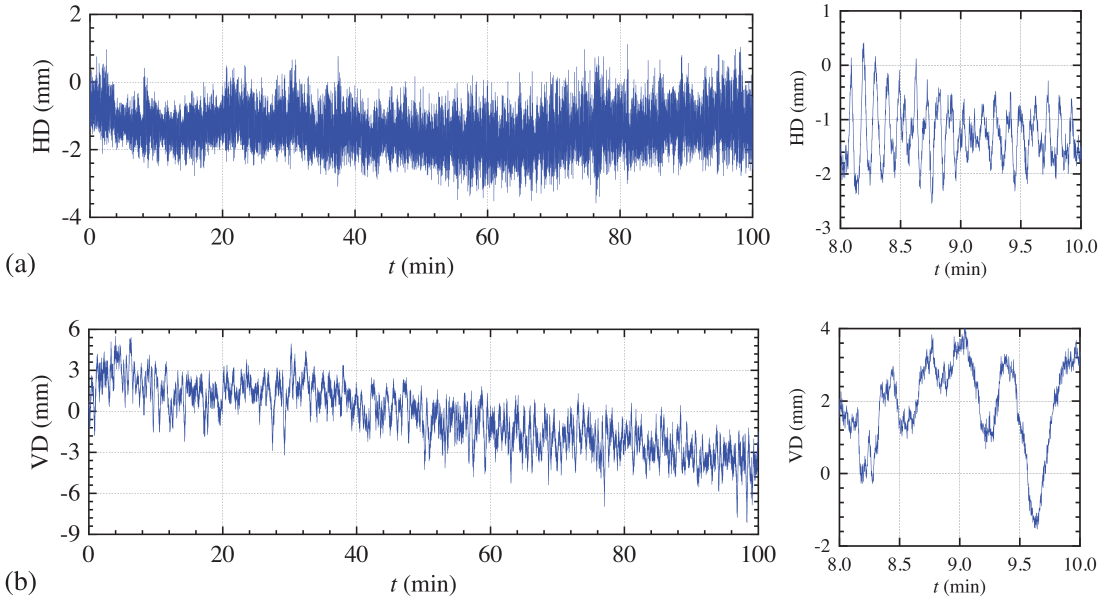


FIGURE 18 Measurement result of midspan target of Xinghaiwan Cross-sea Bridge: (a) HD and (b) VD.

degree of noise, the quality of the VD was satisfactory for infrastructure measurement.

## 7 | CONCLUSION

This study presents a new algorithm called efficient match slimmer (EMS), aimed at reducing the computational demands of advanced template matching algorithms. By integrating EMS with gradient matching via voting (GMV) algorithm, we have created a compact and low-cost vision-based system, Raspberry Vision, for real-time bridge displacement measurement. The main conclusions drawn from this research are as follows:

1. The compactness and portability of Raspberry Vision facilitate a practical measurement strategy for long-span and cross-sea bridge displacement measurement.
2. The proposed measurement strategy not only minimizes the camera-to-target distance but also simplifies the camera calibration process in bridge displacement monitoring, thereby enhancing measurement accuracy.
3. To achieve a balance between computational time and measurement accuracy in Raspberry Vision, it is recommended to set the system parameters as follows for bridge applications: $S_{RS_T}$ at 0.5, $p$ at 0.9, $a_1$ at 1.2, and $a_2$ at 2.0.
4. Compared to the previously developed system, Raspberry Vision exhibits more cost-effective features and shows enhanced applicability in challenging measurement scenarios, particularly in long-span and cross-sea bridge applications.

## ACKNOWLEDGMENTS

The authors are grateful for the financial support of the National Science Fund for Distinguished Young Scholars (Grant Number: 52125805) and the Engineering and Physical Sciences Research Council (EPSRC), UK, via the Programme Grant EP/W005816/1.

## REFERENCES

Bajwa, R., Coleri, E., Rajagopal, R., Varaiya, P., & Flores, C. (2020). Pavement performance assessment using a cost-effective wireless accelerometer system. *Computer-Aided Civil and Infrastructure Engineering*, *35*(9), 1009–1022.

Bidez, M., Lemons, J., & Isenberg, B. (1986). Displacements of precious and nonprecious dental bridges utilizing endosseous implants as distal abutments. *Journal of Biomedical Materials Research*, *20*(6), 785–797.

Boothby, T. E., Domalik, D. E., & Dalal, V. A. (1998). Service load response of masonry arch bridges. *Journal of Structural Engineering*, *124*(1), 17–23.

Brownjohn, J. M. W., Xu, Y., & Hester, D. (2017). Vision-based bridge deformation monitoring. *Frontiers in Built Environment*, *3*, 23.

Bunce, A., Hester, D., Taylor, S., Brownjohn, J., Huseynov, F., & Xu, Y. (2023). A robust approach to calculating bridge displacements from unfiltered accelerations for highway and railway bridges. *Mechanical Systems and Signal Processing*, *200*, 110554.

Feng, D., & Feng, M. Q. (2016). Vision-based multipoint displacement measurement for structural health monitoring. *Structural Control and Health Monitoring*, *23*(5), 876–890.

Feng, D., Feng, M. Q., Ozer, E., & Fukuda, Y. (2015). A Vision-based sensor for noncontact structural displacement measurement. *Sensors*, *15*(7), 16557–16575.

Feng, K., Casero, M., & González, A. (2023). Characterization of the road profile and the rotational stiffness of supports in a bridge based on axle accelerations of a crossing vehicle. *Computer-Aided Civil and Infrastructure Engineering*, *38*(14), 1935–1954.

Fryer, J., Clarke, T., & Chen, J. (1994). Lens distortion for simple C-mount lenses. *International Archives of Photogrammetry and Remote Sensing*, *30*, 97–101.

Fukuda, Y., Feng, M. Q., Narita, Y., Kaneko, S., & Tanaka, T. (2013). Vision-based displacement sensor for monitoring dynamic response using robust object search algorithm. *IEEE Sensors Journal*, *13*(12), 4725–4732.

Ge, L., Koo, K. Y., Wang, M., Brownjohn, J., & Dan, D. (2023). Bridge damage detection using precise vision-based displacement influence lines and weigh-in-motion devices: Experimental validation. *Engineering Structures*, *288*, 116185.

Guo, T., Liu, J., Zhang, Y., & Pan, S. (2015). Displacement monitoring and analysis of expansion joints of long-span steel bridges with viscous dampers. *Journal of Bridge Engineering*, *20*(9), 04014099.

Hampshire, T. A., & Adeli, H. (2000). Monitoring the behavior of steel structures using distributed optical fiber sensors. *Journal of Constructional Steel Research*, *53*(3), 267–281.

He, Z., Li, W., Salehi, H., Zhang, H., Zhou, H., & Jiao, P. (2022). Integrated structural health monitoring in bridge engineering. *Automation in Construction*, *136*, 104168.

Hoag, A., Hoult, N., & Take, A. (2017). Assessment of a bascule lift bridge using digital image correlation. *Proceedings of the Institution of Civil Engineers—Bridge Engineering*, *170*(3), 168–180.

Hoskere, V., Park, J.-W., Yoon, H., & Spencer, B. F. Jr. (2019). Vision-based modal survey of civil infrastructure using unmanned aerial vehicles. *Journal of Structural Engineering*, *145*(7), 04019062.

Hu, B., Chen, W., Zhang, Y., Yin, Y., Yu, Q., Liu, X., & Ding, X. (2023). Vision-based multi-point real-time monitoring of dynamic displacement of large-span cable-stayed bridges. *Mechanical Systems and Signal Processing*, *204*, 110790.

Javadinasab Hormozabad, S., Gutierrez Soto, M., & Adeli, H. (2021). Integrating structural control, health monitoring, and energy harvesting for smart cities. *Expert Systems*, *38*(8), e12845.

Jeong, J. H., & Jo, H. (2022). Real-time generic target tracking for structural displacement monitoring under environmental uncertainties via deep learning. *Structural Control and Health Monitoring*, *29*(3), e2902.

Jung, J., Kim, D.-J., Palanisamy Vadivel, S. K., & Yun, S.-H. (2019). Long-term deflection monitoring for bridges using X and C-band time-series SAR interferometry. *Remote Sensing*, *11*(11), 1258.

Kim, K.-H., & Jung, H.-K. (2022). Development of a remote displacement measuring laser system for bridge inspection. *Sensors*, *22*(5), 1963.

Le, H. V., & Nishio, M. (2019). Structural change monitoring of a cable-stayed bridge by time-series modeling of the global thermal deformation acquired by GPS. *Journal of Civil Structural Health Monitoring*, *9*(5), 689–701.

Liu, C., Yuan, Y., & Zhang, M. (2016). Uncertainty analysis of displacement measurement with Imetrum Video gauge. *ISA Transactions*, *65*, 547–555.

Liu, F., Gao, S., & Chang, S. (2021). Displacement estimation from measured acceleration for fixed offshore structures. *Applied Ocean Research*, *113*, 102741.

Luo, L., & Feng, M. Q. (2018). Edge-enhanced matching for gradient-based computer vision displacement measurement. *Computer-Aided Civil and Infrastructure Engineering*, *33*(12), 1019–1040.

Luo, L., Feng, M. Q., & Wu, J. (2020). A comprehensive alleviation technique for optical-turbulence-induced errors in vision-based displacement measurement. *Structural Control and Health Monitoring*, *27*(3), e2496.

Luo, L., Feng, M. Q., & Wu, Z. Y. (2018). Robust vision sensor for multi-point displacement monitoring of bridges in the field. *Engineering Structures*, *163*, 255–266.

Lydon, D., Lydon, M., Taylor, S., Del Rincon, J. M., Hester, D., & Brownjohn, J. (2019). Development and field testing of a vision-based displacement system using a low cost wireless action camera. *Mechanical Systems and Signal Processing*, *121*, 343–358.

Ma, Z., Choi, J., & Sohn, H. (2022). Real-time structural displacement estimation by fusing asynchronous acceleration and computer vision measurements. *Computer-Aided Civil and Infrastructure Engineering*, *37*(6), 688–703.

Ma, Z., Choi, J., & Sohn, H. (2023). Continuous bridge displacement estimation using millimeter-wave radar, strain gauge and accelerometer. *Mechanical Systems and Signal Processing*, *197*, 110408.

Msaewe, H. A., Psimoulis, P. A., Hancock, C. M., Roberts, G. W., & Bonenberg, L. (2021). Monitoring the response of Severn Suspension Bridge in the United Kingdom using multi-GNSS measurements. *Structural Control and Health Monitoring*, *28*(11), e2830.

Nakamura, S.-I. (2000). GPS measurement of wind-induced suspension bridge girder displacements. *Journal of Structural Engineering*, *126*(12), 1413–1419.

Narazaki, Y., Gomez, F., Hoskere, V., Smith, M. D., & Spencer, B. F. (2021). Efficient development of vision-based dense three-dimensional displacement measurement algorithms using physics-based graphics models. *Structural Health Monitoring*, *20*(4), 1841–1863.

Oh, B. K., Kim, K. J., Kim, Y., Park, H. S., & Adeli, H. (2017). Evolutionary learning based sustainable strain sensing model for structural health monitoring of high-rise buildings. *Applied Soft Computing*, *58*, 576–585.

Pan, B., Xie, H., & Wang, Z. (2010). Equivalence of digital image correlation criteria for pattern matching. *Applied Optics*, *49*(28), 5501–5509.

Pan, X., Yang, T., Xiao, Y., Yao, H., & Adeli, H. (2023). Vision-based real-time structural vibration measurement through deep-learning-based detection and tracking methods. *Engineering Structures*, *281*, 115676.

Park, S., Park, H. S., Kim, J. H., & Adeli, H. (2015). 3D displacement measurement model for health monitoring of structures using a motion capture system. *Measurement*, *59*, 352–362.

Pehlivan, H. (2022). Identification of structural displacements utilizing concurrent robotic total station and GNSS measurements. *Smart Structures and Systems*, *30*(4), 411–420.

Perez-Ramirez, C. A., Amezquita-Sanchez, J. P., Valtierra-Rodriguez, M., Adeli, H., Dominguez-Gonzalez, A., & Romero-Troncoso, R. J. (2019). Recurrent neural network model with Bayesian training and mutual information for response prediction of large buildings. *Engineering Structures*, *178*, 603–615.

Pezeshki, H., Adeli, H., Pavlou, D., & Siriwardane, S. C. (2023). State of the art in structural health monitoring of offshore and marine structures. *Proceedings of the Institution of Civil Engineers—Maritime Engineering*, *176*(1), 89–108.

Pezeshki, H., Pavlou, D., Adeli, H., & Siriwardane, S. C. (2023). Modal analysis of offshore monopile wind turbine: An analytical solution. *Journal of Offshore Mechanics and Arctic Engineering*, *145*(1), 010907.

Shao, Y., Li, L., Li, J., Li, Q., An, S., & Hao, H. (2023). Monocular vision based 3D vibration displacement measurement for civil engineering structures. *Engineering Structures*, *293*, 116661.

Shen, N., Chen, L., & Chen, R. (2023). Multi-route fusion method of GNSS and accelerometer for structural health monitoring. *Journal of Industrial Information Integration*, *32*, 100442.

Shuai, J., Zhao, J., Lei, L., Zeng, P., Wu, X., & Sun, L. (2018). Accelerate multi-thread path-dependent digital image correlation by minimizing thread competition for real-time deformation measurement. *Optics and Lasers in Engineering*, *111*, 98–107.

Su, X., Kang, H., Chen, J., Guo, T., Sun, C., & Zhao, Y. (2019). Experimental study on in-plane nonlinear vibrations of the cable-stayed bridge. *Nonlinear Dynamics*, *98*, 1247–1266.

Tian, L., & Pan, B. (2016). Remote bridge deflection measurement using an advanced video deflectometer and actively illuminated led targets. *Sensors*, *16*(9), 1344.

Shajihan, S. A. V., Hoang, T., Mechitov, K., & Spencer, B. F., Jr. (2022). Wireless SmartVision system for synchronized displacement monitoring of railroad bridges. *Computer-Aided Civil and Infrastructure Engineering*, *37*(9), 1070–1088.

Wang, M., Ao, W. K., Bownjohn, J., & Xu, F. (2022a). A novel gradient-based matching via voting technique for vision-based structural displacement measurement. *Mechanical Systems and Signal Processing*, *171*, 108951.

Wang, M., Bownjohn, J., Xu, F., & Ma, Z. (2022). A novel in-plane displacement signal generation technique for testing the measurement accuracy of vision-based displacement system. *Experimental Techniques*, *47*, 921–927.

Wang, M., Ao, W. K., Bownjohn, J., & Xu, F. (2022b). Completely non-contact modal testing of full-scale bridge in challenging conditions using vision sensing systems. *Engineering Structures*, *272*, 114994.

Wang, M., Koo, K.-Y., Liu, C., & Xu, F. (2023). Development of a low-cost vision-based real-time displacement system using Raspberry Pi. *Engineering Structures*, *278*, 115493.

Wang, M., Xu, F., Xu, Y., & Brownjohn, J. (2022). A robust subpixel refinement technique using self-adaptive edge points matching for vision-based structural displacement measurement. *Computer-Aided Civil and Infrastructure Engineering*, *38*(5), 562–579.

Weng, Y., Shan, J., Lu, Z., Lu, X., & Spencer, B. F. Jr. (2021). Homography-based structural displacement measurement for large structures using unmanned aerial vehicles. *Computer-Aided Civil and Infrastructure Engineering*, *36*(9), 1114–1128.

Xu, Y., Brownjohn, J., & Kong, D. L. (2018). A non-contact vision-based system for multipoint displacement monitoring in a cable-stayed footbridge. *Structural Control & Health Monitoring*, *25*(5), 23.

Xu, Y., & Brownjohn, J. M. W. (2018). Review of machine-vision based methodologies for displacement measurement in civil structures. *Journal of Civil Structural Health Monitoring*, *8*(1), 91–110.

Xu, Y., Brownjohn, J. M. W., Hester, D., & Koo, K. Y. (2017). Long-span bridges: Enhanced data fusion of GPS displacement and deck accelerations. *Engineering Structures*, *147*, 639–651.

Yang, A., Wang, P., & Yang, H. (2021). Bridge dynamic displacement monitoring using adaptive data fusion of GNSS and accelerometer measurements. *IEEE Sensors Journal*, *21*(21), 24359–24370.

Ye, X., Ni, Y., Wai, T., Wong, K., Zhang, X., & Xu, F. (2013). A vision-based system for dynamic displacement measurement of long-span bridges: Algorithm and verification. *Smart Structures and Systems, An International Journal*, *12*(4), 363–379.

Yoon, H., Shin, J., & Spencer, B. F. Jr. (2018). Structural displacement measurement using an unmanned aerial system. *Computer-Aided Civil and Infrastructure Engineering*, *33*(3), 183–192.

Yu, Q., Yin, Y., Zhang, Y., Chen, W., Hu, B., & Liu, X. (2023). Displacement measurement of large structures using nonoverlapping field of view multi-camera systems under six degrees of freedom ego-motion. *Computer-Aided Civil and Infrastructure Engineering*, *38*(11), 1483–1503.

Zhang, H., Zhu, Y., Xiong, W., & Cai, C. (2023). Point cloud registration methods for long-span bridge spatial deformation monitoring using terrestrial laser scanning, *Structural Control and Health Monitoring*, *2023*, 2629418.

Zhang, J., Zhou, L., Tian, Y., Yu, S., Zhao, W., & Cheng, Y. (2022). Vortex-induced vibration measurement of a long-span suspension bridge through noncontact sensing strategies. *Computer-Aided Civil and Infrastructure Engineering*, *37*(12), 1617–1633.

Zhao, W., Zhang, G., & Zhang, J. (2020). Cable force estimation of a long-span cable-stayed bridge with microwave interferometric radar. *Computer-Aided Civil and Infrastructure Engineering*, *35*(12), 1419–1433.

Zhao, Y., & Yu, C. (2020). Instantaneous dynamic deformation monitoring on Jiaozhou Bay Cross-Sea Bridge with LVR-GPS technology. *Journal of Physics: Conference Series, IOP Publishing*, *1626*, 012105.

Zhu, H., Gao, K., Xia, Y., Gao, F., Weng, S., Sun, Y., & Hu, Q. (2020). Multi-rate data fusion for dynamic displacement measurement of beam-like supertall structures using acceleration and strain sensors. *Structural Health Monitoring*, *19*(2), 520–536.

**How to cite this article:** Wang, M., Xu, F., Koo, K.-Y., & Wang, P. (2024). Real-time displacement measurement for long-span bridges using a compact vision-based system with speed-optimized template matching. *Computer-Aided Civil and Infrastructure Engineering*, *39*, 1988–2009. https://doi.org/10.1111/mice.13177