A Robust Non-Contact Bridge Displacement Tracking Method via KCF-ORB with Scale-Space and Occlusion Handling

Xiaolin Ma*, Hongju Hu
College of Intelligent Construction, Sichuan Vocational and Technical College, Suining, 629000, China
E-mail: maxiaolinww@126.com, huhongju198604@163.com
*Corresponding author

Keywords: bridge engineering; displacement monitoring; KCF; ORB; gaussian pyramid; scale space theory

Received: March 26, 2025

With the growing demand for health monitoring in bridge engineering, non-contact displacement measurement techniques have received extensive attention. A novel bridge displacement monitoring method combining spatially constrained ORB feature extraction with kernel correlation filtering (KCF) and longshort-term tracking learning detection (TLD) algorithms is proposed in the study. The overall method is divided into two stages. First, the traditional ORB feature matching is improved by introducing spatial location constraints to enhance the accuracy of feature point detection and description. Second, the improved features are combined with the KCF tracking framework and a Gaussian pyramid (GP) is introduced to adapt to the scale change. Meanwhile, the TLD algorithm is integrated to deal with the occlusion problem to realize robust displacement tracking. The performance of the proposed method is experimentally validated on the OTB and LaSOT datasets. The results showed that with the optimal parameters (descriptor length of 256 and Gaussian scale number of 4), the feature extraction and tracking accuracy of the new model was close to 0.95, and the detection time was as short as 20ms. The tracking loss rate of the proposed model under 50% occlusion was reduced to 15%. Compared with the state-of-the-art models such as YOLOv5, Mask R-CNN, and Faster R-CNN, the proposed method performed better in terms of precision rate (92.56%), recall rate (90.11%), F1 score (91.10%), and average displacement error (0.01mm). The results show that the proposed method has higher precision, stronger robustness, and better detection efficiency in the complex bridge environment, which provides an effective and reliable technical path for bridge engineering displacement monitoring.

Povzetek: Predlagana je metoda za brezkontaktno zaznavanje pomikov mostov, ki združuje prostorsko omejeni ORB, KCF s prilagoditvijo merila (GP) in TLD za robustno sledenje kljub okluzijam.

1 Introduction

Engineering surveying is an indispensable and important link in fields such as civil engineering, construction engineering, transportation, and energy engineering. Its task is to collect and analyze information on the geometric shape, location, and dimensions of terrain, buildings, structures, and various engineering facilities through scientific and precise measurement methods [1]. Engineering surveying not only provides basic data for the planning and design of projects but also supports positioning, layout, and monitoring after completion during the construction process. In this context, health monitoring and displacement measurement of bridge engineering have gradually become important application areas based on image processing technology. As an important transportation hub, the structural health of bridges is directly related to public safety. Traditional bridge displacement measurement often uses contact sensors such as displacement sensors, accelerometers, etc. However, these methods are affected by complex bridge environments, dynamic loads, and other factors in practical applications, making it difficult to achieve efficient and long-term monitoring [2]. In recent years, non-contact displacement measurement methods based on image processing has received wide attention in bridge health monitoring. Du et al. proposed a new strategy combining template matching and feature point detection for the problem of insufficient accuracy of traditional region-of-interest measurements. This strategy was experimentally verified to have good accuracy and stability in laboratory models, high-speed railroad bridges, and cable-stayed bridges [3]. Colombani and Andrawes improved feature-based image alignment parameters and proposed a new detection method that maintains measurement accuracy within the error range of traditional displacement sensors. The error of this method during laboratory testing was only 3.3% [4]. Han et al. combined adaptive regions of interest with adaptive binarization methods to construct a new detection model to address the issues of interference and large accuracy fluctuations in visual displacement measurement. This model performed well in infrastructure testing at different scales, with high accuracy and feasibility [5]. In addition, to enhance the long-term monitoring effectiveness, Shajihan et al. developed a wireless synchronized SmartVision system, which achieved 94.33% accuracy in the real test of cable-stayed bridges, verifying the

feasibility of wireless visual monitoring [6]. For specific technical applications, kernel correlation filtering (KCF) has become a research hotspot in recent years for its efficient frequency domain target tracking capability [7-8]. Liu et al. combined KCF with log-polar coordinate transformation in ship traveling tracking control, which effectively improved the tracking robustness and accuracy [9]. Oriented-fast and rotated brief (ORB), as a lightweight feature extraction and description algorithm, also shows its potential in bridge monitoring. For example, Bianchi et al. proposed rigid, deformable, and hybrid image alignment strategies based on ORB, which significantly improved the reliability of damage evolution monitoring [10]. Meanwhile, Du et al. designed a visual displacement measurement scheme adapted to different lighting environments by combining ORB with a UAV platform, which verified its efficiency and accuracy in bridge monitoring [11]. Overall, although the existing methods have made some progress in improving the accuracy of non-contact monitoring, there are still problems such as accuracy degradation and unstable tracking in complex environments, especially in coping with target scale changes and occlusion interference. In response to these shortcomings, this article proposes to improve ORB feature extraction by combining spatial constraints with KCF, Gaussian pyramid (GP), and tracking learning detection (TLD) algorithms. This article constructs a robust displacement detection framework for complex engineering scenarios to further enhance the adaptability and reliability of the system. The study gives a comparison summary table of the above different methods, as shown in Table 1.

Table 1: Indicator test results for different models.

Method	Reference	Precision (%)	Recall (%)	F1 Score (%)	Average displacement error (ADE)
Template Matching + Feature Detection	Du et al. [3]	88.75%	85.32 %	86.99%	0.08mm
Feature Image Registration Method	Colombani Andrawes [4]	94.67%	92.43 %	93.54%	0.04mm
Adaptive Region of Interest and Binarization Method	Han et al. [5]	91.20%	89.50 %	90.34%	0.10mm
Wireless Synchronized SmartVision System	Shajihan et al. [6]	94.33%	93.10 %	93.71%	0.05mm
KCF with Logarithmic Polar Coordinate Transformation	Liu et al. [9]	89.35%	87.14 %	88.23%	0.12mm
ORB Combined with Drone	Du et al. [11]	91.88%	90.12 %	90.99%	0.06mm

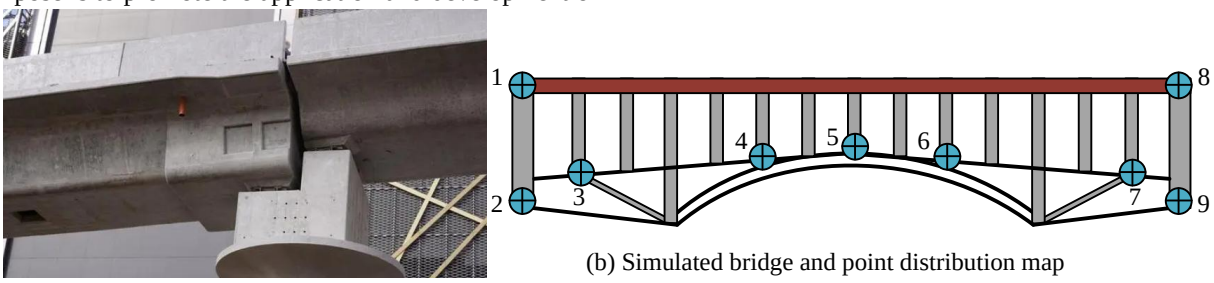
In summary, although there are various computer vision methods applied to bridge displacement detection, there are two major bottlenecks in the existing mainstream technologies. One of them is poor adaptability to changes in target scale. When the camera angle or target distance changes, the detection accuracy is prone to decrease. The second issue is that under partial target occlusion or environmental interference, the tracking robustness is insufficient, which can easily lead to loss or misidentification. These problems seriously limit the practicality and reliability of the visual displacement monitoring system in complex engineering environments. To address the above problems, this paper innovatively proposes a new bridge displacement detection method based on spatial constraints to improve ORB feature extraction, combining KCF with the long short-term TLD algorithm. The overall design is divided into two stages. Firstly, the ORB feature matching process was optimized by introducing a spatial position constraint mechanism, which improved the accuracy and anti-interference ability of feature detection and description. Secondly, the improved features are combined with the KCF framework and combined with GP to enhance the adaptability to scale changes, as well as the introduction of the TLD module to enhance the target recovery ability under occlusion conditions, thereby achieving robust displacement tracking in complex environments. In terms of methodology, this study prioritizes ORB and KCF algorithms that require less computation and are suitable for real-time applications, rather than deep detection methods that rely on large-scale data training. This study is mainly based on literature, indicating that ORB has excellent feature extraction speed and scalability, while KCF is both efficient and stable in small-scale target tracking, with flexible deployment and low cost. It should be noted that the research algorithm outperforms You Only Look Once version 5 (YOLOv5), Mask Region-Based Convolutional Neural Network (Mask R-CNN), and Faster Region-Based CNN (Faster R-CNN) in most of the metrics in the OTB and LaSOT benchmarks. Meanwhile, some of the metrics such as Average Precision (AP) values are close to the other methods in lightly loaded scenarios, so they are not completely superior to other methods in all

comparisons. Focusing on the research motivation, this paper establishes three major objectives. The first is to improve the accuracy and robustness of ORB feature matching through spatial constraint strategies. The second is to integrate KCF with GP and TLD modules to build a robust tracking system adapted to scale change and occlusion environments. The third is to systematically evaluate the performance of the improved model under different loading, occlusion, and scale change conditions. Further specific research questions are proposed, including: how much does the introduction of GP improve the tracking robustness under multi-scale image conditions; The impact of TLD module fusion on detection and restoration under occlusion; The degree of improvement in feature extraction accuracy between spatial constraint matching strategy and standard ORB. Through system modeling, experimental verification, and comparative analysis, the aim is to provide a more accurate, robust, and real-time solution for non-contact displacement monitoring in bridge engineering. The purpose is to promote the application and development of structural health monitoring based on image processing technology in engineering practice.

2 Methods and materials

2.1 Feature extraction and description of engineering images based on improved ORB

Bridge displacement monitoring is a vital component of the bridge health monitoring system. The purpose is to track the displacement of key parts of the bridge in real time, and detect and warn potential structural problems of the bridge promptly [12-13]. This study first simulates the distribution of real bridge structures and their key points. These monitoring points are mainly distributed in areas such as the support structure, mid-span, bridge deck, and connection points of the bridge that are prone to stress concentration and deformation. The simulation diagram is shown in Figure 1.

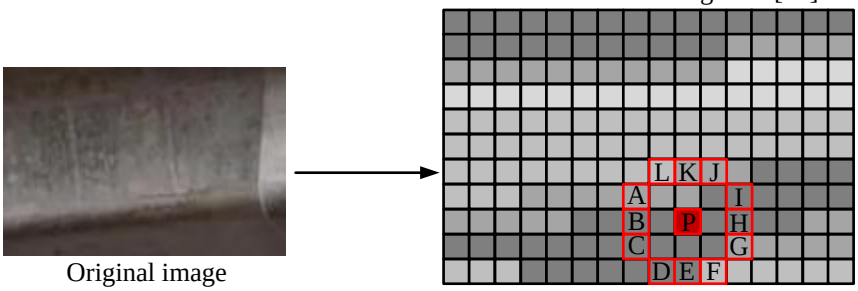


(a) Bridge displacement measurement

(b) Simulated bridge and point distribution map

Figure 1: Schematic diagram of bridge displacement simulation and distribution of monitoring points

Figure 1(a) shows the measured displacement of the bridge under actual stress conditions. The vertical displacement in the mid-span region is more obvious, showing the difference in stress distribution. Figure 1(b) shows the distribution of monitoring points of the simulated bridge. The reasonable distribution of points covers the support structure, the mid-span, and the deck area of the bridge, which can comprehensively reflect the displacement changes of the bridge under different stress conditions. Traditional contact sensors are limited by the environment and use. It is difficult to obtain accurate data for a long time and at a high frequency. This study provides a more efficient and stable solution through video monitoring technology. Traditional contact sensors are limited by environmental conditions and find it difficult to obtain accurate data over a long period of time and at high frequencies. To solve this problem, this study introduces video monitoring technology, which records the displacement changes of bridges under different load conditions through cameras, and combines image processing algorithms to automatically analyze the displacement data. Among them, ORB is a fast and robust feature extraction and matching algorithm. Compared with traditional algorithms, ORB has the advantages of low computational complexity and fast speed, mainly segmented into feature point detection and description [14-15]. Among them, the feature point detection part uses the features from the accelerated segment test (FAST) algorithm to detect corners in the image. Corner detection is shown in Figure 2 [16].



Original image

Angle point detection

Figure 2: Schematic diagram of FAST corner detection process

Figure 2 shows a schematic of the FAST corner point detection process. The working principle of this algorithm is to select a pixel point P in the image, draw a circle around it, and calculate the brightness difference between all pixels on the circle and point P. The algorithm is based on the number of pixel points in the circle. If the number of pixel points within that circle with a larger difference from point P exceeds 0.75 times the circumference of the circle, point P is considered as a key feature point. In this way, FAST corner point detection can effectively identify the significant feature points in an image and provide the basis for subsequent feature matching. The calculation formula for FAST corner detection is shown in equation (1).

$$|I(P)-I(p_i)|>T \quad (1)$$

In equation (1), $I(P)$ is the brightness value of pixel P. $I(P)$ is the brightness value of the i-th pixel on the circumference. T is the preset brightness difference threshold. The corners detected by FAST are further accelerated by ORB using an integral graph to calculate the principal direction of each corner and achieve rotational invariance, as shown in equation (2).

$$\theta=\tan^{-1}\left(\frac{\sum_y (y\cdot I(x,y))}{\sum_x (x\cdot I(x,y))}\right) \quad (2)$$

In equation (2), θ and (x,y) are the main direction and coordinate position of the key point. $I(x,y)$ is the pixel value of the image near the key point. After completing corner detection and main direction assignment, ORB uses the binary robust independent elementary features (BRIEF) algorithm to generate feature descriptors, as shown in equation (3) [17].

$$BRIEF(p_a,p_b)=\begin{cases}1 & if I(p_a)<I(p_b)\\ 0 & otherwise\end{cases} \quad (3)$$

In equation (3), p_a and p_b are the two pixels to be compared. $I(p_a)$ and $I(p_b)$ are the brightness values of pixels p_a and p_b. However, when the distribution of feature points is sparse or the scene is complex, the matching accuracy of the ORB algorithm may be affected. Therefore, this study introduced SCD to improve ORB. The core idea of SCD is to add constraints on spatial position information during the feature point matching process, ensuring that the matched feature points are not only similar in descriptors but also have a certain degree of correlation in spatial position. The improved ORB engineering image feature extraction and description process is shown in Figure 3.

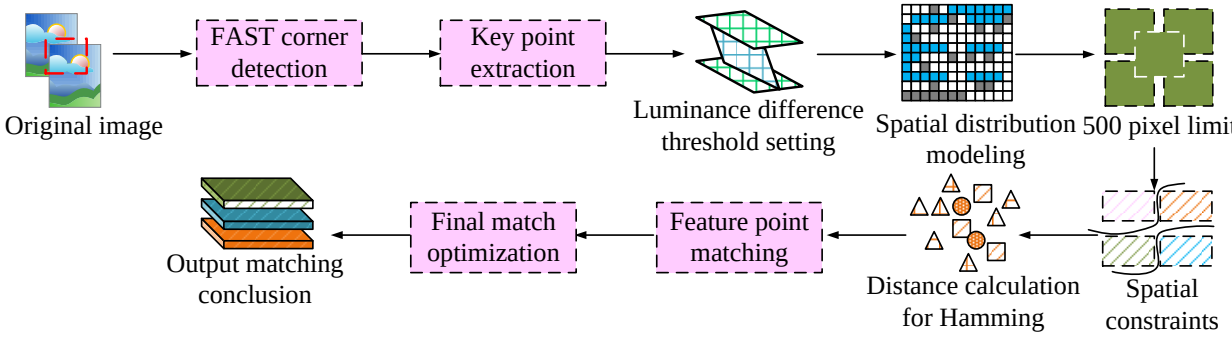


Figure 3: Improved ORB engineering image feature extraction and description process.

In Figure 3, the first step is to use FAST corner detection to extract key points from the image, and set a brightness difference threshold $T=300$ for corner detection. For the detected key points, the second step is to establish a spatial model, record their coordinate positions, and ensure that these points conform to the expected distribution within a 500-pixel range. Feature matching first uses BRIEF to generate feature descriptors, with a descriptor length set to 256 bits. By calculating the Hamming distance of descriptors, preliminary matching pairs are selected, and a matching threshold of 50 bits is set. Then, the spatial position information of the matching points is compared with the expected distribution. Matching points with a spatial distance exceeding 30 pixels will be filtered out to ensure spatial consistency. The spatial distance constraint formula is shown in equation (4).

$$d_s=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2} \quad (4)$$

In equation (4), d_s is the spatial distance between two matching points. (x_1,y_1) and (x_2,y_2) are the coordinates of the matching points in the two images. When the spatial distance d_s is less than the set threshold, such as 30 pixels, it indicates that the matching is effective. The calculation formula for measuring the similarity of feature descriptors using Hamming distance is shown in equation (5).

$$d_h=\sum_{i=1}^{n}(b_1^i\oplus b_2^i) \quad (5)$$

In equation (5), d_h is the Hamming distance between two descriptors. b_1^i and b_2^i are the i-th bits of descriptor 1 and descriptor 2. n is the length of the descriptor.

When the Hamming distance d_h is less than the set threshold, such as 50 bits, the matching is valid.

2.2 Construction of engineering displacement tracking and detection model integrating KCF-ORB

After completing the feature extraction and description of engineering images, this study introduces KCF for target tracking to meet the displacement monitoring requirements in bridge engineering. KCF has significant advantages over other tracking algorithms in fast computation speed and high tracking accuracy. The basic principle is to calculate the response map of the target in the image frequency domain through a correlation filter after initialization, and then find the optimal position of the target [18-20]. The learning formula for the relevant filter is expressed in equation (6).

$$\hat{H} = \frac{\hat{Y} \otimes \hat{X}*}{\hat{X} \otimes \hat{X}* + \lambda} \quad (6)$$

In equation (6), $\hat{H}$ is the frequency domain of the filter. $\hat{X}$ is the frequency domain feature of the target area. $\hat{Y}$ is the ideal output response diagram. λ is the regularization parameter. * is a complex conjugate. The formula for the target response graph is shown in equation (7).

$$R = \eta^{-1}(\hat{H} \oplus \hat{Z}) \quad (7)$$

In equation (7), R is the response graph of the target. η^{-1} is the inverse fast Fourier transform. $\hat{Z}$ is the frequency domain feature of the target area in the current frame. The target update formula is shown in equation (8).

$$\hat{H}_{t+1} = (1-\gamma)\hat{H}_t + \gamma \hat{H}_{new} \quad (8)$$

In equation (8), $\hat{H}_{t+1}$ is the filter for frame $t+1$. γ is the learning rate. $\hat{H}_{new}$ means the novel filter learned in the current frame. The KCF algorithm typically describes the appearance of the target object through ahistogram of oriented gradient (HOG) features. HOG only calculates gradient histograms for local grid images at a single scale, and its computational efficiency is generally poor when the scale is variable. Therefore, this study introduces the theory of scale space. This theory uses GP, which means gradually reducing the image and applying Gaussian smoothing to form multiple levels of image hierarchy, each of which can be used to track the target [21-22]. The GP structure diagram is shown in Figure 4.

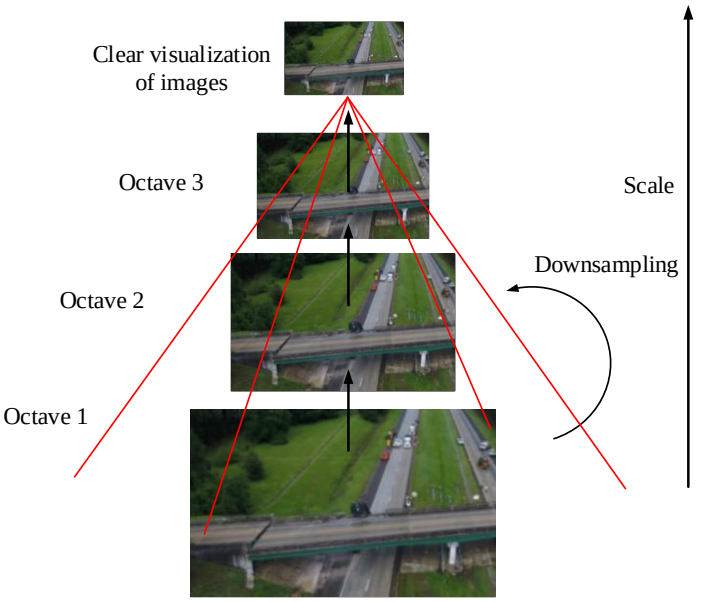


Figure 4: GP principal schematic.

In Figure 4, GP downsamples the original image multiple times to generate hierarchical images of different scales step by step. During the downsampling process, each level of theimage undergoes Gaussian smoothing to remove noise and detail information, ensuring that larger image features remain clear and visible in higher-scale images. Specifically, as the scale increases, the resolution of the image gradually decreases, and the features in the image are also gradually simplified. The GP construction formula is equation (9).

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (9)$$

In equation (9), $L(x, y, \sigma)$ denotes the image in the scale space. $G(x, y, \sigma)$ is a Gaussian kernel. σ is the standard deviation. The calculation formula for downsampling is shown in equation (10).

$$L_{down}(x, y) = L(2x, 2y, \sigma) \quad (10)$$

In equation (10), $L_{down}(x, y)$ is the downsampled image. $L(2x, 2y, \sigma)$ is the image after Gaussian smoothing, and the selected pixel coordinates are sampled every two pixels of the original image. In

addition, to solve the occlusion problem of the KCF algorithm in displacement image feature tracking and detection of bridge engineering, TLD is introduced in this study. Compared to other methods, the learning module in TLD can dynamically update the model based on changes in the appearance of the target during tracking, adapting to changes in features such as shape, brightness, and texture of the target. The KCF-TLD process after combining the two is shown in Figure 5.

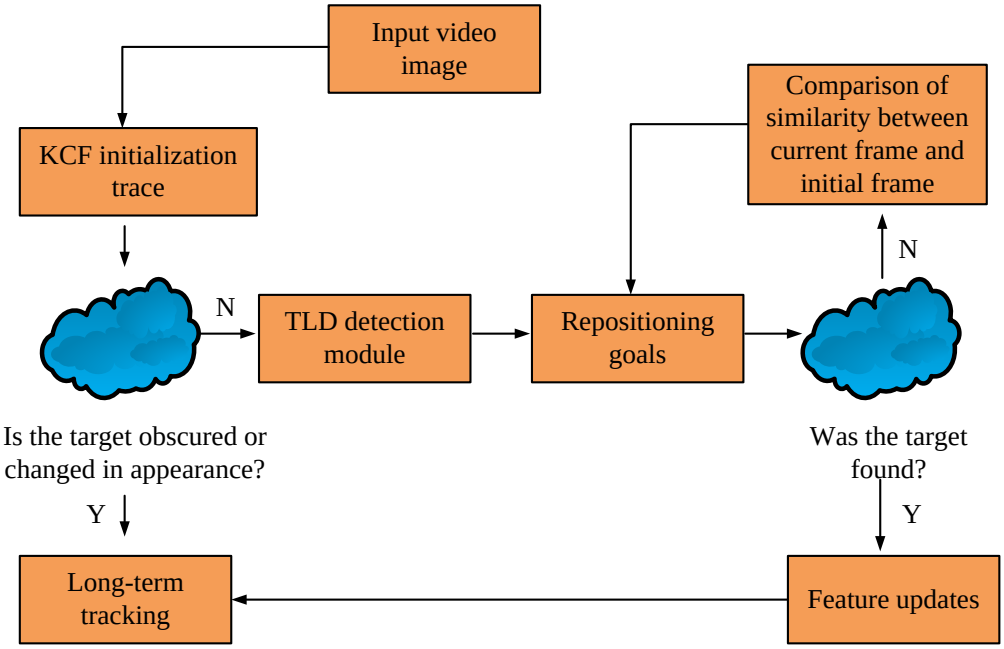


Figure 5: KCF-TLD process.

In Figure 5, step 1 is to conduct a preliminary analysis of the input video and use KCF to initialize and track the target. When there is no obstruction or obvious appearance change of the target, KCF can track the target stably and quickly. However, the KCF algorithm may fail in the presence of target scale changes or partial occlusion. For example, in some test sequences, when the target undergoes rapid motion or is partially occluded, the response value of the KCF may decrease below a threshold, resulting in tracking failure. In this case, the detection module of the TLD is triggered to relocate the target by comparing the similarity with the initial frame. The similarity is usually calculated by cosine similarity or Euclidean distance, and if the similarity is high, the target is considered to be successfully recovered. The key parameters of TLD include the learning rate, the similarity threshold, and the occlusion area threshold. In this study, the learning rate is set to 0.1, the similarity threshold is 0.8, and the occlusion area threshold is 50%.Next, if the TLD detector successfully finds the target, the TLD learning module will learn and update the appearance features of the new target to adapt to subsequent changes in the appearance of the target, ensuring the accuracy of subsequent tracking. Finally, the successfully updated target information will be fed back to KCF for stable target tracking. If the detection fails, the process will end. TLD relocates the target during the detection phase by comparing the similarity between the target in the present and the initial frames. The similarity is calculated as in equation (11).

$$S = \frac{\sum_{i=1}^{n^*}(F_{current}(i)\cdot F_{reference}(i))}{\sqrt{\sum_{i=1}^{n^*}(F_{current}(i))^2}\cdot\sqrt{\sum_{i=1}^{n^*}(F_{reference}(i))^2}} \quad (11)$$

In equation (11), S is the similarity value between the current and reference frames, that is, the initial frame. $F_{current}(i)$ is the feature vector of the target area in the current frame. $F_{reference}(i)$ is the feature vector of the target area in the reference frame. $n*$ is the dimension of the eigenvector. In addition, the expression for updating the objectives of online learning is shown in equation (12).

$$M_{new} = (1-\alpha)M_{old} + \alpha F_{current} \quad (12)$$

In equation (12), M_{new} and M_{old} are the updated and previous frame's target models. α is the learning rate. In summary, this study combines the improved ORB and improved KCF to propose a novel engineering displacement tracking and detection model, as shown in Figure 6.

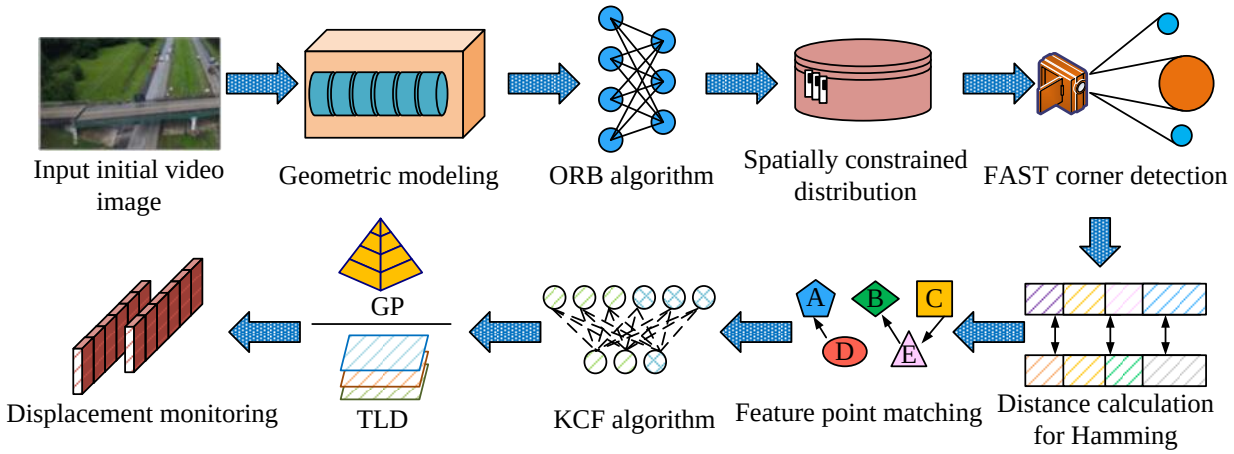


Figure 6: New engineering displacement tracking detection modeling process.

In Figure 6, the processing flow of the new model is mainly separated into 7 steps. Step 1 is to construct a geometric model of the bridge, simulating the stress and deformation of the bridge under actual operating conditions. Step 2 is to identify the key monitoring areas of the bridge, including supporting structures, spans, and bridge decks, to ensure that displacement detection covers critical areas. Step 3 is to use SCD to improve the ORB algorithm, extract key feature points from the image through FAST corner detection, and generate BRIEF descriptors. Step 4 is to calculate the Hamming distance of feature points and introduce spatial position constraints to ensure the accuracy and robustness of feature point matching. Step 5 is to use the KCF algorithm to track the extracted feature points in real time and handle the scale change problem through GP. Step 6, when KCF tracking fails or the target is occluded, the TLD module relocates the target through similarity detection. Step 7, the TLD learning module conducts online learning on changes in the appearance of the target, dynamically updates the target model, and ensures the accuracy of subsequent tracking. The pseudo-code of the algorithm for improving ORB-KCF-GP-TLD is shown in Figure 7.

```
# Step 1: ORB Feature Extraction with Spatial Constraints
def extract_orb_features(image):
    keypoints = detect_keypoints(image)     # Detect keypoints using FAST
    descriptors = compute_orb_descriptors(keypoints)  # Compute ORB descriptors
    constrained_keypoints = apply_spatial_constraint(keypoints)  # Apply spatial constraints
    return constrained_keypoints, descriptors

# Step 2: KCF Tracking
def track_with_kcf(image, target):
    tracker = initialize_kcf_tracker(image, target)  # Initialize KCF tracker
    position = track_target(tracker, image)  # Track the target
    if tracking_failed(position):
        return None  # KCF fails
    return position

# Step 3: TLD for Occlusion Recovery
def recover_with_tld(image, last_position):
    detector = initialize_tld_detector(image, last_position)  # Initialize TLD
    position = detect_target_with_tld(detector, image)  # Detect with TLD
    if position is not None:
        return position
    return None

# Step 4: GP for Scale Adjustment
def adjust_scale_with_gp(image):
    return apply_gaussian_pyramid(image)  # Generate image pyramid for scale change

# Step 5: Combine ORB, KCF, TLD, and GP
def robust_tracking(image, target):
    keypoints, descriptors = extract_orb_features(image)  # Extract ORB features
    kcf_result = track_with_kcf(image, target)  # Track with KCF

    if kcf_result is None:
        tld_result = recover_with_tld(image, target)  # Recover with TLD if KCF fails
        return tld_result if tld_result else None

    # Scale adjustment with GP
    scaled_images = adjust_scale_with_gp(image)
    for img in scaled_images:
        kcf_result = track_with_kcf(img, target)  # Re-track at different scales
        if kcf_result:
            return kcf_result

    return None  # Return None if tracking fails

# Example usage
image = load_image("bridge_frame.jpg")  # Load image
target = define_target(image)  # Define target region

# Run robust tracking
result = robust_tracking(image, target)
if result:
    print("Tracking successful:", result)
else:
    print("Tracking failed.")
```

Figure7: Algorithmic Pseudocode for Improving ORB-KCF-GP-TLD

3 Results

3.1 Performance testing of engineering displacement tracking detection model

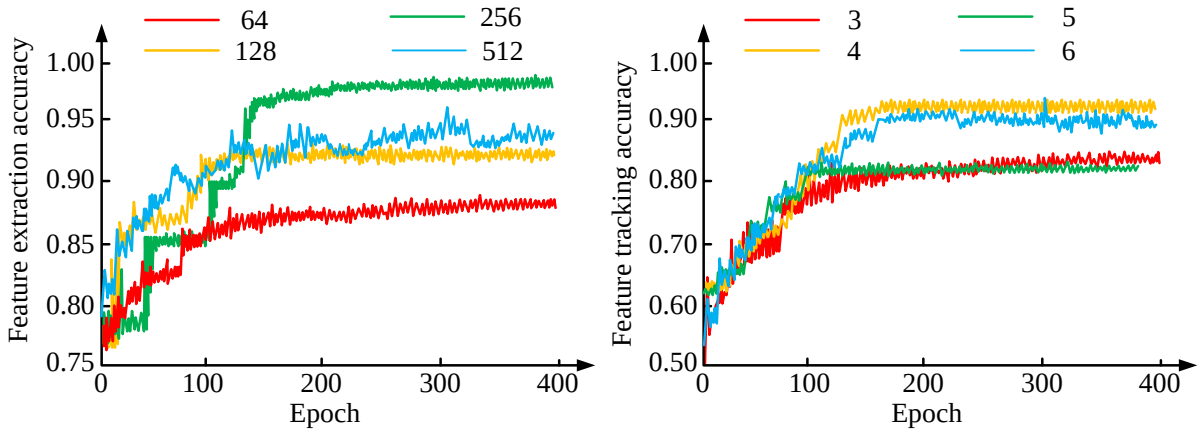
This study sets up a suitable experimental environment, with Intel Core i7-12700K CPU, NVIDIA GeForce RTX 3080 GPU, 32GB DDR4 memory, and Windows 10 64 bit operating system. The Object Tracking Benchmark (OTB) and Large-scale Single Object Tracking Dataset (LaSOT) are used as test data sources. The OTB dataset contains video sequences of 50 different scenarios, and the LaSOT dataset contains 1,400 video sequences covering a wide range of target classes and complex environments. The experimental equipment includes a 20-megapixel Basler ace acA2040-55um industrial camera equipped with a 25-mm lens, a frame rate of 30 fps, and a resolution of 1920×1080. In addition, the bridge model simulated in the experiment adopts standard industrial design. The lighting conditions are natural ambient light and adjustable artificial light sources to ensure experimental accuracy. The programming language used for the model is Python 3.8, and the OpenCV library is used for image processing. The experimental environment and parameters are shown in Table 2.

Table 2: Experimental environment and parameter situation

Parameter Symbol	Algorithm Component	Parameter Description	Setting Value/Explanation
γ (gamma)	KCF	Learning rate used for target update	Default value: 0.2, optimized selection
λ (lambda)	KCF	Regularization parameter controlling filter update smoothness	Default value: 0.01
α (alpha)	KCF	Learning rate for target response, affecting accuracy and stability	Default value: 0.01, optimized value: 0.05
T (threshold)	KCF, ORB	Matching distance threshold for target matching	KCF: 50-bit Hamming distance, ORB: 50-bit (optimized choice)
KCF Kernel	KCF	Type of kernel function used for the correlation filter	Gaussian kernel (Gaussian Kernel)
ORB Pyramid Levels	ORB	Number of pyramid levels for handling scale variation	4 levels (optimal choice based on experiments)
ORB Feature Count	ORB	Maximum number of features to extract	500 features (optimal choice for speed and accuracy)
ORB Descriptor Length	ORB	Length of ORB descriptors, affecting matching accuracy and computation speed	256 bits (optimal choice after experiment)
Gaussian Pyramid σ (sigma)	Gaussian Pyramid (GP)	Standard deviation for Gaussian smoothing in pyramid image generation	Set σ = 1.0 as the standard value
Gaussian Pyramid Levels	Gaussian Pyramid (GP)	Number of levels in the GP	4 levels (providing good scale adaptability in experiments)
ORB Spatial Constraint Range	ORB	Spatial constraint range for feature matching consistency	30 pixels (spatial constraint when matching features)

From Table 2, first, the learning rate γ of the KCF algorithm is set to 0.2, a setting that effectively balances the speed and accuracy of the target update. The regularization parameter λ is 0.01, which aims to ensure smoothness and prevent overfitting during filter updating. The learning rate α of the target response is selected to be 0.01 and optimized to be 0.05 to ensure the accuracy and stability of tracking. The distance threshold T for matching is an important parameter in the KCF and ORB algorithms. For KCF, the threshold is set to 50-bit Hamming distance, and the matching threshold for ORB is also 50-bit, which is optimized through experiments to ensure high efficiency and accuracy of target matching. For the KCF kernel function, Gaussian Kernel is chosen because it has better smoothing, which can reduce the influence of noise in the target tracking process and improve robustness. The pyramid level of the ORB algorithm is set to 4 layers, which makes the algorithm better adapt to changes in the scale of the target and improves the adaptive ability to multi-scale targets. The number of features is set to 500, which is an experimentally verified balance point that ensures speed and accuracy without over-consuming computational resources. The descriptor length of ORB is set to 256 bits, a setting that provides high matching accuracy as well as good computational efficiency. The standard deviation (σ) in the GP is set to 1.0, and the image after Gaussian smoothing can effectively remove the noise and highlight

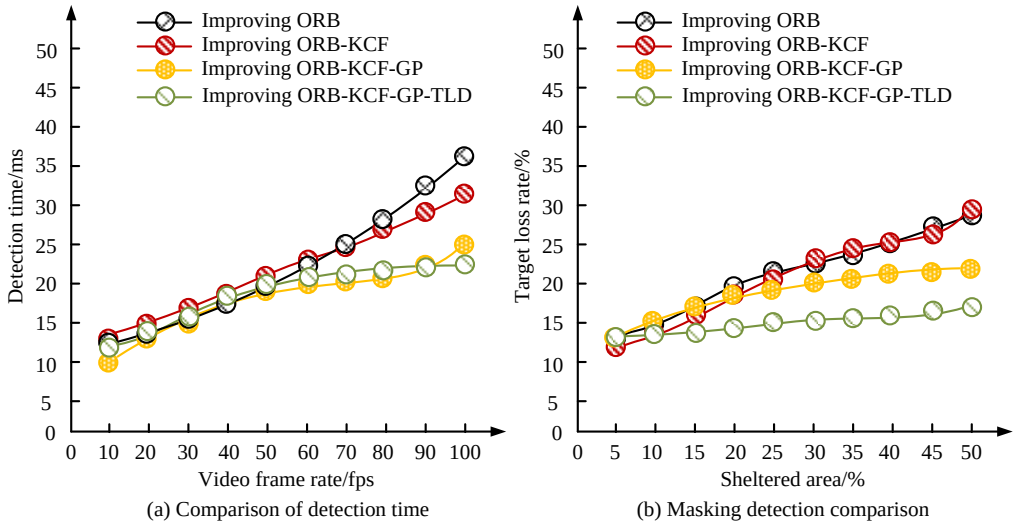
the larger scale features, which helps the algorithm to better deal with scale changes. The number of layers in the GP is set to 4, which is the optimal value chosen based on the experimental results and helps to improve the scale adaptation and image smoothing. The spatial constraint range of the ORB is set to 30 pixels, and this constraint helps to maintain the spatial consistency during the feature matching process, avoiding the mis-matching due to the positional differences. This study first conducts value selection tests on two types of hyperparameters, the BRIEF descriptor length and the GP scale number, to ensure the optimal testing performance of the final model. The result is shown in Figure 8.



(a) BRIEF descriptor length (b) Scale numbers of Gaussian pyramids

Figure 8: Hyperparameter selection test.

Figures 8 (a) and (b) show the test results of selecting values for BRIEF descriptor length and GP degree. In Figure 8 (a), when the descriptor length of BRIEF is 256, the feature extraction accuracy rapidly improves after 100 iterations and maintains high accuracy above 0.95 in subsequent training, demonstrating good stability and convergence. The descriptor with a length of 128 also achieves an accuracy close to 0.90, but its performance is slightly inferior compared to 256. The other two types of BRIEF descriptors perform poorly, especially in terms of 64-bit accuracy and stability. In Figure 8 (b), when the scale of GP is 4, the feature tracking accuracy rapidly improves and stabilizes at around 0.95 in subsequent training, demonstrating the best feature tracking performance. When the scale is 3, 4, and 5, although the accuracy is higher, the overall performance is slightly lower than when the scale is 4. In summary, the optimal length for the BRIEF descriptor is 256, and the optimal number of scales for GP is 4. Under the setting of these two types of hyperparameters, the model achieves optimal performance in feature extraction and tracking accuracy. Figure 9 displays the ablation test results of the final model. Among them, the loss rate is defined as the ratio of the number of frames in which the model fails to successfully identify the target position in the current frame during continuous tracking to the total number of frames. That is, when the intersection integral ratio (IoU) is less than 0.3, the target area is considered a tracking failure area.



(a) Comparison of detection time (b) Masking detection comparison

Figure 9: Ablation test results.

Figure 9(a) shows the results of the ablation test in terms of detection time and Figure 9(b) shows the results in terms of target loss rate. The detection time increases linearly as the video frame rate increases. At 30fps, the detection time of the improved ORB model is 35ms, which decreases to 30ms with the addition of KCF, 25ms with the addition of GP, and 20ms with the combination of TLD, showing the significant effect of the combination of KCF, GP, and TLD in improving the efficiency. In Figure 9(b), the target loss rate increases as the occlusion area increases. Under 20% occlusion, the loss rate of the improved ORB model is 25%, which decreases to 15.4% with the addition of KCF, GP, and TLD. Under 50% occlusion, the loss rate of the improved ORB is %, while the loss rate of the combination of TLD is 15.0%. These results show that TLD effectively reduces target loss in the occlusion case. Spatial constraint distribution (SCD) significantly improves the stability of feature matching, especially under occlusion and motion blur conditions. SCD reduces mis-matching and enhances robustness by ensuring the spatial consistency of the matched features.KCF performs excellently in tracking accuracy and real-time performance. GP enhances the adaptability of the system to targets of different scales. The TLD module effectively solves the occlusion problem. The three together improve the stability and accuracy of the model in complex environments. In addition, the paper utilizes precision (P), recall (R), F1 score, and average displacement error (ADE) as indicators to introduce a more cash-based algorithm model for comparison. For example, YOLOv5, Mask R-CNN, and Faster R-CNN are tested. Table 3 lists the specific results. Among them, the parameters of YOLOv5 are set, withan input size of 640×640, batch size of 16, learning rate of 0.001, using default pre-training weights. Mask R-CNN uses ResNet50 backbone, with an input size of 800×800, batch size of 16, learning rate of 0.0025, using COCO pre-training weights. Faster R-CNN uses ResNet50 backbone, withan input size of 800×800, batch size of 16, learning rate of 0.0025, using ImageNet pre-training weights. CNN uses a ResNet50 backbone with an input size of 600×600, batch size of 8, learning rate of 0.001, using ImageNet pre-training weights. The parameters of all baseline methods are kept in the same training conditions to ensure a fair comparison.

Table 3: Indicator test results for different models.

Data set	Model	P/%	R/%	F1/%	ADE/mm	*p*
OTB	YOLOv5	87.91	88.71	88.31	0.02	0.003
	Mask R-CNN	89.84	86.52	88.18	0.02	0.006
	Faster R-CNN	90.17	88.45	89.31	0.03	0.012
	Research model	92.56	89.29	90.93	0.01	/
LaSOT	YOLOv5	88.24	85.27	86.76	0.02	0.004
	Mask R-CNN	90.63	86.58	88.61	0.02	0.009
	Faster R-CNN	90.89	88.72	89.81	0.02	0.015
	Research model	92.08	90.11	91.10	0.02	/

As shown in Table 3, the proposed model demonstrates superior performance on both OTB and LaSOT datasets. On the OTB dataset, the accuracy of the proposed model is 92.56%, the recall is 89.29%, the F1 score reaches 90.93%, and the ADE is only 0.01 mm. On the LaSOT dataset, the P value is 92.08%, the R value is 90.11%, and the F1 score is 91.10%. Compared with the state-of-the-art methods such as YOLOv5, Mask R-CNN, and Faster R-CNN, the present model shows significant advantages in the two key indexes, namely, F1 score and ADE. It is further verified by t-test that there are statistically significant differences in multiple performance metrics at the $p<0.01$ level, indicating that the present model possesses stronger displacement tracking accuracy and robustness in complex environments.

3.2 Simulation testing of engineering displacement tracking detection model

In this study, the experimental simulation is conducted using a 20-megapixel industrial camera equipped with a 25mm lens, model Basler ace acA2040-55um, with a frame rate of 30fps and a resolution of 1920×1080.This camera is used to simulate the displacement variations of the bridge structure under different stress conditions. The simulation environment is built based on the Unity 3D engine and simulates various forces and environmental conditions of the bridge, including light load, medium load, and heavy load. The experimental simulation scenario is exhibited in Figure 10. Light, middle, and heavy loads represent different bridge stresses. Light and small loads simulate the stress of bridges under normal traffic flow. Middle load refers to simulating the stress of a bridge under high traffic flow or when light vehicles pass through it with a medium load. Heavy load represents simulating the stress of bridges under heavy vehicles or extreme weather conditions using large loads.

Figure10: Experimental simulation scenarios.

Based on this environment, this study conducts tests using AP and actual monitoring time as indicators. Figure 11 shows the detection comparison of four models.

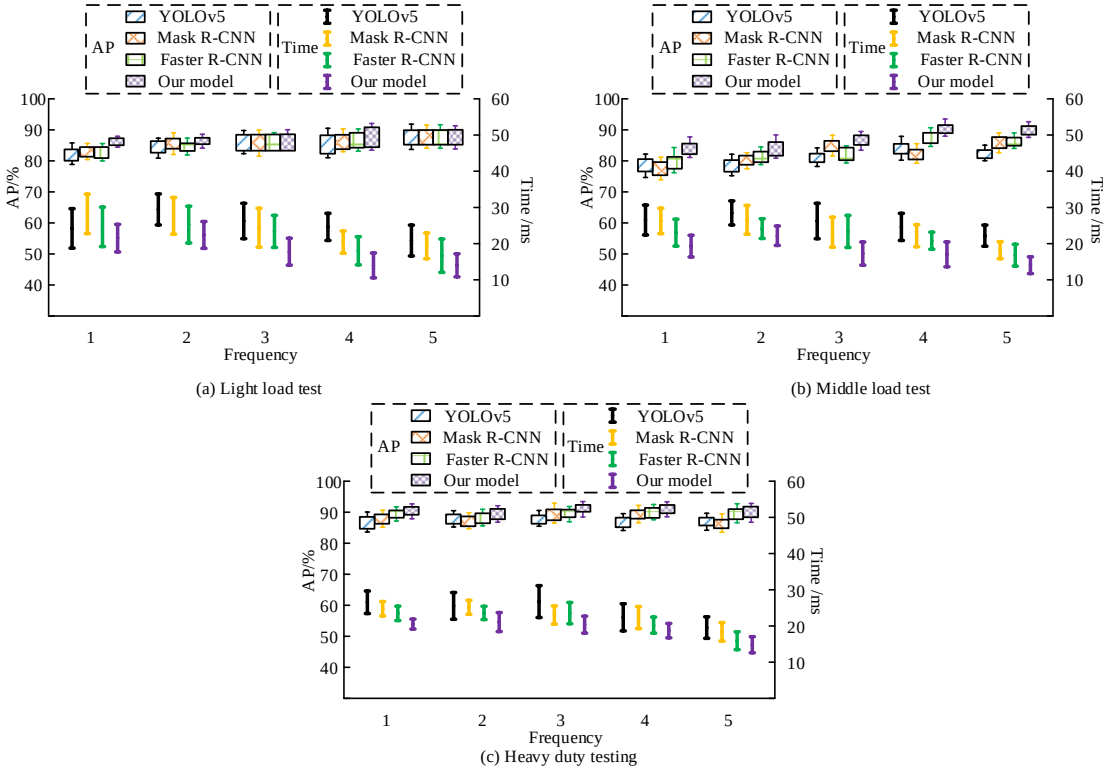


Figure 11: Displacement monitoring tests under different loads.

Figures 11 (a), (b), and (c) show the test results of model indicators under light, middle, and heavy loads. In Figure 11, under light load conditions, the AP values of YOLOv5, Mask R-CNN, and Faster R-CNN are 88.21%, 89.34%, and 89.91%, respectively. The research model reaches 92.17%, which is about 2.5% higher than the average, while the detection time is 15ms, which is about 40% faster than the average time of other models. Under middle load conditions, the AP values of YOLOv5 and Mask R-CNN are 85.38% and 86.72%, respectively, while the AP value of Faster R-CNN is 90.03%. The AP value of the research model is 92.06%, leading by about 1.5% to 6.5% respectively. The detection time is 30ms, which is nearly 25% faster than the average of the other three. Under heavy load environment, the APs of YOLOv5 and Mask R-CNN are 83.41% and 85.14% respectively, and 88.06% for Faster R-CNN. The research model still reaches 91.33%, with a lead of up to 7.9%, and the detection time is controlled at 40ms, which is much lower than the other models with a processing delay of about 55ms. Overall, the research model maintains high detection accuracy and fast operational efficiency under all three load conditions. Especially with the increase of load, its accuracy advantage becomes more significant, indicating that this method has stronger robustness and engineering adaptability to complex stress scenarios. This study tests displacement and absolute error (AE) as indicators, as listed in Table 4.

Table 4: Displacement and absolute error testing.

Conditional	Model	True displacement value/mm	Monitoring of displacement values/mm	AE/mm	*p*
Light load	YOLOv5	3.91	3.14	0.37	0.007
	Mask R-CNN	3.91	3.43	0.26	0.021
	Faster R-CNN	3.91	3.77	0.18	0.063
	Research model	o	3.89	0.07	0.000
Middle load	YOLOv5	9.24	9.01	0.28	0.009
	Mask R-CNN	9.24	9.12	0.16	0.025
	Faster R-CNN	9.24	9.33	0.15	0.039
	Research model	9.24	9.18	0.09	0.002
Heavy load	YOLOv5	18.87	18.21	0.44	0.006
	Mask R-CNN	18.87	18.64	0.28	0.014
	Faster R-CNN	18.87	18.57	0.31	0.018
	Research model	18.87	18.81	0.11	0.001

From Table 4, under light loading conditions, the monitored displacement values of the research model are closest to the real values, with an AE of 0.07 mm, p=0.000, which is significantly better than the other models, and the YOLOv5 error is the largest of 0.37 mm. Under middle loading conditions, the AE of the research model is 0.09 mm, p=0.002, which is lower than that of Mask R-CNN (0.16 mm, p=0.025) and Faster R-CNN (0.15 mm, p=0.039), and the error of YOLOv5 is still high at 0.28 mm. Under heavy loading, the AE of the research model is 0.11 mm (p=0.001), which still has a higher error than that of YOLOv5 (0.44 mm), Mask R-CNN (0.28 mm), and Faster R-CNN (0.31 mm). This indicates that its displacement measurement accuracy and consistency under multi-load conditions are better than the comparison models. Under the same loading conditions and shooting angles, the deviation of the simulation data from the measured data in terms of the number of features extracted at key points is less than 4%, and the ADE is controlled within 0.15 mm. This indicates that the constructed simulation environment has good engineering realism, and it can be an effective alternative to some of the real test scenarios for the verification of the algorithm. Overall, the research model has the highest displacement monitoring accuracy and the smallest AE under different loads, indicating that it has stronger displacement detection capability and robustness.

4 Discussion

The proposed displacement detection model based on spatial constrained ORB feature matching combined with KCF-GP-TLD exhibits performance advantages over existing mainstream methods in multiple indicators, mainly due to the introduction of three key improvements. First, the spatial constrained strategy effectively enhances the accuracy and noise resistance of ORB feature point matching, especially in the presence of motion blur or complex background texture, significantly reducing the false matching rate. Second, the GP structure improves the stability of KCF under target scale changes, ensuring that high tracking accuracy can still be maintained under different camera distances, viewpoints, or target scaling. Third, the fusion of the TLD module improves the recovery ability of the model in occluded scenes, and the loss rate is still lower than 20% when the occluded area reaches 50% in the experiments, which is significantly better than that of the model without the introduction of TLD. In the comparison with advanced models such as YOLOv5, Mask R-CNN, and Faster R-CNN, the research model has more than 2% advantages in AP value and F1 score on both OTB and LaSOT datasets. ADE has been reduced by approximately 0.01-0.02 mm, and the fastest detection time can reach 20 ms, achieving a better balance between accuracy, real-time performance, and robustness. This performance improvement does not come from deep learning training, but relies on the optimized combination among lightweight algorithms, which is particularly suitable for real-time engineering monitoring scenarios. In addition, the model performs particularly well in simulating heavy loads and obstructed environments, maintaining high accuracy and low error even under high-stress or low-visibility conditions. This indicates its applicability and low engineering

deployment cost, particularly suitable for embedded monitoring systems in small and medium-sized bridge sites.

However, this study did not introduce real video data from real bridge engineering. At present, the experimental evidence mainly comes from the OTB and LaSOT standard datasets, as well as the comprehensive simulation platform built on Unity 3D, which cannot fully cover the effects of real problems such as sensor noise, lighting changes, and viewpoint occlusion. Therefore, future research should be further extended to practical engineering scenarios, and cross-environment generalization validation should be performed in combination with real surveillance devices to further verify the practicality and scalability of the method.

5 Conclusion

In response to the shortcomings of traditional displacement measurement techniques in complex environments in bridge engineering, this study introduced the ORB feature matching technology of SCD and combines GP and TLD with KCF to optimize feature point detection and tracking. A new non-contact displacement monitoring method has been proposed. The experiments showed that the feature extraction accuracy and feature tracking accuracy of the model could reach about 0.95 in the OBR dataset with the settings of BRIEF descriptor length of 256 and the number of GP layers of 4. The ablation experiment further validated the performance improvement of the sub-module. The complete model that integrated KCF, GP, and TLD had the best performance under occlusion and scale change conditions, with a detection time of the shortest 20 ms and an average target loss rate of 15%. In the OTB and LaSOT datasets, the model had the highest detection precision of 92.56%, the highest recall of 90.11%, the highest F1score of 91.10%, and the lowest ADE of 0.01 mm, which is better than the mainstream models such as YOLOv5, Mask R-CNN, and Faster R-CNN.This indicated that the new model had higher tracking accuracy and smaller displacement detection errors in complex scenarios. Simulation tests have found that the monitoring AP value and monitoring time of the research model are significantly better under light, middle, and heavy load environments. Especially under heavy load conditions, the average error was only 0.11mm, which was better than traditional algorithms such as YOLOv5 and Faster R-CNN. Although the research model has significantly improved in performance, there are still some robustness issues when dealing with extreme lighting changes and prolonged monitoring. Future research will continue to optimize the model's light adaptability and further integrate more intelligent detection algorithms to improve the system's performance in complex dynamic environments.

References

[1] Z. Ma, J. Choi, and H. Sohn, (2022)"Real-time structural displacement estimation by fusing asynchronous acceleration and computer vision measurements,"*Computer-Aided Civil and Infrastructure Engineering*, vol. 37, no. 6, pp. 688-703. https://doi.org/10.1111/mice.12767

[2] B. Kim, J. Lee, S. H. Sim, S. Cho, and B. H. (2022) park, "Computer vision-based remote displacement monitoring system for in-situ bridge bearings robust to large displacement induced by temperature change,"*Smart Structures and Systems*, vol. 30, no. 5, pp. 521-535, https://doi.org/10.12989/sss.2022.30.5.521

[3] W. Du, D. Lei, Z. Hang, Y. Ling, P. Bai, and F. Zhu, (2023)"Short-distance and long-distance bridge displacement measurement based on template matching and feature detection methods,"*Journal of Civil Structural Health Monitoring*, vol. 13, no. 2, pp. 343-360, https://doi.org/10.1007/s13349-022-00637-6

[4] I. A. Colombani, and B. Andrawes, (2022) "A study of multi-target image-based displacement measurement approach for field testing of bridges,"*Journal of Structural Integrity and Maintenance*, vol. 7, no. 4, pp. 207-216, https://doi.org/10.1080/24705314.2022.2088071

[5] Y. Han, G. Wu, and D. Feng, (2022) "Vision-based displacement measurement using an unmanned aerial vehicle,"*Structural Control and Health Monitoring*, vol. 29, no. 10, pp. 3025-3027, https://doi.org/10.1002/stc.3025

[6] S. A.V. Shajihan, T. Hoang, K. Mechitov, and B. F. Spencer Jr, (2022)"Wireless SmartVision system for synchronized displacement monitoring of railroad bridges,"*Computer-Aided Civil and Infrastructure Engineering*, vol. 37, no. 9, pp. 1070-1088, https://doi.org/10.1111/mice.12846

[7] G. Liu, (2024) "Review and prospect of research on structural health monitoring technology for bridges,"*Journal of Architectural Research and Development*, vol. 8, no. 3, pp. 2208-2214, https://doi.org/10.26689/jard.v8i3.7169

[8] P. Bai, Y. Ni, W. Du, and D. Lei, (2022) "Bridge displacement measurement method based on digital image feature detection,"*Experimental Technology & Management*, vol. 39, no. 7, pp. 86-88, https://doi.org/10.16791/j.cnki.sjg.2022.07.014

[9] H. Liu, X. Xu, X. Chen, C. Li, and M. Wang, (2022)"Real-time ship tracking under challenges of scale variation and different visibility weather conditions,"*Journal of Marine Science and Engineering*, vol. 10, no. 3, pp. 444-447, https://doi.org/10.3390/jmse10030444

[10] E. L. Bianchi, N. Sakib, C. Woolsey, and M. Hebdon, (2023) "Bridge inspection component registration for damage evolution,"*Structural Health Monitoring*, vol. 22, no. 1, pp. 472-495, https://doi.org/10.1177/14759217221083647

[11] W. Du, D. Lei, F. Zhu, P. Bai, and J. Zhang, (2024) "A non-contact displacement measurement system based on a portable smartphone with digital image methods,"*Structure and Infrastructure Engineering*,

vol. 20, no. 9, pp. 1322-1340, https://doi.org/10.1080/15732479.2022.2141268

[12] H. Zhang, Y. Zhu, W. Xiong, and C. S. Cai, (2023) "Point cloud registration methods for long-span bridge spatial deformation monitoring using terrestrial laser scanning,"*Structural Control and Health Monitoring*, vol. 2023, no. 1, pp. 2618-2621, https://doi.org/10.1155/2023/2629418

[13] G. Jeon, S. Kim, S. Ahn, H. Kim, and H. Yoom, (2022) "Vision-based automatic cable displacement measurement using Cable-ROI Net and Uni-KLT,"*Structural Control and Health Monitoring*, vol. 29, no. 8, pp. 2977-2983, https://doi.org/10.1002/stc.2977

[14] S. Zhuge, X. Xu, L. Zhong, S. Gan, B. Lin, X. Yang, and X. Zhang, (2022) "Noncontact deflection measurement for bridge through a multi-UAVs system,"*Computer-Aided Civil and Infrastructure Engineering*, vol. 37, no. 6, pp. 746-761, https://doi.org/10.1111/mice.12771

[15] M. Wang, F. Xu, Y. Xu, and J. Brownjohn, (2023) "A robust subpixel refinement technique using self-adaptive edge points matching for vision-based structural displacement measurement,"*Computer-Aided Civil and Infrastructure Engineering*, vol. 38, no. 5, pp. 562-579, https://doi.org/10.1111/mice.12889

[16] M. Breccolotti, and M. Natalicchi, (2022) "Bridge damage detection through combined quasi-static influence lines and weigh-in-motion devices,"*International Journal of Civil Engineering*, vol. 20, no. 5, pp. 487-500, https://doi.org/10.1007/s40999-021-00682-0

[17] G. I. Zarate Garnica, E. O. L. Lantsoght, and Y. Yang, (2022) "Monitoring structural responses during load testing of reinforced concrete bridges: A review,"*Structure and Infrastructure Engineering*, vol. 18, no. 10, pp. 1558-1580, https://doi.org/10.1080/15732479.2022.2063906

[18] D. Cherid, N. Bourahla, M. S. Laghoub, and A. Mohabeddine, (2022) "Sensor number and placement optimization for detection and localization of damage in a suspension bridge using a hybrid ANN-PCA reduced FRF method,"*International Journal of Structural Integrity*, vol. 13, no. 1, pp. 133-149, https://doi.org/10.1108/IJSI-07-2021-0075

[19] J. Chen, Q. Wang, H. H. Cheng, W. Peng, and W. Xu, (2022) "A review of vision-based traffic semantic understanding in ITSs,"*IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 19954-19979, https://doi.org/10.1109/TITS.2022.3182410

[20] H. Mokayed, T. Z. Quan, L. Alkhaled, and V. Sivakumar, (2023) "Real-time human detection and counting system using deep learning computer vision techniques,"*Artificial Intelligence and Applications*, vol. 1. no. 4, pp. 221-229, https://doi.org/10.47852/bonviewAIA2202391

[21] C. Pan, H. Zhao, and M. Sun. (2024) "Real-time target detection system in scenic landscape based on improved YOLOv4 algorithm,"*Informatica*, vol. 48, no. 8, pp. 35–48, August, DOI: 10.31449/inf. v48i8.5700.

[22] M. Li, H. Dong, F. Zhang, and others. (2024) "Method for top view pedestrian flow detection based on small target tracking,"*Informatica*, vol. 48, no. 11, pp. 59–70, November, DOI: 10.31449/inf. v48i11.6033.