Preprints are preliminary reports that have not undergone peer review. They should not be considered conclusive, used to inform clinical practice, or referenced by the media as validated information.

Vehicle video stabilization algorithm based on grid motion statistics and adaptive kalman filtering

Chengcheng Li (ChengChengLi_TY@outlook.com)
Yuan Tian
Lisen Ma
yunhong Jia
yuqi bi

Research Article

Keywords: Digital image stabilization, ORB, Grid motion statistics, Adaptive Kalman filtering, PSNR

Posted Date: October 17th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3438423/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Signal, Image and Video Processing on December 16th, 2023. See the published version at https://doi.org/10.1007/s11760-023-02890-8.

Vehicle video stabilization algorithm based on grid motion statistics and adaptive kalman filtering

Chengcheng Li[1], Yuan Tian[2,3,4], Lisen Ma[1], Yunhong Jia[2,3,4] and Yueqi Bi[2]

ABSTRACT Owing to the impact of vibration on the carrier of a car-mounted camera, video images are shaking, resulting in decreased or failed recognition accuracy based on visual-target detection. To solve this problem, a video stabilization algorithm based on grid motion statistics and an adaptive Kalman filter is proposed. To satisfy the real-time and precision requirements of the vehicle video stabilization algorithm, the Oriented Fast and Rotated Brief (ORB) feature point detection algorithm was selected to detect and describe the obtained video frames. In addition, the accuracy of the motion estimate is increased by deleting the erroneous match points using an erroneous match-removal algorithm based on grid motion statistics (GMS). The matching accuracy of the GMS-based feature matching algorithm increased by 2.3% and 4.1% compared to conventional feature-matching algorithms based on scale-invariant feature transform (SIFT) and ORB, respectively. However, the matching time between adjacent video frames was reduced by 76% and 16%, respectively. Considering the possible jitter of the vehicle-mounted camera, an adaptive Kalman filtering algorithm was used to smooth the acquired motion trajectory and solve the problem of classical Kalman filtering being sensitive to the initial value. The mean Peak Signal-to-Noise Ratio (PSNR) after stabilization rose by 10.27 dB in comparison to the video stability before stabilization. Therefore, this algorithm exhibits good stability.

INDEX TERMS Digital image stabilization; ORB; Grid motion statistics; Adaptive Kalman filtering; PSNR;

I. INTRODUCTION

Visual-based image perception technology is an important method for unmanned vehicles to perceive changes in their surrounding environment [1]. The harsh road conditions of underground coal mines and the inherent vibration of the vehicle itself can result in an vehicle camera capturing shaky images, which adversely affects the accuracy of the subsequent image recognition. Therefore, video stabilization technology is required to remove shaking in a video [2,3]. Video stabilization is the process of improving the video quality by eliminating shaking. They can be divided into mechanical, optical, and digital stabilization systems [4]. Mechanical stabilization devices generally have larger sizes and higher space requirements for stabilization, making them typically used in airborne, marine, and large-scale weapon guidance systems. Optical stabilization has the characteristics of small size, light weight, and no need for a stable platform, which can effectively eliminate or alleviate video shake and is often used in medical equipment, space observation, aviation, and military fields [5-6]. However, they are expensive, require high-quality materials, and have complex system architectures. To maintain accuracy, color discrepancies must be removed before usage and stability must be rectified regularly. This is not suitable for stabilizing cameras in underground coal-mine operations. Digital image stabilization (DIS) technology utilizes image-processing techniques to analyze images and obtain the original motion trajectory of the camera. Through motion smoothing, active motion in shaking videos is retained, whereas random shaking in video sequences is eliminated, resulting in a stable video sequence [7]. Because of its advantages of low cost and lack of additional mechanical structures, many domestic and foreign scholars have conducted extensive research on DIS [8].

Based on the use of motion models, digital image stabilization (DIS) algorithms can be divided into two main categories: 2D and 3D DIS algorithms [9-10]. The 2D stabilization algorithm tracks the movement of pixels in the image to obtain the offset between adjacent frames, accumulates the camera motion trajectory, and then removes the jitter component in the motion trajectory through motion

ChengchengLi
ChengChengLi_TY@outlook.com
[1] China Coal Research Institute, Beijing 100013, China
[2] Shanxi Tiandi Coal Mining Machinery Co., Ltd., Taiyuan 030006, China
[3] CCTEG Taiyuan Research Institute Co., Ltd., Taiyuan 030006, China
[4] China National Engineering Laboratory for Coal Mining Machinery., Taiyuan 030006, China

smoothing. Finally, an affine transformation is applied to the video frames to generate a stable video [11]. The 3D stabilization algorithm uses the Structure from Motion (SFM) method to estimate the three-dimensional information structure of the image and then performs global motion estimation to obtain the motion trajectory of the camera. The computational complexity of SFM is high and requires sufficient motion information from the video. In addition, when parallax information is lacking in a video, it is difficult to construct an accurate three-dimensional scene. Therefore, for most scenarios, the 2D stabilization algorithm is more robust and has a wider range of applications.

Currently, research on DIS has mainly focused on the motion estimation and motion-smoothing stages. Shujiao et al. [12] proposed using the BRISK operator to extract feature points between adjacent frames in a video image, and established matching relationships based on the Hamming distance between descriptors to obtain a rough estimate of the motion vector. However, the matching relationship constructed based on the Hamming distance is a coarse match, which can easily lead to inaccurate motion estimations when there are mismatching points. Kejriwal et al. [13] proposed a feature point tracking-based image stabilization algorithm that extracts feature points using the Shi-Tomasi algorithm and then tracks and matches the feature points in adjacent frames using the Lucas-Kanade optical flow to obtain motion vectors. However, the optical flow method is based on the assumption of constant brightness between adjacent frames and is significantly affected by changes in the lighting conditions. Rodriguez et al. [14] proposed a method for stabilizing continuous image sequences based on feature matching and subpixel correlation technology, which effectively eliminates high-frequency jitter in shaky videos. Bing et al. [15] used a scale-invariant feature transform (SIFT) operator to extract and describe feature points; however, the high dimensionality of SIFT descriptors affects the real-time performance of feature matching. Shene et al. [16] proposed the use of the Speeded Up Robust Feature (SURF) algorithm to extract and describe feature points and remove incorrect matching points through cross-validation, which improves the speed of feature matching compared to the SIFT algorithm. However, owing to the repeated computation of the Haar wavelet response values when computing the main orientation of the feature points, the real-time performance of image stabilization requires further improvement.

II. CAMERA MOTION MODELING

The main reason for video sequence jitter is irregular camera motion, which causes large jumps in the image pixels and results in a lack of smooth transitions between adjacent frames. The essence of video stabilization is to obtain accurate motion vectors and use specific image transformation models to achieve video stabilization. During the motion of an vehicle device, the movement of the camera consists mainly of rotation and translation. The pixel position can be effectively corrected and the image structure can be restored using an affine transformation model. The affine transformation model between image frames can be expressed as:

$$\begin{bmatrix} X' \\ Y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} + \begin{bmatrix} \Delta X \\ \Delta Y \end{bmatrix} \tag{1}$$

Where $\begin{bmatrix} X & Y \end{bmatrix}^T$ and $\begin{bmatrix} X' & Y' \end{bmatrix}$ represent the corresponding coordinates of the same feature point between the two frames of the jittering video, θ is the rotation angle parameter caused by the jitter, and $\begin{bmatrix} \Delta X & \Delta Y \end{bmatrix}^T$ represents the motion vector in the horizontal and vertical directions between the two frames. For video image sequence $I_0, I_1, \cdots, I_{t-1}, I_t$, the motion model from frame t to frame $t+1$ is defined as H_t according to the affine transformation model between frames. The motion path of the camera at time t can be expressed as:

$$C_t = H_{t-1} \times H_{t-2} \times \ldots \times H_1 \tag{2}$$

After obtaining the camera motion trajectory, a smooth path can be obtained using methods such as path smoothing and fitting. Motion compensation can then be performed based on the relationship between the smoothed paths before and after the smoothing. After completing the motion compensation, image transformation can be performed according to the affine transformation model, and the cropping ratio can be set to remove the black margin that appears after image transformation. The process of generating a stabilized video is illustrated in Fig. 1.

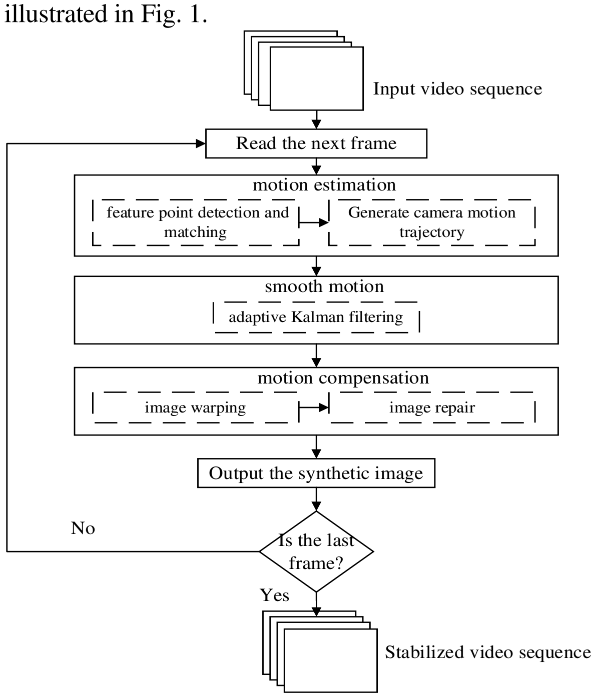


FIGURE 1. Image stabilization algorithm flowchart

III. FUSION OF GMS FOR FEATURE POINT DETECTION AND MATCHING

A. DETECTION AND DESCRIPTION Of ORB FEATURE POINTS

The ORB algorithm is a feature detection and description algorithm that combines Features from Accelerated Segment Test (FAST) feature point detection with BRIEF descriptors. It has the characteristics of fast computation and good real-time performance and is a combined algorithm [25]. The purpose of the FAST algorithm is to find representative pixels. It randomly selects a pixel point in the image as the center and selects 16-pixel points on a circle with a radius of 3. The feature points were determined by comparing the grayscale values of the 16-pixel points. Once the image keypoints have been determined, they are differentiated based on their corresponding descriptors. The ORB algorithm uses a binary, robust, and independent feature descriptor called BRIEF to describe detected keypoints [26]. 256-pixel pairs were randomly selected within the neighborhood range centered around each keypoint. The grayscale values of each pixel pair are then compared, as shown in Equation (3). Where $p(x)$ and $p(y)$ are the gray values of a pair of pixels randomly selected around the feature point. If $p(x) < p(y)$ is used, the binary bit corresponding to the feature point is 1; otherwise, it is 0. After comparing them individually with 256 pairs of pixels randomly selected around the feature point, a descriptor of the feature point was generated.

$$\tau(x, y) = \begin{cases} 1, p(x) < p(y) \\ 0, p(x) > p(y) \end{cases} \tag{3}$$

Although the BRIEF algorithm can generate binary descriptors, it cannot satisfy the rotation invariance of images. Under different working conditions, the vehicle-mounted camera device must detect feature points in images with side tilts and pitch angles. Therefore, the vector pointing from the FAST corner point to the centroid of the surrounding rectangular area can be used as the orientation vector of the ORB feature point. The direction of the feature point can be determined using the orientation vector, thereby solving the problem of rotation invariance. The feature point detection results for the coal mine roadway images are shown in Fig.2. The radius size reflects the offset angle, the line segment in the circle represents the primary direction of the feature point, and the center is the location of the extracted feature point.

FIGURE 2. Detection results of feature points in roadway images

B. INITIAL MATCHING OF FEATURE POINTS

Feature matching is the process of finding the same feature points for the same object in the adjacent frames of an image sequence. The ORB algorithm was used to detect and describe the features in the images, and feature vector registration was performed based on the Hamming distance between descriptors. Assume that A and B represent the descriptors of the two matching feature points, x and y represent the corresponding binary descriptors of the matching feature points, and the Hamming distance obtained by calculating the bitwise AND operation on the descriptors is denoted by D, as shown in Equations (4) and (5). The smaller the Hamming distance between two feature points, the higher the similarity. In the experiments, a match was considered successful when the distance is less than 128.

$$\begin{cases} A_1 = x_0 x_1 x_2 \cdots x_{255} \\ B_1 = y_0 y_1 y_2 \cdots y_{255} \end{cases} \tag{4}$$

$$D(A_1, B_1) = \sum_{i=0}^{255} x_i \oplus y_i \tag{5}$$

In the image feature matching process, it is difficult to achieve absolute accuracy in the feature extraction and description. In addition, images captured by vehicle-mounted cameras are susceptible to factors such as image noise, lighting changes, and image rotation, which can cause changes in the image features and result in erroneous matches during the initial matching process. Fig.3 shows the initial matching results of two adjacent frames in a shaky video that contains erroneous matching point pairs that must be removed in subsequent processing.

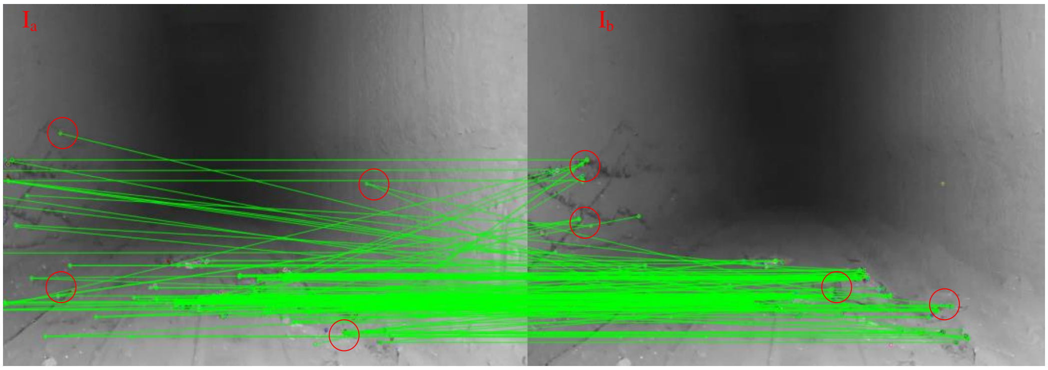


FIGURE 3. Matching results of feature points; Ia : Jitter the first frame of the video; Ib: Jitter the second frame of the video

C. ELIMINATION OF INCORRECT MATCHING POINTS BASED ON GMS

After initial matching, there were pairs of incorrectly matched points. Removing erroneous matching points from the initial matching can improve matching accuracy and reduce matching errors. In addition, it can save computing resources and image processing time, which is a key step in improving motion estimation accuracy between frames in shaky videos. The Random Sample Consensus (RANSAC) algorithm is a robust method for removing erroneous matching points. The main idea is to randomly select a subset of data from the dataset to estimate the model parameters and then verify all data to filter out the required dataset. However, the RANSAC algorithm requires multiple random samplings and model validation, resulting in a high computational complexity and poor real-time performance. To address this issue, a grid-based motion statistics (GMS) algorithm for erroneous matching removal is proposed to improve the erroneous matching removal ability while shortening the overall running time of the algorithm.

The GMS algorithm is an image registration algorithm based on grid partitioning and statistical properties of motion. Its core idea is to use the ORB algorithm for image feature extraction and description and to distinguish between correct and incorrect matches using neighborhood support estimation based on grid motion statistics. [27] Assuming that M and N feature points exist in the matching image I_a and the image to be matched I_b, respectively, their corresponding feature point sets are denoted as $\{M, N\}$. $\chi = \{x_1, x_2, \cdots, x_i, \cdots x_N\}$ is a set of all feature point-matching pairs from image I_a to image I_b, where x_i represents a matching pair of points. By analyzing the local support of feature point matching, we divide χ into a true match set and a false match set. Compared to incorrect matching pairs, correct matching pairs have more features in the neighborhood that fit the matching relationship. Given this difference, incorrect matching points can be eliminated, as shown in Fig. 4.. The neighborhood feature support estimate is given by Equation (6).

$$S_i = |\chi_i| - 1 \tag{6}$$

Here, S_i represents the value of neighborhood support, $\chi_i \in \chi$ is the matching subset of the matching region $\{a, b\}$ corresponding to x_i, and -1 denotes the removal of the original feature points from the matching set. To speed up the discrimination between correct and incorrect matches, the image was divided into $G = g \times g$ grids and a 3×3 3×region, as shown in Fig. 5, was selected. The neighborhood support S_{ij} for each unit to $\{i, j\}$ can be obtained as follows:

$$S_{ij} = \sum_{k=1}^{k=9} \left| x_{a^k b^k} \right| \tag{7}$$

Where K represents the number of regions for cell division, and $\left| x_{a^k b^k} \right|$ represents the matches between cells $\{i^k, j^k\}$. As the grid is partitioned, the probability of differentiation between correct and incorrect matches gradually increases. According to the neighborhood support degree S_{ij} calculated using Equation (7) and the empirical threshold τ, the feature matching pairs can be divided into a correct matching set and an incorrect matching set using Equation (8).

$$cell - pair\{i, j\} \in \begin{cases} T, if \ S_{ij} > \tau_i = \alpha\sqrt{n_i} \\ F, other \end{cases} \tag{8}$$

where $\{T, F\}$ are two sets of correct matches and false matches, respectively, and $\{i, j\}$ represents a pair of matching grid regions, representing the i-th and j-th grid regions in the images I_a and I_b, respectively. n_i is the total number of features in a 3×3 grid, and α is the weight value set to $\alpha = 6$ in the experiment. Feature matching pairs with neighborhood support estimation S_{ij} greater than τ_i in the grid region were retained as reliable feature matches that satisfied the conditions.

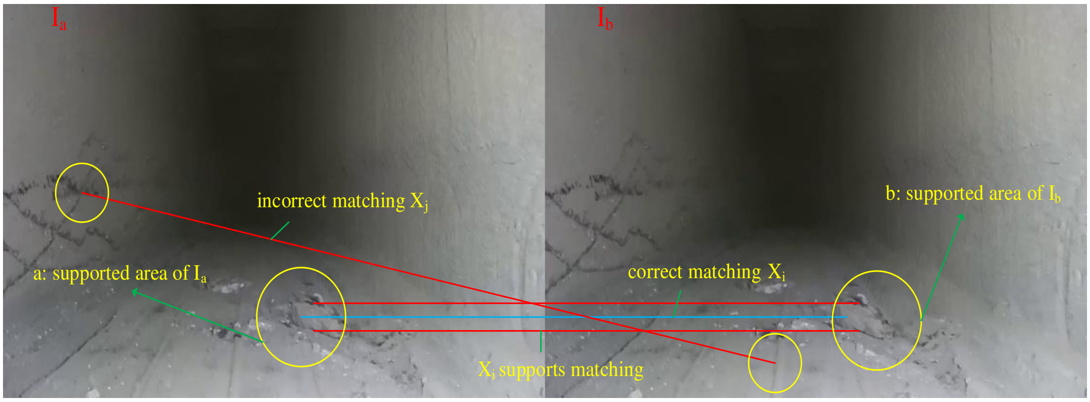


FIGURE 4. Neighborhood matching scoring model; I_a : Jitter the first frame of the video ; I_b : Jitter the second frame of the video

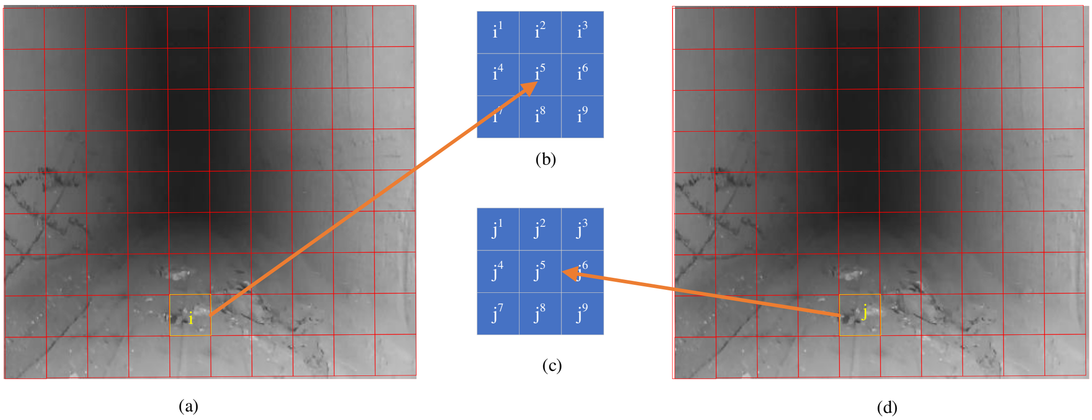


FIGURE 5. Smooth grid movement ; (a) Image Ia is meshed; (b) The area used to evaluate the degree of neighborhood support for cell {i}; (c) Image Ib is meshed; (d) The area used to evaluate the degree of neighborhood support for cell {j}

After using an error-matching removal algorithm to remove error-matching points, error-matching points may still exist. Based on the distance relationship between all matching pairs, the numbers of correct and incorrect matching pairs were selected after image feature matching. The ability of the error-matching rejection algorithm to reject error-matching points was analyzed based on the proportion of incorrectly matched pairs. Table 1 summarizes the number of error-matching points after feature matching, and the matching time for two adjacent image frames. The analysis compared the processing effects of different feature matching algorithms on two adjacent frames of images in the same video. Under equivalent conditions, although the number of feature-matching pairs selected by the algorithm in this study was the lowest, it paid more attention to the quality of the feature matching. The matching accuracy and matching time are better than those of traditional methods. Table 1 shows that the ORB+GMS algorithm processed two adjacent frames of images with the highest matching accuracy, with an average of 99.5%, and the fastest matching times, with an average of 0.025 s. The matching accuracy was 2.3% higher than that of the SIFT+RANSAC algorithm, the matching time was 76% lower, the matching accuracy was 4.1% higher than that of the ORB+RANSAC algorithm, and the matching time was 16% lower. Thus, the feature matching algorithm with GMS fusion has the characteristics of high matching accuracy and good capacity to reduce incorrect matching spots, according to a comparative analysis of the results from numerous group tests. The accuracy of motion estimation between shaky video frames was also improved, whereas the running time of the algorithm was significantly decreased.

TABLE 1. COMPARISON OF TEST RESULTS OF DIFFERENT ALGORITHMS

	SIFT+ RANSAC	ORB+RANSAC	ORB+GMS
matching number	327	327	206
Number of incorrect matches	9	15	1
match accuracy (%)	97.2%	95.4%	99.5%
matching time (s)	0.108	0.030	0.025

IV. TRAJECTORY SMOOTHING BASED ON ADAPTIVE KALMAN FILTERING

A. KALMAN FILTERING

In video stabilization systems, it is necessary to remove random shaking contained within through filtering to obtain the true motion trajectory of the camera. Common filtering methods include the mean, Gaussian, curve-fitting, and Kalman filtering. The implementation of the mean filter is simple; however, it easily produces an overstabilization phenomenon. Gaussian filtering and curve-fitting filtering exhibit poor real-time performance and cannot satisfy the requirements for real-time image stabilization.

The Kalman filter can generate the motion estimation value of the current frame using only the predicted values of the motion parameters from the previous frame and the measured values of the motion parameters from the current frame [28]. Because it does not require prestored data, it exhibits good real-time performance and is widely used in the field of stabilization. However, the classical Kalman filter often needs to set fixed parameters in advance, and cannot adaptively update them with external noise changes. The quality of an algorithm depends on people's understanding of the prior information and their understanding of the corresponding function and stochastic models. If the system noise covariance Q is constant, the tracking features of the filter are greater when the measurement noise covariance R is smaller and the smoothing characteristics are stronger when the measurement noise R is larger.

B. IMPROVE ADAPTIVE KALMAN FILTERING BASED ON SAGE-HUSA

To address the problem that the traditional Kalman filter cannot update the system noise covariance Q and measurement noise covariance R in real time according to the working conditions, the Sage-Husa adaptive Kalman filter was introduced into the stabilization system and improved [29]. Owing to the influence of the environment, the noise distribution in the video images captured by vehicle cameras during video image acquisition is not necessarily normal distribution. Therefore, the means of the process noise and measurement noise in the Adaptive Kalman Filter (AKF) is not zero and are denoted as $\hat{q}_k$ and $\hat{r}_k$, respectively. The state and observation equations are shown in Equations (9) and (10), respectively.

$$X_k = FX_{k-1} + W_{k-1} + \hat{q}_{k-1} \tag{9}$$

$$Z_k = H_k X_k + V_k + \hat{r}_{k-1} \tag{10}$$

In these equations, $E\left[\hat{q}_{k-1}\right] = q_k$ and $E\left[\hat{r}_k\right] = r_k$ were used. X_k is the state vector of the system at time k and F is a state transition matrix. where Z_k represents the observation vector of the system at time k. The H_k is a systematic observation matrix. W_{k-1} represents the noise in the system state, which follows a normal distribution $N(0, Q_{k-1})$. V_K represents the noise in the system measurement, which follows a normal distribution $N(0, R_k)$. The AKF mainly consists of prediction and correction. The specific steps are as follows:

(1) The state–space equation of the system is established. If the estimated value of the state at moment k-1 is $\hat{x}_k$, then the estimated value of the state $\hat{x}_k^-$ at the next moment k is given by Equation (11).

$$X_k = FX_{k-1} + q_{k-1} \tag{11}$$

(2) The prior estimation error covariance representing the system state uncertainty at time k was calculated. If the posterior estimation error covariance at time k-1 is P_{k-1}, the prior estimation error covariance matrix P_k^- at time k is given by Equation (12).

$$P_k^- = FP_{k-1}F^T + Q_{k-1} \tag{12}$$

(3) The Kalman filter gain, K, was updated. As shown in Equation (13), the Kalman filter gain represents the proportion of model prediction and measurement errors in the process of optimal state estimation. The smaller the measurement noise covariance R, the larger is the gain k.

$$K_k = P_k^- H^T (HP_K^- H^T + \hat{R}_{k-1})^{-1} \tag{13}$$

(4) Compute the residual ε_k.

$$\varepsilon_k = Z_k - H\hat{X}_k^- - \hat{r}_{k-1} \tag{14}$$

(5) The state vector is updated as follows:. According to the state-estimated value X_{k-1} obtained in the prediction process, the state-estimated value X_k at moment k was corrected by combining the difference between the actual observed value and the estimated observed value.

$$\hat{X}_k = \hat{X}_k^- + K_k\left(Z_k - H\hat{X}_k^-\right) \tag{15}$$

(6) Update the posterior estimation error covariance matrix

$$P_k = (I - K_k H)P_k^- \tag{16}$$

(7) Compute the weight of the exponential decay memory, denoted by d_k, as shown in Equation (17). Here, b is the decay factor, set to 0.98 in our experiments.

$$d_k = d_{k-1} / \left(d_{k-1} + b\right) \tag{17}$$

(8) Update $\hat{q}_k$ $\hat{Q}_k$ $\hat{r}_k$ $\hat{R}_k$

$$q_k = \left(1 - d_k\right)q_{k-1} + d_k\left(X_k - X_{k-1}\right) \tag{18}$$

$$Q_k = \left(1 - d_k\right)\hat{Q}_{k-1} + d_k\left(K_k\varepsilon_k\varepsilon_k^T K_k + P_k - FP_{k-1}F^T\right) \tag{19}$$

$$r_k = \left(1 - d_k\right)r_{k-1} + d_k\left(Z_k - H_k X_k\right) \tag{20}$$

$$R_k = \left(1 - d_k\right)R_{k-1} + d_k\left(\varepsilon_k\varepsilon_k^T - H_k P_k^- H_k^T\right) \tag{21}$$

The Sage-Husa adaptive Kalman filter algorithm constantly corrects the prediction values using observation values, ultimately achieving the purpose of reducing the error of the traditional model and improving the accuracy of the filter. From Equation (21), if the measurement noise of the actual system is smaller than the estimated value of the theoretical model, the $\varepsilon_k\varepsilon_k^T$ will be relatively small. If the initial state noise is set to a large value, it will cause $H_k P_k^- H_k^T$ to be relatively large. In combination with the two situations described above, it is easy to result in $\varepsilon_k\varepsilon_k^T - H_k P_k^- H_k^T < 0$, which in turn causes R_k to lose positive definiteness and leads to filter divergence. To avoid this problem and ensure the convergence and stability of the filter, the size of the R_k should be restricted in combination with the degree of camera shaking. The specific steps are as follows:

(1) First, the deviation between the estimated value of the previous frame and the observed value is calculated to measure the degree of camera shaking. Specifically, the formula shown in Equation (22) was used.

$$J = J_k - J_{k-1} \tag{22}$$

(2) When the degree of camera shaking on the imaging platform is large, R_k should be assigned as small a value as possible. Otherwise, when the degree of camera shaking is low, R_k should be assigned a higher value. To avoid producing filtering divergence, a filtering convergence check condition was introduced to limit the value range of the measurement noise covariance. The check condition for the value range of R_k according to Equation (21) is abbreviated as t and is shown in Equation (23).

$$t = \varepsilon_k \varepsilon_k^{\ T} - H_k P_k^- H_k^{\ T} \tag{23}$$

(3) According to Equations (12) and (13), the filter gain K and error covariance P is related to the measurement noise covariance R_k. Therefore, in conjunction with the degree of camera shake, the measurement noise covariance R_k is calculated.

$$\hat{R}_k = \begin{cases} (1-d_k)\hat{R}_{k-1} + d_k R_{\min} & t < R_{\min} \\ R_{\max} & abs(J[k]) > a \text{ or } t > R_{\max} \\ (1-d_k)\hat{R}_{k-1} + d_k t & \text{other} \end{cases} \tag{24}$$

In this equation, parameter a is a threshold value used to determine the degree of camera shake. In the experiment, 'a' is set to 3. When the degree of jitter is too large, increase the value of 'R' to achieve the desired smoothing effect. Here, $[R_{\min}, R_{\max}]$ is the minimum and maximum range of values used to measure noise. By the aforementioned processing, the AKF (Adaptive Kalman Filter) has improved adaptive capability and reliability. In addition, it is theoretically difficult to analyze the observability and stability of adaptive filtering. As can be seen from Equations (13) and (15), the filtering gain K_k and error covariance P_k are mainly related to the measurement noise $\hat{R}_k$. Compared to the system noise covariance $\hat{Q}_k$, $\hat{R}_k$ has a greater impact on filtering. To ensure the stability of filtering, the number of adaptive parameters should be reduced in experiments, and the Sage-Husa algorithm should be simplified by only updating the measurement noise covariance $\hat{R}_k$ adaptively, reducing the computational burden of the filtering system and improving the real-time performance of filtering. The modified Equation (18) would be: $\hat{Q}_k = Q$ .In the experiment, Q was set to 0.03.

Following the acquisition of the camera motion trajectory, low-pass filtering was used to eliminate high-frequency shaking. Fig. 6 shows the motion trajectory graphs of the camera in various motion directions after filtering and smoothing. Fig.6(a), 6(c), and 6(e) show the motion trajectory graphs of the camera in the horizontal, vertical, and rotational directions, respectively, after filtering and smoothing using the Sage-Husa adaptive Kalman filter. Fig.6(b), 6(d), and 6(f) show the motion trajectory graphs after improving the Sage-Husa adaptive Kalman filter. From Fig.6(a) and 6(e), it can be observed that the reasons for the vibration of the camera carrier in underground mines are mainly focused on the horizontal and rotational directions. Because the underground environment of mines is relatively severe, video acquisition equipment is easily affected by noise such as camera vibration, pixel error, and nonlinear distortion. These noises affect the quality of the video images, thereby affecting motion smoothing. In addition, the narrow space of underground mines and the different coupling and propagation of electromagnetic interference compared with the ground make the situation worse. The shape, size, medium, metal support, and ventilation of the tunnel affect the coupling and propagation of the electromagnetic interference. Therefore, when the vehicle is driving, the sources of electromagnetic interference that they receive will also be different. Electromagnetic interference affects the quality of video signal transmission, thereby affecting motion smoothing. When the camera moved horizontally, as shown in Fig.6(a), the maximum gain K of the Kalman filter in the green box region a, green box region b, and green box region *c* is -50.2, 206.2, and -41, respectively. When the camera moves rotationally, as shown in Fig.6(e), the maximum gain K of the Kalman filter in the green box region d and green box region e is 19.64 and -10, respectively. Owing to the interference of noise, some abnormal values of the Kalman filter gain K occur during the filtering process, resulting in an excessive influence of the observation data on the state estimation. This causes the state estimation to be unstable, thereby reducing the accuracy of the filter. The measurement covariance R_k is related to gain K of the Kalman filter. When gain K occurs abnormally, it causes the measurement covariance R to lose its positive definiteness, resulting in filtering divergence. Therefore, to obtain reliable motion-smoothing effects, the filter parameters should be optimized in conjunction with the degree of camera vibration during the filtering process. When filtering divergence occurs, the measurement noise covariance usually has an abnormally large or a small value. According to Equations. (23) and (24), the current filtering divergence is determined based on whether it satisfies the convergence conditions. If the conditions are met, the Sage-Husa adaptive Kalman filter is updated. Otherwise, the degree of camera vibration was calculated using Equation (22), and the range of R_k was limited in conjunction with the degree of camera vibration to suppress the possibility of filtering divergence. This ensures the appropriate size of the filtering adaptability while also ensuring the stability and effectiveness of smoothing.

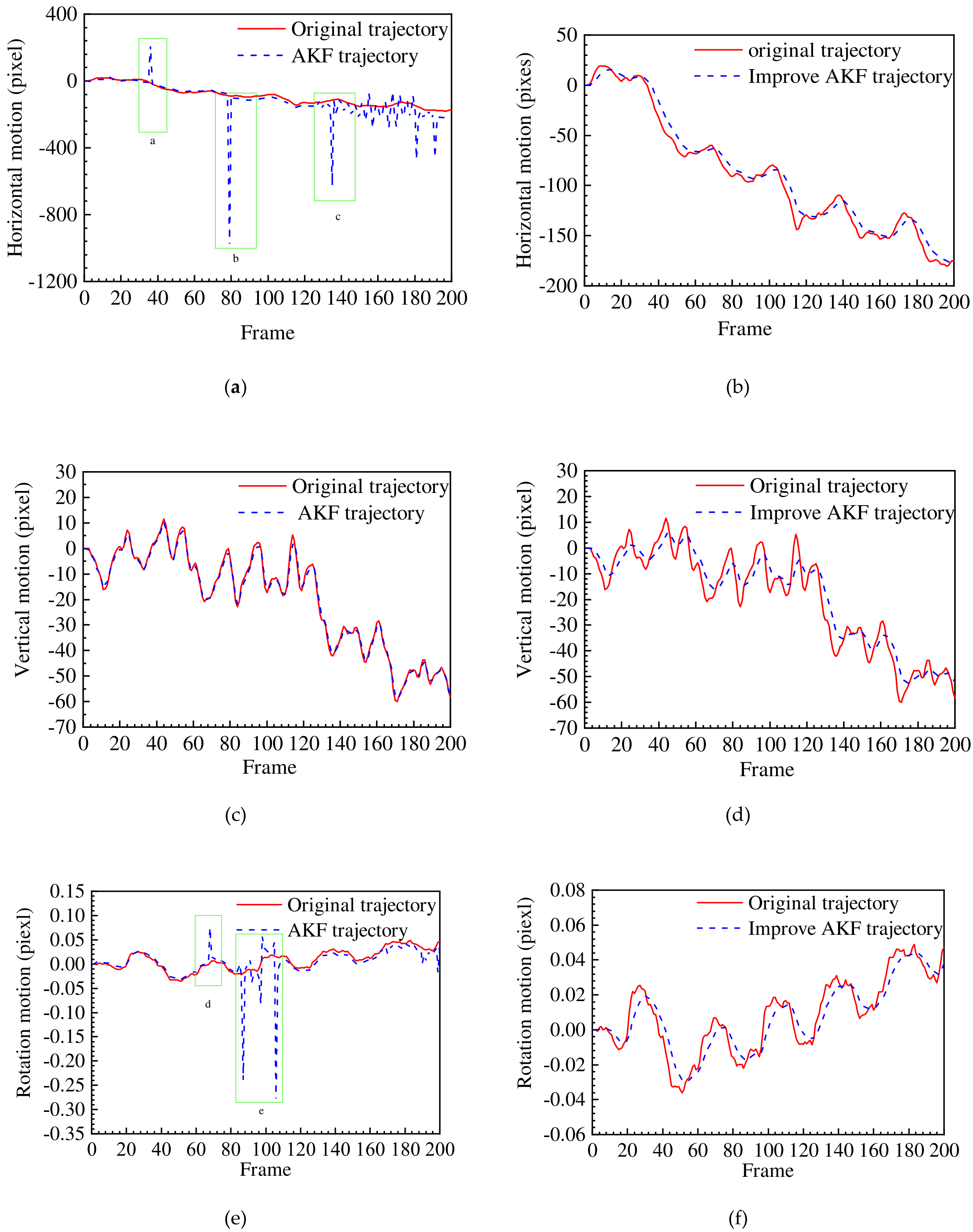


FIGURE 6. **AKF filtering effect before and after improvement. (a) Horizontal motion trajectory filtered by AKF; (b) Improved AKF filtered horizontal motion trajectory; (c) Vertical motion trajectory filtered by AKF; (d) Improved AKF filtered vertical motion trajectory; (e) Rotation motion trajectory filtered by AKF ;(f) Improved AKF filtered rotation motion trajector**

V. TEST AND ANALYSIS

To verify the effectiveness of the algorithm in this study, an Automated Guided Vehicle (AGV) image acquisition system was used to collect jittery video sequences of its movement in a coal mine tunnel. Because the coal mine environment is relatively special, explosion-proof requirements exist for all the electronic devices. The structure of the AGV car is shown in Fig.7, with explosion-proof cameras on both sides of the vehicle body, and lighting equipment in the middle. The experimental conditions used for the tests are illustrated in Fig.8. From Fig.8, it can be observed that the road conditions of the coal mine lane are relatively harsh, and the vehicle is prone to jitter while driving. The jitter video captured by the onboard camera was processed using Opencv and the VSCode programming tool on a PC with an Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz. The algorithm proposed in this study was implemented in Python, and a stability algorithm based on grid motion statistics and adaptive Kalman filtering was validated.

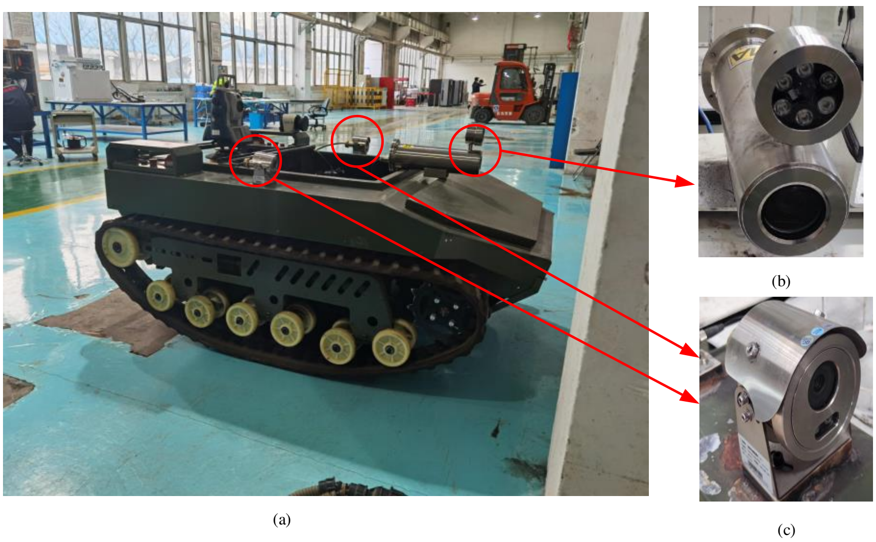


FIGURE 7. Experimental equipment; (a) driverless vehicle equipped with an image acquisition system. (b) Lighting equipment with explosion-proof function, (c) Camera with explosion-proof function.

FIGURE 8. Experimental environment; (a) Driverless vehicle driving tests. (b) Road conditions in coal mine roadways.

A. FEATURE POINTS OFFSET ANALYSIS BEFORE AND AFTER STABILIZATION

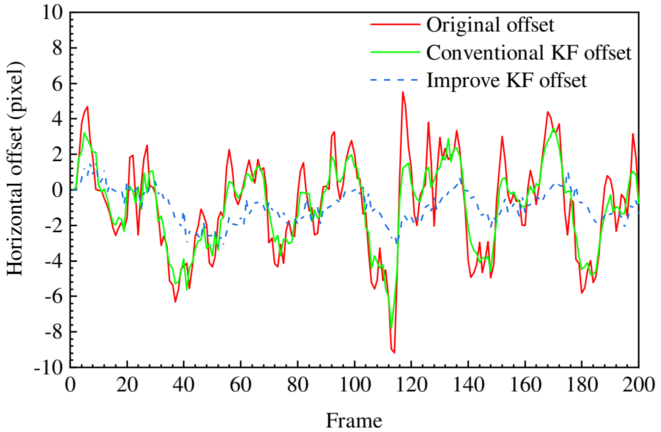
After stabilizing the shaky video acquired by the vehicle camera, the image stabilization effect was evaluated using the offset of the image feature points before and after the stabilization. Image feature points are often located at key locations in a video image, such as the edges, corners, and centers. The position data of the feature points were presented in the video image before video stabilization. The positional data of the feature points vary with the video frame. To stabilize a shaky video, it is necessary to first identify the image feature points and calculate their position information of those feature points. Algorithms for video image processing can identify and extract the position data of feature points before stabilizing the video. Before stabilizing the video, the offset of the feature points can be calculated using their location information of feature points once they have been identified and extracted. The offset of the feature points before stabilizing the video is shown in Equation (25), where p_{xi} and p_{yi} represent the positions of the image feature points in the video sequence of frames i and j, respectively. After calculating the offset of the feature points before stabilizing the video, the same principle is used to calculate the offset of the image feature points after stabilizing the video.

$$\Delta x_{ij} = \left| p_{xi} - p_{yj} \right| \tag{25}$$

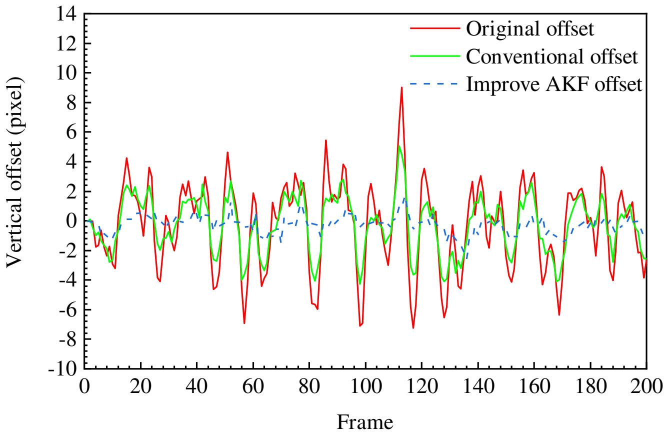
The feature point offset between the corresponding positions in successive images in a shaking video series reflects the intensity of shaking. Shaking decreased as the feature point offset decreased. Table 2 and Fig.9 show the feature point offsets in the horizontal, vertical, and rotational directions before and after stabilization of the shaking video sequence. According to Fig. 9, the feature point offset trajectory swings about the horizontal axis, and the more intense the camera platform shakes, the higher is the fluctuation. The feature point offset is substantially smaller and typically smoother after image stabilization than before, and is much closer to zero. The mean feature point offset in the horizontal direction decreased by 52%, mean feature point offset in the vertical direction decreased by 79%, and mean pixel deviation in the rotational direction decreased by 75% after the shaking video sequence was processed using the proposed stabilization algorithm. The video processed using the algorithm in this study had a 41% decrease in horizontal mean deviation, 69% decrease in vertical mean deviation, and 68% decrease in rotational mean deviation when compared to the jitter video smoothed using the conventional Kalman filtering algorithm. The experiment shows that the proposed method has a good impact on stability and successfully lowers the amount of shaking in the video sequence.

TABLE2. ANALYSIS OF MEAN OFFSET OF CORRESPONDING FEATURE POINTS BEFORE AND AFTER STABILIZATION

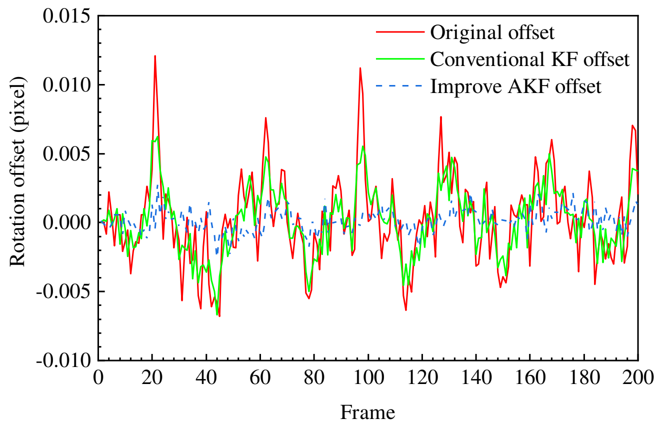
	Horizontal offset/pixel	**Vertical offset/pixel**	**Angular offset/pixel**
Before stabilization	2.28	2.32	0.0024
After conventional KF stabilization	1.86	1.55	0.0019
After improving AKF stabilization	1.08	0.48	0.0006



(a)



(b)



(c)

FIGURE 9. Feature point offset analysis before and after video stabilization. (a) the feature point offsets horizontally; (b) the feature point offsets vertically; (c) the feature point offset in the rotation

B. SIMILARITY ANALYSIS BETWEEN FRAMES BEFORE AND AFTER STABILIZATION

The peak signal-to-noise ratio (PSNR) is often used as a quality factor for evaluating stabilization accuracy. This reflects the similarity between adjacent frames of an image sequence after stabilization processing. The similarity between neighboring video frames increases with increasing PSNR, which improves the video stabilization [30]. The PSNR was calculated using Equation. (26).   represents two adjacent frames of images and MSE is the Mean Square Error, which represents the difference between the two images. The MSE calculation is given by Eq. (27).

$$PSNR\left(f_k, f_{k-1}\right) = 10\lg \frac{255^2}{MSE\left(f_k, f_{k-1}\right)} \tag{26}$$

$$MSE\left(f_k, f_{k-1}\right) = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} \left(f_k\left(m,n\right) - f_{k-1}\left(m,n\right)\right)^2 \tag{27}$$

In order to verify the effectiveness of the algorithm, we selected the classic YouTube Stabilizer[20] and the vehicle-mounted fast video stabilization algorithm that uses Kalman filtering for motion smoothing[31] to compare. Table 3 presents the mean PSNR values of the different stabilization algorithms before and after the processing. The experiment revealed that the stabilization procedure significantly increased the PSNR values over the original videos, indicating that the video became more stable and the disparity in grayscale across video sequences decreased. The PSNR mean values of the jittery videos processed by the algorithms in references [20] and references [31] were 28.83 dB and 27.63 dB, respectively, which were increased by 9.18 dB and 8.48 dB compared to the original video. The PSNR mean value of the jittery video sequence processed by the algorithm proposed in this article was increased by 10.77 dB compared to the original video, which was increased by 1.09 dB and 2.29 dB compared to the algorithms in [20] and [31], respectively. Therefore, the algorithm proposed in this study is more suitable for processing jittery video images on mobile car platforms in underground coalmines.

TABLE 3. COMPARISON OF PSNR MEAN VALUES AFTER IMAGE STABILIZATION BY DIFFERENT ALGORITHMS

	Original video	YouTube [20]	KF [31]	Proposed method
PSNR mean (db)	19.15	28.83	27.63	29.92

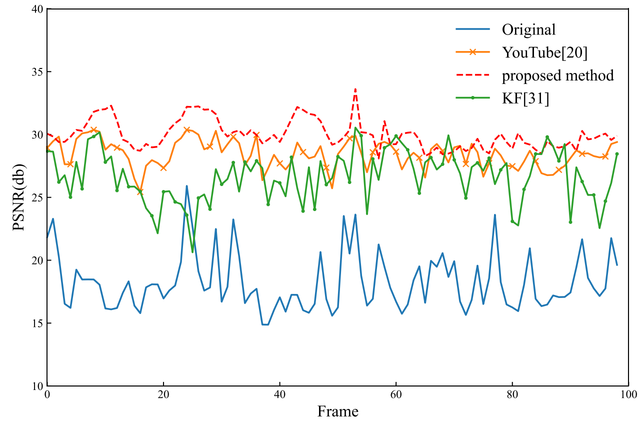


FIGURE 10. Comparison of PSNR mean values after image stabilization by different algorithms.

VI. CONCLUSIONS

A video stabilization technique based on grid motion statistics and adaptive Kalman filtering was proposed to address the problem of image blur in automotive video-monitoring systems. The algorithm was applied to collect video data from coal mine tunnels by using an AGV mobile car. Methods for feature point recognition and matching have been researched based on the properties of automobile video data. The research focused on how to better separate the jitter noise during motion and obtained the following conclusions:

(1) The accuracy and real-time performance of the stabilization were enhanced by employing a fusion GMS algorithm to reduce erroneous matching. The research revealed that the feature matching method with fusion GMS boosted the matching accuracy compared to the conventional SIFT and ORB matching algorithms by 2.3% and 4.1%, respectively, while decreasing the matching time between adjacent video frames by 76% and 16%, respectively.

(2) Using an improved adaptive Kalman filter algorithm based on Sage-Husa effectively improved the stabilization effect of jitter videos. The experiment showed that the PSNR mean value of the stabilized jitter video increased by 10.27 dB, thereby proving the effectiveness and correctness of the proposed algorithm.

Underground images of a coal mine are characterized by low illumination and high dust. Therefore, the quality of the acquired images is generally poor. To obtain a better video image stabilization effect, we plan to integrate on-board IMU information based on this algorithm to achieve video image stabilization.

Declarations

Ethical approval All applicable institutional and/or national guidelinesfor the care and use of animals were followed.

Funding This work was supported by the Shanxi Province Key R&D Program under Grant (2020XXX001), National Key R&D Program under Grant (2020YFB1314003).

Conflict of interest The authors declare that they have no conflict of interest.

Availability of data and materials

Data availability The raw data can be shared if the researchers need to do research on relevant topic and cite it in their papers

Code availability The code can be shared in the near future for the sake of development

REFERENCES

[1] Cui. Y, Liu. S. Yao, J.; Gu, C. “Integrated positioning system of unmanned automatic vehicle in coal mines,” *IEEE Transactions on Instrumentation and Measurement*, vol.70, pp.1-13,2021

[2] Wang G, Ren H, Zhao G, "Research and practice of intelligent coal mine technology systems in China", *International Journal of Coal Science & Technology*, vol.9, pp.1-24, 2022.

[3] Liang Z, "Thoughts on the Development Status and Application Direction of the Driverless Industry", *International Journal of Scientific Engineering and Science*, vol. 4, no.3, pp.15-16, 2020.

[4] Auysakul J, Xu H, Pooneeth V, "A hybrid motion estimation for video stabilization based on an IMU sensor", *Sensors*, vol.18, no.8, pp.2708, 2018.

[5] Song M G, Baek H W, Park N C, "Development of small sized actuator with compliant mechanism for optical image stabilization", *IEEE Transactions on Magnetics*, vol.46, no.6, pp.2369-2372, 2010.

[6] Song M G, Hur Y J, Park N C, "Design of a voice-coil actuator for optical image stabilization based on genetic algorithm", IEEE Transactions on Magnetics, vol.45, no.10, pp.4558-4561, 2009.

[7] M R E S, Maia H A, Pedrini H, "Survey on digital video stabilization: concepts, methods, and challenges", ACM Computing Surveys, vol.55, no.3, pp.1-37, 2022.

[8] Yu J, Ramamoorthi R, "Selfie video stabilization," in ECCV, 2018, pp.551-566

[9] Wu R, Xu Z, Zhang J, "Robust global motion estimation for video stabilization based on improved k-means clustering and superpixel", Sensors, vol.21 , no.7, pp.2505, 2021.

[10] Liu F, Gleicher M, Jin H, "Content-preserving warps for 3D video stabilization", ACM Transactions on Graphics, vol.28, no.3, pp.1-9, 2009.

[11] Jeon S, Yoon I, Jang J, "Robust video stabilization using particle keypoint update and l 1-optimized camera path", Sensors, vol.17, no.2, pp.337, 2017.

[12] Shujiao J, Ming Z, Yanmin L, "Video stabilization with improved motion vector estimation", Optics and Precision Engineering, vol.23, no.5, pp.1458-1465, 2015.

[13] Kejriwal L, Singh I, "A hybrid filtering approach of digital video stabilization for UAV using kalman and low pass filter", Procedia Computer Science, vol.93, pp.359-366, 2016.

[14] RodriguezPadilla I, Castelle B, Marieu V, "A simple and efficient image stabilization method for coastal monitoring video system", Remote sensing, vol.12, no.1, pp.70, 2019.

[15] Zhong B, Li Y, " Image feature point matching based on improved SIFT algorithm," in 2019 IEEE 4th International Conference on Image, Vision and Computing, , 2019, pp.489-493

[16] Shene T N, Sridharan K, Sudha N, "Real-time SURF-based video stabilization system for an FPGA-driven mobile robot", IEEE Transactions on Industrial Electronics, vol.63, no.8, pp.5012-5021, 2016.

[17] Ling Q, Deng S, Li F, "A feedback-based robust video stabilization method for traffic videos", IEEE Transactions on Circuits and Systems for Video Technology, vol.28, no.3, pp.561-572, 2016.2016.

[18] Ren Z, Chen C, Fang M, "Electronic image stabilization algorithm based on smoothing 3D rotation matrix," in IEEE International Conference on Computer and Communications, 2017, pp.2752-2755.

[19] M R E S, Maia H A, Pedrini H, "Survey on digital video stabilization: concepts, methods, and challenges", ACM Computing Surveys, vol.55(3), pp.1-37, 2022.

[20] Grundmann M, Kwatra V, Essa I, " Auto-directed video stabilization with robust l1 optimal camera paths", CVPR, pp.225-232, June.2011.

[21] Qu H, Song L, "Video stabilization with L1–L2 optimization," in IEEE International Conference on Image Processing,2013, pp.29-33

[22] Souza M R, Pedrini H, "Digital video stabilization based on adaptive camera trajectory smoothing", EURASIP Journal on Image and Video Processing, vol.2018, pp.1-11, 2018.

[23] Liu S, Yuan L, Tan P, "Bundled camera paths for video stabilization", ACM transactions on graphics, vol.32, pp.1-10, 2013.

[24] Zhai B, Zheng J, Li B, "Digital image stabilization based on adaptive motion filtering with feedback correction", Multimedia Tools and Applications, vol.75, pp.12173-12200, 2016.

[25] Mukherjee D, M J W Q, Wang G, "A comparative experimental study of image feature detectors and descriptors", Machine Vision and Applications, vol.26, pp.443-466, 2015.

[26] Bansal M, Kumar M, Kumar M, "2D object recognition: a comparative analysis of SIFT, SURF and ORB feature descriptors", Multimedia Tools and Applications, vol.80, pp.18839-18857, 2021.

[27] Bian J W, Lin W Y, Matsushita Y, "Grid-based motion statistics for fast, ultra-robust feature correspondence," in Proceedings of the IEEE conference on computer vision and pattern recognition,, 2017, pp.4181-4190

[28] Dong J, Liu H, "Video stabilization for strict real-time applications", IEEE Transactions on Circuits and Systems for Video Technology, vol. 27(4), pp.716-724, 2016.

[29] Narasimhappa M, Mahindrakar A D, Guizilini V C, "MEMS-based IMU drift minimization: Sage Husa adaptive robust Kalman filtering", IEEE Sensors Journal, vol.20(1), pp.250-260, 2019.

[30] Souza M R, Pedrini H, " Digital video stabilization: algorithms and evaluation," in Anais Estendidos do XXXII Conference on Graphics, Patterns and Images, SBC, 2019, pp.35-41

[31] Wu-Chih Hu, Chao-Ho Chen, Tsong-Yi Chen, Min-Yang Peng, and Yi-Jen Su, "Real-Time Video Stabilization for Fast-Moving Vehicle Cameras," Multimedia Tools and Applications, vol. 77, pp. 1237-1260, Jan 2018.