

# Research on Video Electronic Stabilization Technology Based on ShiTomasi Corner Detection

Yong Deng, Yutao Zhang

School of Mechatronic Engineering, Southwest Petroleum University, Chengdu 610500, China

**Abstract:** Video stabilization technology based on ShiTomasi corner detection has broad applications in intelligence surveillance, medical examination, and security monitoring. The research on video stabilization technology is crucial due to the inevitable mechanical vibrations affecting embedded cameras during use, leading to video jitter and compromising image quality. This paper proposes a video stabilization scheme based on the ShiTomasi corner detection method. The ShiTomasi algorithm is robust and accurate, effectively extracting corner features in videos. To further improve stabilization accuracy, a new weighted voting mechanism is designed, combining three-dimensional directional information to avoid mismatches in corner matching and enhance the accuracy of block matching. Specifically, this paper employs an adaptive search area and a multi-level feature detection method to process each video frame, ensuring the accuracy and robustness of corner detection. Additionally, an improved Kalman filter is introduced to reduce the impact of mismatched points on the stabilization results. In summary, the ShiTomasi algorithm-based video stabilization system designed in this paper not only enhances the speed and efficiency of video processing but also improves the precision and stability of stabilization.

**Keywords:** ShiTomasi Corner Detection; Block Matching; Kalman Filter.

## 1. Introduction

Electronic image stabilization technology mainly consists of three major steps: motion estimation, motion filtering, and image compensation. The system flowchart for stabilization is shown in Figure 1. Motion estimation is the calculation of

the global motion of the imaging device. Due to the impact of irregular and random vibrations on the imaging device, the computed global motion needs to be processed through motion filtering to separate intentional scanning motion from unintentional random jitter. In the image compensation stage, by eliminating the random jitter, a stable output video frame sequence is achieved.

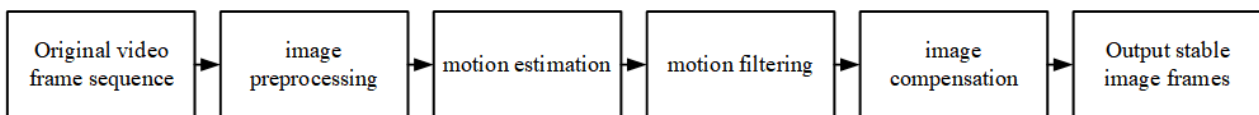


Fig 1. System Flowchart of Electronic Image Stabilization Technology

## 2. Algorithm Principle

The algorithm achieves video stabilization through a series of technical measures. First, it uses the ShiTomasi algorithm to detect corner points in the previous frame and finds the corresponding corner points in the current frame through block matching. Then, it estimates the global offset between the two frames and uses Kalman filtering to smooth the predicted offset of the current frame, reducing the impact of noise. By combining the accumulated offsets from previous frames, the motion compensation for the current frame is calculated. Finally, the current frame is translated to eliminate jitter, resulting in a stabilized image frame. Through these methods, the algorithm effectively eliminates video jitter, enhancing the stabilization effect and system performance.

### 2.1. Algorithm Composition

#### 2.1.1. Corner Detection

Detecting the corner points  $kp1$  in the previous frame (preFrame). In the first step of video stabilization, it is necessary to detect small regions with significant features from the previous frame image (preFrame). These regions are referred to as corner points. Corner point detection is implemented using the ShiTomasi corner detection algorithm. The ShiTomasi algorithm is an improved version of the Harris

corner detection method. It calculates the grayscale variation matrix for each pixel in the image and sorts the eigenvalues of the matrix, selecting the larger eigenvalues as corner points. These corner points can maintain relatively stable positions and shapes in consecutive frames of the video and are referred to as  $kp1$ .

#### 2.1.2. Block Matching

Based on the corner points detected in the preFrame, corresponding points  $kp2$  in the current frame (curFrame) are found using the block matching method (excluding vanishing points). Block matching involves searching for blocks in the curFrame that are similar to the regions around the corner points in the preFrame. The size of the search area can be adaptively adjusted to improve the accuracy of the matching. Through this method, changes in the positions of the corner points due to motion can be eliminated, while excluding the influence of vanishing points.

#### 2.1.3. Global Offset Estimation

Using the obtained  $kp1$  and  $kp2$ , estimate the global offset  $dx$  and  $dy$  between the preFrame and curFrame. By calculating the displacement vectors between  $kp1$  and  $kp2$  and statistically processing all the vectors, the average offset  $dx$  and  $dy$  for the entire image is obtained. This offset reflects the camera's motion changes between the two frames.

### 2.1.4. Kalman Filtering

Using Kalman filtering, obtain the predicted values transX and transY for dx and dy of the curFrame. To further smooth the global offset and reduce noise impact, Kalman filtering is introduced. Kalman filtering is a recursive algorithm that estimates the system state by combining the system's motion model and measurements. In video stabilization, Kalman filtering uses the offsets dx and dy from the previous frames to predict the offset for the current frame, resulting in the predicted values transX and transY. This prediction effectively reduces jitter caused by mismatches or noise.

### 2.1.5. Motion Compensation

Using the predicted values transX and transY along with the accumulated values of dx and dy from all previous frames before curFrame, sum\_transX and sum\_transY, obtain the motion compensation for the current frame, diff\_transX and diff\_transY. By combining the Kalman filter-predicted transX and transY with the accumulated offsets sum\_transX and sum\_transY from all previous frames, the motion compensation amounts diff\_transX and diff\_transY for the current frame can be obtained. This step ensures that the motion compensation not only considers the predicted offset of the current frame but also incorporates the historical motion information of the entire video sequence, making the

compensation more accurate and smooth.

### 2.1.6. Translation Transformation

Using diff\_transX and diff\_transY, perform a translation transformation on curFrame to obtain smoothFrame. Finally, based on the calculated motion compensation amounts diff\_transX and diff\_transY, perform a translation transformation on the current frame (curFrame). This step translates curFrame along the direction of the compensation amounts, thereby eliminating the jitter caused by camera movement and producing the stabilized image frame smoothFrame. This translation transformation process can effectively improve the visual stability of the video, enhancing the viewing experience and the accuracy of subsequent processing.

### 2.1.7. Summary

Through the above steps, the video stabilization system proposed in this paper utilizes the ShiTomasi corner detection algorithm, block matching, and an improved Kalman filter to successfully eliminate svideo jitter, enhancing the stabilization effect and the overall performance of the system.

## 3. Algorithm Design

### 3.1. Algorithm Structure

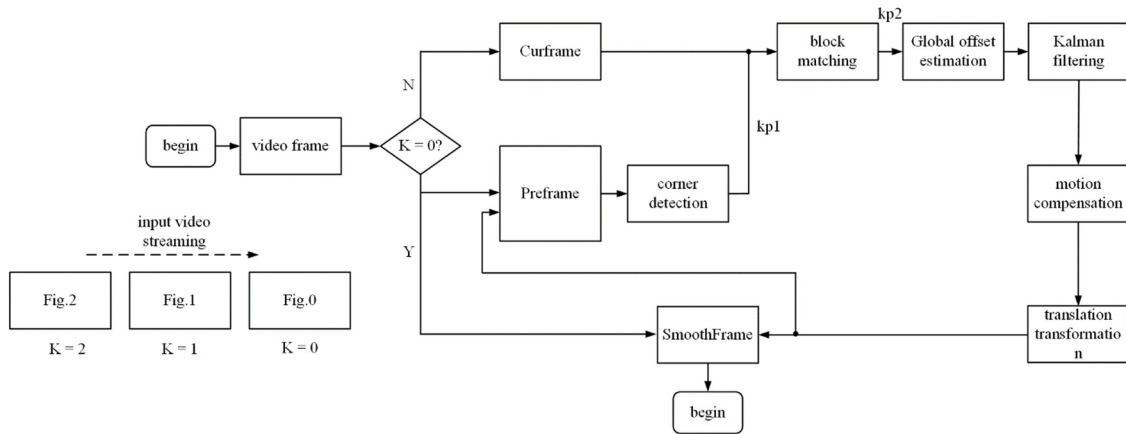


Fig 2. Algorithm Flowchart

Note: In the document, all indices of images and vectors start from 0. The height and width of the image are denoted by height and width, respectively.  $I(y, x)$  represents the pixel value of the image  $I$  at position  $(y, x)$ , where  $y$  is the row coordinate and  $x$  are the column coordinate.

### 3.2. Corner Detection

Use the Shi-Tomasi algorithm to detect corner points in preFrame. The specific calculation process is as follows:

#### 3.2.1. Gradient Calculation and Maximum Gradient Value Computation

Use the Scharr operator to compute the gradients in the dx and dy directions:

$$\text{kernel}_x = \begin{bmatrix} -3 & 0 & 3 \\ -10 & 0 & 10 \\ -3 & 0 & 3 \\ -3 & -10 & -3 \\ 0 & 0 & 0 \\ 3 & 10 & 3 \end{bmatrix} \quad \text{kernel}_y =$$

The calculation formulas are as follows, with zero-padding at the boundaries.

$$\text{gradient}_x(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 \text{preFrame}(y + i, x + j) \times \text{kernel}_x(i + 1, j + 1)$$

$$\text{gradient}_y(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 \text{preFrame}(y + i, x + j) \times \text{kernel}_y(i + 1, j + 1)$$

The entire process is illustrated in the diagram below:

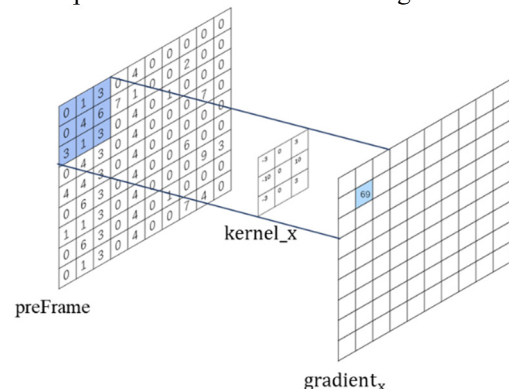


Fig 3. Diagram illustrating the gradient calculation in the x direction

$\text{gradient}_x(x, y)$ ,  $\text{gradient}_y(x, y)$  Represent the gradients in the x and y directions at the point(x, y).

### 3.2.2. Calculation of the Corner Response Function R

Calculate the Shi-Tomasi corner response value R at (x, y) using the  $\text{gradient}_x$  and  $\text{gradient}_y$  obtained in (1). The corner response function is as follows:

$$R(x, y) = (a + c) - b^2 - (a - c)^2$$

where a, b and c are defined as follow:

$$a = \text{fix}\left(\frac{\text{gradient}_x(x, y) \times \text{gradient}_x(x, y) * 2^{20}}{256 * 16 * 256 * 16}\right)$$

$$b = \text{fix}\left(\frac{\text{gradient}_x(x, y) \times \text{gradient}_y(x, y) * 2^{20}}{256 * 16 * 256 * 16}\right)$$

$$c = \text{fix}\left(\frac{\text{gradient}_y(x, y) \times \text{gradient}_y(x, y) * 2^{20}}{256 * 16 * 256 * 16}\right)$$

### 3.2.3. Interval Maximum Value Calculation and Selection and Global Corner NMS

**Interval Maximum Value Calculation:** Divide the R map into 10 (rows) \* 20 (columns) intervals. Calculate one maximum value for each interval. If the maximum value is not zero, it is considered a potential strong corner point. If the row and column coordinates of this maximum value are respectively.

[search\_range+patch\_radius, height search\_rangepatch\_radius], [search\_range+patch\_radius, width search\_rangepatch\_radius], then it is considered not to be a boundary corner point, and this point is stored in pool\_corners (an array). The search\_range is the search radius (initially set to 16), and the patch\_radius is the neighborhood radius of the matching block (initially set to 3).

**Selection:** Find the maximum value of the corner response function in pool\_corners, then perform threshold segmentation on pool\_corners based on the set minimum quality level of corners (qualityLevel). Points with corner response function values less than the threshold are set to 0, and points with values greater than or equal to the threshold remain unchanged. The threshold is determined by the following formula:

$$T = \text{fix}\left(\frac{102 \times \text{maxvalue}_R}{1024}\right)$$

where the value of qualityLevel is 102/1024 (0.1), and maxvalue\_R represents the maximum value of the corner response function in pool\_corners. Then, store the corners with response function values greater than the threshold T into coarse\_corners (an array) and sort coarse\_corners in ascending order based on the y-coordinate.

**NMS:** The search ranges of the corners stored in coarse\_corners may overlap. Use the NMS (Non-Maximum Suppression) algorithm to eliminate overlapping points. The specific process is as follows:

1) Take the i-th corner from coarse\_corners, where the value of i-th ranges from 0 to coarse\_corners.size() - 1. The coordinates of the i-th corner are (x1, y1), and initialize a coordinate point m\_corners = (x1, y1).

2) Check if R (x1, y1) is equal to 0. If it is not equal to 0, then proceed with the following steps: 3). If it is equal to 0, set m\_corners to (1, 1), and then jump to 5)

3) Check if the j-th corner in coarse\_corners and. 1) Check the distance between the i-th corner taken from coarse\_corners and the j-th corner in coarse\_corners, where j ranges from i+1 to coarse\_corners.size() - 1. The coordinates of the j-th corner are (x2, y2). If the distance between the two corners satisfies the following condition, then execute:4), If the following condition is not met, then j++ and repeat 3):

$$|x1x2| \leq 2 * \text{search\_range}, |y1y2| \leq 2 * \text{search\_range}$$

(Both of the following conditions are satisfied simultaneously)

4) If R (x1, y1) > R (x2, y2), then set R (x2, y2) equal 0, j++, return 3), else set R(x1,y1) equal 0, and set m\_corners equal (1,1), jump to 5)

5) Check if m\_corners is equal to (1, 1). If it is not equal, then m\_corners is a true strong corner and should be stored in corners (an array for storing strong corners). If m\_corners is equal to (1, 1), it indicates that this point is not a true strong corner and it should not be stored. Finally, increment i. If i is greater than corners.size() - 1, the NMS algorithm is complete; otherwise, return and repeat the process. 1)

### 3.3. Block Matching

Based on the corner points detected in the preFrame, the coordinates in the curFrame are calculated through block matching (excluding the vanishing points). The detailed calculation process is as follows:

(1) Define a data type vector<P\_N> named kpn2, where P\_N is a structure containing three variables: two Point type variables p1 and bestmatch, and an int type variable sad. The initial value of sad is 12496. kpn2[i].p1 stores corners[i], and bestmatch is the best matching point for corners[i].

(2) Centered at kpn2[i].p1, perform block matching row by row. Traverse curFrame line by line, and for each line, check if it participates in the block matching of the point kpn2[i].p1 (i = 0, 1, 2, ... kpn2.size()-1). If the line participates in the block matching calculation for the point kpn2[i].p1, perform the following calculations:

1. Extract image blocks from preFrame and curFrame, denoted as patch1 and patch2 respectively. patch1 is the rectangular region centered at kpn2[i].p1 in preFrame with dimensions (2\*patch\_radius+1) \* (2\*patch\_radius+1). patch2 is the rectangular region centered at the matching point participating in the block calculation in curFrame, also with dimensions (2\*patch\_radius+1) \* (2\*patch\_radius+1).

2. Calculate the SAD value between patch1 and patch2. If the SAD value is less than kpn2[i].sad, assign the calculated SAD value to kpn2[i].sad, and assign the coordinates of the matching point to kpn2[i].bestmatch. The SAD calculation is as follows (with SAD values ranging from 0 to 12495):

$$SAD = \sum_{i=0}^n \sum_{j=0}^n |patch1(i, j) - patch2(i, j)|$$

After completing the traversal of the entire image, the matches stored in 'kp2.bestmatch' are the matching points for 'kp2.p1'. The matching process is shown below:

In Fig.4, the red area in curFrame represents the search range for points in preFrame, and the green area represents the patch block used for block matching. For each row in curFrame, we determine whether the row falls within the search range of any point in corners. If it is not within the search range of any point in corners, we move to the next row. If the row is within the search range of some points in corners, we match the corresponding patch blocks patch1 of these points with the matching points in the row. We then compare the calculated SAD values with the existing SAD values of these points. If the calculated SAD value is smaller than the original SAD, we assign the coordinates of the matching point to bestmatch for the corresponding point and update the SAD value for that point accordingly.

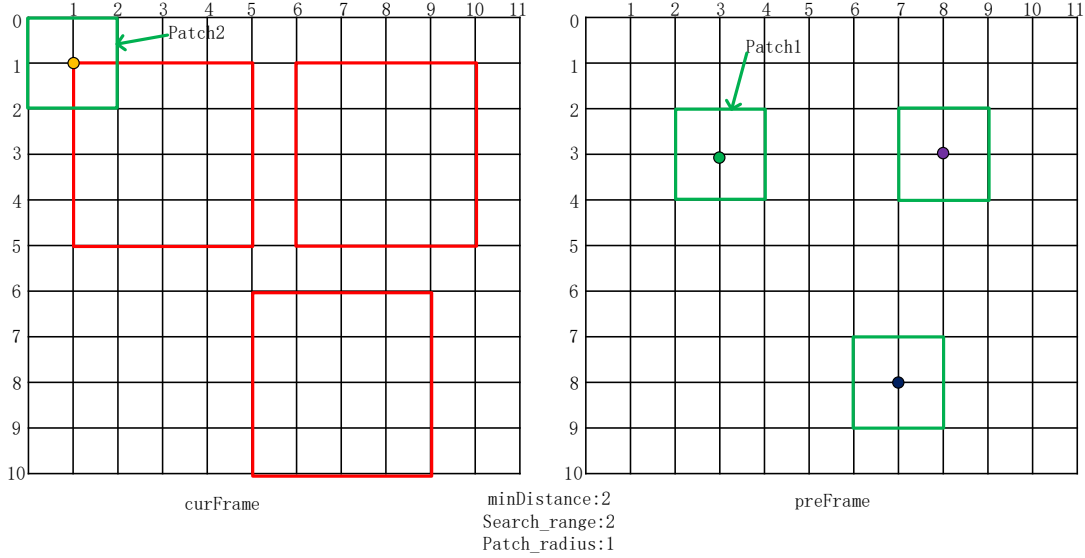


Fig 4. block matching

### 3.4. Global Offset Estimation

First, define two arrays vector int dx and vector int dy to store the offsets between successfully matched corner points:

```
dx_.pushback( kp2.bestmatch.x kp2.p1.x)
dy_.pushback( kp2.bestmatch.y kp2.p1.y)
```

Next, perform a 1D median filtering of size 3 on dx\_ and dy\_. This involves applying a median filter independently to each array dx\_ and dy\_ with a window size of 3.

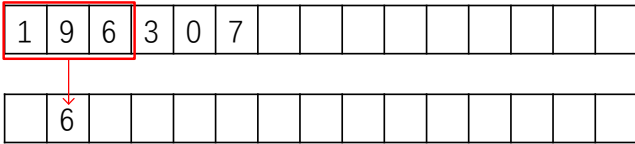


Fig 5. Median Filtering

Finally, calculate the average of the arrays dx\_ and dy\_ after median filtering to obtain the global offsets dx and dy between preFrame and curFrame.

$$dx = \text{round}\left(\frac{\text{sum}(dx\_)}{\text{size}(dx\_)} * 2^{15}\right)$$

$$dy = \text{round}\left(\frac{\text{sum}(dy\_)}{\text{size}(dy\_)} * 2^{15}\right)$$

### 3.5. Kalman Filtering

The specific calculation process of Kalman filtering is as follows:

In Kalman filtering, the first frame is considered to be a stable frame, which is not processed and starts from the second frame.

### 3.6. Motion Compensation

Kalman filtering will output the optimal estimated values of dx and dy for curFrame, which are the updated transX and transY. Since transX and transY are the optimal estimates relative to the first frame, the following calculation is performed to obtain the motion compensation relative to preFrame:

$$\text{diff\_transX} = \text{round}\left(\frac{\text{transX} - \text{sum\_transX}}{2^{15}}\right)$$

$$\text{diff\_transY} = \text{round}\left(\frac{\text{transY} - \text{sum\_transY}}{2^{15}}\right)$$

diff\_transX, diff\_transY (This provides the motion compensation for the horizontal and vertical directions.)

### 3.7. Translation Transformation

Translate the current frame curFrame using the motion-compensated diff\_transX and diff\_transY. If diff\_transX or diff\_transY exceeds Limit\_radius, truncate them to stay within the specified range.

$$\text{diff\_transX} \in [-\text{Limit\_radius}, +\text{Limit\_radius}]$$

$$\text{diff\_transY} \in [-\text{Limit\_radius}, +\text{Limit\_radius}]$$

The process of translation transformation is as follows,  $x' \in [0, \text{width}], y' \in [0, \text{height}]$ :

$$x' = x + \text{diff\_transX}$$

$$y' = y + \text{diff\_transY}$$

If derived from the above formula,  $x \in [0, \text{width}], y \in [0, \text{height}]$ :

$$\text{smoothFrame}(x', y') = \text{curFrame}(x, y)$$

else:

$$\text{smoothFrame}(x', y') = 0$$

( $x', y'$ ) representing points in smoothFrame., ( $x, y$ ) representing coordinates projected back into curFrame.

## 4. Algorithm Demonstration

Shi-Tomasi corner detection in the first frame: The first image has been processed to detect corner points, which are marked with red dots.

Shi-Tomasi corner detection is used to identify key points or corners in an image, which exhibit good stability and recognizability under geometric transformations. In stereo vision and motion detection, detecting and matching these corners can achieve image registration and object tracking. As shown in the figure, the corners reflect areas in the image that are rich in information, and detecting these corners helps in understanding the geometric structure and content of the image. Compared to Harris corner detection, Shi-Tomasi corner detection is more robust to image noise and can locate corners more accurately.

Block matching corner detection in the second frame: The second image shows the corner points detected through block matching, also marked with red dots.

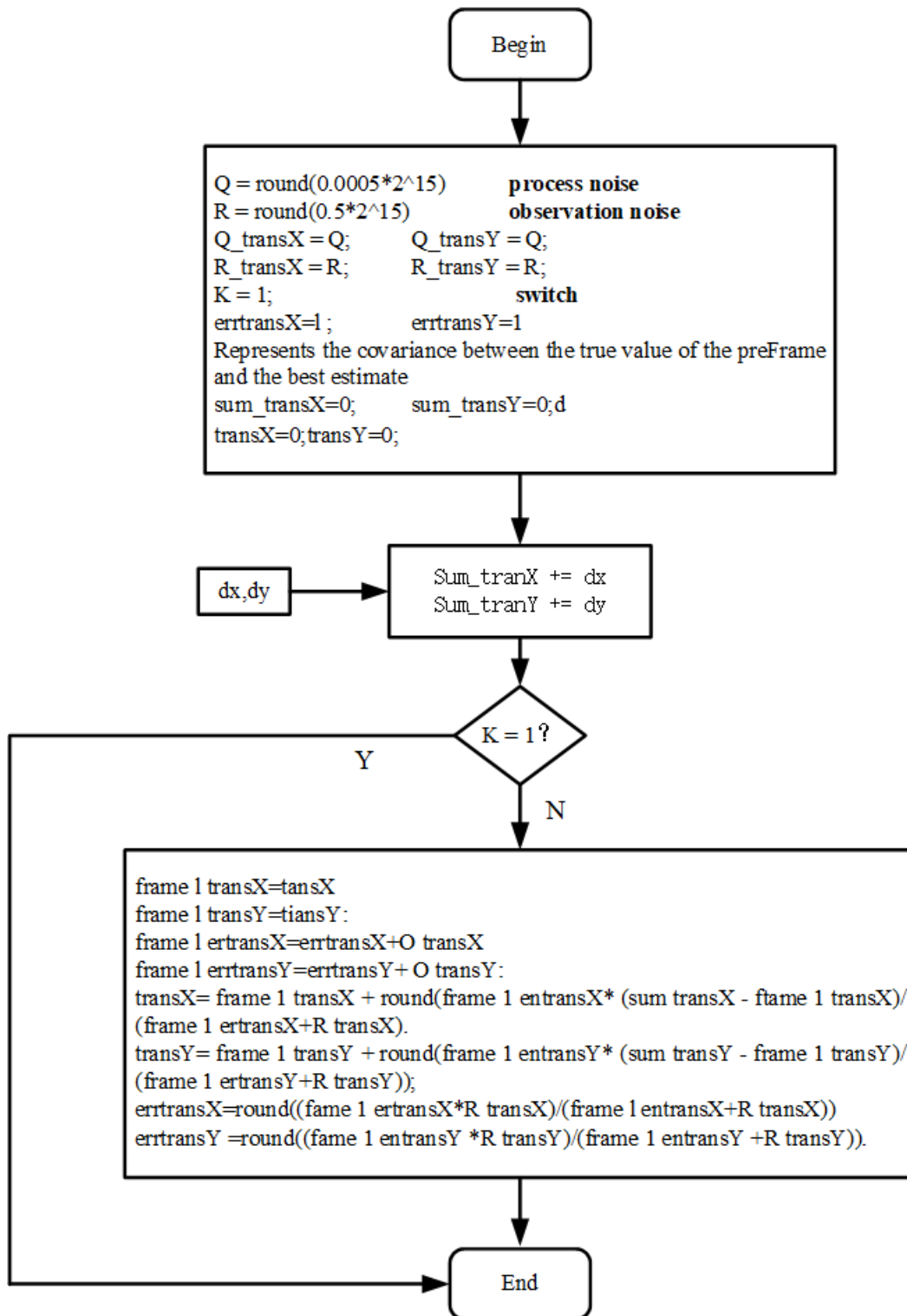


Fig 6. Kalman Filter Process



Fig 7. Reference Frame



Fig 8. Current Frame

Shi-Tomasi Corner Detection with Red Corners



Fig 9. Corner Detection kp1

Block Matching Corner Detection in Second Frame



Fig 10. Block Matching Resulting in kp2

Global Offset Estimation  $dx = 13.78$   $dy = 5.53$   
 $transX = 0.40$ ,  $transY = 0.16$

Smoothed frame after motion compensation: The second frame has been motion compensated based on the predicted offset, resulting in a smoothed image.

Smooth Frame



Fig 11. Smoothed Output Results

## 5. Results Analysis

### 5.1. Analysis of Horizontal and Vertical Profiles Before and After Smoothing

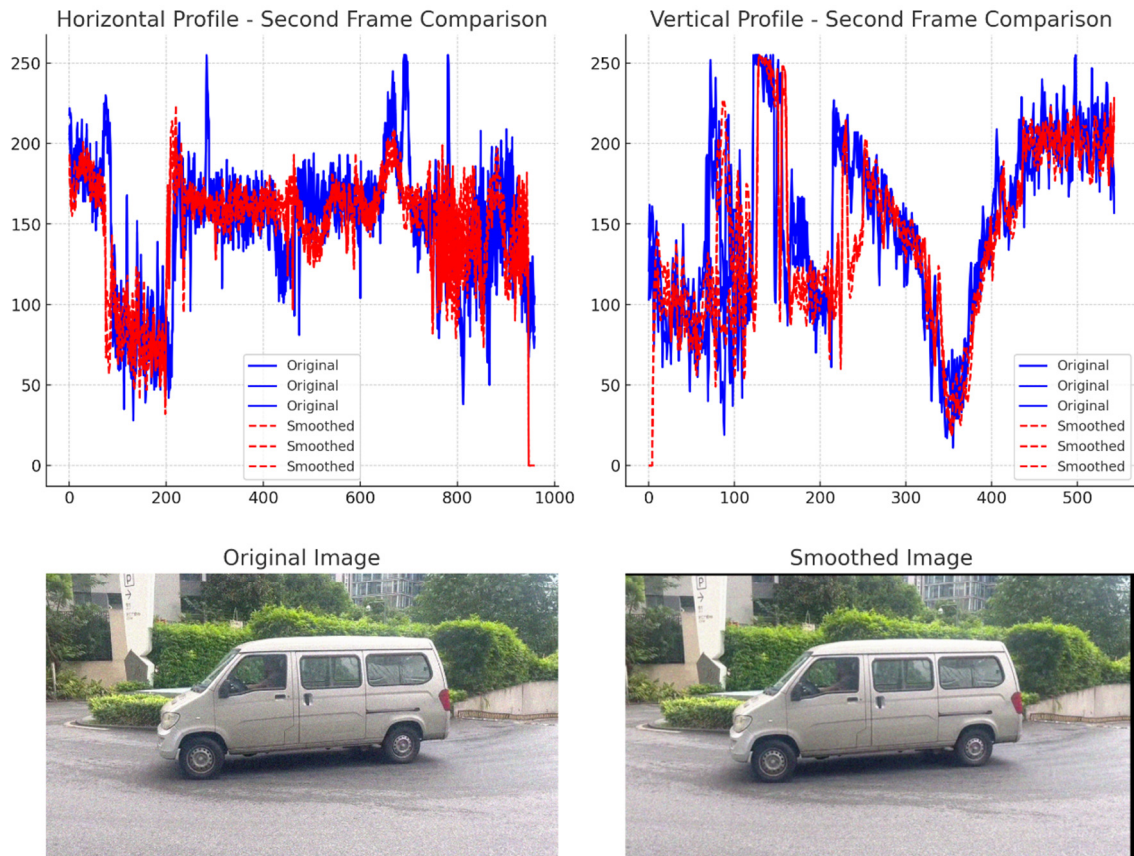


Fig 12. Vertical horizontal analysis diagram

(1) Horizontal Profile:

- Original Image (blue line): The intensity values of the middle row in the original image show significant fluctuations, indicating the presence of prominent edges and noise.

- Smoothed Image (red line): The profile line of the smoothed image is more stable, indicating reduced noise and smoother transitions. The overall pattern is similar, showing

that the main structure has been preserved.

(2) Vertical Profile:

- Original Image (blue line): Like the horizontal profile, the vertical profile of the original image shows noticeable variations, reflecting the presence of edges and noise.

- Smoothed Image (red line): The profile line of the smoothed image is more stable, indicating that the smoothing

process successfully reduced noise while retaining the main features.

## 5.2. Visual Comparison

Original Image: The original image contains visible noise and sharp transitions between different intensity levels.

Smoothed Image: The smoothed image appears more refined with less intensity variation, highlighting the effectiveness of the smoothing process.

## 5.3. Summary

Comparison of the horizontal and vertical profiles indicates that the smoothing process effectively reduced noise while preserving the fundamental features of the image. This is further confirmed by a visual inspection of the original and smoothed images. The main structures and details are retained, but the overall image appears cleaner and more consistent.

## References

- [1] Ning Z, Yuan Y ,Jianhua W , et al. Vehicular Electronic Image Stabilization System Based on a Gasoline Model Car Platform [J]. Chinese Journal of Mechanical Engineering, 2022, 35 (1).
- [2] SHAN X ,PAN M, ZHAO D , et al. Maritime Target Detection Based on Electronic Image Stabilization Technology of Shipborne Camera:Regular Section [J]. IEICE Transactions on Information and Systems, 2021, E104.D (7): 948-960.
- [3] Fabio B, Marco F ,Carlo C , et al. Experiencing with electronic image stabilization and PRNU through scene content image registration [J]. Pattern Recognition Letters, 2021, 145 (prepublish): 8-15.
- [4] Personal Computer Companies; Patent Issued for Combined Optical and Electronic Image Stabilization (USPTO 9596411) [J]. Computer Weekly News, 2017,1357-.
- [5] Patents; "Combined Optical and Electronic Image Stabilization" in Patent Application Approval Process (USPTO 20160360111) [J]. Computer Weekly News, 2016.
- [6] Ho J K . Electronic Image Stabilization for Portable Thermal Image Camera [J]. Journal of the Korea Institute of Military Science and Technology, 2016, 19 (3): 288-293.
- [7] Xiaodong L ,Tianze W ,Jun Z , et al. Experiment research on inertia-aided adaptive electronic image stabilization of optical stable platform [J]. Northwestern Polytechnical Univ. (China); China North Institute of Electronic Equipment (China);China Aerodynamics Research & Development Ctr. (China); Institute of High Energy Physics (China); Shanghai Institute of Technical Physics (China);Shanghai Institute of Optics and Fine Mechanics (China), 2016, 10255 102553R-102553R-6.
- [8] Chao W ,Ming J ,Ying Z , et al. The electronic image stabilization technology research based on improved optical-flow motion vector estimation [J]. Xi'an Institute of Applied Optics (China), 2016, 9796 97962U-97962U-7.
- [9] OmniVision releases 4K video sensor [J]. Telecomworldwire, 2015.
- [10] Zhu J J ,Zheng L W ,Guo L B . An Overview and Analysis of Electronic Stabilization for Cameras on Moving Vehicles [J]. Applied Mechanics and Materials, 2015, 3822 (740-740): 612-618.