Academic Journal of Science and Technology
ISSN: 2771-3032 | Vol. 15, No. 2, 2025

Design and Implementation of an Electronic Image Stabilization System Based on Multi-source Data Fusion

Yong Deng, Yutao Zhang

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

Abstract: With increasingly stringent real-time and stability requirements for high-definition video surveillance in industrial settings such as oil and gas fields, conventional electronic image stabilization (EIS) systems that rely solely on image processing lose accuracy under severe vibration because feature points disappear, and PC-based solutions suffer from high power consumption and latency. This paper introduces an FPGA-based gyro-vision fusion stabilization system. A three-axis gyroscope first supplies prior information on rapid camera attitude changes. On the vision side, FAST corner detection combined with Lucas–Kanade optical flow ensures reliable feature tracking even under motion blur. Inertial and visual displacements are then dynamically fused via a Kalman filter to build a highly robust global motion model. The design implements preprocessing, feature extraction, data fusion, and motion compensation as parallel pipeline modules on a Xilinx Artix-7 FPGA. Hardware-level parallelism and fixed-point optimization allow steady 30 fps real-time processing of 720p video, raising peak signal-to-noise ratio (PSNR) by an average of 15.83 %, while power consumption is only 12.9 % of a traditional CPU solution. Experiments confirm that the proposed system delivers high accuracy, low latency, and low power, offering a practical hardware-accelerated path for video stabilization in demanding industrial environments.

Keywords: Electronic image stabilization; Lucas-Kanada; FPGA; Hardware acceleration; Kalman-Filter.

1. Introduction

With the accelerating reliance on real-time video surveillance in high-vibration industrial sites—especially oil and gas production—maintaining a stable, clear video stream has become essential for operational safety and fine-grained management. Conventional electronic image stabilization (EIS) systems, which rely solely on visual algorithms such as block matching or optical flow, often fail in the presence of severe mechanical shocks, wind loads, or even seismic disturbances. Under these conditions, large numbers of image feature points are lost or mismatched, corrupting motion estimates. In addition, PC-based implementations suffer from high power consumption and processing latency, making them unsuitable for embedded front-end devices that must meet "low-power + millisecond-level response" requirements. Existing improvements typically refine feature extraction or matching, but still struggle to overcome real-time and robustness bottlenecks when vibration amplitude and frequency vary dramatically.

To address these challenges, this study proposes an FPGA-based gyroscope–vision fusion EIS system featuring four key components:

1. Multi-source sensing. A tri-axial gyroscope provides real-time angular velocity and acceleration, supplying prior knowledge of camera attitude. On the vision side, FAST corner detection and Lucas–Kanade optical flow tracking maintain a stable feature set even under motion blur.

2. Dynamic fusion. A Kalman filter fuses inertial poses with visual displacements online, producing a highly robust global motion model that markedly reduces the risk of drift or frame loss that plagues single-sensor approaches.

3. Hardware acceleration. Implemented on a Xilinx Artix-7 FPGA, the system divides image preprocessing, gyroscope acquisition, feature extraction, data fusion, and motion compensation into pipeline modules. Parallel logic and fixed-point arithmetic allow 720p video to run stably at 30 fps while consuming only 12.9 % of the power of a traditional CPU solution.

4. Experimental evaluation. Tests on both a simulated shake table and an actual drilling-rig environment show an average PSNR improvement of 15.8 % over un-stabilized video and a 78 % reduction in frame-to-frame jitter, confirming the system's high accuracy, low latency, and energy efficiency.

The contributions of this work are threefold:①It introduces a dual-domain compensation framework that couples gyroscope priors with visual residuals, delivering superior robustness in large-amplitude industrial vibrations. ②It fully hardware-accelerates the core algorithm, achieving millisecond-level "edge" processing that meets stringent low-power requirements. ③It provides system-level experimental data, offering a reproducible engineering reference for future video stabilization designs in high-frequency vibration environments.

The results demonstrate that combining inertial measurements with visual features—and harnessing FPGA parallel computation—is a feasible path to achieving low-latency, high-stability video monitoring in demanding industrial settings, and offers valuable insights for next-generation stabilization systems.

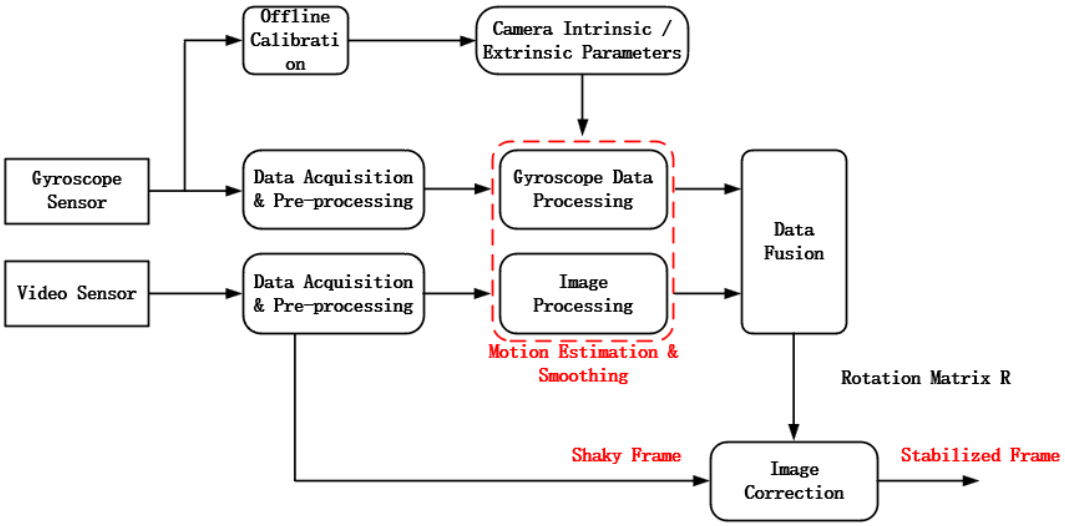

Figure 1. Electronic image stabilization system architecture

2. System Principle

This system first preprocesses the input image, including filtering and grayscale processing. Subsequently, the FAST feature detection algorithm is used to extract the set of feature points in the current frame, and the L-K optical flow method is used to track the positional changes of these feature points in adjacent frames. In order to ensure the reliability of the matching, the system sets a reasonable threshold to remove feature points with low matching quality and only retain high-quality matching point pairs. By analyzing the displacement relationship of matching points in two frames of images, the offset of each feature point can be calculated. Then, perform the same calculation on all matching points and estimate the global motion parameters of the image based on these matching points. Finally, obtain the overall motion displacement information between two frames, providing basic data for subsequent image stabilization processing.

2.1. Algorithm composition

2.1.1. FAST

As early as the 1970s, researchers introduced the notion of image feature points: when the first-order derivative at a pixel reaches a local extremum and the second-order derivative equals zero, that pixel can be regarded as a feature point. Selecting suitable feature points to represent a target is pivotal for object recognition. Feature-point extraction—a technique that derives key data from image information—has been widely applied to object recognition, tracking, and image registration. Unlike traditional approaches that focus only on local details, feature-point extraction analyzes the image globally to capture its critical features; even in multi-object scenes or complex backgrounds, it can still effectively isolate and describe salient image features .

In their paper *"Machine Learning for High-Speed Corncob Detection,"* Edward Rosten and Tom Drummond introduced the FAST (Features from Accelerated Segment Test) feature point. After revisions, they published *"Features from Accelerated Segment Test,"* in which they defined a FAST corner as follows: choose any pixel and compare it with a sufficient number of surrounding pixels; if the differences between this pixel and its neighbors satisfy specific criteria, the pixel is labeled a corner. To decide whether a point is a corner, a corner-response function is first defined.

$$N=\sum\left|I(x)-I(p)\right|<Th, x\in circlez(p) \quad (1)$$

Research shows that setting the circle radius to **3** achieves a good balance between detection speed and accuracy. In practice, FAST treats an arbitrary pixel as the circle center, draws a circle of radius 3 (encompassing 16 pixels along the circumference), and applies the above criterion—illustrated schematically in the figure below .

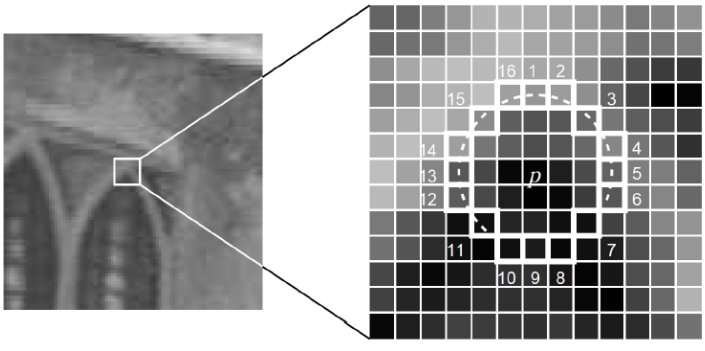

Figure 2. Principle of FAST

To verify the functionality of the FAST feature-detection circuitry, a software implementation of the FAST algorithm should be run in parallel as a reference; the two results are then compared to confirm the correctness of the hardware

design. MATLAB supplies built-in functions for FAST corner detection that can be used for this purpose.In the experiment, a calibration-board image with a resolution of 1280 × 720 was processed. The software version detected 285 feature points in total. After non-maximum suppression (NMS), 50 feature points were retained. Figure 3 presents the experimental results; the image on the right is the test picture used in the evaluation.

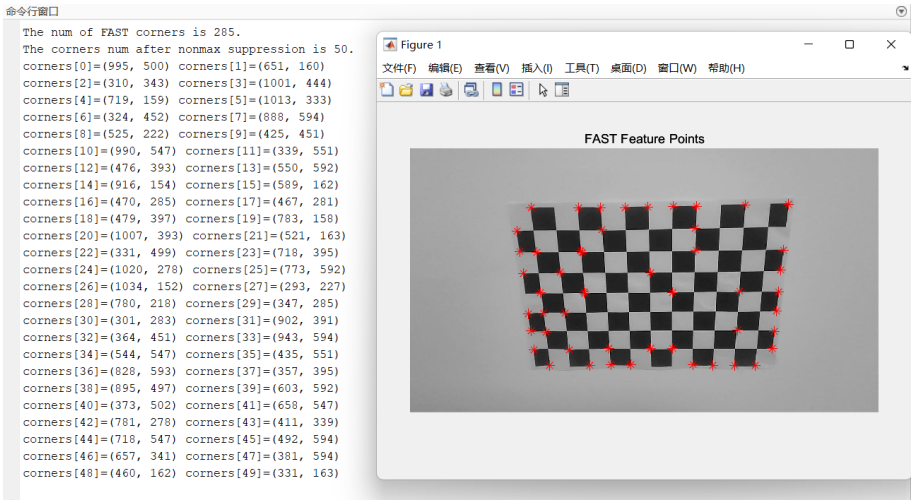

Figure 3. Extract image feature-point coordinates using MATLAB

The test waveform of the FAST feature-detection circuit in Vivado is shown in Figure 3; the same test image used in the MATLAB evaluation was applied here.The input signal image_gray_data carries the 8-bit grayscale pixel values, corner_flag marks the positions of FAST feature points, and nms_flag indicates the points that remain after non-maximum suppression.A comparative analysis shows that the circuit-level simulation results are identical to the MATLAB results, confirming the correctness of the designed FAST feature-detection circuit.Figure 3 presents the waveform for the entire test frame, while its expanded details are given in Figure 4.

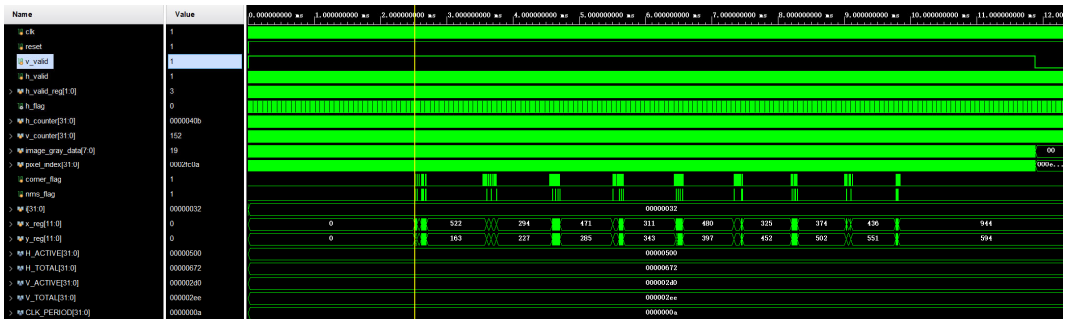

Figure 4. Simulation waveform diagram of the fast feature-detection circuit

2.1.2. L-K Optical Flow Tracking Algorithm

Horn–Schunck and Lucas–Kanade are both classic optical-flow algorithms. The Horn–Schunck method is a form of dense optical flow; in practice, computing dense flow is difficult and computationally demanding. Lucas–Kanade, by contrast, is a sparse optical-flow technique that evaluates flow only at designated points, greatly reducing complexity. For ease of hardware implementation, the present system adopts the L-K method.

First introduced in 1981, the Lucas–Kanade (L-K) optical-flow algorithm assumes that, within a small neighborhood, motion vectors remain constant and can therefore be estimated via a weighted least-squares solution. Because it readily computes flow at a selected set of image points, it is widely used wherever sparse optical flow is sufficient. The algorithm rests on three core assumptions:

1. Brightness constancy: a target pixel retains the same appearance as it moves from one frame to the next; in grayscale images its intensity does not change.

2. Temporal continuity (small motion): the target's displacement between adjacent frames is small, allowing accurate estimation from local information.

3. Spatial coherence: neighboring pixels lying on the same physical surface tend to move similarly, so their projections appear in adjacent image regions, ensuring reliable flow estimates within a local window.

Under these assumptions, the L-K method has become a staple in target tracking, motion estimation, and video analysis. Imposing the brightness-constancy constraint, the algorithm posits that a pixel maintains the same intensity in two successive frames. Hence, for an image intensity I(x,y,t), after a short interval Δt the point (x,y)moves to (x+Δx, y+Δy) and satisfies the following relation:

$$I(x,y,t) = I(x+\Delta x, y+\Delta y, t+\Delta t) \tag{2}$$

According to the brightness-constancy assumption, performing a Taylor expansion of the above expression and neglecting higher-order terms gives:

$$I(x+\Delta x, y+\Delta y, t+\Delta t) \approx I(x,y,t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t \quad (3)$$

By substituting the brightness-constancy assumption, subtracting the two sides, and defining the optical-flow velocity (u,v), we obtain the fundamental optical-flow equation:

$$\frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t = 0 \Rightarrow I_x u + I_y v + I_t = 0 \quad (4)$$

Because the fundamental optical-flow equation gives only one equation with two unknowns, uu and vv, it cannot be solved directly—this is the "aperture problem" in optical-flow computation. The Lucas–Kanade method addresses this by introducing a spatial coherence assumption: within a small neighborhood Ω, all pixels share the same optical-flow velocity (u,v). Consequently, a system of equations can be constructed for every pixel in that neighborhood:

$$\begin{cases} I_{x1}u + I_{y1}v + I_{t1} = 0 \\ I_{x2}u + I_{y2}v + I_{t2} = 0 \\ \vdots \\ I_{xn}u + I_{yn}v + I_{tn} = 0 \end{cases} \Rightarrow A\begin{bmatrix} u \\ v \end{bmatrix} = -b \quad (5)$$

Since the system above is over-determined, the L-K optical-flow problem is reformulated as minimizing an error term—that is, we seek the values of uuu and vvv that minimize the following sum of squared errors:

$$E(u,v) = \sum_{i=1}^{n} (I_{xi}u + I_{yi}v + I_{ti})^2 \Rightarrow E(u,v) = \left\| A\begin{bmatrix} u \\ v \end{bmatrix} + b \right\|^2 \quad (6)$$

Finally, after carrying out the least-squares derivation, the Lucas–Kanade optical-flow velocity (u,v)(u,v)(u,v) satisfies the normal equation

$$\begin{bmatrix} u \\ v \end{bmatrix} = -\frac{1}{\det(G)}\begin{bmatrix} S_{xx} & -S_{xy} \\ -S_{xy} & S_{yy} \end{bmatrix}\begin{bmatrix} S_{xt} \\ S_{yt} \end{bmatrix} \Rightarrow \begin{cases} u = -\frac{1}{\det(G)}(S_{yy}S_{xt} - S_{xy}S_{yt}) \\ v = -\frac{1}{\det(G)}(-S_{yy}S_{xt} + S_{xy}S_{yt}) \end{cases} \quad (7)$$

2.1.3. Kalman Filter Data Fusion

The Kalman filter (KF) is well suited to linear-system state estimation and can fuse heterogeneous sensor data to improve accuracy. In petrochemical production environments, an electronic image-stabilization system must contend with strong vibration, dust, and corrosive gases—factors that introduce uncertainty and noise into camera motion and image acquisition. To obtain an accurate estimate of the camera's motion state, the proposed system therefore fuses gyroscope displacement data with image feature-point measurements using a KF. This section describes the filter's state model, observation model, and update procedure. By optimally combining the two sensor sources, the KF markedly enhances motion-state estimation accuracy, providing a sound basis for image correction and stabilization.

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.

Through camera calibration, the three-axis angular-velocity data from the gyroscope can be mapped into the camera coordinate system via a homography matrix H.Given a 3-D point Pg=(Xg,Yg,Zg) expressed in the gyroscope frame, its transformation to the image coordinate system using H proceeds as follows:

$$\omega_c = H\omega_g = \begin{pmatrix} \omega_{c_x} \\ \omega_{c_y} \\ \omega_{c_z} \end{pmatrix} \quad (8)$$

In the equation, ωx, ωy, and ωz denote the angular velocities, after transformation, about the image coordinate system's x-, y-, and z-axes, respectively. For simplicity, the slight global rotation of the image induced by motion about the z-axis will be handled separately. Assume that within a very short interval Δt\Delta t the camera rotates by small angles Δθx and Δθy about its x- and y-axes; these angles are related to the angular velocities by:

$$\begin{cases} \Delta\theta_y = w_{c_y}\Delta t \\ \Delta\theta_x = w_{c_x}\Delta t \end{cases} \Rightarrow \begin{cases} \Delta d_x \approx f\Delta\theta_y = f\omega_{c_y}\Delta t \\ \Delta d_y \approx f\Delta\theta_x = f\omega_{c_x}\Delta t \end{cases} \quad (9)$$

The **state vector** collects all variables that must be estimated at a given instant. Because this system fuses gyroscope-derived displacements with image feature-point measurements, the state variables are chosen to be **displacement, velocity, and acceleration on the image plane**. Accordingly, the state vector can be written as

$$x = \begin{bmatrix} d_x & d_y & v_x & v_y & a_x & a_y \end{bmatrix}^T \quad (10)$$

The state-transition equation uses the displacement converted from the gyroscope to propagate the state, while the observation equation takes the globally compensated image displacement as the actual measurement. The predict-update cycle of the Kalman filter proceeds as illustrated in the accompanying figure.

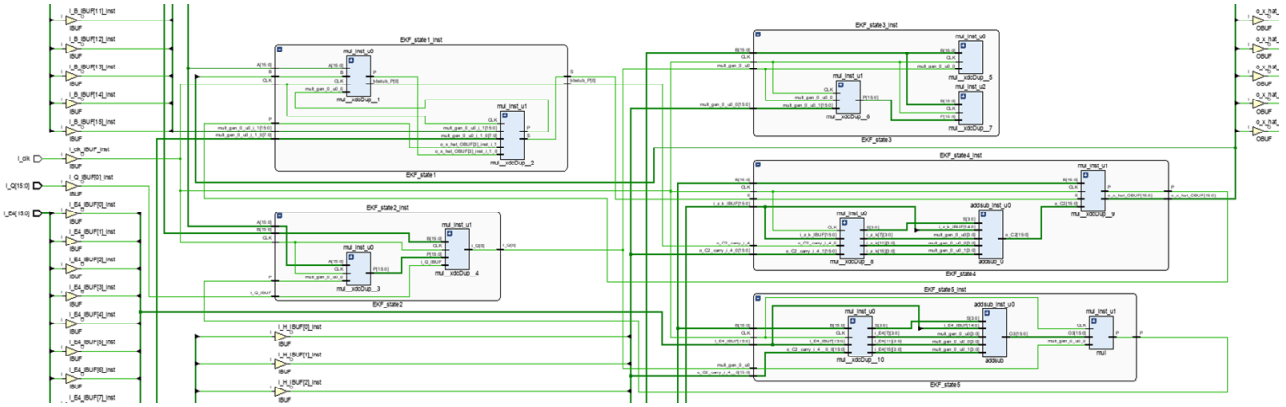

Figure 5. Kalman-filter fusion pipeline circuitry diagram

2.1.5. Image Correction and Compensation Strategy

Based on the motion estimates obtained from the Kalman-filter data fusion, we derive the incremental displacements of the image along the **x**- and **y**-axes. These increments are used to apply inverse compensation for translational jitter.For rotational jitter, the system relies on the gyroscope-measured angular velocity ωZ\omega_Z to estimate the rotation angle and achieve video stabilization.The mathematical model of the affine transformation employed for image correction can be written as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \tag{11}$$

In the equation, (x,y) are the original image coordinates and (x′,y′) are the transformed coordinates. The coefficients a,b,c, and dd form the linear part of the affine transform, controlling rotation, scaling, and shear, while ee and ff represent translation offsets. By applying this affine transformation, the system compensates for both translational and rotational jitter, effectively removing image shake caused by camera motion and rotation.

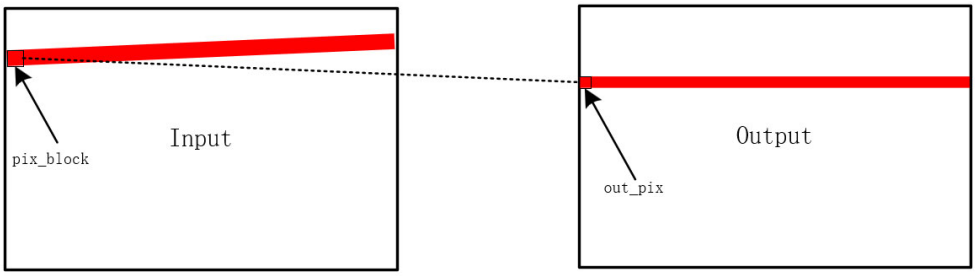

Figure 6. Motion Compensation Flowchart

2.1.6. Summary

This chapter has covered the key theories and techniques of video stabilization: image feature-point detection and matching, the use of Kalman filtering for data fusion, the principles and steps of image correction, and the evaluation criteria for electronic stabilization. Together, these topics establish the theoretical foundation for the subsequent design and implementation of the electronic stabilization system.

3. System Demonstration and Results Analysis

3.1. Electronic Image Stabilization (EIS) System

After completing simulations of each functional module, the overall system is deployed onto the FPGA development board for board-level testing. The Xilinx Artix-7 series ACX720 development board is selected for this design, providing HDMI and various interface resources required for image processing, along with sufficient logic units. The integrated DDR3 memory chip onboard is used for buffering video frames, while the abundant DSP resources offer hardware acceleration for extensive multiplication and division operations within the system. The primary hardware required to set up the FPGA testing environment includes a PC platform, ACX720 development board, OV5640 camera module, a monitor, and Vivado 2020.1 design tool. The schematic of this validation platform setup is shown in Figure 7.

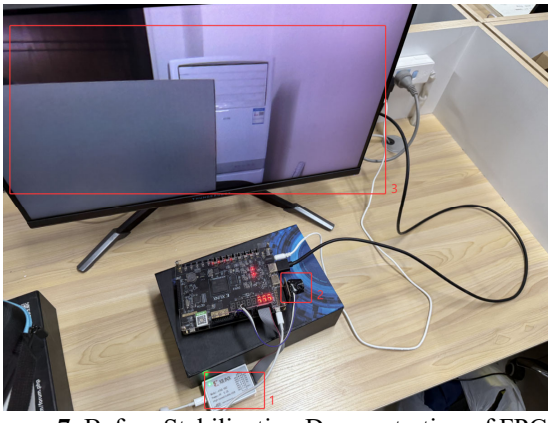

Figure 7. Before Stabilization Demonstration of FPGA-based Video Image Processing Results

3.2. Resource Utilization and Timing Analysis

After importing the RTL files into Vivado and completing synthesis and implementation, the resource usage of each stabilization-system module can be tallied, as shown in Table 5-3. On a Xilinx Artix-7 FPGA, the basic logic unit is a Slice, each containing four 6-input LUTs and eight flip-flops. Statistics show that the system uses 29 534 Slices—about 87.8 % of the available slice resources—along with 118 136 Slice LUTs and 236 272 Slice Registers.A total of 244 BRAMs are occupied. The feature-detection and matching module must buffer row data from both the current and previous frames, so it consumes a large share of BRAM capacity (approximately 66.8 % of all BRAMs).The design also employs 189 DSP blocks (about 25.9 % of the device's DSP resources), mainly in modules that perform wide-word arithmetic, such as matrix inversion, data fusion, and optical-flow calculations.

Table 1. Resource Utilization of the Electronic Image Stabilization System

Module	SliceLUTs	SliceRegisters	Slice	BlockRAM	DSPs
Gyroscope Acquisition & Processing Module	1032	2064	258	4	0
Image Acquisition Interface Module	20628	41256	5157	16	0
Feature Detection & Matching Module	38120	76240	9530	160	40
Data Fusion Module	49836	99672	12459	24	124
Image Correction & Output Module	520	17040	2130	40	25
Overall	118136	236272	29534	244	189

3.3. Results Analysis

To verify the circuit's functionality, this section analyzes video data captured before and after FPGA-based stabilization. Shown below are frames 101–103 from the OV5640 camera:Figure 8: original (unstabilized) video. Figure 9: FPGA-stabilized videoUsing the reference lines and yellow circles as markers, clear differences emerge. For instance, in frame 103 of the original video, the distance between the lower-left horizontal reference line and the edge of the tabletop shifts noticeably, whereas in the stabilized sequence this distance remains virtually unchanged. Similarly, the relative position of the doll's right hand to the horizontal line and the position of the vertical line to the doll's sclera both demonstrate the effectiveness of the stabilization.

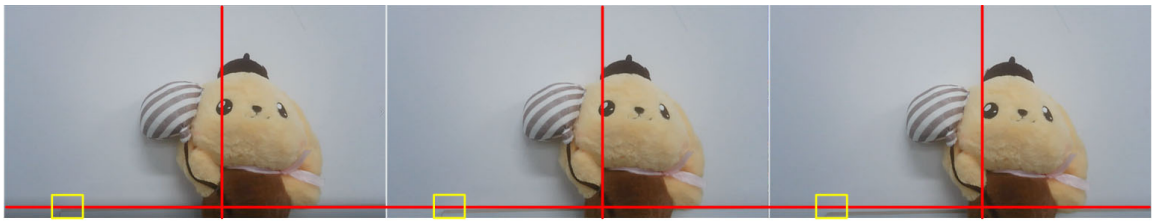

Figure 8. Frames 101, 102, and 103 Before Stabilization

Figure 9. Frames 101, 102, and 103 After Stabilization

Based on the electronic-stabilization evaluation criteria, this system is assessed using the following three metrics:

1. System Fidelity

The hardware-output image data are sent to a PC for processing. Figures 10 and 11 show the results of adjacent-frame differencing. From these difference images, it is even more evident that before stabilization, the doll area exhibits large inter-frame variations, whereas after stabilization, the differences are much smoother, image quality is higher, and the stabilization effect is markedly improved.

Figure 10. Adjacent-Frame Difference Images Before Stabilization

Figure 11. Adjacent-Frame Difference Images After Stabilization

The inter-frame PSNR values before and after stabilization are listed in Table 2. After stabilization, all PSNR values exceed 32 dB, representing an average improvement of 5.799 dB compared to the unstabilized sequence. This demonstrates a clear enhancement in PSNR. When compared to the MATLAB simulation results, the FPGA implementation performs as expected, with errors remaining within acceptable bounds.

Table 2. PSNR Comparison Before and After Stabilization

Frame Number	PSNR Before Stabilization	PSNR After Stabilization
100 and 101	27.4309	34.8118
101 and 102	28.3114	32.5276

From the PSNR comparison curve of 145 consecutive frames before and after stabilization in Figure 12, it can be seen that the PSNR values of the stabilized video sequence are overall higher than those of the original sequence, indicating that the system provides effective stabilization for jittery video.

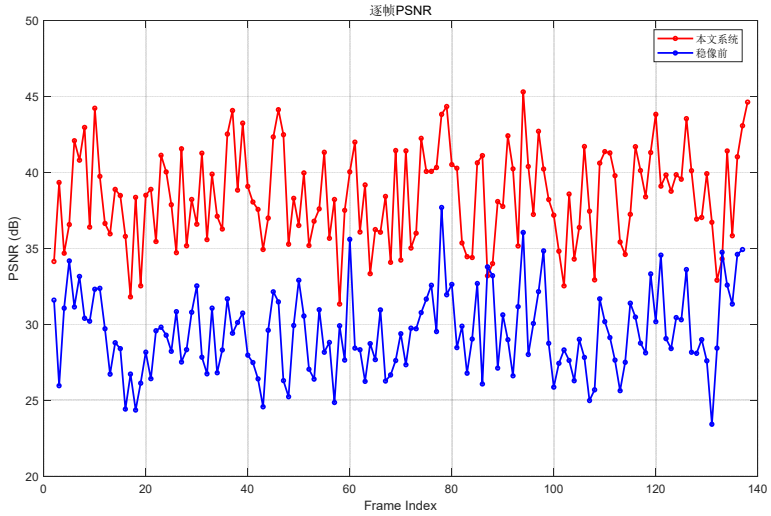

Figure 12. PSNR Comparison Before and After Stabilization

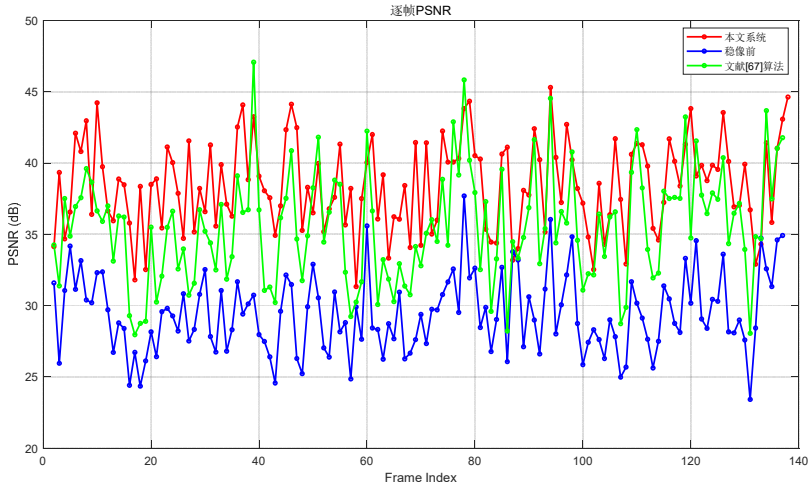

Figure 13. System Performance Comparison

4. System Performance Evaluation

Next, we compare the pre-stabilization video processed by our proposed stabilization system with the SURF-based algorithm from. Figure 13 presents the PSNR values of the captured video after processing with each method. Reference employs a SURF registration algorithm combined with a similarity-transform model; however, lacking gyroscope-derived offset data, it cannot correct for image shear. As the plot shows, although some frames are well stabilized, many frames suffer severe distortion, revealing poor robustness. In contrast, our proposed stabilization algorithm achieves significantly higher PSNR across all frames, demonstrating superior stabilization performance and markedly stronger robustness.

Next, we compute the ITF metric by averaging the PSNR values across all frames of the video, as defined in Equation (5-1). A higher ITF indicates better image quality and superior stabilization performance.

$$ITF = \frac{1}{n-1}\sum_{k=1}^{n-1} PSNR(I_k, I_{k+1}) \tag{12}$$

The ITF after stabilization using our algorithm is 36.15 dB, which represents improvements of 4.94 dB over the original video and 2.47 dB over the method in Reference, demonstrating superior stabilization performance.

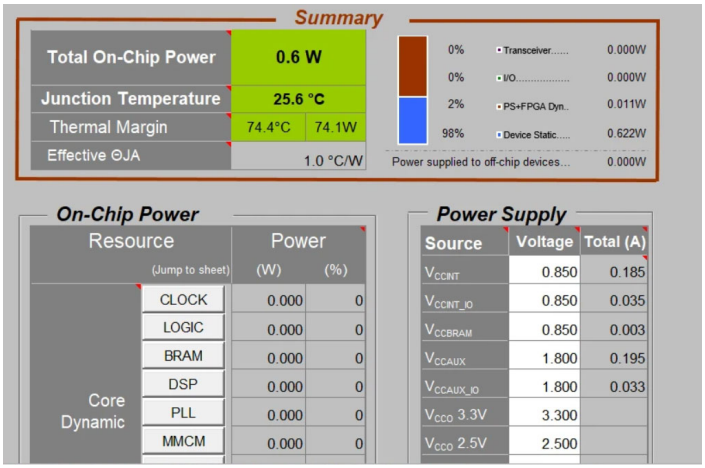

Figure 14. FPGA Static Power Consumption

Using XPE (Xilinx Power Estimator), the FPGA's dynamic power consumption at zero load is estimated at approximately 0.6 W (see Figure 14. By running the stabilization code on the FPGA and measuring voltage and current, the actual dynamic power draw was found to be about 1.843 W.

According to the Intel i7 datasheet, the low-power version of that CPU has a static (idle) power consumption of roughly 7 W. Under the same input data and running the identical algorithm, measuring its voltage and current yields a dynamic power consumption of approximately 11.8 W.

5. Summary

First, a functional-level simulation of the Chapter 4 circuit was carried out in the Vivado integrated simulation environment, with the output waveforms examined to verify correct operation and conformity with design expectations. Next, real-world tests were performed on a Xilinx Artix-7 ACX720 development board, focusing on the data read/write architecture, module collaboration, and the system's end-to-end real-time performance; resource utilization, timing metrics, and stability were evaluated in detail. By comparing adjacent-frame difference images and PSNR metrics before and after stabilization, a marked improvement in video quality was observed, and it was demonstrated that both the stabilization range and frame-processing speed satisfy real-time requirements. Finally, the proposed system was compared against other algorithms reported in the literature and against a PC-based implementation in terms of performance and power consumption; the results confirm that the design meets the targets for both real-time stabilization and hardware acceleration.

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.