

Research on Ball Strike Judgement Improvement Using AKAZE Algorithm under Multi-perspective Video Fusion

Huimin Shi^{1,a,*}

¹North China University of Technology, Beijing, China

a. 501306560@qq.com

*corresponding author

Abstract: Addressing the issue of judgment bias in baseball games due to view occlusion from a single camera angle, this study proposes the use of multi-camera video fusion and real-time motion monitoring technologies to merge the visual information from two cameras into a unified video frame, thereby enhancing the judgment accuracy. To achieve faster and more precise image matching, an improved AKAZE algorithm for efficient image registration is proposed. Firstly, in the feature extraction phase, the AKAZE algorithm constructs a nonlinear scale space and uses adaptive descriptors to effectively describe feature points. Techniques such as superpixel segmentation and illumination adjustment are employed to better adapt to complex environmental conditions. Secondly, the FLANN method is used for pre-matching the feature points, quickly calculating the nearest neighbors for each feature point to accelerate the matching process. The ratio test proposed by Lowe is used to compare the distance differences between the nearest and next-nearest neighbors, discarding unlikely matching feature points. Finally, further selection of matches is performed using the RANSAC algorithm to remove erroneous matches, thus improving the accuracy and reliability of the final match. To validate the effectiveness of the proposed method, experiments were conducted using a real-time video stream dataset collected by cameras, compared with the SIFT, ORB, and AKAZE algorithms. The results showed that the proposed method maintains high accuracy while achieving rapid image registration and demonstrating robustness under changes in illumination, blur, and compression transformations. The image fusion algorithm enables seamless visualization and tracking of the transferred attention category across different camera views, meeting the real-time detection needs for ball strike judgements in baseball.

Keywords: Multi-camera video fusion, Image stitching, AKAZE algorithm, Adaptive descriptor

In baseball games, disputes over the strike zone have always been a highly scrutinized topic. The misjudgment of balls and strikes affects the outcome of a game and is a stumbling block for baseball's inclusion in the Olympics. With the growing popularity of baseball tournaments, more and more games are introducing video arbitration systems to assist referees in making strike zone decisions. Traditionally, these systems rely on multiple cameras to capture the game footage from various angles. However, due to the limitations of camera perspectives, especially in fast pitches and curveballs,

decisions are often affected by occlusion—quick ball movements or the batter’s swing frequently cause parts of the ball’s trajectory to be blocked, leading to judgment errors [1].

To overcome these issues, this study proposes a multi-camera video fusion technology. By capturing game footage in real-time from multiple cameras at different angles, the method effectively eliminates the visual blind spots caused by a single camera perspective. Using synchronized processing of the real-time video streams captured by multiple cameras, each camera’s footage is aligned and merged into a wide-angle view. This view covers more areas of the playing field, ensuring that regardless of the ball’s flight path, it can be tracked and captured in real-time. Additionally, a motion detection algorithm is introduced to identify moving areas in each video frame, supporting further target analysis. However, a critical challenge in this approach lies in accurately and efficiently performing image stitching and matching due to synchronization issues between cameras, varying image quality, and different perspectives.

Over the years, researchers have made numerous advancements in key point detection and description algorithms to address different image matching tasks. Lowe et al. were the pioneers of the Scale-Invariant Feature Transform (SIFT) algorithm, which has undergone various optimizations [2]. However, the traditional SIFT algorithm uses 128-dimensional descriptors [3-4], resulting in high computational complexity and low efficiency. To address this, Bay et al. introduced the Speeded-Up Robust Features (SURF) algorithm [5], which accelerates the calculation of Harris wave response using integral images. While this improved efficiency, the SURF algorithm still faced limitations in meeting the high-speed execution requirements due to advancements in machine vision technology [6]. To further enhance efficiency, Rublee et al. developed the Oriented FAST and Rotated BRIEF (ORB) algorithm, which improves execution efficiency by one to two orders of magnitude compared to SIFT and SURF [8]. However, the ORB algorithm does not include rotational and directional information, thus lacking rotational invariance and limiting its application in some scenarios. To address this, Leutenegger et al. proposed the Binary Robust Invariant Scalable Keypoints (BRISK) algorithm [9], utilizing adaptive general-purpose accelerated segmented testing (BRISST) [10-11] to enhance rotational invariance and reduce computational complexity. Despite this, the BRISK algorithm remains sensitive to noise, making it susceptible to interference in practical applications and resulting in limited widespread usage.

To achieve efficient and precise image stitching and feature matching, this study proposes an improved AKAZE algorithm for high-efficiency and precise image registration. The AKAZE (Accelerated KAZE) algorithm is a feature extraction method based on a nonlinear scale space, suitable for images with complex lighting changes, blur, or noise. Traditional AKAZE algorithms may exhibit low computational efficiency when processing large-scale images; therefore, we optimized it in several ways:

- (1) Introducing adaptive descriptors, which adjust the calculation method of descriptors based on the local features of the image to better accommodate significant lighting and texture changes.

- (2) Employing superpixel segmentation technology to divide the image into multiple blocks, processing these small blocks to reduce the impact of global lighting changes. Lighting adjustment techniques are also used to balance differences across different regions, thus improving feature point stability and matching accuracy.

- (3) Using the FLANN algorithm to quickly pre-match the extracted feature points. The Lowe ratio test method is then applied to filter out unreliable matching points, enhancing matching accuracy.

- (4) Combining the RANSAC algorithm to further filter the matching points, removing those that do not conform to geometric transformations, thereby improving matching accuracy.

1. Introduction to Relevant Algorithms

AKAZE is an accelerated version of the KAZE algorithm, specifically designed for feature point detection and descriptor extraction. This algorithm utilizes nonlinear diffusion filtering to construct a scale space, allowing for the extraction of image features at multiple scales and possessing characteristics such as scale invariance and rotational invariance. The core idea is based on scale space theory and image pyramids, generating the scale space quickly through a nonlinear diffusion process, and detecting key points within this space. The identification of key points relies on a regional approach, typically finding local extrema in the image. To describe the surrounding information of feature points, AKAZE employs an improved M-LDB descriptor, which is more effective in representing image features than the traditional LDB descriptor.

In the specific process of the algorithm, the image is first grayscale-converted and denoised to ensure more accurate feature extraction. Then, a nonlinear diffusion filter is used to construct a scale space, where key points are identified at different scales through a process of finding local extrema. Subsequently, AKAZE encodes the surrounding area of each key point using the M-LDB descriptor. Finally, by comparing the M-LDB descriptors between different images, feature matching is completed, thereby identifying similar regions [12].

The AKAZE algorithm effectively extracts feature points at different scales and provides good rotational invariance through an accelerated nonlinear diffusion process. However, while it shows strong robustness against noise and interference, it remains sensitive to changes in lighting and local detail variations, which may lead to misdetection or omissions.

2. Research Methodology

2.1. Video Fusion and Inter-frame Synchronization Methods

In this study, we deploy two cameras to capture real-time scenes of a baseball game. The cameras used are part of an IP camera array. This system transmits visual data to computing devices using an encoding format, which is typically used to compress original video and audio data so they can be sent over a network connection at much lower bandwidth than other methods. The encoded video signal is decoded and decompressed at the receiving end of the communication channel to restore the original visual information. The multiple IP cameras are mounted on a single pole to capture both the side and top contours of the playing field, with visual data being transmitted directly to the computing device via the Internet Protocol.

The two IP cameras are divided into a master camera and a side camera. The master camera is positioned above the catcher, facing the outer field area. It captures the view that includes the batter and the catcher, and typically also includes the pitcher. The real-time video feed from the master camera is used to extract the coordinates of the baseball and home plate (if visible in the frame). The side camera tracks the position of the baseball relative to the batter's stance at home plate, thereby continuously defining the strike zone. By analyzing the trajectory of the baseball and key points of the batter's stance, such as the knees, elbows, etc., and the relative position when the ball crosses home plate, it is possible to judge if a pitch is a strike or a ball. An image fusion algorithm is used to merge the visual data from the two cameras into a single video frame, enabling seamless visualization and tracking across the different camera views. Once a complete view of the field is obtained, the next step involves eliminating unnecessary areas from the video frames. Removing irrelevant areas reduces the overall image size, allowing faster processing using an ML model.

Real-time media streams from both cameras, based on the Real-Time Streaming Protocol (RTSP), are decoded to extract the frames at the same moment. An improved feature point set matching algorithm is used to match these frames between the two cameras. A transformation matrix is then

calculated based on the matched feature points, and the image from the master camera is transformed accordingly. The image from the master camera is then stitched with the image from the side camera by coordinate alignment and output, achieving a seamless visual representation. The process of extracting video frames is repeated, bypassing the feature point set matching step, and directly using the transformation matrix calculated earlier to perform matrix transformation and output the stitched video. The main process flow is illustrated in Figure 1 below.

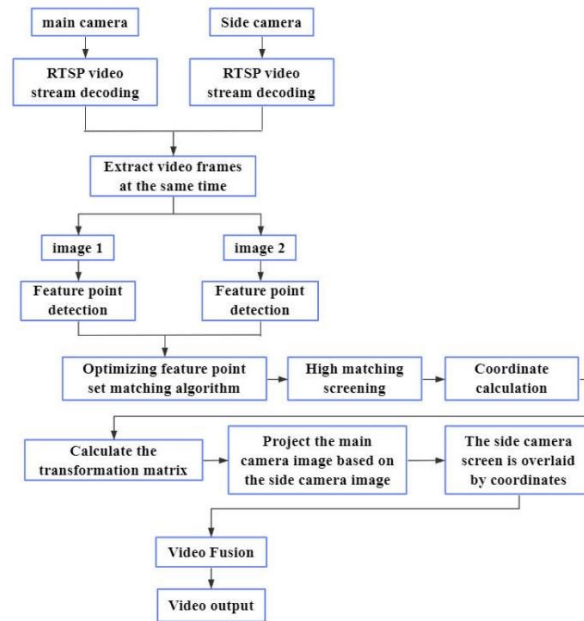


Figure 1: Video Fusion Technology Process Flow Diagram

The challenge in multi-camera video fusion lies in accurately aligning and stitching images from different angles. Due to the varied image transmission and processing speeds of different cameras, video streams may become out of sync, resulting in excessive time differences between frames. To address this, timestamps for each frame are recorded, and the synchronization of video frames is achieved by comparing the timestamps from the left and right camera streams to ensure that the time difference between frames read from both video streams falls within an acceptable range.

When reading each frame, the timestamp for each video stream is recorded, and the time difference between the two video streams is calculated. If the difference between the timestamps exceeds a certain threshold (G), the system will re-read the frame and synchronize the timestamps again until the time difference between the left and right frames is within the allowable range. The detailed process flow for timestamp synchronization is illustrated in Figure 2 below.

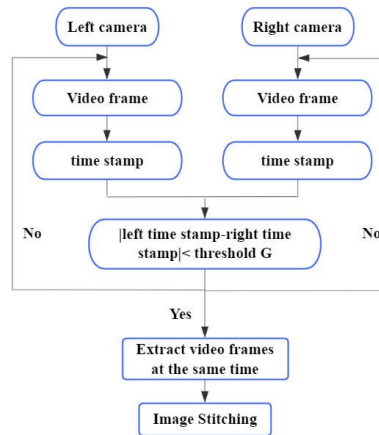


Figure 2: Timestamp Synchronization Process Flow Diagram

2.2. Real-time Motion Detection Method

To detect dynamic objects in the images, a motion detection algorithm is introduced. The basic idea of motion detection is to compare changes between consecutive frames to determine if there is movement. In the motion detection process, the system real-time processes each frame of the image and converts it into a grayscale image, followed by Gaussian blur to remove noise. By calculating the differences in the images, the system can detect moving regions within the image. When motion is detected, the system draws a bounding box around the moving region to help referees focus on important areas and reduce distractions. Once a moving region is detected, the system marks the location of the moving object with a red bounding box in the video stream. In this way, moving objects are tracked and marked in real time, aiding in subsequent target analysis. Through motion detection, we can track the trajectory of a baseball and the associated player movements in the real-time video stream. By capturing the motion regions and analyzing the trajectory of the ball, we can assist in determining whether a pitch is a strike or a ball.

2.3. Image Registration Method Based on an Improved AKAZE Algorithm

To achieve efficient and precise image stitching and feature matching, this study employs an improved AKAZE algorithm for image registration. Traditional AKAZE algorithms suffer from lower computational efficiency when handling large-scale images. Therefore, we optimized it primarily through the following improvements.

2.3.1. Introduction of Adaptive Descriptors

In the traditional AKAZE algorithm, descriptors are computed based on the gradient information of local regions. This approach is ineffective in handling factors such as illumination changes, texture variation, and viewpoint changes in complex scenes, which makes the feature point descriptors unstable and affects the subsequent matching and stitching results.

The key idea of adaptive descriptors is to use different descriptor computation strategies depending on the local features. The local mean and local variance of the image are used to adjust the computation of descriptors. Specifically, different neighborhood sizes are used for calculation depending on the local texture and illumination conditions. For regions with significant illumination changes, the neighborhood size can be increased to capture more surrounding information, while for areas with complex textures and small local changes, the neighborhood size can be reduced. The

Histogram of Oriented Gradients (HOG) is used as the descriptor for local gradient direction histograms. The calculation formula is as follows:

$$D = \sum_{i=1}^N w_i * \phi_i \quad (1)$$

Where, D is the descriptor, w_i is the weight for each direction, ϕ_i is the gradient direction information for each direction, and N is the total number of directions. For adaptive descriptors, the weights and direction selection are adjusted based on local illumination changes and texture information, using the formula:

$$W_i' = f(W_i, L, T) \quad (2)$$

Where, W_i' is the adjusted weight, L is the local illumination information, T is the local texture information, and f is a function used to adjust the weights based on local features.

2.3.2. Introduction of Superpixel Segmentation and Illumination Adjustment Methods

In the process of image stitching and matching, the complexity of the background and illumination changes often affect the stability of feature points. Traditional matching algorithms often fail to achieve ideal results. By introducing superpixel segmentation, the image is divided into multiple smaller regions with similar textures. These small regions are then processed to reduce the impact of global illumination changes on feature matching.

Superpixels refer to dividing the image into several regions, each with similar color or texture information. By using superpixel segmentation, the influence of the background in the image is effectively reduced, making the computation of feature points more precise and localized. The segmentation method employs the SLIC algorithm. The core idea of this algorithm is to use K-means clustering to divide the image pixels into several small blocks, each containing pixels with similar color and texture information. The formula for this is:

Assuming the pixel points of an image are $P=(p_1, p_2, p_3, \dots, p_n)$, the SLIC algorithm divides them into superpixel regions by calculating the color and spatial distances of the pixels. Each superpixel region S_i contains multiple pixel points $P_i \in S_i$, as shown in the following formula:

$$D(P_i, C_i) = \sqrt{C(P_i - C(C_i))^2 + S(P_i - S(C_i))^2} \quad (3)$$

Where, $C(P_i)$ and $S(P_i)$ are the color and spatial information of the pixel point P_i , and $C(C_i)$ and $S(C_i)$ are the color and spatial information of the superpixel center C_i .

Differences in illumination within an image can lead to variations in feature point descriptors; therefore, image illumination adjustment is necessary. The illumination adjustment method used is histogram equalization, which enhances the contrast of the image by adjusting the distribution of grayscale values. Assume $I(x, y)$ is the grayscale value of a pixel in the image. The equalization process can be completed using the following formula:

$$I_{eq}(x, y) = \frac{\sum_{k=0}^{I(x, y)} P_k}{N} \quad (4)$$

Where, P_k is the frequency of the grayscale value k and N is the total number of pixels in the image.

2.3.3. Introduction of FLANN and Lowe Ratio Test

During the process of feature point matching, mismatches may occur due to changes in local features or viewpoint differences. The FLANN algorithm is used for quick pre-matching of feature points,

leveraging its efficient nearest neighbor search functionality. Then, the Lowe ratio test is applied to eliminate unreliable matching points and improve the matching accuracy.

FLANN is an efficient approximate nearest neighbor search algorithm. By constructing a hierarchical tree structure, FLANN significantly accelerates the matching of feature points in large datasets [13]. The matching process using FLANN can be represented as follows:

$$NN(x) = \arg \min_{y \in Y} ||x - y|| \quad (5)$$

Where, Y is the set of feature points, and $||x-y||$ is the distance between feature points x and y.

The Lowe ratio test is used to filter out mismatched feature points. Specifically, for each feature point x, the nearest and the second nearest neighbors y1 and y2 are found, and their distance ratio is calculated:

$$r = \frac{dist(x,y_1)}{dist(x,y_2)} \quad (6)$$

If $r < \text{threshold}$, then y1 is considered a valid match; otherwise, the match is rejected.

2.3.4. RANSAC Algorithm to Optimize Matching Results

In the process of feature matching, erroneous matches may occur, affecting subsequent image stitching and object detection. To remove these erroneous matches, the RANSAC algorithm is used for optimization.

The core idea of RANSAC is to randomly select the minimum number of points from the matching points to perform hypothesis verification and determine whether other points conform to this hypothesis. If most points conform to the hypothesis, this set of points is considered to be correctly matched [14]. Assuming the feature points to be matched are $\{x_1, x_2, x_3, \dots, x_n\}$, the optimization process of RANSAC can be expressed using the following formula:

$$\hat{\theta} = \arg \min_{\theta} \sum_{i=1}^n \rho(||y_i - f(x_i, \theta)||) \quad (7)$$

Where, ρ is a robust loss function, $f(x_i, \theta)$ is the prediction of the transformation model for point x_i , and $\hat{\theta}$ is the optimal transformation parameters.

3. Experiment Results and Analysis

3.1. Experimental Environment and Dataset

The experimental platform is Windows 10, equipped with 32 GB of memory and an NVIDIA RTX2080 GPU. The experiments were conducted using a PyTorch 2.2.1 environment and accelerated by CUDA 11.6. All experiments were conducted under the same hyperparameter settings, with 100 training rounds, a batch size of 16, and image size of 640.

To verify the effectiveness of the proposed methods, the dataset was captured in real-time using a camera during competitions.

3.2. Image Stitching Accuracy and Efficiency Comparative Experiments

To further illustrate the effectiveness of the improved AKAZE method for image stitching on video streams from left and right cameras, comparative experiments were conducted using the root mean square error to evaluate image stitching accuracy and stitching time to evaluate efficiency. The smaller the root mean square error value, the higher the registration accuracy and the better the image stitching quality. The root mean square error can be expressed as:

$$Q_{\text{rmse}} = \sqrt{\frac{1}{m_c} \sum_{i=1}^{m_c} ||(x_i, y_i) - (\tilde{x}_i, \tilde{y}_i)||} \quad (8)$$

Where, m_c is the number of correctly matched points, (x_i, y_i) are the coordinates of feature points in the master camera, and $(\tilde{x}_i, \tilde{y}_i)$ are the coordinates of the feature points in the image transformed using the homography matrix.

Table 1: Image Stitching Accuracy and Efficiency Comparison Results

Feature Point Detection Algorithms	Qrmse	T/s
SIFT	0.489	2.69
ORB	0.852	0.56
AKAZE	0.635	2.54
The method herein	0.583	2.19

Based on the comparative experimental results shown in Table 1, the table illustrates the image stitching precision and efficiency of SIFT, ORB, AKAZE, and the proposed method. In terms of stitching precision, the SIFT method achieved the highest precision, while the ORB method exhibited relatively poor precision. Regarding stitching efficiency, the SIFT method took the longest time. Compared to the original AKAZE method, the proposed improved method increased stitching precision and efficiency by 14% and 8%, respectively. Although the ORB method achieved the highest stitching efficiency, its performance in feature detection, matching accuracy, and stitching precision was inferior. Thus, the improved method proposed in this paper balances stitching precision and efficiency, making it more suitable for real-time video stream image stitching in baseball games.

3.3. Experimental Results

The experimental results demonstrate that the improved AKAZE algorithm achieved a matching accuracy of over 90% across all tested datasets, maintaining an accuracy of over 95% under conditions of varying illumination and image blur. In terms of registration speed, the improved AKAZE algorithm is approximately 20% faster than the traditional SIFT algorithm, making it more suitable for real-time video processing requirements.

With multiple cameras working collaboratively, the system effectively eliminates blind spots and occlusion issues caused by a single camera, accurately capturing the trajectory and position of the baseball. Compared with traditional single-camera systems, the multi-camera video fusion technology significantly improves the accuracy and reliability of judgment.

4. Conclusion

This paper proposes a real-time method for determining baseball strikes and balls based on multi-camera video fusion and an improved AKAZE algorithm. By integrating multi-camera video streams with motion detection technology, the method addresses the limitations of single-camera systems. The improved AKAZE algorithm demonstrates higher precision and efficiency in image stitching and feature matching, maintaining strong robustness in complex environments.

Future work will focus on further optimizing the motion detection algorithm and exploring applications with more camera configurations. Additionally, as deep learning technology advances, combining deep learning with image analysis and judgment will become a key direction for future improvements.

References

- [1] Chen, Y. P. (2022). *A Baseball Strike Zone Recognition System Based on Deep Learning* [Master's thesis, National Central University].
- [2] Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 91–110.
- [3] Suo, C. B., Yang, D. Q., & Liu, Y. P. (2014). A multi-angle comparison of SIFT, SURF, BRISK, ORB, and FREAK algorithms. *Beijing Surveying and Mapping*, 28(5), 23–26.
- [4] Niu, J. W., Hao, X. Y., & Liu, S. L. (2014). An improved SIFT feature extraction algorithm. *Journal of Geomatics Science and Technology*, 31(2), 173–176.
- [5] Bay, H., Tuytelaars, T., Gool, L., et al. (2006). SURF: Speeded up robust features. *European Conference on Computer Vision*, 404–417.
- [6] Li, Y. F. (2019). *An image matching algorithm based on SURF feature extraction and brute-force search*. *Beijing Surveying and Mapping*, 33(11), 1352–1355.
- [7] Rublee, E., Rabaud, V., Konolige, K., et al. (2011). ORB: An efficient alternative to SIFT or SURF. In *Proceedings of the International Conference on Computer Vision* (pp. 2564–2571).
- [8] He, X. (2019). *Research on Image Feature Matching Algorithms and Their Hardware Implementation in SLAM* [Master's thesis, Xi'an University of Technology].
- [9] Leutenegger, S., Chli, M., & Siegwart, R. (2011). BRISK: Binary robust invariant scalable keypoints. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2540–2555).
- [10] Yao, Y. (2019). Optimization of image matching process design for low-altitude UAVs. *Beijing Surveying and Mapping*, 33(4), 423–426.
- [11] Yao, J. (2017). *Research on Panoramic Image Stitching Based on BRISK Image Registration* [Master's thesis, Changchun University of Technology].
- [12] Cui, C. Y., Fang, Y., Gao, W. W., et al. (2024). A wind turbine blade crack image stitching method based on AKAZE and PROSAC. *Electronic Measurement Technology*, 47(7), 177–183.
- [13] Dang, Y. Y., & Chen, Z. X. (2023). Research on color card automatic extraction method based on AKAZE-FLANN algorithm. *Software Engineering*, 26(7).
- [14] Zhang, D. Z. (2024). Comparative research on low-altitude UAV photogrammetry image matching algorithms. *Surveying and Spatial Geographic Information*, 47(9), 192–196.