RESEARCH ARTICLE

Visual Feature Extraction and Tracking Method Based on Corner Flow Detection

Jiaxi Li[1], **Binbin Wang**[1], **Huijun Ma**[1,*], **Longfei Gao**[1] **and Heran Fu**[1]

[1] National Engineering Laboratory for Agri-product Quality Traceability, BTBU, Beijing, China

Abstract

Front-end feature tracking based on vision is the process in which a robot captures images of its surrounding environment using a camera while in motion. Each frame of the image is then analyzed to extract feature points, which are subsequently matched between pairwise frames to estimate the robot's pose changes by solving for the variations in these points. While feature matching methods that rely on descriptor-based approaches perform well in cases of significant lighting and texture variations, the addition of descriptors increases computational cost and introduces instability. Therefore, in this paper, a novel approach is proposed that combines sparse optical flow tracking with Shi-Tomasi corner detection, replacing the use of descriptors. This new method offers improved stability in situations of challenging lighting and texture variations while maintaining lower computational cost. Experimental results, validated using the OpenCV library on the Ubuntu operating system, demonstrate the algorithm's effectiveness and efficiency.

Academic Editor:
Weimin Zhang

Submitted: 09 January 2024
Accepted: 10 May 2024
Published: 15 May 2024

Vol. 1, **No.** 1, 2024.
10.62762/TIS.2024.136895

***Corresponding author:**
✉ Huijun Ma
mahuijun@th.btbu.edu.cn

Keywords: computer vision, feature tracking, optical flow method, visual features, visual tracking.

1 Introduction

Visual tracking technology is a prominent topic in the field of computer vision. It comprises a range of related techniques, including moving object detection from continuous image sequences, feature extraction, classification and recognition, tracking and filtering, and behavior recognition. These methods are used to accurately estimate a target's motion parameters—such as position, velocity, acceleration, and trajectory—and to perform the corresponding processing and analysis for understanding the target's behavior. With the advancement of information technology and intelligent systems, computer vision has become a crucial component in robotics [1] and unmanned vehicles [2], making it one of the key technologies in this domain. Incremental learning-based approaches have further demonstrated the adaptability of visual tracking systems in dynamic environments [13], while surveys on aerial object detection have highlighted the breadth of computer vision applications across intelligent platforms [14].

At present, visual tracking technology can be broadly classified into two main categories. The first is based on feature points in images, which involves matching these feature points across different images to determine their motion patterns. The second approach directly detects and recognizes the target and then tracks its motion across frames. This

Citation
Li, J., Wang, B., Ma, H., Gao, L., & Fu, H. (2024). Visual Feature Extraction and Tracking Method Based on Corner Flow Detection. *ICCK Transactions on Intelligent Systematics*, 1(1), 3–9.

method must overcome challenges in video object recognition and typically relies on deep learning techniques. Its framework is relatively complex and requires significant computational time to implement [3, 4]. Therefore, approaches that first extract feature points from images offer considerable advantages in practical systems.

2 Related Works

In the field of computer vision, feature detection and matching are essential components. The process generally begins with extracting distinctive features from an image and establishing reliable correspondences between images based on these features. In this context, a "feature" refers to a salient location or structure in an image, typically including point features, line features, and other geometric primitives. The joint exploitation of these complementary feature types has been shown to enhance the robustness of visual SLAM systems [5]. Among them, point features—also known as key points, points of interest, or corner points—are the most commonly used.

Feature points can be classified into the following types:

1. Pixels corresponding to local maxima of the first derivative (i.e., image intensity gradients);
2. Intersections of two or more edges;
3. Points in an image where both the magnitude and direction of the gradient change rapidly;
4. Corner points, where the first derivative is maximal and the second derivative is zero, indicating discontinuities in the object boundary direction.

Common methods for extracting corners include the Harris Corner [6] and the Shi-Tomasi Corner [15]. The Shi-Tomasi corner extraction algorithm builds upon the foundation of the Harris Corner technique. The Harris Corner algorithm utilizes an autocorrelation matrix, specifically, a matrix M associated with the autocorrelation function. The eigenvalues of the M-matrix represent the first-order curvature of the correlation function. If both curvature values are high, the point is identified as a corner feature. Shi and Tomasi developed an improved method, which considers a point a strong corner if the smaller of the two eigenvalues exceeds a certain threshold. Compared to the Harris method, the Shi-Tomasi approach is generally more robust.

The effectiveness of corner detection and tracking significantly influences the performance of video tracking systems. Optical flow is a pivotal concept in motion detection, analyzing the movement of objects, surfaces, or edges relative to the observer. Optical flow is extensively used for motion detection [7] and tracking [8], making it one of the most prevalent methods currently in use. Many researchers have noted that using the optical flow method for extensive corner tracking and repeated calculations adds complexity, making it challenging to maintain real-time video processing [9]. Consequently, this paper proposes a hybrid approach that combines the corner method with the optical flow technique. It employs a sparse optical flow method to streamline the operations of the optical flow technique while still ensuring effective tracking of these corners.

3 Methodology

The Shi-Tomasi corner detection algorithm is one of the most widely used corner detection methods in computer vision. It identifies corner points by computing a corner response value for each pixel in the image. The core idea is to select pixels whose minimum eigenvalue of the local structure tensor exceeds a predefined threshold, as these points exhibit significant intensity changes in multiple directions. Specifically, for each pixel, the algorithm constructs a 2×2 gradient covariance matrix (structure tensor) using the horizontal and vertical image gradients within a local window. The eigenvalues of this matrix are then calculated, and the smaller eigenvalue serves as the corner response measure.

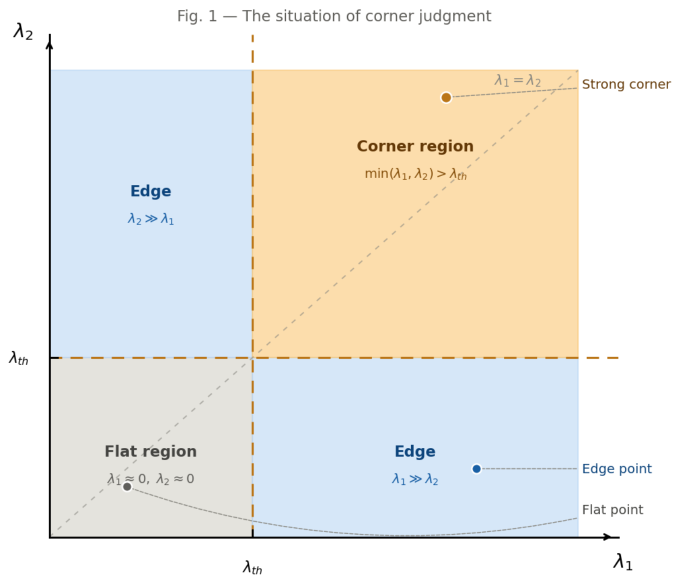

Figure 1. The situation of corner judgment.

As shown in Figure 1, during the corner selection process, a threshold can be set, and only pixels with feature response values greater than the threshold are selected as corner points. By adjusting the threshold, the number and quality of corners can be controlled. At a point (x, y) in the image, the self-similarity after a translation of (u, v) can be expressed using the gray-level variation function $E(u, v)$.

$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u, y+v) - I(x,y)]^2 \quad (1)$$

where $w(x, y)$ is a window function, and $I(x+u, y+v)$ is the pixel gray level after translation, expressed as Equation (2)

$$I(x+u, y+v) = I(x,y) + I_x u + I_y v + O(u^2, v^2) \quad (2)$$

Substituting Equation (2) into Equation (1) gives:

$$E(u,v) = [u,v] \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \quad (3)$$

When the local displacement (u, v) is very small, Equation (3) can be approximated by Equation (4) as follows

$$E(u,v) \cong [u,v] M \begin{bmatrix} u \\ v \end{bmatrix} \quad (4)$$

where M is the gradient covariance matrix of the image. The matrix M is represented by Equation (5),

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (5)$$

At this point, $E(u, v)$ can be expressed in elliptic form as shown in Figure 2:

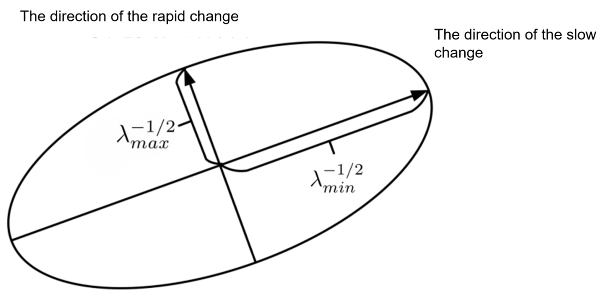

Figure 2. Representation of gray variation function.

Let the response function of characteristic points R be:

$$R = \min(\lambda_1, \lambda_2) \quad (6)$$

As shown in Figure 2, the autocorrelation matrix M can be decomposed into two eigenvalues (λ_1, λ_2) and their corresponding eigenvectors. Because smaller eigenvalues lead to greater uncertainty, i.e., $a_0^{-1/2}$, good feature points can be identified by finding the maximum value of the smallest eigenvalue. Based on the selected feature points, this paper introduces the Kanade-Lucas-Tomasi (KLT) algorithm, KLT algorithm is a traditional sparse optical flow tracking method, which solves the offset to match the image. In the past decades, KLT algorithm has been widely used in various motion tracking examples.

KLT is an algorithm used to track the same feature points across two consecutive images. The KLT algorithm relies on several assumptions: (1) image brightness remains constant; (2) spatial consistency, meaning neighboring points exhibit similar motion and remain adjacent; and (3) temporal continuity of motion, i.e., the motion is continuous or involves only small displacements. These assumptions are generally satisfied in most practical scenarios.

Let I and J denote two consecutive images in the image sequence. The grayscale value of a point (x, y) in the image is denoted as $I(x, y)$ and $J(x, y)$ Based on the assumptions of the KLT algorithm, set the coordinates of a feature point in the image I as u,the feature point correctly matched with u in image j is v, point v is obtained by moving point u a short distance d,so there are $v = u + d$, point v at time $t + 1$ corresponds to point u at time t displaced by d.The purpose of the KLT algorithm is to solve for the change in displacement d. When solving KLT, it is equivalent to searching within a certain neighborhood around the feature points. Let u expand the window range by ω_x and ω_y, the window size is $(2\omega_x + 1) \times (2\omega_y + 1)$. So finding d can be transformed into finding the minimum value of function $\varepsilon(d)$, as shown in Equation (7):

$$\varepsilon(d) = \sum_{x=u_x-\omega_x}^{x=u_x+\omega_x} \sum_{y=u_y-\omega_y}^{y=u_y+\omega_y} (I(x,y) - J(x+d_x, y+d_y))^2 \quad (7)$$

Equation (7), expressed in integral form, can be equivalent to:

$$\varepsilon(d) = \int\int (J(x+\frac{d}{2}) - I(x-\frac{d}{2}))^2 w(x) dx \quad (8)$$

The minimum value of $\varepsilon(d)$ can be obtained by solving the partial derivative of vector d to make it equal to 0, resulting in:

$$\frac{\partial_\varepsilon}{\partial_d} = 2\int\int (J(x+\frac{d}{2})-I(x-\frac{d}{2}))(\frac{\partial J(x+\frac{d}{2})}{\partial d} - \frac{\partial J(x-\frac{d}{2})}{\partial d})w(x)dx \quad (9)$$

Equation (9) is expanded using the Taylor series:

$$J(\xi) \approx J(a) + (\xi_x - a)\frac{\partial J}{\partial x}(a) + (\xi_y - a_y)\frac{\partial J}{\partial y}(a) \quad (10)$$

According to Equation (10), expanding the Taylor formula on $J(x+\frac{d}{2})$ and $I(x-\frac{d}{2})$ yields:

$$J(x+\frac{d}{2}) \approx J(x) + \frac{d_x}{2}\frac{\partial J}{\partial x}(x) + \frac{d_y}{2}\frac{\partial J}{\partial y}(x)$$

$$I(x+\frac{d}{2}) \approx I(x) + \frac{d_x}{2}\frac{\partial I}{\partial x}(x) + \frac{d_y}{2}\frac{\partial I}{\partial y}(x) \quad (11)$$

Substituting Equation (11) into Equation (9) gives:

$$\frac{\partial_\varepsilon}{\partial_d} \approx \int\int (J(x) - I(x) + g^T d)g(x)w(x)dx \quad (12)$$

Among them, the expression for g is:

$$g = [\frac{\partial}{\partial x}(\frac{I+J}{2})\frac{\partial}{\partial y}(\frac{I+J}{2})]^T \quad (13)$$

What ultimately needs to be solved is:

$$\frac{\partial_\varepsilon}{\partial_d} = \int\int (J(x) - I(x) + g^T d)g(x)w(x)dx = 0 \quad (14)$$

After opening it:

$$\int\int [J(x) - I(x)]g(x)w(x)dx = -\int\int g^T(x)dg(x)w(x)dx \quad (15)$$

Factoring out d from the right-hand side yields:

$$-\int\int g^T(x)dg(x)w(x)dx = -[\int\int g(x)g^T(x)w(x)dx]d \quad (16)$$

Equation (16) can be simplified as:

$$Zd = e \quad (17)$$

where z and e are:

$$Z = \int\int g(x)g^T(x)w(x)dx$$

$$e = \int\int [I(x) - J(x)]g(x)w(x)dx \quad (18)$$

For d to have a unique solution, the matrix Z must be invertible. Equation (17) calculates the offset d iteratively, which can obtain a more accurate offset. The iterative method is:If the displacement obtained from the $(k-1st)$ iteration is $d^{k-1} = [d_x^{k-1} d_y^{k-1}]$, then $J(x,y)$, which is the $k-th$ iteration, can be represented by the $(k-1st)$ iteration as follows:

$$J(x,y) = J(x + d_x^{k-1}, y + d_y^{k-1}) \quad (19)$$

By substituting the iterative update of Equation (19) into the minimization framework, the k-th displacement d^k can be obtained as follows

$$\varepsilon(d) = \sum_{x=u_x-w_x}^{u_x+w_x}\sum_{y=u_y-w_y}^{u_y+w_y}(I(x,y) - J(x+d_x^{k-1}, y+d_y^{k-1}))^2 \quad (20)$$

$$d^k = Z^{-1}e_k \quad (21)$$

We assume that after k iterations, Equation (20) converges, resulting in d:

$$d = \sum_{k=1}^{K} d^k \quad (22)$$

By solving for d, we determined the matching relationship between the feature points on image I and image J, as well as the motion relationship between the two frames of images.

In the experiment, visualized optical flow tracking results were obtained, as shown in Figure 3. This figure shows a still frame from a video in which a bus and a car pass in front of a stationary camera. Since the background remains static, no optical flow trajectories appear on stationary objects. The red markers indicate feature points on the moving bus and car, while the green lines represent the tracking trajectories of the feature points.

The video image processing pipeline is as follows:

1. Firstly, the first frame image is detected, and 200 Shi-Tomasi corners are extracted from the first frame image using the goodFeaturesToTrack function in Opencv. When extracting corners, the interval between corners is more than 30 pixels to ensure the uniform distribution of feature points. Then, each feature point is tagged;
2. The image frames after the first frame are tracked by KLT sparse optical flow, and the untracked feature points are eliminated;

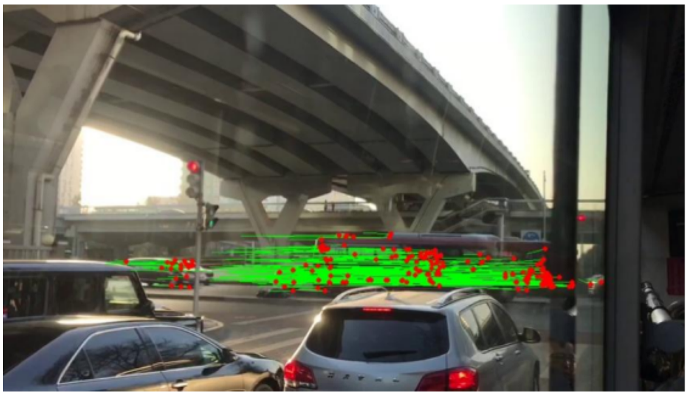

Figure 3. Visualization results of optical flow tracking.

3. The tracking between two frames also produces mismatched points. The outlier points are removed by 2 point-RANSAC method;
4. Then, compute the number of remaining feature points k, and increment the successful tracking count for each of these points. A mask is then applied within a 30-pixel neighborhood around the remaining feature points, preventing the selection of additional points in these regions. This process ensures a uniform spatial distribution of the extracted features;
5. 200-k Shi-Tomasi corners are extracted in the current frame, so that the total number of feature points is still 200;
6. Go to the next frame and repeat the cycle as above.

Figure 4. Image of extracting feature points without mask.

Figure 5. Image of extracting feature points after mask is set.

In the front-end feature tracking algorithm pipeline, when extracting feature points in the first frame, a constraint is imposed to ensure that the distance between feature points is greater than 30 pixels. During the feature tracking process, a mask is applied so that no new feature points are added within a 30-pixel neighborhood around already tracked points. This strategy ensures a uniform spatial distribution of feature points in the image, which is beneficial for subsequent pose estimation. Without this operation, the feature points would be unevenly distributed, as shown in Figure 4. After applying the mask, the extracted image feature points are shown in Figure 5.

4 Experiments

To verify the effectiveness of the proposed method, we compared it with the traditional descriptor-based feature matching method. The comparison involved conducting experiments on the same dataset to assess both methods in terms of processing time and translation error. The experiments were conducted using a Lenovo ThinkPad E470 computer equipped with an i5-7200U CPU at 2.50 GHz, 8 GB RAM, and running the Ubuntu 16.04 operating system, along with Kinetic for ROS. The experiment is carried out on a set of vehicle data collected from Kwun Tong Community, Chaoyang District, Beijing. The vehicle speed is from 0 km / h to 30 km / h.

The experimental results of video-based feature matching are shown in Figure 6, where red denotes the results of the proposed method and blue denotes those of traditional feature matching. Compared with the traditional feature matching method [10], the experimental time consumption and translation error are presented in Figure 7. This figure illustrates the comparison of computational time between the two methods, where the horizontal axis represents the number of extracted feature points. The vertical axis is the time needed. It can be seen that with the increase of the number of extracted feature points, the calculation time of the two methods for feature extraction and data association is also increased, but the optical flow tracking method proposed in this paper is less time-consuming than the feature matching method. The horizontal axis represents the number of extracted feature points, and the vertical axis represents the translation error. It can be seen from Figure 8 that the error of the proposed method is less than that of the feature point matching method within 800 feature points. Experimental results show that compared with feature point matching methods adopted in video processing applications [11, 12], the proposed method is less time-consuming and has lower translation error.

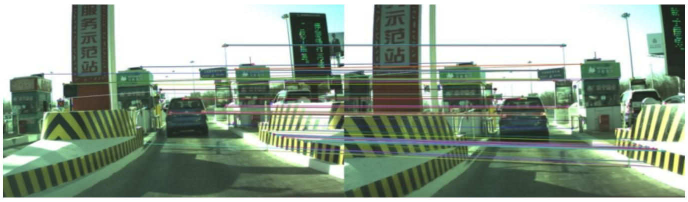

Figure 6. Result of feature matching.

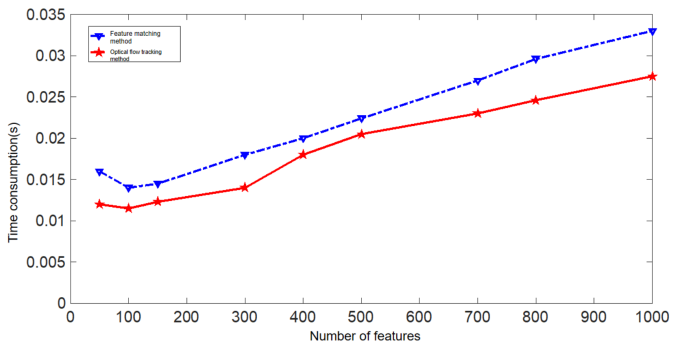

Figure 7. Computational time comparison between the proposed optical flow tracking method and the traditional feature matching method with varying numbers of feature points.

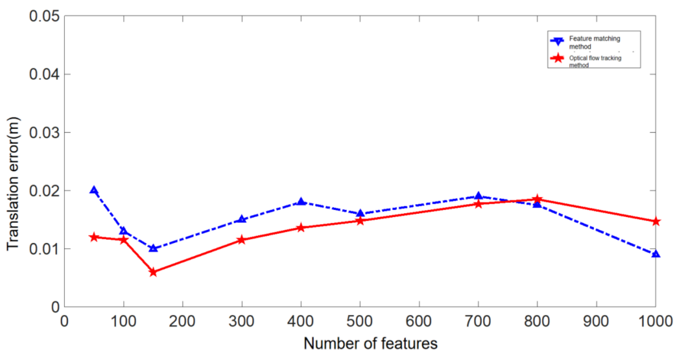

Figure 8. Translation error comparison between the proposed optical flow tracking method and the traditional feature matching method with varying numbers of feature points.

5 Conclusion

This paper addresses the challenges of high computational complexity and instability associated with descriptor-based feature point matching in visual tracking. To overcome these limitations, we propose a novel approach that integrates Shi-Tomasi corner detection with sparse optical flow (KLT) tracking.

The proposed method first employs the Shi-Tomasi algorithm to extract high-quality corner features, which effectively reduces redundant information, lowers computational cost, and enhances feature stability and accuracy. Second, we optimize the optical flow tracking process by incorporating motion-based feature selection and outlier rejection mechanisms. This helps eliminate low-quality matches, mitigates mismatching issues, and further improves the robustness of the tracking algorithm. Experimental results, compared with traditional descriptor-based feature matching methods, demonstrate that the proposed approach achieves superior performance in both computational efficiency and translation accuracy.

In summary, the integration of Shi-Tomasi corner detection and sparse optical flow tracking provides a more stable and efficient solution for front-end visual feature tracking. The method exhibits strong robustness under significant variations in illumination and texture, while enabling real-time robot pose estimation.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported in part by the National Natural Science Foundation of China under Grant 62173007, Grant 62006008, and Grant 62203020; in part by the Project of Humanities and Social Sciences (Ministry of Education in China, MOC) under Grant 22YJCZH006.

Conflicts of Interest

The authors declare no conflicts of interest.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Papanikolopoulos, N. P., Khosla, P. K., & Kanade, T. (1993). Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision. *IEEE transactions on robotics and automation*, *9*(1), 14-35. [CrossRef]

[2] Li, S., & Yeung, D. Y. (2017, February). Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 31, No. 1). [CrossRef]

[3] Jiao, L., Wang, D., Bai, Y., Chen, P., & Liu, F. (2021). Deep learning in visual tracking: A review. *IEEE transactions on neural networks and learning systems*, *34*(9), 5497-5516. [CrossRef]

[4] Marvasti-Zadeh, S. M., Cheng, L., Ghanei-Yakhdan, H., & Kasaei, S. (2021). Deep learning for visual tracking: A comprehensive survey. *IEEE Transactions on Intelligent Transportation Systems*, *23*(5), 3943-3968. [CrossRef]

[5] Haifeng, L., Zunhe, H., & Xinwei, C. (2017). PLP-SLAM: A visual SLAM method based on point line and surface feature fusion. *Robot*, *39*(02), 214-220.

[6] Noble, J. A. (1988). Finding corners. *Image and vision computing*, *6*(2), 121-128. [CrossRef]

[7] Agarwal, A., Gupta, S., & Singh, D. K. (2016, December). Review of optical flow technique for moving object detection. In *2016 2nd international conference on contemporary computing and informatics (IC3I)* (pp. 409-413). IEEE. [CrossRef]

[8] Nie, G. Y., Bodda, S. S., Sandhu, H. K., Han, K., & Gupta, A. (2022). Computer-vision-based vibration tracking using a digital camera: A sparse-optical-flow-based target tracking method. *Sensors*, *22*(18), 6869. [CrossRef]

[9] Plyer, A., Le Besnerais, G., & Champagnat, F. (2016). Massively parallel Lucas Kanade optical flow for real-time video processing applications. *Journal of Real-Time Image Processing*, *11*(4), 713-730. [CrossRef]

[10] Sinha, S. N., Frahm, J. M., Pollefeys, M., & Genc, Y. (2006, May). GPU-based video feature tracking and matching. In *EDGE, workshop on edge computing using new commodity architectures* (Vol. 278, p. 4321).

[11] Abdullah, L. M., Tahir, N. M., & Samad, M. (2012, July). Video stabilization based on point feature matching technique. In *2012 IEEE Control and System Graduate Research Colloquium* (pp. 303-307). IEEE. [CrossRef]

[12] Kulkarni, S., Bormane, D. S., & Nalbalwar, S. L. (2017, January). Video stabilization using feature point matching. In *Journal of Physics: Conference Series* (Vol. 787, No. 1, p. 012017). IOP Publishing. [CrossRef]

[13] Lim, J., Ross, D., Lin, R. S., & Yang, M. H. (2004). Incremental learning for visual tracking. *Advances in neural information processing systems*, *17*.

[14] Cazzato, D., Cimarelli, C., Sanchez-Lopez, J. L., Voos, H., & Leo, M. (2020). A survey of computer vision methods for 2d object detection from unmanned aerial vehicles. Journal of Imaging, 6(8), 78. [CrossRef]

[15] Shi, J. (1994, June). Good features to track. In *1994 Proceedings of IEEE conference on computer vision and pattern recognition* (pp. 593-600). IEEE. [CrossRef]

Jiaxi Li received the B.S. degree in Automation from Beijing Technology and Business University, Beijing, China, in 2022. He is currently pursuing a Master's degree in Control Engineering at Beijing Technology and Business University. His main research interests focus on navigation and control, as well as multimodal object detection technologies and their applications. (Email: lijiaxi@st.btbu.edu.cn)

Binbin Wang received the M.S. degree in Control Engineering from Beijing Technology and Business University, Beijing, China, in 2019. His main research interests include pattern recognition and information fusion, unmanned vehicles, machine learning, and other related fields. (Email: wangbinbin@st.btbu.edu.cn)

Huijun Ma received the M.S. degree in Atomic and Molecular Physics from the Changchun Institute of Optics, Fine Mechanics and Physics, Changchun, China, in 2010. She is currently pursuing a Ph.D. in Systems Science at Beijing Technology and Business University. Her research interests include complex system modeling, pattern recognition and information fusion, and machine learning. (Email: huijunma@btbu.edu.cn)

Longfei Gao received the B.S. degree in Electrical Engineering and Automation from the University of Jinan, Shandong, China, in 2022. He is currently pursuing a Master's degree in Control Engineering at Beijing Technology and Business University. His main research interests focus on image detection and pattern recognition. (Email: gaolongfei@st.btbu.edu.cn)

Heran Fu received the B.S. degree in Automation from Beijing Technology and Business University, Beijing, China, in 2022. He is currently pursuing a Master's degree in Control Engineering at Beijing Technology and Business University. His main research interests include image classification and detection, pattern recognition, and deep learning. (Email: fuheran@st.btbu.edu.cn)