China Instrument and Control Society
Instrumentation
OPEN ACCESS
INSTRUMENTATION, Vol. 12, No. 3, September 2025 https://doi.org/10.15878/j.instr.202500296

Article

A Real-time Stitching Method Based on FLANN Algorithm of Multi-UAV for Remote Sensing Image

Hanxiang Qian, Xiaojun Guo*, Shaojing Su, Xiaoyong Sun, Feiyang Liu

Country College of Intelligence Science and Technology, National University of Defense Technology, Changsha, Hunan, China

* Corresponding author email: jeanakin@nudt.edu.cn

Abstract: The proliferation of low-altitude economic activities has precipitated a significant expansion in the application domains of Unmanned Aerial Vehicle (UAV), transcending the constraints of singular platform operation. Multi-UAV cooperative online remote sensing image stitching involves several challenges, including image blurriness and deformation caused by platform vibrations, as well as managing vast amounts of data and addressing transmission delays. This paper introduces a real-time system for multi-UAV video stitching that integrates hardware acceleration, pose-guided ROI filtering, and temporal consistency constraints. Data from the onboard GPS and Inertial Measurement Unit (IMU) are used to estimate the overlap between two aerial images. Subsequently, the Graphics Processing Unit (GPU) within an edge computing platform is tasked with swiftly detecting SIFT features within the overlapping image sectors. Feature point matching is performed using a position-guided matching algorithm to compute the homography matrix. The results demonstrate that the RK3588-based edge computing platform exhibits rapid image decoding and stitching capabilities, with an average stitching latency of 36.43ms. Furthermore, this study developed a ground-based human-machine interaction system to present stitched video streams in real-time, thereby improving the operational efficiency and visual quality of multi-UAV aerial surveillance.

Keywords: image stitching; aerial image; homograph matrix; sift feature; edge computing

Citation: Hanxiang Qian, Xiaojun Guo, Shaojing Su, Xiaoyong Sun, Feiyang Liu. "A Real-time Stitching Method Based on FLANN Algorithm of Multi-UAV for Remote Sensing Image." Instrumentation 12, no.3 (September 2025). https://doi.org/10.15878/j.instr.202500296

1 Introduction

Image stitching, a fundamental task in computer vision, has been extensively studied over the past few decades. Its primary objective is to combine multiple video sources with overlapping regions using algorithms that feature point matching to create broad-view images. This technology has wide applications in video surveillance, autonomous driving, virtual reality, and medical treatment.

Recently, with the emergence of the low-altitude economy, drones are being increasingly deployed. Their ability to collaborate and capture images from multiple viewpoints provides users with a comprehensive global perspective. Compared to surveillance cameras, unmanned aerial vehicles face higher real-time demands and are constrained by limited computing and communication resources. They must also address challenges such as platform instability and motion.

To address these challenges, this paper proposes a method for large-view video splicing that utilizes UAV position data and an improved SIFT feature extraction algorithm. During reconnaissance missions, a single UAV's field of view is limited; thus, multiple units must be used to gather video from various angles and stitch these together to form a comprehensive wide field of view. This approach provides commanders with high-resolution, large-angle video or image data of battlefield environments. As shown in Fig. 1, traditional video

splicing algorithms primarily rely on feature point matching, mismatch elimination, and single transformation matrices. However, in unknown environments, traditional algorithms struggle with images characterized by weak textures, low-light conditions, or repeated patterns, as these provide fewer distinguishable features. Consequently, feature extraction becomes difficult, affecting both image quality and splicing accuracy. Additionally, feature extraction is time-consuming, presenting challenges for meeting real-time video requirements.

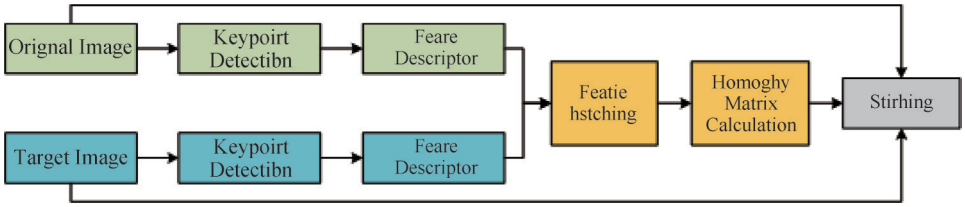


Fig.1 Traditional video stitching algorithm structure

The splicing of aerial images typically involves three steps: acquiring the images to be spliced, aligning the images, and fusing the images. Among these, image alignment is the core part of the entire image stitching process, as it is directly related to the quality of the final result. Its main purpose is to obtain the homography matrix between two images. The current UAV image alignment technology is mainly based on feature point matching, which utilizes local features extracted from images for matching. Its key steps are the extraction and matching of image features. However, directly extracting features from the original image and performing matching can produce many invalid detection points, which significantly increases computational demands. This project proposes an aerial image stitching method that combines UAV position information with local feature point matching. This method enables real-time image processing while maintaining accuracy.

In summary, to address the challenges of multi-UAV large field of view splicing in terms of splicing speed, data transmission, and stitching quality, this paper proposes using the RK3588 as the edge computing platform. It comprehensively utilizes GPS and IMU information, adopts Vulkan-SIFT as the baseline, and accelerates feature extraction by combining it with a priori knowledge regarding regions of interest (ROIs). Additionally, it applies position information as constraints to build a video streaming ground station visualization and control system. The data flow and system modules are shown in Fig. 2,

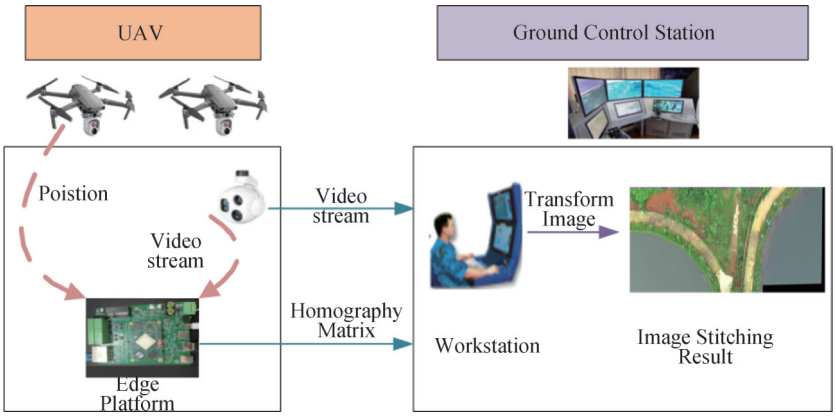


Fig.2 Data flow of the real-time stitching framework

The contributions of this paper are summarized as follows:

(1) A high-speed video decoding method based on the RK3588 chip is implemented, utilizing the Multimedia Processing Platform (MPP) and GPU for rapid processing of H.264 streams from drone sensors.

(2) An improved Fast Library for Approximate Nearest Neighbors (FLANN) algorithm is proposed, integrating pose prior knowledge and ROI data from previous frames to achieve lightweight, fast, and high-quality image stitching.

(3) A software system based on the VUE framework is developed, displaying stitching results via WebStreamer and UDP protocol.

2 Related Work

Image stitching is a classical computer vision problem with a wide range of applications, including expanding the detection field of view and supporting remote sensing measurement and control. It is defined as the alignment of images taken from different viewpoints into a common coordinate system, followed by the blending of these images to avoid deformation at the seams. Stitching methods are broadly categorized into traditional feature point detection and deep learning-

based approaches. Traditional methods generally employ homography transforms, which apply a pairwise transformation to stitch images into a unified canvas based on invertibility. Image stitching typically consists of two main steps: feature detection and feature matching.

In the domain of traditional image stitching, the SIFT algorithm introduced by Brown and Lowe in 2003 represented a significant advancement[1]. This algorithm constructs a Gaussian pyramid of images and uses 128-dimensional vectors to describe feature points, enabling effective adaptation to scale and rotation changes. In 2008, Herbert Bay and colleagues introduced the SURF algorithm, which significantly improved feature detection efficiency by utilizing Hessian matrix-based metrics and distribution-based descriptors[2]. In 2011, Rublee et al. developed the ORB algorithm, which employs BRIEF-based rapid binary descriptors and operates two orders of magnitude faster than SIFT[3].

Inspired by developments in visual tasks such as optical flow estimation and stereo matching, recent image stitching research based on deep learning has focused on constructing more efficient model architectures, designing specialized stitching scenarios, and optimizing training strategies. From early convolutional neural network (CNN) based approaches[4] to more recent unsupervised learning techniques[5], deep learning methods have effectively addressed challenges such as low-texture scene alignment and large disparity handling—issues with which traditional techniques often struggle. Notable contributions include models such as SuperGlue[6] and LightGlue[7].

In the field of aerial image stitching, researchers are increasingly integrating multiple sources of information to enhance both the quality and speed of the stitching process, in contrast to approaches that rely solely on image features. Huang Yingdong and colleagues proposed a UAV image stitching method primarily based on feature point matching, focusing on addressing outlier feature points[8]. Guo et al. introduced a method that combines geographic coordinates with image features, improving reliability in long-sequence aerial photography[9]. Wei and colleagues developed the Deep-Assisted Incremental Image Stitching (DAIIS) method, which significantly reduces stitching errors caused by large depth variations using single-view depth estimation and planar fitting, making it suitable for complex urban environments[10]. Ren et al. presented a fast stitching technique based on a multilayer perceptron, initially applying UAV pose information for coarse registration and then using a multilayer perceptron for rapid correction, achieving an optimal speed-accuracy tradeoff[11]. Wang et al. proposed a spatial arrangement-preserving algorithm based on triangular similarity transformations, which maintains spatial consistency in stitched images, particularly effective in homogeneous scenes such as farmlands[12]. Li and associates introduced a real-time UAV video stitching framework that employs keyframe selection and an optimization method minimizing weighted projection error to produce high-quality panoramas in real time[13]. Mo et al. proposed a UAV hyperspectral image stitching method based on deep feature matching, utilizing a robust VGG-style network for feature point extraction and a graph neural network for feature point matching[14].

These methods leverage diverse forms of information integration to meet high real-time requirements and address challenges posed by weak texture features and significant depth variations in UAV image stitching. Traditional algorithms continue to face limitations, including the need to enhance the robustness and real-time performance of inter-frame feature matching techniques. Additionally, large video frame dimensions reduce the efficiency of feature matching, while downsampled video frames result in lower resolution in the stitched images.

3 Methods

3.1 Overall Structure

As shown in Fig.3, we propose an efficient and fast aerial image stitching method based on an improved FLANN algorithm. The main process of this method includes video stream decoding, feature point detection, feature point matching, image fusion, and result output. The green box represents the operation on the unmanned aerial edge computing platform, and the blue box represents the operation on the ground station. After decoding the video streams from the left and right drones, use RGA for color conversion. Next, SIFT feature extraction and improved FLANN feature point matching algorithm are applied, combined with prior pose data to generate a homography matrix. Next, the video stream from the drone is directly decoded at the ground station, and then the image is transformed through homography matrix to obtain visualization results at the ground station. Our architecture adopts a cloud edge fusion structure to minimize display latency to the greatest extent possible.

3.2 Homography Modeling from UAV Pose

Feature matching is an $O(N^2)$ problem that grows quadratically with the number of input features to be matched. In this project, an a priori feature filtering algorithm is employed to pre-filter feature points prior to matching. This algorithm reduces computational complexity and enhances accuracy. The a priori knowledge used includes the following assumptions: the video to be matched has minimal relative motion, the cameras remain relatively stationary or exhibit only small relative movements, and scene transformations are neither drastic nor rotational.

Firstly, the theoretical basis of UAV video stitching is introduced. In addition to the general feature point

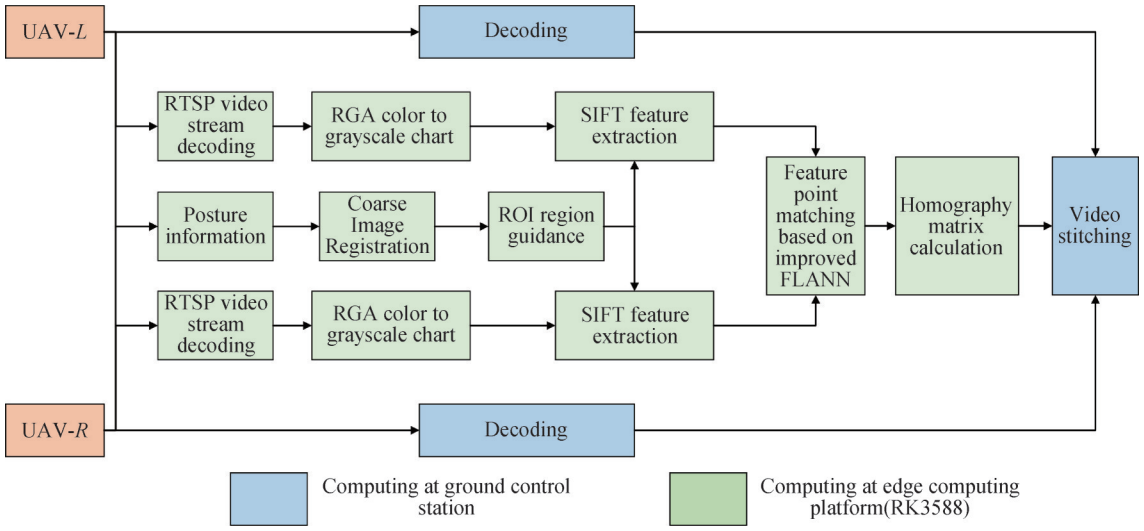


Fig.3 Overall structure

matching, the homography matrix can also be calculated by using the pose information of the UAV. Although this will bring some errors, we can use it as a priori information to efficiently guide feature point matching.

As shown in Fig. 4, the UAV takes images of the ground scene at O_1 and O_2, thus obtaining images I_1 and I_2. $(\mathrm{X},\mathrm{Y},\mathrm{Z})^T$is a point in the field; $(\mathrm{X}_1,\mathrm{Y}_1,\mathrm{Z}_1)^T$ and $(\mathrm{X}_2,\mathrm{Y}_2,\mathrm{Z}_2)^T$denote the camera coordinates of M at two positions; $\mathrm{m}_1(\mathrm{u}_1,\mathrm{v}_1,1)^T$ and $\mathrm{m}_2(\mathrm{u}_2,\mathrm{v}_2,1)^T$ are the points of M on images I_1 and I_2 respectively. If the matrix H can satisfy:

$$m_2 = sHm_1 \tag{1}$$

The matrix H is the unary homography matrix of the image I_1 to I_2. where s is a nonzero constant term.

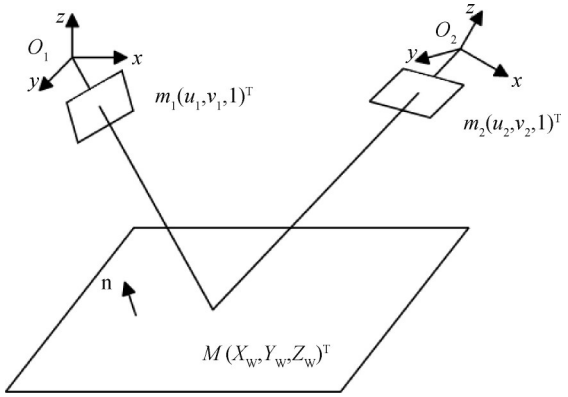


Fig.4 Aerial images from different viewpoints

Based on the imaging principle of the camera, the homograph matrix between two aerial images can be expressed as follows:

$$\mathrm{H}=\mathrm{KR}\left(\mathrm{I}-\frac{\mathrm{t}}{\mathrm{d}}\mathrm{n}^T\right)\mathrm{K}^{-1}=\mathrm{KRK}^{-1}-\frac{1}{\mathrm{d}}\mathrm{KRtn}^T\mathrm{K}^{-1} \tag{2}$$

where K is the internal parameter matrix of the camera. The motion between O_1 and O_2 is $R[I,-t]$, where R is a 3×3 rotation matrix and t is the translation vector. X is the point on the ground, n is the unit normal vector of the ground, and d is the distance from the coordinate origin to the earth plane.

In this manner, the homography matrix between aerial images can be obtained. The O where the drone is located can be represented by six parameters, which are recorded as $\mathrm{O_i}(\varphi_\mathrm{i},\theta_\mathrm{i},\psi_\mathrm{i},\mathrm{x_i},\mathrm{y_i},\mathrm{z_i})$. The UAV is equipped with an inertial navigation unit and GPS that can output the attitude angle and coordinates of the UAV in real time. The rotation matrix R and translation vector t are calculated from the differences in Euler angles (roll, pitch, yaw) and GPS coordinates between two viewpoints.

Thus, the transformation matrix between two aerial images can be calculated using Eq. 3 and Eq. 4. It can be observed from the formula that deriving a single transformation matrix between aerial images requires all points in the image to be at the same depth. That is, the entire scene under the viewpoint should lie within a single imaging plane; otherwise, a single transformation between images cannot be satisfied. In practice, it is not strictly necessary for all pixels to have the same depth, which is one of the main sources of stitching error in aerial imagery.

$$\begin{aligned}
&\mathrm{R}=\mathrm{R}_\varphi\mathrm{R}_\theta\mathrm{R}_\psi\\
&\mathrm{R}_\varphi=\begin{bmatrix}1&0&0\\0&\cos\varphi&\sin\varphi\\0&-\sin\varphi&\cos\varphi\end{bmatrix}\\
&\mathrm{R}_\theta=\begin{bmatrix}\cos\theta&0&-\sin\theta\\0&1&0\\\sin\theta&0&\cos\theta\end{bmatrix}\\
&\mathrm{R}_\psi=\begin{bmatrix}\cos\psi&\sin\psi&0\\-\sin\psi&\cos\psi&0\\0&0&1\end{bmatrix}
\end{aligned} \tag{3}$$

$$\mathrm{t}=\begin{bmatrix}\mathrm{x}_2-\mathrm{x}_1&\mathrm{y}_2-\mathrm{y}_1&\mathrm{z}_2-\mathrm{z}_1\end{bmatrix}^T \tag{4}$$

where (ϕ,θ,ψ) denote the roll, pitch, and yaw angles.

The image stitching method based solely on position

information involves processing UAV positional data to compute a homography matrix for image stitching. This method is highly efficient and fast, with processing completed in milliseconds. As it eliminates the need to extract image features and applies only a one-shot transformation to the image, both the computational load and processing time are significantly reduced. This rapid processing makes the position-based stitching method particularly advantageous for real-time or fast-processing scenarios.

3.3 ROI-Guided Feature Matching with Temporal Consistency

In this section, we propose an improved FLANN-based feature point matching algorithm based on prior knowledge. Its core idea is to determine the overlapping regions using prior knowledge and then search for matching points only within the corresponding overlapping areas.

FLANN is a fast nearest-neighbor search algorithm that performs well with high-dimensional features. The FLANN algorithm improves parallel search performance by employing a hierarchical clustering tree and building multiple trees simultaneously. The method proposed in this paper adopts an R-FLANN approach. While the original FLANN determines matching quality by calculating Euclidean distance between feature points, it includes mismatched and irrelevant feature points in this process. Additionally, the RANSAC algorithm requires repeated iterations to identify the most probable model, which does not satisfy the demands of real-time aerial video processing.

To address this issue, this project proposes an improved FLANN matching algorithm tailored for real-time aerial video. This algorithm incorporates principles from the RANSAC method to eliminate incorrectly matched or unmatched feature points, thus improving matching speed to meet the requirements of real-time aerial video stitching. Due to the low distinctiveness and complexity of terrain features such as grass or water surfaces, a large number of non-informative SIFT feature points are often detected. To reduce this redundancy, the original video image sequence is downsampled prior to SIFT feature point extraction.

As shown in Fig.5, we obtained roughly overlapping regions from pose prior information and corresponding regions for feature points from historical stitching results. Considering that the relative pose relationship of the drone will not undergo significant changes within a short time delay (<40ms), we only search for matching points near the matching area of the previous frame for the feature points.

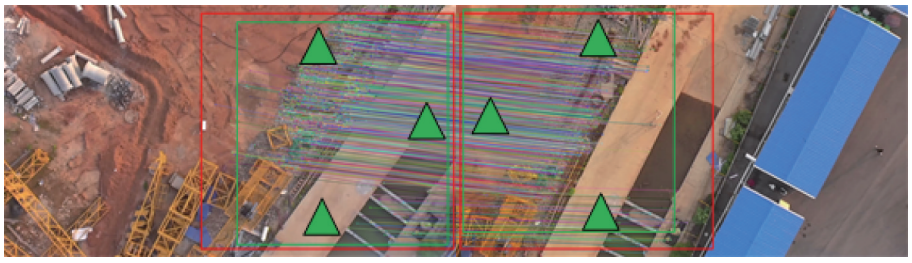


Fig.5 Feature point matching based on prior knowledge guidance

Additionally, before calculating the Euclidean distance between feature points, a preliminary screening based on their gradients is applied. If the gradients of two feature points have the same sign, it indicates that they exhibit similar contrast changes in the same direction. The flow of the improved algorithm is shown in Fig.6.

The principle of improved FLANN matching algorithm is as follows:

(1) In the original FLANN algorithm, the RANSAC algorithm is used to increase the prior information of feature point matching to reduce the amount of pairing calculation. By using the matching point pairs obtained from the previous frame image, the RANSAC algorithm is used to calculate the mapping relationship between the matching points. Unlike the traditional RANSAC iteration cycle, here only four groups of feature points are selected from the matching feature points of the previous frame, and their mapping matrix H is calculated. In this way, the final model can be obtained without iteration. The following equation is shown:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{5}$$

where (x', y') are the coordinates of the mapping points of the feature points (x, y), and H is the mapping matrix. The solution of the mapping matrix H can be obtained by expanding the above equation as follows:

$$\begin{cases} h_{11}x + h_{12}y + h_{13} - h_{31}xx' - h_{32}yx' - h_{33}x' = 0 \\ h_{21}x + h_{22}y + h_{23} - h_{31}xy' - h_{32}yy' - h_{33}y' = 0 \end{cases} \tag{6}$$

Assume that $h_{33} = 1$, in order to solve the problem that the transformation matrix can be scaled arbitrarily. Each group of matched eigenpoints can get two sets of equations, so 4 sets of equations can be obtained from 4 groups of eigenpoints. By solving these 4 sets of equations, the 8 parameters of the matrix can be calculated, and the transformation matrix can be obtained. Utilize the obtained mapping matrix. If the current frame is the first frame of the video, or the pairing point was not found in the previous frame, then skip this

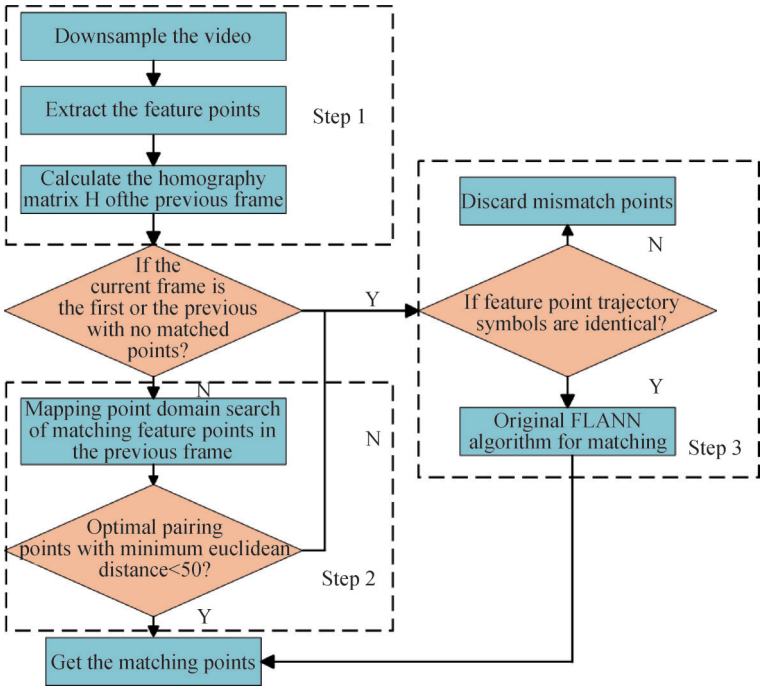


Fig.6 ROI guided FLANN matching algorithm

step and use the original FLANN algorithm to solve the pairing point directly.

(2) The algorithm increases the prediction region of the pairing points according to the RANSAC algorithm and searches for the pairing points. By calculating the predicted pairing point of feature point $p_0(x_0, y_0)$, the prediction area of the predicted point is taken as the prediction region of the pairing point in the field j of the predicted point. j is the radius of the neighborhood, where j is verified by the empirical value, and the specific value of j is determined by the method of cross-checking. Firstly, the coordinates of the mapping point p_0 are calculated using Eq. (5), and then the search for the pairing point in field j of the mapping point is prioritized. The pair of points with the minimum Euclidean distance from the domain of p_0 is found. Suppose that two m-dimensional feature vectors are$p(x_{p1}, x_{p1}, \ldots, x_{pm})$、$q(x_{q1}, x_{q1}, \ldots, x_{qm})$, then their Euclidean distances D_{pd} are as follows:

$$D_{pd} = \left[\sum_{i-1}^{m} \left(x_{pi} - x_{qi} \right)^2 \right]^{1/2} \tag{7}$$

In the field of p_0 mapping points, there is a great probability that the best matching point with the minimum Euclidean distance less than the threshold can be found. The threshold determines the specific value according to the empirical analysis and cross-validation method of aerial image data set. Otherwise, step (3) and step (4) continue to call the original FLANN algorithm to search for the remaining feature points.

(3) Before FLANN algorithm matching, the trace symbols of feature points are classified to reduce the amount of pairing calculation. By determining whether the trace symbols of two feature points are the same or not, we can proceed to the next step, otherwise, we will directly determine that they are different points and skip the Euclidean distance calculation. The following is shown as follows:

$$\begin{cases} \text{Trace}\,(H_1) \cdot \text{Trace}\,(H_2) \geq 0 \rightarrow \text{next} \\ \text{Trace}\,(H_1) \cdot \text{Trace}\,(H_2) < 0 \rightarrow \text{jump} \end{cases} \tag{8}$$

(4) By judging whether the ratio of the minimum Euclidean distance and the sub-minimum Euclidean distance of the feature point is lower than the T threshold to screen whether there is a unique matching point for the feature point, the following is a system of equations for screening feature point matching pairs.

$$\begin{cases} D_{pd} \cdot D_{pk} \leq T \rightarrow p's \text{ adaptable point} \\ D_{pd} \cdot D_{pk} > T \rightarrow \text{not } p's \text{ adaptable point} \end{cases} \tag{9}$$

Where D_{pd} is the minimum Euclidean distance from feature point p, and D_{pk} is the sub-minimum Euclidean distance from feature point p. T is the ratio threshold, where T is verified by empirical value, and then the specific value of T is determined by cross-validation method.。

The effect is shown in Fig.7, which shows the initial matching points and the points optimized by RANSAC, and we can see that the slopes are basically the same, which is consistent with the physical phenomenon of splicing in this environment.

3.4 Edge Acceleration with RK3588

This section focuses on optimizing the use of MPP hardware decoding and GPU processing to enhance drone edge computing platform computational efficiency. The main function of MPP hardware decoding is to convert

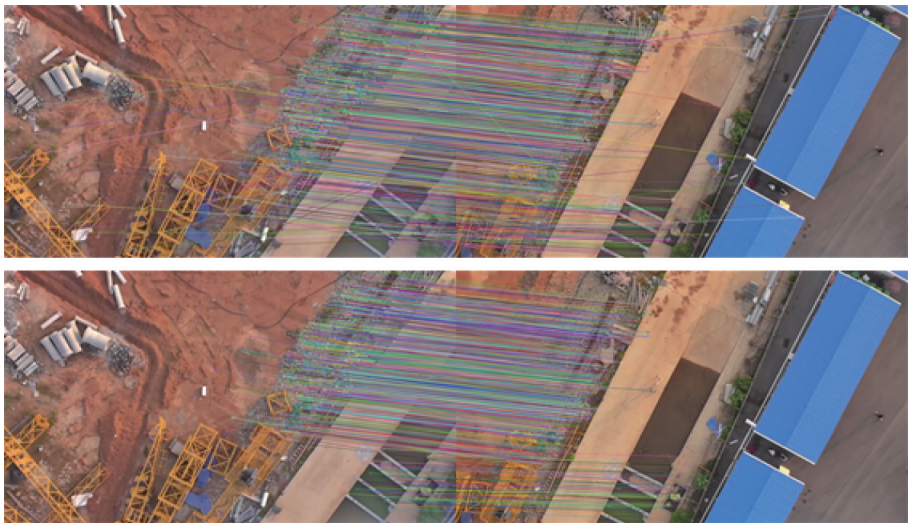


Fig.7 Feature point matching based on improved FLANN

the H.264 real-time video stream from the optoelectronic pod into image frames that can be directly processed by OpenCV. The GPU is primarily responsible for extracting image feature points.

Fig.8 illustrates the flow diagram of MPP hardware decoding. MPP is a dedicated module of the onboard RK3588 chip, which activates the hardware decoding module to locate a complete frame structure after receiving the real-time stream data and then decodes the video data.

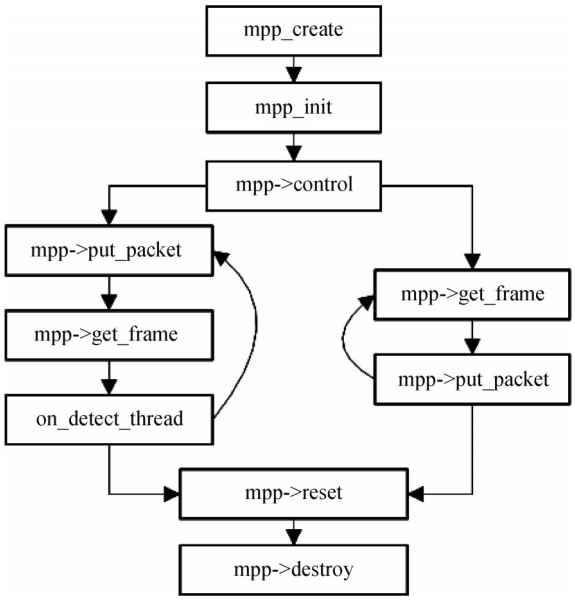


Fig.8 Diagram of H.264 video stream multimedia processing platform decoding

The video data are typically stored in H. 264 encoding format, which must be decoded before stitching. In this project, the RK3588 hardware decoding module (VPU) is employed to decode two video channels in real time. The VPU module supports a maximum of 32-channel video decoding, with a maximum decoding capability of 8K/60fps, which satisfies the experimental requirements of this study. The decoding process is illustrated in Fig.9. Initially, the video file to be decoded is opened, and its resolution and encoding format are read. The available decoders are then located and registered, followed by setting their decoding format and resolution. The decoders for each channel are then activated. Subsequently, the video buffer pool and buffer blocks are initialized according to the decoding format and resolution. The system checks whether undecoded video images are available. If so, they are decoded; otherwise, decoding fails and the thread exits. The decoded image is then sent to the output channel. Finally, the system checks whether decoding is complete; if so, the thread exits, otherwise decoding continues in a loop.

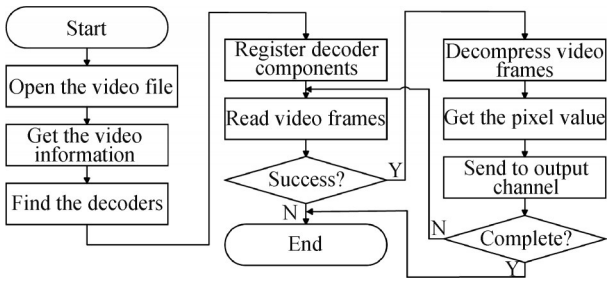


Fig.9 Video decoding process

The acquisition of multi-channel video data in this study is completed using the RK3588 MPP and the intelligent hardware accelerator (RGA). Development based on RGA enables hardware-level program acceleration, reduces processing time on the ARM side, and allocates more resources to the stitching module. During video data acquisition, the decoding function in MPP is first used to decode multiple video channels. Then, the fast memory copy module (DMA) in RGA transfers the decoded video images to the user space. The image format conversion module in RGA converts the YUV-format video data into RGB format for use in the stitching module. The video acquisition process is depicted in Fig.10. The specific execution steps are as follows:

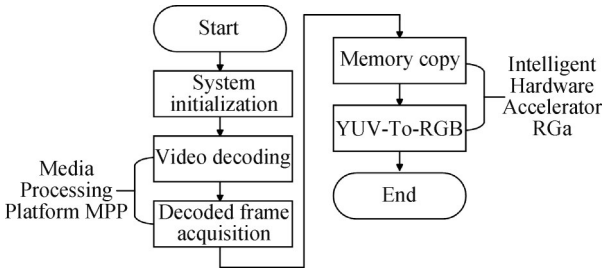


Fig.10 Multimedia processing platform decoding process

(1) Prior to data acquisition, the imaging system is initialized. This includes loading the MPP and RGA devices and initializing the video buffer. When the output format of the video decoding is YUV420P, the number of video cache blocks (BlkCnt) and block size (BlkSize) can be calculated using Eq.(10), where W is the video width and H is the video height.

$$\begin{cases} \text{BlkSize} = W^{*} H^{*}3/2 \\ \text{BlkCnt} = 10 \end{cases} \tag{10}$$

(2) In MPP, functions are called to create six video decoding channels. The video data are input into each channel, and the decode_get_frame function is invoked to initiate the decoding proces.

(3) The MppBufferGroup function in MPP is used to read the decoded video image data from the video buffer pool.

(4) The MPIIVE_DMA function of the RGA hardware accelerator copies the acquired video image data into user space.

(5) The function ops->go(dec_rga) in RGA converts each YUV420P-format video image to RGB888 format for stitching.

The five video images decoded in real time at the same moment are shown in Fig.11.

Fig.11 Real-time decoded display of video

3.5 Real-Time Visualization Ground Station

To better visualize the UAV video stream stitching results on the ground, this paper develops a Ground Control Station based on the VUE architecture. The client adopts VUE. js as the front-end framework and Spring Boot as the back-end framework. As shown in Fig.12, the system first includes a browser-based H. 264 stream decoding module, which invokes the decoding unit of the Nvidia GPU to perform browser-side decoding. Simultaneously, a multi-threaded decoding module is employed to retrieve the UAV flight status and homography matrix in real time.

Based on delay calculations, the average computation time for the homography matrix is approximately 30.4 ms, while video stream decoding requires about 15 ms. Therefore, when considering edge-side video decoding, the computation time of the homography matrix is only approximately 20 ms slower than that of the video stream. For 30 FPS video, this delay is less than one frame, thus it does not result in visible misalignment due to processing lag.

As shown in Fig.13, the display interface is divided into three sections: the left UAV perspective, the right UAV perspective, and the synthesized perspective.

Fig. 14 shows that the control interface is also capable of displaying the latitude, longitude, and positional information of each UAV, which aids in rational planning of flight trajectories.

4 Experiments and Results

This section validates the wide-field stitching system for UAVs from two perspectives. First, the performance of the edge computing module in real-time video decoding is evaluated. Subsequently, the proposed feature point detection and matching algorithm is compared with other existing algorithms to assess the achieved improvements.

4.1 Video Decoding Performance

Fig. 15 illustrates the method for calculating the RTSP video stream decoding delay. In this setup, a laptop is configured to display time information accurate to milliseconds, and remote desktop control of the edge

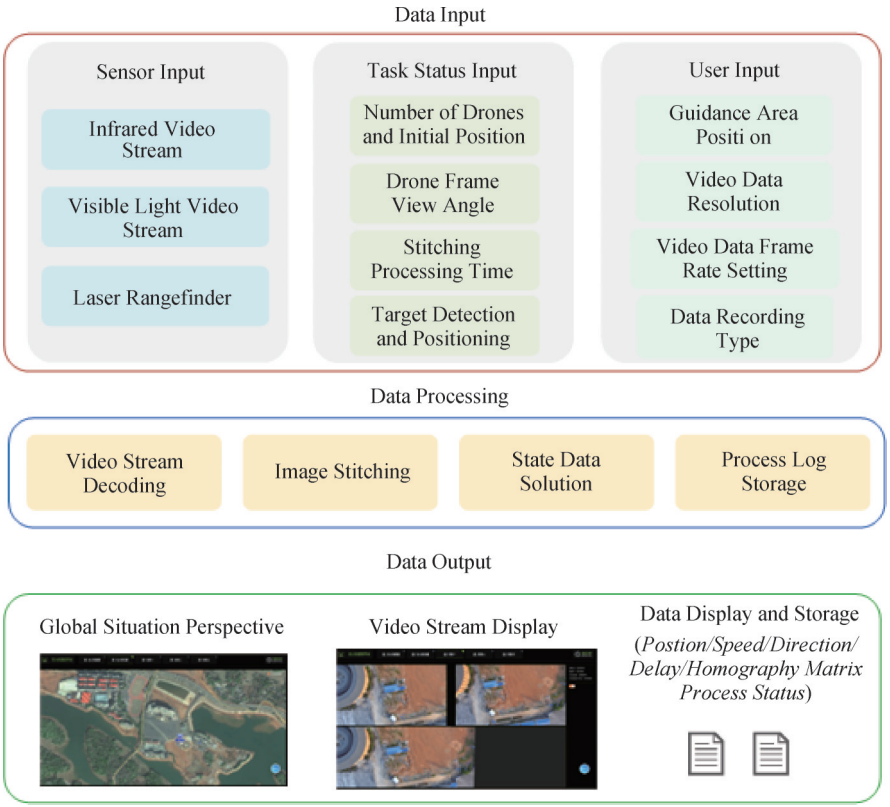


Fig.12 Ground control station software architecture

Fig.13 Large field of view splicing interface

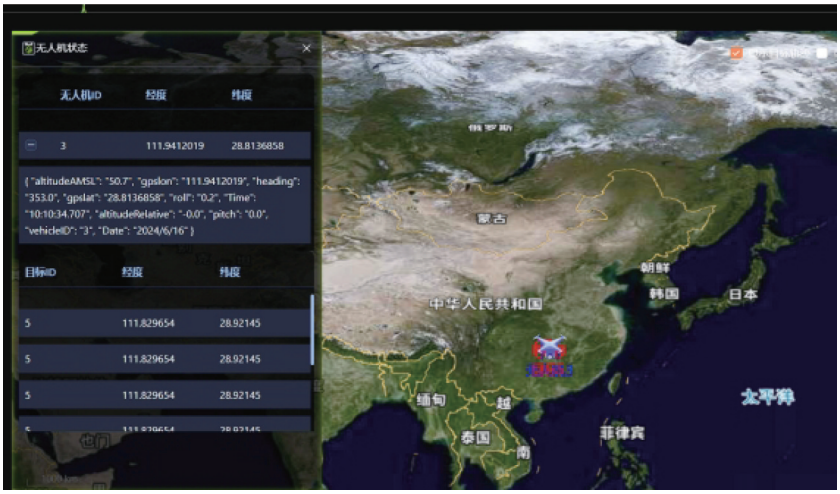


Fig.14 UAV status monitoring interface

computing platform is established via remote tools. The time display on the laptop screen is captured by a UAV node equipped with an optoelectronic pod. Meanwhile, the edge computing platform is controlled to display the captured data. The difference between the two-time values represents the decoding delay of the RTSP video stream.

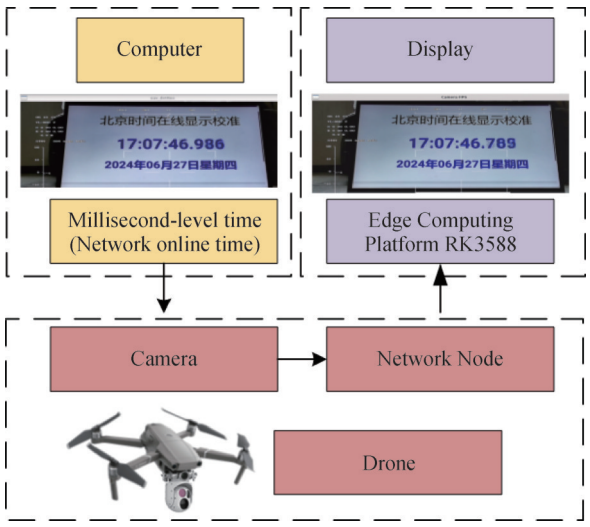


Fig.15 UAV communication delay statistics

The data stream is processed by both the camera and the RK3588 hardware decoding process on the PC. MPP hardware decoding efficiency is evaluated by analyzing the decoding time differences between the two schemes.

This video decode method demonstrates significant efficiency improvements compared to OpenCV-based software decoding. The comparison, shown in Table 1, indicates that the time required for hardware decoding is only 1.3% of that required for software decoding, highlighting the superior performance of hardware decoding. The RK3588 platform includes a hardware accelerator, which also provides strong advantages in YUV420P-to-RGB888 conversion and 1080P-to-720P image scaling tasks, with implementation times at 66.7% and 77.8%, respectively, of those using OpenCV.

Table 1 Video decode time comparison

Method	Time Cost
OpenCV(Soft Decode)	0.983s
FFmpeg(Soft Decode)	0.962s
MPP+FFmpeg(Hard Decode)	0.019s
MPP+ZLMedia(Hard Decode)	0.013s

4.2 Keypoints Detection and Matching

In this project, a large field-of-view stitching experiment was conducted. The flight route covered various terrain types, including water surface, woodland, grassland, and building areas. The flight altitude was 200 m, the field of view for a single UAV was 1032 m^2, and the dual-UAV configuration covered 1548 m^2, with a side overlap rate of 50%. We extracted 200 image pairs for quantitative analysis.

Table 2 Image transfer time comparison

Operation	Method	Time Cost
YUV420P to RGB888	OpenCV	0.012s
	RGA	0.008s
1080P to720P	OpenCV	0.009s
	RGA	0.007s

We used two objective indicators, i. e., Mean Matching Accuracy (MMA)[15] and image stitching time, for comparison. The larger the MMA, the higher the matching accuracy of the representative feature points.

The MMA represents the average percentage of correct feature correspondences per image pair. For a query image i, its keypoints are projected onto the reference image using the homography matrix. Matches with reprojection error below a threshold t (typically 1-10 pixels) are classified as correct. MMA is then computed as the average correct match rate across all pairs.

$$\mathrm{MMA}(t)=\frac{1}{N}\sum_{i=1}^{N}\left(\frac{\sum_{j=1}^{N^f}\delta\left(t-\left\|H_i\left(P_{A,i,j}^f\right)-P_{B,i,j}^f\right\|_2\right)}{N^f}\right) \tag{11}$$

Where N represents the number of image pairs; N^fdenotes the number of predicted matches; $\delta(\cdot)$ is the binary indicator function, which outputs 1 for non-negative values and 0 otherwise; t is the threshold for reprojection error, which is set as 5 in this paper; $H_i(\cdot)$represents the keypoints from the i query image warped to the reference image using the homography matrix; $\left(P_{A,i,j}^f, P_{B,i,j}^f\right)$ denotes the pixel coordinates of the j match in the i image pair.

As summarized in Table 3, SIFT+Improved FLANN (our method) achieves 79.2 accuracy with a processing time of 36.43 ms. We evaluate three conventional feature

Table 3 Quantitative comparisons with video stitching methods

Arithmetic	MMA	Average time(ms)
SIFT + BF	51.2	61.79
SIFT + FLANN	60.4	74.17
ORB + BF	36.4	55.60
ORB + FLANN	42.9	63.52
AKAZE + BF	59.5	83.18
AKAZE + FLANN	63.6	167.52
LightGlue	**82.6**	275.36
SIFT + Improved FLANN (our)	79.2	**36.43**

detectors: SIFT, ORB, and AKAZE[16], with LightGlue utilizing SIFT features. Our approach demonstrates a 25.36 ms speed improvement and 18.8 accuracy gain over the baseline SIFT+FLANN combination. Although LightGlue yields higher accuracy (82.6%), its computational latency of 275.36 ms exceeds real-time requirements.

The execution times of each component are presented in Table 4. It should be noted that these three components operate in parallel. Consequently, the output frequency of the homography matrices is constrained by the execution time of the Keypoint Detection and Matching module, which is 36.43 ms.

Table 4 End-to-end latency breakdown

Step	Average Time
Video Decode	13.02 ms
Keypoint Detection and Match	36.43 ms
Homography Matrix Compute	0.91 ms

The ground station stitches video streams using homography matrices transmitted from the drone, which inherently exhibit temporal latency. Although these matrices represent past drone states (Δt <20 ms), the maximum relative displacement during this interval is constrained to 0.2 m under typical operational conditions (200 m altitude, 5 m/s). This results in ⩽10-pixel projection errors that do not materially impact visual output quality.

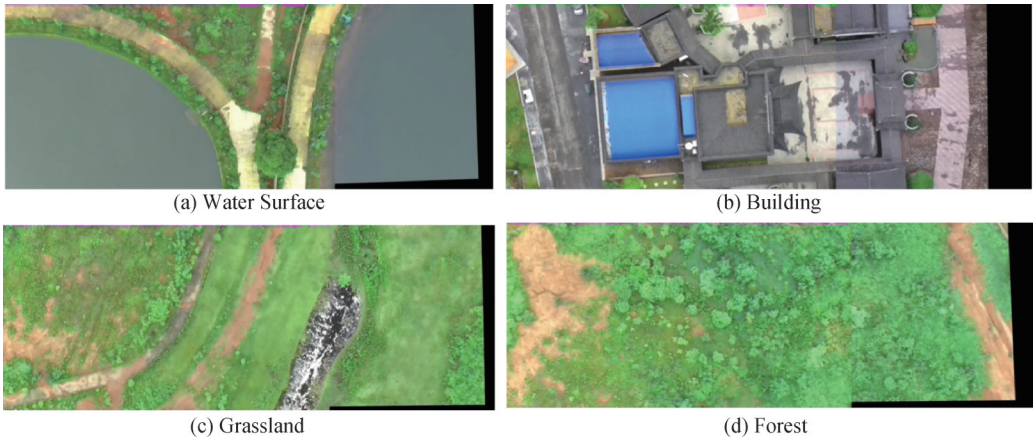
4.3 Discussion

The proposed framework leverages edge devices to compute the homography matrix, while the ground control station decodes the video stream and performs image transformation and stitching based on this matrix. This air-ground collaborative computing approach significantly enhances system real-time performance. However, certain system limitations and challenges under degenerate inputs remain.

First, the system lacks specific handling for GPS-denied environments or drift scenarios, primarily because the target operational domain assumes reliable satellite signals. To enhance robustness and scalability, a viable strategy is to automatically switch to vision-only stitching when the drone's status indicates GPS confidence falls below a predefined threshold. Second, rapid attitude changes pose a significant challenge to real-time image stitching systems. This limitation can be mitigated by incorporating constraints derived from IMU data, which will be addressed in future work. Furthermore, low-texture environments (e. g., water surfaces, deserts) present a major challenge for vision -only stitching solutions. The scarcity of feature points and their lack of distinctiveness in such scenarios necessitate increased reliance on GPS/IMU-based methods when the number of detected features falls below a specified threshold.

4.4 Visualization

In order to better demonstrate the splicing effect, we selected four types of splicing results underground (forest, grassland, water surface, building). From the image, it can be seen that there is basically no deformation in the spliced image, and the seams match well. The experimental results show that the algorithm proposed in this paper can achieve lightweight and high-quality video stitching.



(a) Water Surface (b) Building

(c) Grassland (d) Forest

Fig.16 Display of stitching images in different scenarios

6 Conclusion

In this paper, we design a multi-UAV large field-of-view splicing system based on the improved FLANN algorithm, which mainly completes the edge computing hardware construction, the splicing algorithm design based on multi-frame time series ROI area transfer, and the development of user-oriented display and control software. With normal network communication, the system is able to provide large field-of-view splicing images with a user delay of <40ms and maintain good splicing quality by conducting relevant experiments in

areas such as forests, water surfaces, and architectural complexes. Subsequently, it can support a series of applications such as multi-UAV cooperative ground positioning and multi-UAV object tracking.

Author Contribution:

Conceptualization: Hanxiang Qian; Data curation: Hanxiang Qian; Formal analysis: Hanxiang Qian; Funding acquisition: Xiaojun Guo; Investigation: Hanxiang Qian; Methodology: Hanxiang Qian; Project administration: Hanxiang Qian; Resources: Hanxiang Qian; Software: Hanxiang Qian; Supervision: Shaojing Su, Xiaojun Guo; Validation: Feiyang liu, Runze Guo; Visualization: Xiaoyong Sun; Roles/Writing - original draft: Hanxiang Qian; and Writing - review & editing: Hanxiang Qian.

Funding Information:

This research was funded by Graduate innovation program of National University of Defense Technology, grant number:XJZH2024016.

Data Availability:

The authors declare that the main data supporting the findings of this study are available within the paper and its Supplementary Information files.

Conflicts of Interest:

The authors declare no competing interests.

Dates:

Received 18 May 2025; Accepted 31 August 2025; Published online 30 September 2025

References

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision* , vol. 60, pp. 91-110, **2004**.

[2] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features, " in Computer Vision - ECCV 2006, vol. 3951, *A. Leonardis* , H. Bischof, and A. Pinz, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404-417. doi: 10.1007/11744023_32.

[3] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF, " in 2011 International Conference on Computer Vision, Barcelona, Spain: IEEE, Nov. 2011, pp. 2564-2571. doi: 10.1109/ICCV.2011.6126544.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks, " *Advances in neural information processing systems* , vol. 25, **2012**.

[5] B. Fan, H. Liu, H. Zeng, J. Zhang, X. Liu, and J. Han, "Deep unsupervised binary descriptor learning through locality consistency and self distinctiveness, " *IEEE Transactions on Multimedia* , vol. 23, pp. 2770-2781, **2020**.

[6] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks, " in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, **2020**, pp. 4938-4947.

[7] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys, "LightGlue: Local Feature Matching at Light Speed," in 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France: IEEE, Oct. 2023, pp. 17581-17592. doi: 10.1109/ICCV51070.2023.01616.

[8] Y. Huang, J. Li, and N. Fan, "Image mosaicing for UAV application," *in 2008 International Symposium on Knowledge Acquisition and Modeling* , IEEE, **2008**, pp. 663-667.

[9] L. Guo, H. Zhu, Y. Liu, X. Sun, and X. Teng, "Aerial Image Stitching Based on Fusion of Geographic Coordinates and Image Features, " in 2022 IEEE International Conference on Unmanned Systems (ICUS), Guangzhou, China: IEEE, Oct.**2022**, pp. 1095-1100. doi: 10.1109/ICUS55513.2022.9986913.

[10] W. Qian, Y. Yang, Y. Xiao, K. Xu, S. Xie, and Y. Xie, "Enhanced Incremental Image Stitching for Low-Altitude UAV Imagery With Depth Estimation," *IEEE Geoscience and Remote Sensing Letters* , vol. 21, pp. 1-5, 2024, doi: 10.1109/LGRS.2024.3483254.

[11] M. Ren, J. Li, L. Song, H. Li, and T. Xu, "MLP-Based Efficient Stitching Method for UAV Images, " *IEEE Geoscience and Remote Sensing Letters* , vol. 19, pp. 1-5, **2022**, doi: 10.1109/LGRS.2022.3141890.

[12] J. Wang, P. Du, S. Yang, Z. Zhang, and J. Ning, "A Spatial Arrangement Preservation-Based Stitching Method via Geographic Coordinates of UAV for Farmland Remote Sensing Image, " *IEEE Transactions on Geoscience and Remote Sensing* , vol. 62, pp. 1-13, **2024**, doi: 10.1109/TGRS.2024.3374075.

[13] R. L iet al., "A Real-Time Incremental Video Mosaic Framework for UAV Remote Sensing," *Remote Sensing* , vol. 15, no. 8, p. 2127, Apr. **2023**, doi: 10.3390/rs15082127.

[14] Y. Mo, X. Kang, P. Duan, and S. Li, "A Robust UAV Hyperspectral Image Stitching Method Based on Deep Feature Matching, " *IEEE Transactions on Geoscience and Remote Sensing* , vol. 60, pp. 1-14, **2022**, doi: 10.1109/TGRS.2021.3123980.

[15] M. Dusmanuet al., *"D2-Net: A Trainable CNN for Joint Description and Detection of Local Features," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* , Long Beach, CA, USA: IEEE, June 2019, pp. 8084-8093. doi: 10.1109/cvpr.**2019**.00828.

[16] P. Alcantarilla, J. Nuevo, and A. Bartoli, "Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces," *in Procedings of the British Machine Vision Conference 2013* , Bristol: British Machine Vision Association, **2013**, p. 13.1-13.11. doi: 10.5244/c.27.13.