IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

NATURAL-INSPIRED DRONE SWARM PROCESSING FOV FOR EFFICIENT MULTI-VIEW MONITORING AND OBJECT DETECTION

[1]Osama ElSayed, [2]Sherine Youssef, [3]Ossama Ismail

[1]Student, [2]Professor, [3]Professor
[1]Department of Computer Engineering, College of Engineering and Technology,
[1]Arab Academy for Science, Technology and Maritime Transport, Alexandria, Egypt

ABSTRACT: The rising interest in aerial swarm drones, a type of multi-agent robotic system, has sparked potential applications. This paper presents the Intelligent Model Using Swarm Drones for Surveillance Applications. It proposes the use of commercially available DJI Tello drones for unmanned aerial vehicle (UAV) swarm missions as a cost-effective alternative to custom-made drones. To enable simultaneous control of multiple drones and introduce real-time video stitching, a camera path estimation and homography refinement method will be used. The paper aims to contribute to the surveillance swarm drone industry by decreasing data transfer, storage, management, and monitoring, using efficient multi-view panoramic imaging and extra compression of surveillance swarm drones' cameras' footage through stitching. To enhance video stitching stability, a unified framework for joint video stitching and stabilization is proposed. This approach involves creating an optimal virtual 2D camera path from the original paths, space-temporal optimization that considers inter and intra motions, grid-based tracking for improved robustness, and mesh-based motion models to handle scene parallax in videos captured simultaneously by multiple moving cameras that may exhibit shaking and artifacts when directly applying image stitching methods.

***IndexTerms*: Unmanned Aerial Vehicles, Drones, Multi-UAV, Object detection, Video stitching, video stabilization.**

1. INTRODUCTION

Unmanned aerial vehicles (UAVs), commonly referred to as drones, are garnering greater attention in both academia and industry. This is due to advances in sensing and computing capabilities, as well as reductions in form factors and costs. Moreover, the development of onboard intelligence and autonomous capabilities has expanded the range of applications for UAV systems [1]. In recent years, the combination of UAV and swarm intelligence technologies has enabled swarms of small-scale and low-cost UAVs to execute complex combat tasks.

Micro aerial vehicles (MAVs), also known as small and light-weight UAVs, have significantly contributed to the development of robotics and unmanned systems. These small robots are widely used in various applications, such as surveillance, search and rescue, and mapping.

One of the significant advantages of MAVs is their portability, which allows for easy transportation and deployment in different locations. Their smaller size and weight also make them more agile, enabling them to maneuver through narrow spaces and reduce the risk of causing damage to their surroundings. However, the small size of MAVs also limits their capabilities. For instance, they have shorter flight times, less on-board sensing and compute power, and lower payloads compared to larger drones [2].

This limitation makes it challenging for MAVs to carry out complex tasks on their own. To address these challenges, researchers have developed the concept of aerial swarms, where multiple MAVs collaborate to overcome the limitations of individual robots. A swarm of MAVs can work together to perform complex tasks, such as mapping large areas, inspecting infrastructure, and even delivering goods [1] [2].

The use of swarm technology in MAVs has opened up new avenues for robotics and unmanned systems. It has also led to advancements in swarm intelligence and collective behavior, which can be applied to other fields, such as transportation and logistics. With the continued development of MAVs and swarm technology, we can expect to see more innovative and exciting applications in the future [1] [2].

A swarm or fleet of Unmanned Aerial Vehicles (UAVs) refers to a group of aerial robots or drones that collaborate to achieve a common objective. Each drone in a swarm is powered by a specific number of rotors and has the capability to vertically hover, take off, and land (VTOL). The flight of these drones can either be manually controlled through remote operations or autonomously managed by employing processors that are installed on the drones [3].

Recently, a swarm consisting of N unmanned aerial vehicles (UAVs) successfully completed a large-scale military-focused evaluation. However, the missions generated a huge amount of data from the swarm's surveillance cameras, resulting in numerous redundant pixels in the footage. Consequently, there is an urgent need to develop more reliable systems with improved compression and panoramic monitoring solutions to aid in object detection. To address these issues, a new model called "An Efficient Multi-View Panoramic Imaging and Extra Compression of Surveillance Cameras' Footage using Stitching" (ECS) has been introduced. The ECS model utilizes feature extraction stitching techniques and geometric relational matrix calculations to create a real-time, effective stitching and reconstruction of panoramas that minimizes the transfer and storage of redundant pixels in the surveillance cameras' footage [4].

The organization of drone swarms can be divided into two main categories: single layered and multi-layered swarms. In a single layered swarm, each drone functions as its own leader, making individual decisions to achieve the overall mission objective. On the other hand, in a multi-layered swarm, dedicated leader drones exist at each layer, reporting to the higher level leader drones. The highest layer in this hierarchy is typically a ground-based server station. By utilizing a multi-layered swarm, the system can more efficiently manage and coordinate the actions of a large number of drones, ultimately leading to increased efficiency and improved mission success rates [3].

With a variety of sensors and the ability to operate autonomously, drones are now seamlessly connected to the internet, providing valuable perspectives from the air that were previously inaccessible. The data gathering capabilities of drones are immense, with the ability to collect up to half a terabyte of data per hour, per drone. This data has enormous potential for big data analysis and could have significant applications across a range of industries. By utilizing this technology to collect and analyze data, we may be able to tackle some of the greatest problems facing humanity [5].

In a swarm configuration, each drone can be assigned specific data collection and processing tasks and equipped with enough computing power to execute these tasks in real-time. However, for more computationally intensive tasks, the central processing may occur on a more powerful server or base station, or even in the cloud. This allows for more efficient data processing and analysis and can enable more complex tasks to be carried out by the swarm as a whole [3].

Fig. 1.a and Fig. 1.b, show a group of drones communicating with the cloud to reach the base station and to upload the collected data, with the two options of collecting the data from an indoor environment and from an outdoor environment. swarm of bees leaves their hive to form a new colony elsewhere, a swarm of mosquitoes may gather in a certain area, and a swarm of tourists might be seen at a popular destination.

Swarm robotics enables collective transport of objects that would be too heavy or large for individual robots to move alone. This is achieved through the cooperation of multiple robots, which can be either homogenous or heterogeneous in nature, forming a swarm of robots. Each robot is equipped with on-board processing, communication, and sensing capabilities that enable them to interact with one another and autonomously respond to the environment. This collective behavior allows the swarm to perform complex tasks that would be difficult or impossible for a single robot to accomplish alone [6].

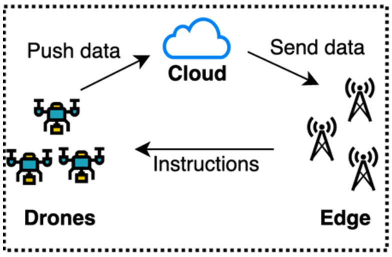


Fig. 1.a. The communication and retrieving of the data from the UAVs.

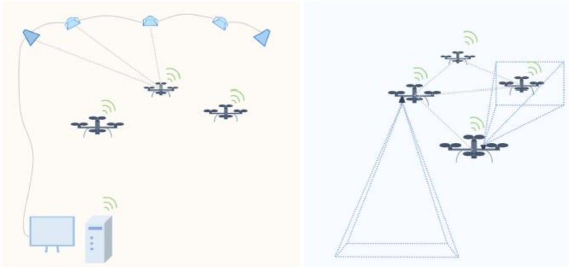


Fig. 1.b. Indoor localization solution (on the left): Motion capture used to track multiple drones' positions. Outdoor localization solution (on the right), both communicates to a base station.

The paper is organized as follows:
Section 2 will highlight the Overview of Aerial Swarm Applications, while section 3 will demonstrate some related work. The proposed model will be introduced in section 4 with description if each phase in the model. Finally, experimental results, discussions and conclusions will be illustrated in section 5 and 6 respectively.

2. OVERVIEW OF AERIAL SWARM APPLICATIONS

2.1. Security and Surveillance

With the increasing adoption of drones in industrial and commercial security applications, the world is on the brink of a significant transformation in the way security is monitored and managed. The recent advancements in drone and AI technologies have made it possible for off-the-shelf drones to be equipped with advanced onboard monitoring sensors like RGB and thermal cameras, which allow for efficient and effective aerial surveillance. Unlike traditional security methods such as fixed CCTV cameras and human patrols, surveillance drones offer several advantages, including the ability to cover larger areas, reduce costs associated with human patrols, and minimize risks to human security personnel [7] [8].

In fact, individual drones have already been used as electronic eyes in both manual and automated operations, providing real-time intelligence on critical security issues. The use of drone swarms in security applications further enhances these advantages by enabling the simultaneous monitoring of multiple areas, resulting in significant time savings and increased operational efficiency. As drone and AI technologies continue to advance, it is expected that the deployment of drones for industrial and commercial security will become increasingly widespread, revolutionizing the way security is managed and maintained around the world [7] [8].

UAVs are not just limited to collecting intelligence information about objects that they detect and identify on their own, they are also capable of self-coordinating and collaborating with each other as a fleet to accomplish surveillance tasks. This includes optimal area coverage considering vehicle and sensor constraints, as well as adapting to changes in the environment or mission objectives. With the help of advanced onboard intelligence and communication capabilities, the UAV fleet can adjust its behavior and strategies in real-time, making it a powerful tool for surveillance operations. Additionally, the use of UAV swarms can further enhance their capabilities by enabling more efficient and effective surveillance over larger areas in a shorter amount of time. [7] [8].

Detecting and tracking objects can be achieved by employing individual sensor technologies such as radio frequency (RF) detection and spoofing, RADAR, optical sensors including RGB and thermal cameras, and acoustic sensors. Furthermore, combining the aforementioned technologies can result in increased accuracy of detection and tracking. Although swarm-based anti-UAV systems have not yet been commercially developed, research efforts are being conducted in this area. The use of UAV swarms can also be advantageous in surveillance applications where large areas need to be covered and searched in shorter periods compared to single UAVs [9] [10].

2.2. Collaborative Transportation

Payload transportation by a swarm of UAVs has gained interest for its potential to overcome the limitations of single UAVs in terms of payload capacity. However, safety and regulation constraints make it difficult to deploy large and heavy UAVs for transporting heavier payloads. In order to transport larger payloads, researchers have demonstrated the use of a group of small UAVs working together. Decentralized control laws were used for each quadrotor, where each quadrotor knows its fixed relative position and orientation with respect to the body and payload goal. The required state estimation of quadrotor positions and velocity was done in a centralized way using an overhead motion capture system. In addition, methods to estimate payload deformation and stabilize it in 3D using a centralized LQR controller were presented for transporting a certain class of flexible structures that were still rigidly attached to the UAVs [11]. The quadrotor positions and velocities were accurately provided by a motion capture system.

The system uses a decentralized approach where each quadrotor estimates its own pose using visual-inertial odometry and communicates with its neighbors to maintain a desired formation. The system is able to transport the rigid rod payload while avoiding obstacles and maintaining formation, demonstrating the feasibility of using onboard sensors for outdoor multi-UAV payload transportation [12]. This is a promising development for the field as it eliminates the need for expensive and cumbersome motion capture systems and opens up possibilities for real-world applications in various outdoor environments.

2.3. Environmental Monitoring

The system consisted of a swarm of UAVs equipped with different sensors such as optical and thermal cameras and GPS receivers. The UAVs autonomously flew over the flood area and captured high-resolution images and videos, which were processed and analyzed in real-time by a ground station. The system provided accurate flood maps and tracked the movement of the floodwater, allowing for timely and effective responses from disaster management authorities [13]. Additionally, the use of a swarm of UAVs enabled the coverage of a larger area in a shorter amount of time, which is particularly important in the case of rapidly evolving flood events.

3. Related works

Many well-known applications, such as Adobe Photoshop, AutoStitch, PTGui, and Image Composite Editor (ICE) effectively stitch multiple overlapping images to generate a wide-angle view. These software packages typically use a combination of feature detection, matching, and alignment algorithms to automatically align and blend multiple images into a seamless panorama [14].

Meanwhile, various 360-degree polydioptric cameras based on panoramic stitching have been released, e.g., Nokia OZO', GoPro Odyssey', Facebook Surround 360', and Samsung Gear360'. These cameras capture a complete 360-degree view of the scene, which can be used for virtual reality (VR) and immersive media applications. The images captured by these cameras are typically stitched together using specialized software to create a seamless 360-degree panorama [14].

Google Street View is a well-known application that uses CMOS cameras to capture video at lower sampling rates, resulting in less continuous videos than regular ones, and it utilizes specially designed head cameras with fixed relative geometries that are synchronized for capturing. Video stitching is a crucial aspect of this application [15].

The stitching process differs on the cameras motion, there are two types of camera positioning:

3.1. Static Camera:

Video stitching is based on the same concept of image stitching specially when it is about static cameras the problem appears when there are moving objects as ghosting of the objects and persons are seen clearly through frames. Video stitching works on the selected frames of the inputs and start generating what is called a stitching template, as long as there are no moving objects and all the objects are static between the overlapping regions and frames taken from the inputs. The subsequence's frames are stitched together based on the stitching template [14]. The stitching template keeps updating when moving objects are included in the scene or selected frames coming from the inputs. Many methods and approaches were followed to address the idea of moving objects. The most famous used approaches to address are the following [16] [17].

One approach is to embed the moving content that were detected into the final stitched images with a standard stitching algorithm and then do an object detection phase afterwards to ensure the embedding phase. The spatially neighboring videos captured by a horizontally swiveling video camera from a stationary location provide reliable information to ensure accurate alignment between the frames. [17] [18].

A surveillance application video stitching model was introduced using coarse to fine process, by creating different layers from the chosen frames that are obtained from the input videos, then the process starts by stitching the background layers using a standard stitching pipeline algorithm as long as there are no moving objects, all the objects are static, across the overlapping regions of the stitched frames [16].

On the same page, many layers that contains variant objects are generated from the clustering of the matched feature pairs where each one of the generated layers have a collection of matched feature pairs consistent with the identical homography, while the videos are totally pre-aligned [16].

To prevent the presence of moving objects from causing missing data, artifacts, or ghosting, the best fit seam is adjusted by analyzing the gradient variations in the overlapping regions [19].

To address both the spatial and temporal domains, a spatial-temporal mesh optimization framework can be employed, which involves formulating an objective function that captures the matching costs in both domains. This is achieved by taking the initially aligned frames of the input videos, based on their estimated spatial and temporal transformations, and using the objective function to improve the geometric alignment between the frames. To obtain the optimal seam and minimize the salient effect, higher weights are assigned to salient regions such as spatial and temporal edges. [17].

3.2. Moving Camera

The use of unmanned aerial vehicles (UAVs) and smartphones often results in moving camera videos that present stitching problems, such as jitters and shakiness. While scan methods can help reduce these issues, they cannot be directly used to stabilize stitched videos. To address the problem of shaky footage, many studies combine image stitching and stabilization techniques [19]. However, capturing video with a moving camera can still be a significant challenge, as camera movement can cause objects in stitched videos to appear distorted and blurred.

Employing image stitching algorithms directly to shaky video frames through a frame-by-frame approach would have two drawbacks. Firstly, the severe perspective distortions in the acquired videos due to shakiness. Secondly, the frame-by-frame stitching process does not take into account the temporal smoothness of the videos, which can result in visible jerks. [16].

Current solutions to this issue often rely on additional technology and require limiting camera movement. Some hardware solutions have been introduced to address the challenges faced by traditional video stitching methods, such as the FC-110 full view camera, which combines 10 CCD (charged-coupled device) cameras in a glass container with optical centering, and the GoPano method, which uses a spherical lens mounted in front of a mobile camera to expand the field of view. However, these methods can be expensive and difficult to implement, as they typically involve using an unstructured camera array or stitching videos under static cameras [15] [18].

To achieve video stitching and stabilization, a two-fold transformation approach is employed. The first step involves an inter-transformation between the two cameras to establish spatial alignment, while the second step is an intra-transformation conducted within each video to maintain temporal smoothness. Additionally, a smooth virtual camera path is synthesized in two steps. Firstly, a mesh-based warping method is used to achieve the required video alignment for stitching [17]. Then, a bundled-paths method is used as a baseline for video stabilization, effectively stitching scenes with a specified degree of parallax while ensuring stable camera movement within certain degrees of freedom. This method initially imposes rough synchronization, with any misalignments arising due to significant depth changes or strong motion blur.

A proposed model aims to find a solution for an objective function that is derived from two terms: a stabilization term and a stitching term. An iterative optimization scheme is then executed to obtain the final output. This model was then updated and extended to get better results. The updated method proposed three steps to get the output video. First, an identification method is used to identify and stitch the input videos' backgrounds. Then, to reduce the mismatches and errors generated from the first step a false match elimination scheme is used. Finally, a fitness function, or what is called a scoring strategy, which is used to evaluate and estimate the stabilization and stitching quality of the output video.

An approach introduced for hand-held cameras involves combining a dense 3D reconstruction and camera pose estimation technique to stitch input videos from cameras. The CoSLAM system requires two steps: recovering 3D camera motions and sparse scene points using overlapping scenes to reconstruct 3D scenes, and constructing a smooth virtual camera path that remains in the middle of all the original paths. The Line-Preserving Video Warp (LPVW) method is then used, which utilizes a mesh-based warping optimization to synthesize the stitched video while simultaneously stabilizing it. However, this method can fail to provide correct correspondences in homogeneous regions and may also fail when overlapping regions have significant depth discontinuity. [20].

Video stitching algorithms typically involve several steps, including the construction of a stitching template by stitching selected frames of original videos with image stitching algorithms. This template is then used to stitch subsequent frames to generate a single wide-angle video. To address potential blurring and ghosting in the stitched video, foreground detection is often employed. This technique aims to identify and isolate moving objects in the scene, allowing for more accurate stitching and reducing the impact of ghosting and blurring caused by object motion during the video capture process [14].

Field of view (FOV) limitations in images and videos is a huge drawback in the field of computer vision and graphics, that's why Image Stitching is widely used to overcome the limitations of FOV. Stitching is used nowadays in the daily lives when constructing a panorama using smartphones, creating wide angle videos for security and surveillance and it is a widely used technique in the field of autonomous vehicles. It involves stitching together multiple overlapping images to create a single, wide-angle view. This technique is essential for improving the accuracy and reliability of perception and navigation systems in autonomous vehicles, and is thus considered a critical technology for their development [16]. Fig. 2, shows the idea of extending the FOV using stitching from two cameras. It is also used in the recent technology cameras to get a 360-degree view from a sequence of images, it is also used in the virtual reality field. To construct a panorama image, several computer vision and image processing techniques are employed, such as key point detection and local invariant descriptors, key point matching, Random Sample Consensus (RANSAC), and perspective warping. Although multi-view image stitching has piqued the interest in the last few decades and it then has attracted the computer vision community, Compared to traditional image stitching, multi-view video stitching, also known as panoramic video stitching, has not received as much attention [17].

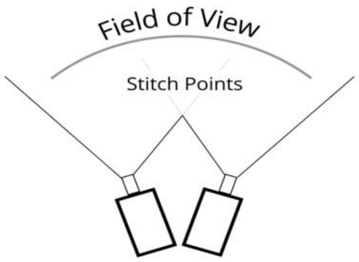


Fig. 2. Increasing the field of view using two cameras' views by image stitching the two views

Multi-view video stitching aims to create a panoramic video by combining multiple overlapping video clips captured by cameras with different relative displacements. Although multi-view videos can be captured from fixed or moving cameras, it is important to maintain a constant relative geometry during the stitching process. A multi-view video stitching program takes multiple overlapping videos captured from different angles as input and generates a panoramic video with a wider field of view by stitching the individual videos together. The resulting video is known as a multi-view video [21].

Multi-view video stitching presents unique challenges compared to multi-view image stitching. While many of the basic challenges such as frame alignment and color correction still apply, multi-view video stitching also involves problems related to its temporal properties, such as video synchronization and video stabilization. These challenges require innovative computer vision solutions that are currently being developed. In some cases, sub-problems such as frame alignment can be addressed more robustly by using multiple frame correspondences from the temporal sequence [21].

Multi-view video stitching can pose additional challenges due to the fact that the input videos may have been captured by mobile devices, which may introduce issues such as jitter. This makes it necessary to address video processing problems in addition to the spatial stitching problems encountered in multi-view image stitching. The combination of these two types of problems can make the overall task even more challenging [21].

Effective handling of camera motions is crucial in video stitching. However, spatial, and temporal artifacts are commonly observed in the results. In contrast, video stabilization techniques can eliminate shaky and jittery movements [21].

Video stitching poses many major problems in literature, including [14] [19] [21]: Video stabilization, Video synchronization, Efficient large-size multi-view video alignment and panoramic video stitching, Color correction, Blurred frame detection and repair.

Mesh networking and redundant data storage makes the overall system very fault tolerant and hard to bring down. Swarms are also highly scalable and self-organizing, so they can execute commands like a multi core processor. Swarms make sense when you have large areas to explore and monitor, expect the environment to be harsh and lose some drones due to accidents, do not expect robots to last more than one mission and you expect high demand. Robotic systems are suitable for hazardous environments where human safety is a concern, such as in nuclear or chemical industries. These systems do not require the same environmental conditions as humans, such as lighting, air conditioning, or noise protection. Moreover, robots have advanced sensors and actuators that can surpass human capabilities in certain aspects.

4. Proposed NiDroneSwarm FOV-MOD Model

In this section, the proposed Natural-Inspired Drone Swarm Processing FOV for Efficient Multi-view Monitoring and Object Detection will be introduced. (NiDroneSwarm FOV-MOD model). The target is to simulate the cooperative process of N drones (with the option to control more). Three experimental scenarios will be carried out: (1) Static Object, Static Cameras; (2) Moving Object, Static Cameras; and (3) Static Object, Moving Cameras. Experiments will be carried out with Two DJI Ryze Tello drones applied on distinct scenarios, Fig. 3, shows the drones used in the model proposed.

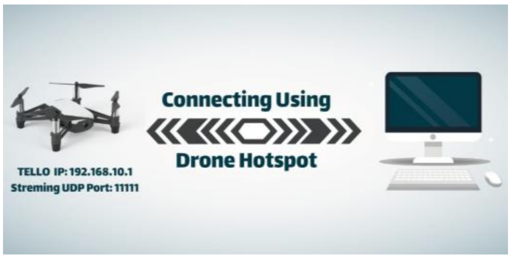
Fig. 3. The drones used are the standard edition of Ryze Tello

Fig. 4 shows the proposed model includes 4 main phases: Receiving of live streams among Drones, video frame processing, stitching phase and panoramic construction phase, and object detection phase.

4.1. Receiving of live streams phase

The Tello drone is a popular and affordable option equipped with a high-definition camera. In this paper, we describe the process of receiving live video streams from multiple Tello drones using a base station. The first step is to establish a communication channel between each Tello drone and the base station. This is achieved by connecting the drones to the base station through a wireless network. Once the connection is established, the base station can request the video stream from each drone. The video stream is sent over a wireless connection as an encoded video stream. The encoded video stream must then be decoded by the base station to obtain the individual frames. This process is essential to display the live video stream from each drone. Finally, the base station can display the video streams from each drone in separate windows.

The Tello quadcopter receives commands via the Wi-Fi hotspot it emits. To control many Tello drones in a swarm, a connection to each drone's Wi-Fi hotspot is made to be able to transmit commands to each drone individually. To do so a Wi-Fi interface for each drone is used to operate it [22]. Fig. 5, shows how each drone is connected to the base station via the drone's hotspot.



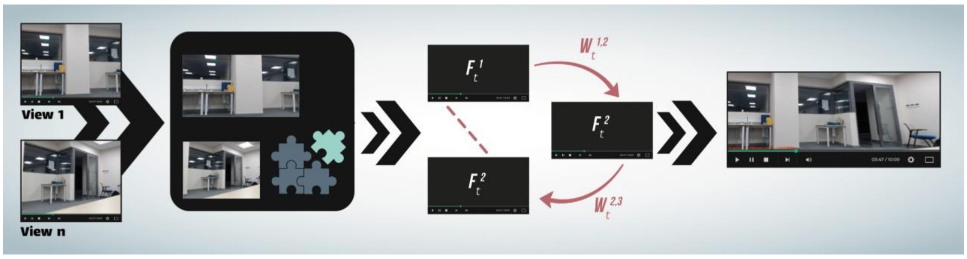


Fig. 4. The flow of the framework from getting the video frames till arriving at the final output.

The methodology followed on this framework is to first start controlling the drones and receive the live streams from the multiple quadcopters, the N drones must follow a constant move or a certain degree of freedom movements, as each two adjacent drones must have overlapping areas so that the stitching process can be done. Then the starting point for stitching is initialized, as the videos must be synchronized, and since live streams are the case here, it's only the matter of starting the stitching process while the drones are in position. It's allowed to have fractions of one second in both static and dynamic cases but it's not preferable.

The two drones communicate with the home station (Ground station for now), as the master node or swarm leader, which will be an equipped quadcopter that will be able to process the data and feed coming to it. Each drone transmits its video live feed and both videos are used to create the panorama wide FOV which will be used to live detect and track objects.

All of the Tello drones have the same IP and UDP port for commanding and receiving live streams, so changing the IP addresses was a huge problem, that was solved using port rerouting method is introduced so that the receiving of multiple feed from multiple drones is available on the same device, also as all the drones are connected using the drone's hotspot which means we have to use a network adaptor for each drone. For achieving both controlling a swarm and retrieving the video feed from the drones, each drone is connected to a raspberry pi and a WIFI adapter to be able to connect to the drone and also have a connection to the access point so that the videos can be retrieved and the commands for the drones can be sent, see Fig. 6.

The whole system is managed using a raspberry PI 4 8G which receives the videos and starts the stitching process, the redirecting of the video streams was done as follows:
Raspberry Pi IP Address: 192.168.1.120
Port Where Video Feed is Received: 11111
Port Where Video Feed is Changed to 11117
Code for implementing the redirecting:

```
sudo iptables -t nat -A PREROUTING -s 192.168.1.120 -p udp --dport 11111 -j REDIRECT -- to-port 11117
```

4.2. Video frame processing phase

The pre-processing stage in real-time video stitching is a critical step that prepares the video streams for stitching. This stage starts after the video streams have been captured and must be performed before the actual video stitching process begins. The objective of the pre-processing stage is to ensure that all video streams have consistent spatial and temporal resolutions to allow for seamless and accurate stitching.

Spatial resolution refers to the size of the video frames and can be represented mathematically as the number of pixels in each frame (W x H), (see Fig. 7). Temporal resolution, on the other hand, refers to the number of frames per second and can be represented mathematically as the frame rate (FPS), (see Fig. 8). When different drones or cameras are used to capture the video streams, they may return different resolutions. To unify these resolutions, the smallest spatial resolution, and the slowest frame rate of all the video streams must be chosen as the standard.

To achieve this, the following steps can be taken:

4.2.1. Spatial resolution:

Let (W1 x H1), (W2 x H2), ..., (Wn x Hn) be the spatial resolutions of the n video streams. The smallest resolution among these can be represented mathematically as (Wmin x Hmin) where Wmin = min (W1, W2, ..., Wn) and Hmin = min (H1, H2, ..., Hn). All other video streams must then be resized to match this standard resolution by using appropriate resizing algorithms such as bilinear interpolation or bicubic interpolation.

For moving camera scenarios Bilinear Interpolation is used, as it uses linear interpolation to estimate the value of a new pixel based on the values of its surrounding pixels. The basic idea behind Bilinear Interpolation is to use the four nearest pixels to a new pixel to estimate its value. The equation for Bilinear Interpolation can be expressed as:

$f(x, y) = (1-\alpha) \cdot (1-\beta) \cdot f(x_1,y_1) + \alpha \cdot (1-\beta) \cdot f(x_2,y_1) + (1-\alpha) \cdot \beta \cdot f(x_1,y_2) + \alpha \cdot \beta \cdot f(x_2,y_2)$

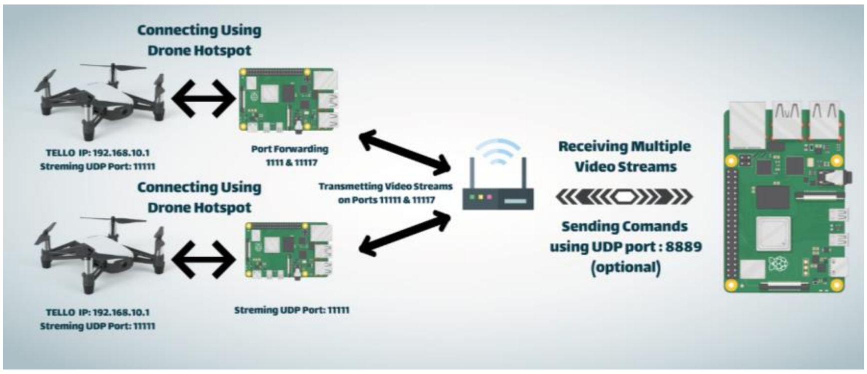

Connecting Using Drone Hotspot
TELLO IP: 192.168.10.1 Streming UDP Port: 11111
Port Forwarding 1111 & 11117
Connecting Using Drone Hotspot
Transmetting Video Streams on Ports 11111 & 11117
TELLO IP: 192.168.10.1 Streming UDP Port: 11111
Streming UDP Port: 11111
Receiving Multiple Video Streams
Sending Comands using UDP port : 8889 (optional)

Fig. 6. Port forwarding to receive multiple video streams at same time.

Where $f(x,y)$ is the estimated value of the new pixel, $f(x_1,y_1)$, $f(x_2,y_1)$, $f(x_1,y_2)$, and $f(x_2,y_2)$ are the values of the four nearest pixels, and α and β are interpolation coefficients determined by the fractional distances between the new pixel and the nearest pixels.

For static camera scenarios Bilinear Interpolation is used, as it uses a cubic polynomial to estimate the value of a new pixel based on the values of the surrounding pixels. Unlike Bilinear Interpolation, Bicubic Interpolation uses a 16-pixel neighborhood to estimate the value of a new pixel, which provides a more accurate representation of the image. The equation for Bicubic Interpolation can be expressed as:

$f (x, y) = \sum_{i=-1} ^ {2} \sum_{j=-1} ^{2} a_{i,j} \cdot f(x_0+i,y_0+j)$

Where $f (x, y) $ is the estimated value of the new pixel, $f(x_0+i,y_0+j)$ are the values of the 16 surrounding pixels, and $a_{i,j}$ are interpolation coefficients determined by the fractional distances between the new pixel and the surrounding pixels. The coefficients $a_{i,j}$ can be calculated using a set of pre-determined cubic polynomials.

In summary, Bilinear Interpolation uses linear interpolation to estimate the value of a new pixel based on the values of its surrounding pixels, while Bicubic Interpolation uses a cubic polynomial to estimate the value of a new pixel based on the values of a larger number of surrounding pixels. Bicubic Interpolation provides higher quality results compared to Bilinear Interpolation but is also more computationally intensive.

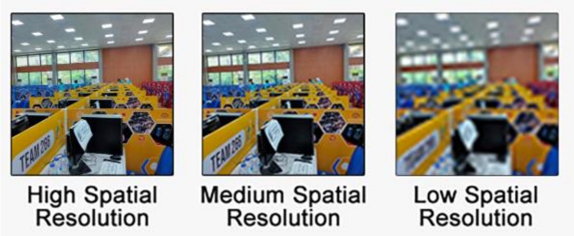


Fig. 7. Spatial resolution examples.

4.2.2.Temporal resolution:

Let FPS1, FPS2, ..., FPSn be the frame rates of the n video streams. The slowest frame rate among these can be represented mathematically as FPSmin where FPSmin = min (FPS1, FPS2, ..., FPSn). All other video streams must then be resampled to match this standard frame rate by using appropriate resampling algorithms such as linear interpolation or spline interpolation.

For moving camera scenarios Linear Interpolation is used, as it uses a straight line to estimate the value of a new pixel based on the values of its two nearest pixels. The equation for Linear Interpolation can be expressed as:

$f(x,y) = (1-\alpha) \cdot f(x_1,y) + \alpha \cdot f(x_2,y)$

Where $f(x,y)$ is the estimated value of the new pixel, $f(x_1,y)$ and $f(x_2,y)$ are the values of the two nearest pixels, and α is an interpolation coefficient determined by the fractional distance between the new pixel and the nearest pixels.

For static camera scenarios Spline Interpolation is used, as it uses a smooth curve to estimate the value of a new pixel based on the values of its surrounding pixels. The basic idea behind Spline Interpolation is to fit a smooth curve through the surrounding pixels and use this curve to estimate the value of the new pixel. The equation for Spline Interpolation can be expressed as:

$f(x,y) = \sum_{i=0}^{n} a_i \cdot \varphi_i(x,y)$

Where $f(x,y)$ is the estimated value of the new pixel, a_i are interpolation coefficients determined by the values of the surrounding pixels, and $\varphi_i(x,y)$ are a set of pre-determined smooth functions.

In summary, Linear Interpolation uses a straight line to estimate the value of a new pixel based on the values of its two nearest pixels, while Spline Interpolation uses a smooth curve to estimate the value of a new pixel based on the values of its surrounding pixels. Spline Interpolation provides higher quality results compared to Linear Interpolation but is also more computationally intensive.

The pre-processing stage of real-time video stitching involves various mathematical techniques that are utilized to harmonize the spatial and temporal resolutions of multiple video streams. By performing this, it is possible to seamlessly stitch together the different video streams into a coherent and visually appealing final product. This pre-processing step is essential for achieving high-quality results in real-time video stitching applications, as it helps to overcome the challenges that arise due to the varying resolutions and frame rates of the input video streams. The ultimate goal of this stage is to ensure that the final stitched video is free from any artifacts or distortions that may detract from its overall quality.

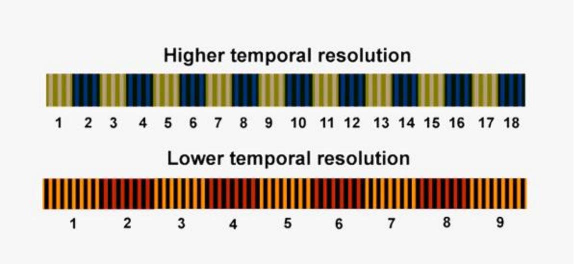


Fig. 8. Temporal resolution examples.

4.3. Stitching and panoramic construction phase

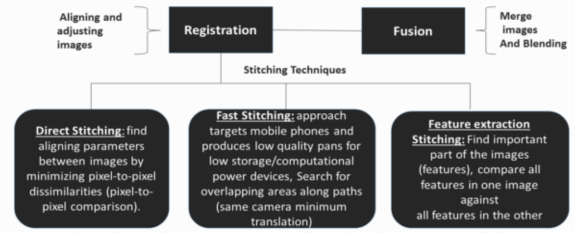
4.3.1. Registration Phase

Stitching registration techniques can be classified into three main approaches: direct, fast, and feature-based. The direct approach involves finding correlation parameters between pixels in different images, by minimizing pixel-to-pixel dissimilarities. This method has a polynomial time complexity with respect to the number of pixels, N [4].

The Fast approach is designed for mobile devices with limited storage and processing power, resulting in lower quality panoramic videos. In contrast, the Brown and Lowe Method, also known as the BLM approach, introduced stitching with invariant features, which is considered the most efficient and high-quality method [24]. This approach involves polynomial time complexity based on the number of extracted features, denoted as n, where n is significantly smaller than the total number of pixels N. The feature-based stitching algorithms typically include two primary steps: registration and blending/fusion. During the registration phase, the algorithms extract and match features [4].

4.3.2. Fusion phase

The fusion phase, in contrast to the registration phase, is a simpler process and involves merging the images and applying blending to ensure seamless stitching. Various methods can be employed for blending, including alpha "feathering" blending, which involves taking a weighted average between the two frames [24]. Alpha blending works best when the image pixels are well-aligned and the differences between the frames lie primarily in the illumination levels. Another popular blending approach is the Gaussian pyramid method, which merges the images at different frequency bands and applies appropriate filtering. The lower frequency bands result in more blurred borders, achieving a smoother blend [4], (see Fig. 9).


Aligning and adjusting images

Registration

Fusion

Merge images And Blending

Stitching Techniques

Direct Stitching: find aligning parameters between images by minimizing pixel-to-pixel dissimilarities (pixel-to-pixel comparison).

Fast Stitching: approach targets mobile phones and produces low quality pans for low storage/computational power devices, Search for overlapping areas along paths (same camera minimum translation)

Feature extraction Stitching: Find important part of the images (features), compare all features in one image against all features in the other

Fig. 9. The stitching involves image registration and image fusion.

The most unique process in the framework which is the estimation of two types of motion not only one, but the common use is also to estimate the motions at the corresponding frames between the multiple different videos, which is referred to by inter motions, and estimate the motions within the same video between neighboring frames, which is referred to as intra motions.

For inter motions, it can be represented as "Tn, m(t)", where 't' is the time of motion frame taken between two footages 'n' and 'm'. The mathematical equation for this can be expressed as:

$Tn, m(t) = f(n(t), m(t))$

Where f is a function that calculates the inter motion between two footages n and m at time t.

For intra motions, it can be represented as "Cn(t)", where 't' is the time and 'n' denotes the view number. The mathematical equation for this can be expressed as:

$Cn(t) = g(n(t), n(t+1))$

Where g is a function that calculates the intra motion between two neighboring frames n at time t and t+1.

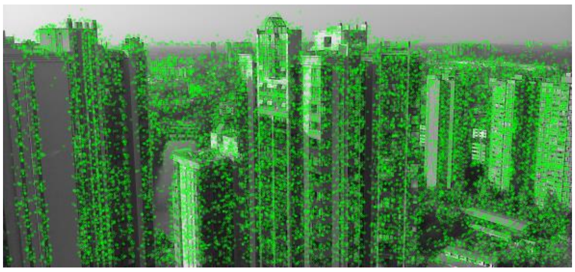


Fig. 10. Feature detection from input frames.

With two major types for path estimation which are a global path vs bundled paths, it's found that bundled paths can reduce and handle jitters, shaky frames and ghosting, which appears because of the parallax which makes global path homography model insufficient, which means some regions in the given frames won't be stabilized right, the bundled path approach is adopted in this framework as its proved to handle all the previous cases as it produces a comparable results to the 3D methods but with respect to the metrics of the 2D methods. As the optimization stage is the core and most important stage in the whole framework its divided into three main components: the first component is to provide the best quality feature extraction on the inter motions and intra motions level, which is fast and rich feature tracking. The second component ensures the generation of the optimal camera path which is perfectly positioned in along all of the original paths to get over the perspective distortions, which is mutually optimal camera path generation. The last component is to use the generated optimal path from the previous stage to start the joint stitching and stabilization processes.

The feature detection (see Fig. 10) and tracking is one the most important phases to apply the stitching algorithm, features from accelerated segment test (FAST) [13], which uses a 16-pixels circle numbered from 1 to 16 to identify whether point P is a corner.

Final step includes point or not, and Kanade–Lucas–Tomasi (KLT) feature tracker [14], makes use of spatial intensity information to direct the search for the position that yields the best match, both are the fastest algorithms to be used for feature detection and tracking, which are used as this is a real time framework. Also, random sample consensus (RANSAC), which can be described as an outlier's detector or as an iterative method to estimate parameters of a mathematical model from a set of observed data that contains outliers.

Grid-based detection method is used instead of the traditional global threshold method, as the global threshold produces few features in low gradient areas, such as the sky, as the threshold is looking after the highly textured areas, that's why grid-based is much better as it gives a local value to each part of the grid so that more features can be produced, FAST is applied on each grid, and then an automatically chosen value for the grid threshold will be assigned and it will update until it reaches the best fit value. If one of the grids produced too many features, then a pruning algorithm is introduced based on the feature detection score of each grid. After the feature detection and setting the local threshold, KLT tracker starts its work on the given frames.

In the feature detection and tracking phase, the FAST (Features from Accelerated Segment Test) algorithm is used to detect corners. The mathematical representation of FAST algorithm can be expressed as:

$$FAST(P) = \{1, 2, 3, ..., 16\}$$

Where P is a point in the image, and the set {1, 2, 3, ..., 16} represents the 16 pixels in a circle around the point P. The algorithm determines whether the point P is a corner based on the intensity values of these 16 pixels.

The KLT (Kanade-Lucas-Tomasi) feature tracker uses spatial intensity information to track features in the video. The mathematical representation of KLT algorithm can be expressed as:

$$KLT(I) = \text{argmin} ||I(x + u) - T(x)||$$

Where I(x) is the intensity of the current frame at position x, T(x) is the intensity of the previous frame at position x, and u is the displacement vector between the two frames. The KLT algorithm finds the displacement vector that minimizes the difference in intensity between the current and previous frames.

The combination of feature tracking and feature matching algorithms gets over the limitations of matching the features on the dominant plane, the rejection of outliers in the feature matching process can be represented mathematically using the random sample consensus (RANSAC) algorithm. Given a set of observed data points that contain outliers, RANSAC estimates the parameters of a mathematical model that best fits the inliers (the non-outlier data points). The algorithm starts by selecting a random subset of the data points and using them to fit the model. Then, it checks the remaining data points to see if they are consistent with the model. If a sufficient number of data points are found to be consistent with the model, the algorithm refits the model using all of the inliers. This process is repeated multiple times to find the best fit model that has the most inliers. The dominant plane matches from the previous frame can be used to guide the search for inliers in the current frame, reducing the computation time and increasing the accuracy of the feature matching process.

A 16x16 grid is generated using the bundled paths algorithm, which wraps each frame with the previous frame, by generating a camera path for each cell in the grid. The bundled paths method reduces the perspective distortions, and it deals with parallax. Bundled-path stabilization has two main advantages: it can manage parallax and correct significant perspective distortions that occur when a single homography is used.

In the final step of the framework, the grid-based detection method is used to detect features in the video. The mathematical representation of this can be expressed as:

$f = g(I, T)$

Where I is the current frame, T is the threshold value for the grid, and g is a function that calculates the features in the video based on the intensity values in the current frame and the threshold value for the grid. The function g can be expressed as:

$g (I, T) = \{p1, p2, ..., pn\}$

Where p1, p2, ..., pn are the feature points detected in the current frame based on the intensity values and the threshold value T.

4.4. Object detection phase

YOLOv4 (You Only Look Once version 4) is a state-of-the-art object detection algorithm that has several advantages over other object detection models. One major advantage of YOLOv4 is its speed. YOLOv4 can process images in real-time, meaning it can detect objects in a video stream at a speed of around 40 frames per second on a GPU. This makes it ideal for applications that require fast and accurate object detection, such as surveillance, traffic monitoring, and robotics.

Another advantage of YOLOv4 is its accuracy. YOLOv4 has been shown to outperform other object detection models on several benchmark datasets, achieving state-of-the-art performance in terms of both accuracy and speed. This is due in part to the use of a highly optimized CNN architecture with many residual blocks, which allows YOLOv4 to extract features from the input image more efficiently than other models.

In addition, YOLOv4 is highly configurable and can be customized to meet the specific needs of different applications. For example, users can adjust the size of the input image, the number of cells in the grid, the number of bounding boxes per cell, and the threshold for objectness scores to optimize the model for different use cases.

Overall, YOLOv4 is a highly effective and efficient object detection algorithm that is well-suited for a wide range of applications. Its combination of speed, accuracy, and configurability make it an attractive choice for researchers and practitioners who need fast and reliable object detection.

YOLOv4 divides the input image into a grid of cells and predicts a fixed number of bounding boxes and associated objectness scores for each cell. The model also associates each bounding box with class probabilities for each class of object it has been trained on.

To extract features from the input image, YOLOv4 uses a convolutional neural network (CNN) architecture with many residual blocks. The CNN features are used to predict the objectness scores, bounding box coordinates, and class probabilities for each cell and bounding box.

For objectness score prediction, YOLOv4 uses logistic regression, represented by the equation: P(object) = sigmoid(score), where "score" is the output of the final layer of the CNN for that bounding box.
For bounding box coordinate prediction, YOLOv4 predicts the coordinates relative to the coordinates of the cell in which the box is located, using the following equations: bx = sigmoid(tx) + cx, by = sigmoid(ty) + cy, bw = pw * exp(tw), bh = ph * exp(th). Here, "tx" and "ty" are the predicted x and y offsets of the center of the bounding box relative to the cell, "tw" and "th" are the predicted widths and heights of the box, "pw" and "ph" are the width and height of the anchor box (used to normalize the width and height predictions), and "cx" and "cy" are the coordinates of the top-left corner of the cell.

Finally, for class probability prediction, YOLOv4 uses softmax regression, represented by the equation: P (class_i | object) = exp(score_i) / sum(exp(score_j)), where "score_i" is the output of the final layer of the CNN for class i, and "sum(exp(score_j))" is the sum of the exponential scores for all classes.

5. Experimental Testing

The input videos were taken from the Tello drones, and the processing is done using RPI 4, the performance metrics used throughout the process were three main metrics:

1) The delay of the output video, we use Python's time module to record the time before and after the processing is done, and then compute the difference between these times.

2) The stability score which is a measure of the smoothness of a stitched video. It is calculated by tracking features on the stitched video and retaining tracks with a length of greater than twenty frames. The energy percentage of the lowest frequencies (2nd to 6th without DC component) is then calculated for these tracks, and the final score is obtained by averaging the energy percentages from all tracks [25]. A high stability score (close to 1) indicates a smooth and stable stitched video.

3) The stitching score which is a measure of the quality of the stitching in a stitched video. It is calculated by calculating the feature reprojection error for each frame. This involves calculating the distance between matched features after they have been transformed. The stitching score for a single frame is obtained by averaging these distances for all feature pairs. The final stitching score is obtained by taking the worst score (largest value) among all frames [26]. A low stitching score indicates a good alignment and high-quality stitching.

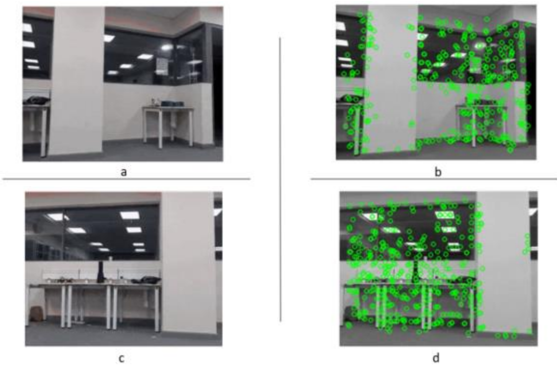


Fig. 11.a Illustrates the right view, Fig. 11.b illustrates the right view with feature detection, Fig. 11.c Illustrates the left view, Fig. 11.d illustrates the left view with feature detection.

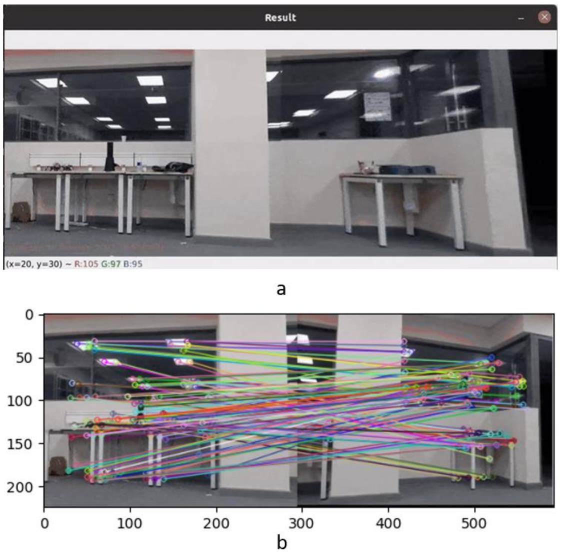


Fig. 12.a Illustrates the panorama view, Fig. 12.b illustrates the panorama view with feature matching.

The tests were conducted on three different scenarios:

5.1 Static object, Static cameras

The scenario of static objects and static cameras is a common one, particularly in the field of image stitching. This scenario involves capturing multiple images of the same scene using cameras that are fixed in position and do not move during the image capture process. The objects in the scene are also static and do not move between the different images captured (see Fig. 11).

This scenario is often encountered in applications such as panoramic photography, where multiple images of a scene are stitched together to create a single large image that captures a wider field of view than a single image. Other applications of this scenario include surveillance systems, where multiple cameras are used to cover a wide area.

The main advantage of this scenario is that it simplifies the image processing task, as the positions and orientations of the cameras are fixed and known, and the objects in the scene do not move. This allows for more accurate and efficient image stitching, as well as other types of image processing such as object detection, tracking, and recognition.

Overall, the scenario of static objects and static cameras is an important and widely used one in computer vision and is particularly useful in the context of image stitching and other image processing tasks, (see, Fig. 12).

5.2 Moving object, Static cameras

In the scenario of moving objects and static cameras involves capturing images of a scene using cameras that are fixed in position and do not move during the image capture process, but where the objects in the scene are in motion, (see Fig. 13 and Fig. 14).

This scenario is encountered in a wide range of applications, including security and surveillance systems in large buildings or outdoor areas, where multiple static cameras are positioned to cover the entire area. The resulting video provides a comprehensive view of the scene and helps in identifying potential security threats.

Another application of this scenario is event videography, where multiple static cameras are used to capture different angles of a performance, ceremony, or sports event. The resulting video provides a wide-angle view of the event, allowing the viewer to see all the important details.

The main challenge in this scenario is to accurately detect and track the moving objects in the scene, while also accounting for any changes in lighting, shadows, or occlusions. This requires the use of advanced computer vision techniques such as object detection, tracking, and recognition, as well as the ability to handle multiple object trajectories and occlusions.

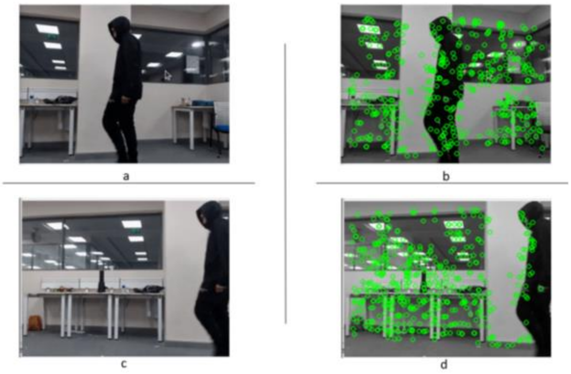


Fig. 13.a Illustrates the right view, Fig. 13.b illustrates the right view with feature detection, Fig. 13.c Illustrates the left view, Fig. 13.d illustrates the left view with feature detection.

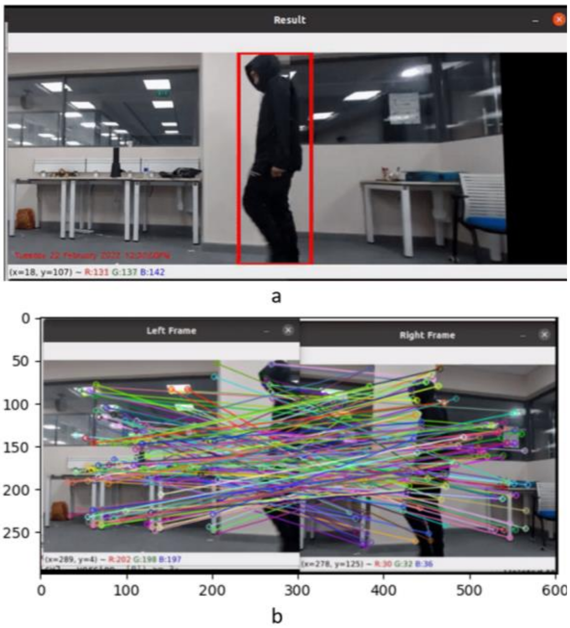


Fig. 14.a Illustrates the panorama view, Fig. 14.b illustrates the panorama view with feature matching.

Overall, the scenario of moving objects and static cameras is an important and challenging one in computer vision and is particularly useful in the context of security and surveillance systems, as well as event videography and other applications where a wide-angle view of a moving scene is required.

5.3 Static object, Moving cameras.

The scenario of static objects and moving cameras involves capturing images of a scene using cameras that are in motion, but where the objects in the scene are static and do not move during the image capture process, encountered in a variety of applications, including virtual tours of real estate properties, where a camera mounted on a drone, or a wearable device is used to capture video as the operator moves through the property. The resulting video provides a seamless view of the property, giving the viewer an immersive experience.

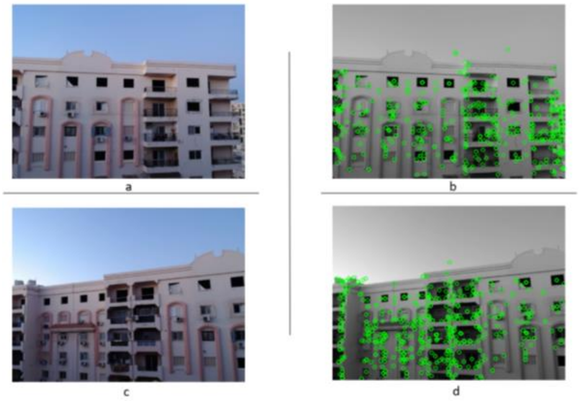


Fig. 15.a Illustrates the right view, Fig. 15.b illustrates the right view with feature detection, Fig. 15.c Illustrates the left view, Fig. 15.d illustrates the left view with feature detection.

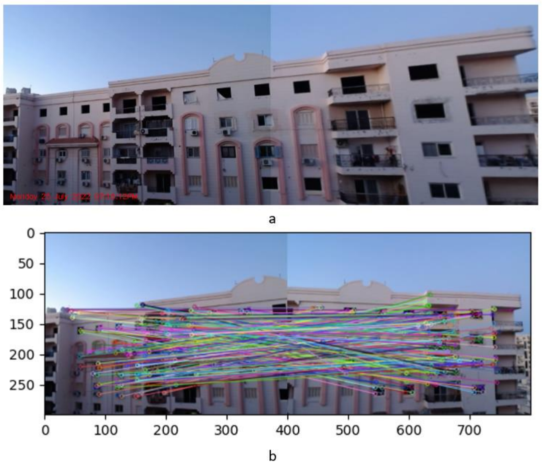


Fig. 16.a Illustrates the panorama view, Fig. 16.b illustrates the panorama view with feature matching.

Table 1. The comparison of the "Stability Score" and "Stitching Score" for different experimental scenarios.

	Scenario I	Scenario II	Scenario III
Stability Score	1.00	0.93	0.90
Stitching Score	0.67	1.01	1.02

Documentary filmmaking is another application of this scenario, where a moving camera is used to capture the entire story. The resulting video provides a seamless view of the scene and helps in creating an immersive experience for the viewer. Action sports is another popular application of this scenario, where a moving camera is used to capture the entire performance, allowing the viewer to see all the important details from multiple angles.

The main challenge in this scenario is to achieve video stabilization, which is the process of removing camera shake and improving the quality of the final video. This can be achieved through the use of advanced computer vision algorithms that analyze the motion of the camera and compensate for any movements or vibrations, resulting in a smoother and more immersive viewing experience.

Overall, the scenario of static objects and moving cameras is an important and challenging one in computer vision and is particularly useful in the context of virtual tours, documentary filmmaking, and action sports. It requires advanced techniques for video stabilization and motion analysis and can provide a more immersive viewing experience for the viewer.

5.4 Results

Table 1. Based on the results of the three trials, we can see that the stability score was consistently high, with the first trial having a perfect score of 1.00 and the other two trials having scores of 0.93 and 0.90, respectively. This indicates that the stitched videos were relatively smooth and stable across all three experiments.

On the other hand, the stitching score varied more across the three trials, with the first trial having a score of 0.67, which is a relatively low score indicating good alignment and high-quality stitching. The second and third trials had higher stitching scores of 1.01 and 1.02, respectively, indicating lower quality stitching with lower alignment than the first trial.

Given that the three experiments were independent and conducted in different environments, it is interesting to note that the stability score remained high across all three trials. However, the stitching score varied, which suggests that the quality of the stitching may be influenced more by the specific environmental factors and conditions of each experiment, rather than the overall stability of the stitched video.

Overall, a high stability score is desirable as it indicates a smoother and more stable stitched video, while a lower stitching score is also desirable as it indicates better alignment and higher quality stitching.

6. Experimental Results and discussion

6.1. Video Stitching Dataset

The model used videos from which were used in other papers to compare the results of the stitching quality with other paper [17], the dataset used includes six videos which represents two videos captured while moving which introduces shaky frames and jitters due to the temporal resolution.

6.2. Object Detection Dataset

The COCO (Common Objects in Context) dataset is a widely used object detection dataset for computer vision research. It contains more than 330,000 images and over 2.5 million object instances across 80 different object categories, the dataset was created to provide a challenging benchmark for object detection algorithms, with images captured in a wide variety of settings and contexts, including both indoor and outdoor environments. The images were sourced from a variety of sources, including Flickr and Microsoft Bing image search, and each image is annotated with object bounding boxes and category labels, the COCO dataset has become a standard benchmark for evaluating the performance of object detection algorithms and has been used in numerous research studies and competitions. It is widely recognized for its high-quality annotations, which are highly accurate and detailed, making it an excellent resource for training and testing computer vision models, overall, the COCO dataset has played a significant role in advancing the state-of-the-art in object detection and remains an important resource for computer vision researchers and practitioners.

6.3. Performance Measures:

6.3.1. Video Stitching metrics:

The stability and stitching scores for the examples tested are summarized in Table 2. Based on these scores, it appears that there has been a significant improvement in stability for all of the examples.

6.3.2. Object detection metrics:

There are many metrics that can be used to evaluate the performance of an object detection model, but two common ones are Mean Average Precision (mAP) and accuracy.

mAP is a metric that measures the average precision at different levels of recall. Precision is defined as the ratio of true positives to the total number of predicted positives, while recall is defined as the ratio of true positives to the total number of actual positives. The mAP is calculated by taking the average of the precision at different recall levels, where the precision is interpolated between different recall levels. In mathematical terms, the mAP is given by:

mAP = (1/N) * ∑[i=1 to N] (AP_i)

where N is the number of object categories, and AP_i is the average precision for category i.

If an object detection model has an mAP value of 89.5%, it means that, on average, the precision at different recall levels is 89.5%.

Accuracy, on the other hand, is a metric that measures the percentage of correctly classified objects. It is defined as the ratio of true positives and true negatives to the total number of objects. In mathematical terms, accuracy is given by:
accuracy = (true positives + true negatives) / (true positives + false positives + true negatives + false negatives)

If an object detection model has an accuracy of 93.28%, it means that, on average, it correctly identified and localized 93.28% of the objects in the images it was tested on.

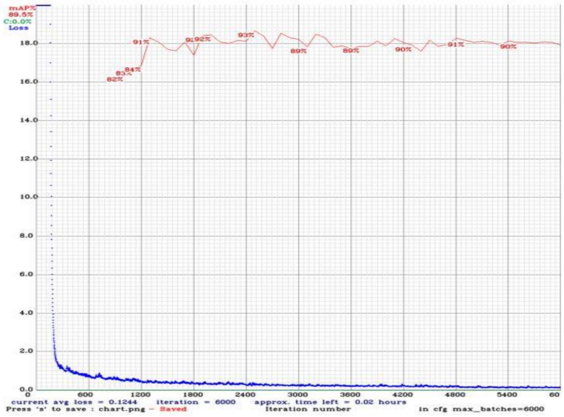


Fig. 17. mAP and loss graph for object detection model.

Table 2. Experimental results for different video datasets.

Video	Length	Frame width	Frame height	Approx. fps	Stitching score		Stability score		
Video 1	0:00:13	1280	720	30 fps	0.99	0.9	0.67	0.83	0.88
Video 2	0:00:12	1280	720	30 fps	0.54	0.51	0.37	0.78	0.82
Video 3	0:00:10	1280	720	30 fps	1.07	1.01	0.65	0.82	0.86
Video 4	0:00:13	1280	720	30 fps	1.01	0.89	0.71	0.88	0.89
Video 5	0:00:15	1280	720	30 fps	0.43	0.37	0.68	0.84	0.88
Video 6	0:00:12	1280	720	30 fps	1.04	0.94	0.63	0.91	0.94

Prediction time is a measure of how long it takes for the model to process an image and output its predictions. In the case of the object detection model, it takes 4.9 milliseconds on average to process an image and output its predictions.

So, in summary, an object detection model with a mAP value of 89.5%, an accuracy of 93.28%, and a prediction time of 4.9 milliseconds is a high-performing and efficient model that can identify and localize objects in images with high precision and speed, (see Fig. 17).

7. Conclusion

With the advancement of technology and the emergence of new requirements, image and video stitching has become an essential aspect of both personal and professional applications. To address the challenges and opportunities presented by this field, various experiments have been conducted. The outcomes demonstrate that the proposed approach is capable of producing panoramic images with improved compression ratios, faster and more precise reconstruction, and enhanced object detection capabilities.

The proposed model works on jointly stitching and stabilizing the live stream from two or more quadcopters. The estimation of inter motion between the live feeds from the cameras and intra motion between the frames of the same video. The entire process is turned into an optimization problem to get the best fit stitching and stabilized video, so the intra motion method assures the temporal smoothness sustainability between the different frames of the same video, and the inter motion method assures the forcing of the spatial alignment between the multiple videos provided by the drones. Handle scenes with parallax, each video frame is divided into smaller cells so that it is easier to use the bundled-path methodology.

The suggested model displayed exceptional object detection performance, achieving high average precision measures across multiple recall levels. Moreover, it exhibited consistent and robust performance in diverse experimental settings, with relatively high stability and stitching scores which indicates how the model is effective.

References

[1] P. A. D. P. S. S. K. V. Chung A, “A survey on aerial swarm robotics,” IEEE Trans Robot., 2018.

[2] T. J. L. S. Xi, “Review of unmanned aerial vehicle swarm communication architectures and routing protocols,” Appl Sci, 2020.

[3] A. Tahir, J. Böling and M.-H. e. a. Haghbayan, “Swarms of unmanned aerial vehicles — A survey. In: Journal of Industrial Information Integration.,” 2019.

[4] S. Y. a. S. F. M. el Shehaby, “"An Efficient Multi-View Panoramic Imaging and Extra Compression of Surveillance Cameras’ Footage Using Stitching,",” 2019.

[5] M. S. W. a. L. Andrew, “Drone swarms—A monograph by school of advanced military studies.,” 2017.

[6] M. a. B. M. Dorigo, “Swarm intelligence,” 2007.

[7] J. A. K. V. P. G. Ahmadzadeh A, “Multi-UAV cooperative surveillance with spatio-temporal specifications.,” Proceedings of the 45th IEEE conference on decision and control, 2006.

[8] V. V. S. M. Petrl´ık M, “Coverage optimization in the cooperative surveillance task using multiple micro aerial vehicles,” IEEE international conference on systems man and cybernetics (SMC), 2019.

[9] V. V. C. J. T. J. L. G. Saska M, “Swarm distribution and deployment for cooperative surveillance by micro-aerial vehicles.,” J Intell Robot Syst., 2016.

[10] T. M. K. A. M. D. M. N. K. V. Mohta K, “QuadCloud: a rapid response force with quadrotor teams.,” Cham: Springer International Publishing, 2016.

[11] D. R. Ritz R, “Carrying a flexible payload with multiple flying vehicles,” IEEE/RSJ international conference on intelligent robots and systems, 2013.

[12] K. V. Loianno G, “Cooperative transportation using small quadrotors using monocular vision and inertial sensing.,” IEEE Robot Autom Lett. , 2018.

[13] S. M. G. M. C. N. C. V. C. C. Abdelkader M, “Optimal multi-agent path planning for fast inverse modeling in UAV-based flood sensing applications.,” international conference on unmanned aircraft systems (ICUAS), 2014.

[14] Z. Z. L. C. Y. Z. Wei LYU, “A survey on image and video stitching,” 2019.

[15] J. W. a. S.-F. C. Y. Wang, ““Camswarm: Instantaneous smartphone camera arrays for collaborative photography.”,” 2015.

[16] W. J. a. J. Gu, ““Video stitching with spatial-temporal content preserving warping,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW),” 2015.

[17] S. L. T. H. S. Z. B. Z. G. M. Heng Guo, “Joint Video Stitching and Stabilization From Moving Cameras. IEEE Trans Image Process.,” 2016.

[18] W. X. J. Z. M. Z. Z. W. a. X. L. J. Li, ““Efficient Video,” 2015.

[19] F. Perazzi et al., ““Panoramic video from unstructured camera arrays,”,” 2015.

[20] H. L. P. T. G. Z. a. H. B. H. Jiang, ““3D reconstruction,” 2012.

[21] K. &. L. S. &. C. L.-F. &. Z. B. Lin, “Seamless Video Stitching from Hand-held Camera Inputs,” 2016.

[22] J. J. &. B. A. Flores, “Let’s Democratize Drones! Using the Ryze Tello Drone as a Tool for Ecological Farm Design & Landscape Ecology Research.,” (2019).

[23] F. &. V. C. Vancea, “ Portable UDP port forwarding in user space.,” 2009.

[24] M. B. a. D. G. Lowe, “Automatic Panoramic Image Stitching using Invariant Features,” 2007.

[25] L. Y. P. T. a. J. S. S. Liu, “Bundled camera paths for video stabilization,” *ACM Trans*, vol. 32, p. 78, 2013.
[26] M. B. a. L. M. C. Buehler, “Non-metric image-based rendering for video stabilization,” *IEEE CVPR*, vol. 2, 2001.
[27] Ardupilot, “ “Swarming,” Mission Planner,” 2018.
[28] Botlink, “Botlink XRD-real time data upload.,” 2017.