Implementing & Integrating Live Mission Recording with Starter Drones

JURGIS ŠERKŠNAS, University of Twente, The Netherlands

Figure 1 -DJI Mini 2 drone

ABSTRACT

The rapid growth of technological advancement present in drones is leading to promising drone-related features. Live Mission Recording is one of those features introduced and limited by DJI only for high-end and high-cost drones. The technology's potential value in tandem with its absence for starter drones offers a foundation for our research. We aim to implement and integrate the Live Mission Recording feature for a starter drone. The research covers 1. the requirements of the feature's functionality, 2. the available elements to support the requirements offered by the DJI mobile SDK, and 3. the possibilities of the Live Mission Recording feature's implementation and integration for the DJI Mini 2 drone.

KEYWORDS

Live Mission Recording, drone, DJI, autonomous, implementation, integration, DJI mobile SDK

1 INTRODUCTION

Drones have become steadily popular throughout recent years due to their advanced sensors, high-quality cameras, and their ability to move in three-dimensional space allowing them to easily access and observe areas that would otherwise be difficult for humans to reach. Their advanced technological improvements have shown to be practical tools for various applications, including various observations. These observations are important for many different fields, such as agriculture [1], and structural integrity [2].

DJI, one of the leading drone manufacturers [3], has made significant technological advancements, making their drones easy to use, and with their provided mobile SDK[1], developers can implement the technical behavior of the drones to their liking. This provides a great opportunity to explore DJI's highly advanced drones to implement different types of features that could streamline and improve the development of complex observation missions.

Live Mission Recording (**LMR**) is a feature implemented by DJI [4] that allows the drone operator to capture the state of the drone in a three-dimensional environment. These states provide all the information necessary for the drone to execute an autonomous inspection mission by reaching each of the captured states. Creating an observation mission using the LMR may be more beneficial than using the traditional means of pre-planning the drone's behavior. The current conventional method of mission planning as seen in the Litchi[2] application is done via the use of a two-dimensional digital map, which could get replaced by the LMR feature. The benefit of LMR can be seen in DJI's effort of showcasing the technology to the public [5].

2 PROBLEM STATEMENT

The LMR feature provided by DJI is available only on the DJI PILOT application. This application is dedicated to high-end drones [6], which come at a significant expense. However, the development environment offered by DJI might enable the possibility to implement this feature on more affordable drones, which could reduce the need to spend large amounts of financial resources.

2.1 Research Question

To implement and integrate the LMR feature on a starter drone, we aim to answer the following research questions (**RQ**):

TScIT 39, July 7, 2023, Enschede, The Netherlands

---

[1] https://developer.dji.com/mobile-sdk-v4/

[2] https://flylitchi.com/

RQ1: What are the requirements for a functional LMR feature?
RQ2: What elements for the LMR feature are supported by the mobile SDK?
RQ3: How can the LMR feature be implemented and integrated on a starter drone using the mobile SDK?

2.2 Methodology

Design Science Methodology is an approach to research that aims to solve research problems by developing new knowledge. Throughout this research, this methodology will be applied to solve the identified problem of LMR accessibility for starter drones. Relevant functional requirements will be defined, which will be used as the basis to design and develop a solution. Afterwards, the developed solution will be tested in parts and evaluated.

2.3 Technology Validation

To validate the implementation of the LMR technology for starter drones, the DJI Mini 2 drone (See Figure 1) will be used. The LMR will be validated by implementing parts of the technology's components that when integrated make the LMR feature complete.

3 RELATED WORK

The approach to finding related work that would be related to Live Mission Recording for drones did not yield any results on any scientific literature databases.

However, using IEEE Xplore and Google Scholar we can find some related works that delve into the implementation and integration of other drone features. For instance, a study by Besada et. al. [7] presents a Mission Definition System (MDS). This system facilitates the definition of drone missions, supports their implementation with autonomous flights, and provides interfaces to implement automated operations for different UAV (drone) autopilots, thereby delving into the technical aspects of autonomous drone missions' implementation and integration. Another notable work by Besada et. al. [8] discusses the concept of drones-as-a-service, which connects the general public to end services offered by autonomous drones. This research provides a comprehensive look into the technical implementation of drone-related features for autonomous missions.

Table 1 - Drone behavior explanation dependent on action execution

Action	Drone Behavior
TakeOffAction	Takes off from the ground
AircraftYawAction	Rotates along the Z (yaw) axis
GoHomeAction	Goes back to the location where the drone took off
HotPointAction	Orbits around a certain point of interest
LandAction	Lands in the current location
GimbalAttitudeAction	Rotates the gimbal with specified behavior
RecordVideoAction	Records a video
ShootPhotoAction	Takes a photograph, or photographs

These works provide an overview of the implementation and integration of drone-related missions and services. However, since the LMR feature has a very specific purpose and functionality, relevant material that would be useful for this research and this technology's implementation has not been found.

Since there is no scientific literature available on the Live Mission Recording feature, no systematic literature review is possible. Thus, it is undeniable that this research is very focused on a particular technology, which requires original insights into the capabilities of implementing and integrating the technology for starter drones.

4 LMR IMPLEMENTATION & INTEGRATION

It is important to define what requirements LMR entails since this will help to narrow down the focus on a specific technology (or technologies) that will fulfill the requirement.

In the context of using LMR for inspection purposes, DJI's implementation of the feature [4] has provided useful insights into the requirements this technology necessitates. Thus, the following requirements (**R**) have been discovered:

R1: The drone has to autonomously navigate to a specific point in the environment.
R2: The drone has to inspect the point of interest.
R3: The aforementioned information has to be captured and derived as the state of the drone.
R4: The captured states can be used to create and repeat a mission for periodical inspections.

4.1 Autonomous Navigation (R1)

The location of a drone in a three-dimensional environment is defined by GPS and the altitude from the place where the drone took off[3]. GPS is used to provide information on latitude and longitude. The mobile SDK derives the latitude, longitude, and altitude of a drone's current position using the getAircraftLocation method, which returns an object of type LocationCoordinate3D with latitude, longitude, and altitude as the only three properties. The LocationCoordinate3D class can be used to create multiple instances that store the information necessary for any three-dimensional location. These instances can be used to save the properties of important locations for the drone to visit during the inspection mission.

4.1.1 Off-The-Shelf Implementation

A common and straightforward way for a DJI drone to reach a specific location utilizing the mobile SDK is by using the GoToAction as a mission action, which makes the drone fly

---

[3] https://forum.dji.com/thread-257153-1-1.html#:~:text=As%20with%20most%20DJI%20drone,and%20be%20at%20%2D120%20feet.

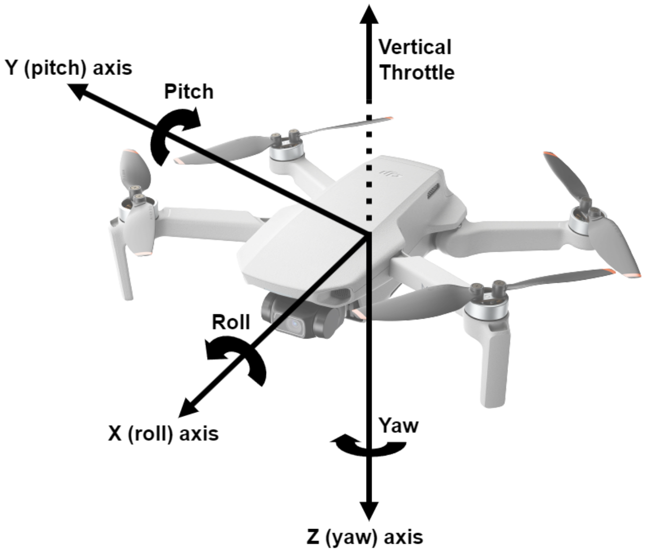


Figure 2 - Roll, Pitch, and Yaw are used either as angles or X, Y, Z axes respectively. Vertical Throttle value represents the Z (yaw) axis.

autonomously to a specified location. This action is used during the construction of a MissionControl object and scheduled as one of the elements to be performed using the scheduleElement method. There are other actions that can also be scheduled (See Table 1).

Unfortunately, the DJI Mini 2 drone firmware does not support the GoToAction [9, 10], nor any support of waypoint missions that would allow the drone to autonomously fly to a designated location. This has been confirmed when the action was implemented in a MissionControl of the DJI Mini 2 drone, and no action has been observed upon the attempted execution of the element. However, the Litchi app does seem to support waypoint missions for unsupported drones, even for DJI Mini 2 [11], which utilizes the autonomous navigation system to reach the locations, thus confirming that there is a way of allowing the starter drone to fly from one location to another in a three-dimensional environment.

4.1.2 Custom Implementation.

Virtual Sticks is a feature offered by the mobile SDK where a DJI drone can be controlled by other means of input other than the standard way of using a physical remote controller. The alternative mean of controlling the drone is by using a virtual representation of the remote controller sticks that are displayed on the mobile phone. The Virtual Sticks provide roll, pitch, yaw, and vertical throttle values that are stored on runtime in the custom drone application. These values are used to inform the drone how it will move around the environment (See Figure 2). To send the values to the drone, a FlightControlData object is created containing the values and forwarded to the sendVirtualStickFlightControlData method for execution. This method is crucial for our custom implementation since it is the only method available that lets us manipulate the drone's movement behavior in the environment. It is important to note that if we want to use this method, the Virtual Sticks have to be set up.

The preconditions (**PR**) of the Virtual Sticks implementation to use the method are:

> **PR1:** Enabling Virtual Stick control by executing the setVirtualStickModeEnabled method and passing the value of **true** as a parameter;
>
> (*optional*) **PR2:** Enabling Virtual Stick Advanced mode and passing the value of **true** as a parameter. This mode allows the drone to compensate for wind when hovering, thus the developer does not need to implement their own custom solution to compensate for the drone's drift due to wind;
>
> **PR3:** Executing the setRollPitchCoordinateSystem method sets the coordinate system in which the values of roll and pitch will be interpreted. There are two coordinate systems options available: GROUND - which enables the north and south to be the X (roll) axis, east and west to be the Y (pitch) axis for the aircraft's horizontal movements, and BODY - which enabled the front and back of the aircraft to be the X (roll) axis, right and left sides of the aircraft to be Y (pitch) axis. In other words, the GROUND enables the cardinal direction to represent the X (roll) and Y (pitch) axes, and the BODY enables the X (roll) and Y (pitch) axes to be relative to the aircraft;
>
> **PR4:** Executing the setRollPitchControlMode method sets how the roll and pitch values will be interpreted. There are two modes available: ANGLE - which interprets the values as the angle in degrees relative to the aircraft's body, and VELOCITY - which interprets the value as velocity in m/s along the X (roll) and Y (pitch) axes;
>
> **PR5:** Executing the setYawControlMode method sets how the yaw value will be interpreted. There are two modes available: ANGLE - which interprets the value as an angle relative to the north, and ANGULAR_VELOCITY - which interprets the value as angular velocity in degrees/s;
>
> **PR6:** Executing the setVerticalControlMode method sets how the vertical throttle value will be interpreted. There are two modes available: VELOCITY - which interprets the value as vertical velocity in m/s, and POSITION - which interprets the value as altitude;
>
> **PR7:** Executing the sendVirtualStickFlightControlData method at the frequency of 5Hz and 25Hz, or once every 40ms to 200ms.

Once all of the preconditions are met, we can successfully use the sendVirtualStickFlightControlData method to input our own values of roll, pitch, yaw, and vertical throttle. By knowing the current location of the drone, and the target location, we can calculate these values and send them to the drone to reach the target location.

The focus of this custom implementation is to reach the target location using the current drone's location. The current drone

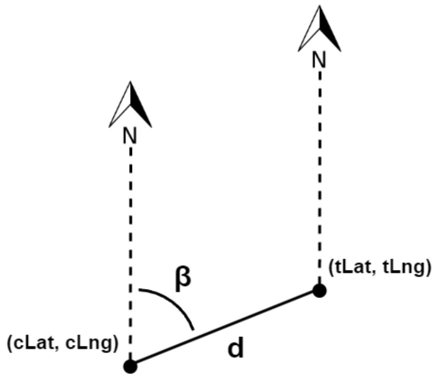


Figure 3 - Bearing angle (**β**), Distance (**d**), Current Latitude (**cLat**), Current Longitude (**cLng**), Target Latitude (**tLat**), Target Longitude (**tLng**)

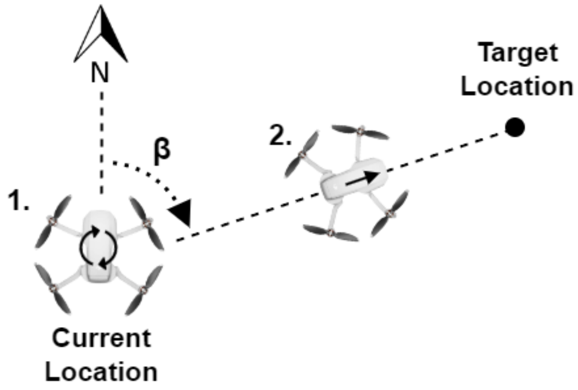


Figure 4 - 1. Drone rotating towards the target location with angle β relative to the north; 2. Drone moving towards the target location

location and the target location consist of latitude (DD[4]), longitude (DD), and altitude (m) values, stored as an instance of the LocationCoordinate3D object. The advanced Virtual Stick mode has been enabled, since then we do not have to worry about wind influence.

To reach the target's altitude, the vertical control mode has been set to POSITION. This allows the drone to ascend or descend depending on its current altitude autonomously. The throttle value is received from the altitude value of the target's LocationCoordinate3D instance and sent to the aircraft via the sendVirtualStickFlightControlData method.

Reaching the target's latitude and longitude coordinates is more complicated since some creative geometry is involved to calculate the pitch and roll values for the drone's horizontal movement behavior. To tackle this issue, our solution included the drone facing the direction of the target location and moving forward until the target location is reached. To face the aircraft towards the target location, the current and target's location latitude and longitude has been used to find the bearing angle β [14] (See Figure 3), using the formula $\beta = atan2(X, Y)$, where:

$$X = cos(tLat) \times sin(|tLng - cLng|)$$

$$Y = Y_1 - Y_2$$

$$Y_1 = cos(cLat) \times sin(tLat)$$

$$Y_2 = sin(cLat) \times cos(tLat) \times cos(|tLng - cLng|)$$

, where *cLat* is the current latitude, *cLng* is the current longitude, *tLat* is the target latitude, and *tLng* is the target longitude. All latitude and longitude values must be converted to radians before the calculations using the formula.

The calculated β result is converted to degrees, and then the angle of the aircraft's rotation relative to the north is known, which will make the drone face towards the target location. The yaw control mode is set to ANGLE, the yaw value is set to the β result converted to degrees and passed to the sendVirtualStickFlightControlData method. After the drone faces towards the target's location, we can move the drone forward by setting the roll and pitch coordinate system to BODY, control mode to VELOCITY, and then set the value of roll to the desired speed we want the drone to move along the X (roll) axis relative to its body.

To know when the drone has reached the target location, the distance (d) between the current drone location and the target location is calculated (See Figure 3) using the haversine formula [14]. Keeping track of the distance between the current drone location and the target location allows us to know when the aircraft has reached the location and stop the drone's horizontal movement behavior.

The end result of combining the β angle, the distance (d), and the technical implementation of using these values to control the drone yields the drone being able to navigate autonomously from one location to another (See Figure 4). This implementation can be applied numerous amounts of times for the aircraft to reach multiple locations, thus this custom implementation replaces the functionality of the GoToAction action.

4.2 Inspecting Point of Interest (R2)

In the context of inspecting an object (or point of interest) using a drone, we will be focusing on the technical capabilities of the DJI Mini 2 drone, and then on the technical implementation of the inspection action.

Figure 5 - Camera gimbal rotation

Beyond the DJI Mini 2 drone's fundamental abilities to navigate within a three-dimensional space, it incorporates an onboard camera that can capture photographs or record videos. This camera is involved in examining a particular point on the observable infrastructure. To accomplish this inspection, the camera is required to orient itself towards the point of interest utilizing the X (roll), Y (pitch), and Z (yaw) axes (See Figure 5).

---

[4] DD - Decimal Degrees

The design of the DJI Mini 2 drone facilitates camera stabilization, both vertically and horizontally, thanks to its gimbal. Nonetheless, the gimbal's design on the DJI Mini 2 constrains the operator's controls only to the adjustments of the camera's pitch axis. Consequently, the camera maintains a fixed position, viewing the drone's frontal area, attributed to the gimbal's lack of movement along the yaw axis.

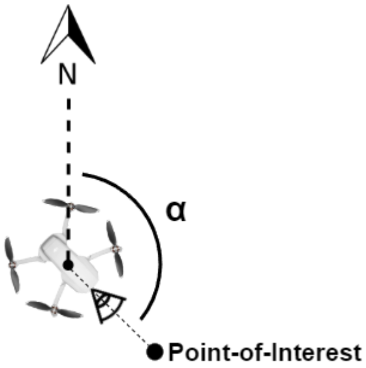


Figure 6 - The angle (**α**) between the north and the point-of-interest, considering the current drone's location

For utilizing the drone for object inspections, the drone must orient itself towards the object and adjust the camera's pitch angle to encompass the point of interest. To achieve this, the angle (α) between the north and the point of interest, considering the target location, must be determined (See Figure 6). Subsequently, this angle can be used to modify the aircraft's yaw using either the sendVirtualStickFlightControlData method or the AircraftYawAction during MissionControl execution. To amend the camera's pitch, the GimbalAttitudeAction can be utilized. Once these adjustments are made, and the point of interest is in the camera's view, the drone can proceed to either capture a photograph or record a video, with the ShootPhotoAction and RecordVideoAction elements respectively.

4.3 Capturing the State of Drone (R3)

The fundamental concept behind the Live Mission Recording (LMR) feature allows the drone operator to maneuver the drone towards a point for inspection, orient the camera towards the point, and upon capturing a photo or recording a video, preserve the drone's state to replicate the precise inspection behavior in future operations.

Within the framework of the LMR feature, 'state' is characterized as the confinement of all relevant data needed to replicate the original inspection process. Hence, adhering to this definition, the state encapsulates:

- The drone's current location and altitude, which will serve as the forthcoming target location;
- The drone's body and/or gimbal position during the inspection;
- And the ongoing inspection activity, either photograph capture or video recording.

The drone's current location and altitude are obtained through the getAircraftLocation method, which yields a LocationCoordinate3D object that encompasses latitude, longitude, and altitude values.

The drone's body and/or gimbal position during inspection can be acquired by retrieving the Attitude from both the Gimbal object and the FlightControllerState object. The Attitude object encapsulates three elements: the roll, pitch, and yaw values of the associated object. This means that the roll, pitch, and yaw values of both body and gimbal position are acquired.

The Attitude obtained from the Gimbal object via the GimbalState conveys the precise current positioning of the camera gimbal, thereby informing us how the camera is oriented towards the point of interest. This data facilitates the execution of the GimbalAttitudeAction element by feeding in the received Attitude. In circumstances where the gimbal's action is constrained, as is the case for the DJI Mini 2, the Attitude derived from the FlightControllerState is used to determine the drone's body yaw value, which represents the angle in degrees in relation to the north. This value can be used with the AircraftYawAction element to turn the drone's front, hence the fixed camera's view, towards the point of interest, thus substituting the lack of gimbal's functionality to turn along the yaw axis.

The ongoing inspection activity can be acquired by tracking the specific feature employed by the operator via the application's user interface (UI). This insight can be leveraged to replicate the previously performed action by deploying the components of ShootPhotoAction or RecordVideoAction as required.

4.4 Mission Creation (R4)

To create a mission, the application should archive the collected data of each state whenever it is captured. The chronology of these captured states must also be retained, as it dictates the order in which locations should be visited, thereby ensuring a consecutive execution of the mission.

Upon completion of the drone's state capture during the inspection process, the stored state information can be utilized to autonomously execute the inspection mission, employing the actions outlined in Table 1 and the autonomous navigation solution in section 4.1.2 Custom Implementation.

4.5 Validation Summary

The implementation of the LMR technology has been validated using the DJI Mini 2 drone. The LMR technology's requirements were used for the implementation testing and validation of its functionalities. Android Studio has been used to set up a development environment for an Android application that had a basic implementation of the DJI mobile SDK. This application is run on a mobile phone which is connected via a type-c cable to the drone's remote controller. Once the drone and the remote controller are turned on, the application is launched, where all the basic functionalities of the drone's control are available, such as the live camera view, take-off and landing functionalities, camera settings, etc. All these functionalities are also available on the DJI FLY application, which is the official application released by DJI to support starter drones.

*4.5.1 Requirement 1 (**R1**)*

As a way for the drone to autonomously navigate to a specific point in the environment, a script was developed for the Android application that turns on Virtual Sticks control, implements all of its preconditions (**PR**), and implements the rest of the custom implementation solution as mentioned in section *4.1.2 Custom Implementation*. The Android application can be accessed at LarvaZZa/LMR-Autonomous-R1.

To test if the custom autonomous navigation implementation works as intended, the script was provided with a pre-defined target location's coordinates and altitude for the drone to reach. The target's position was different than the drone's current position. After the Android application was installed on the phone and the phone connected to the remote controller, the script was executed.

The results were satisfactory. As intended by the script, the drone hovered above the ground when the Virtual Sticks turned on and the drone faced the true north. Afterwards, the drone lifted itself up to the target location's altitude and corrected its yaw position to the calculated bearing angle. The last step that the drone did, was moving forward along the roll axis and checking the distance between the target location and its current location, which eventually led the drone to a full stop and hover. This provided sufficient evidence that the custom autonomous navigation implementation works as intended, and the drone indeed does reach a location autonomously. The only thing to consider was the accuracy with which the drone can reach a specific location in the three-dimensional environment.

*4.5.2 Requirement 2 (**R2**)*

As a way for the drone to inspect a point of interest, another script was developed that executed the AircraftYawAction, GimbalAttitudeAction, ShootPhotoAction, and RecordVideoAction using the MissionControl object as mentioned in section *4.2 Inspecting Point of Interest (R2)*. The Android application can be accessed at LarvaZZa/LMR-Inspection-R2.

The AircraftYawAction and GimbalAttitudeAction elements were provided with predefined values for the main purpose to test if the drone is able to adjust its camera position with these action elements. The ShootPhotoAction was executed once, and the RecordVideoAction was executed with a predefined amount of seconds for the recording duration.

The results were satisfactory. Upon execution of the script, the drone turned its body the predefined amount of degrees relative to the north, the camera's gimbal moved downwards the predefined degrees as well, and the drone shot a photo and a video whose footage was found on the drone's SD card via the DJI FLY application.

*4.5.3 Requirement 3 (**R3**)*

As a way for the drone to capture the necessary information that would be used to execute R1 and R2, a script was developed as mentioned in section *4.3 Capturing the State of Drone (R3)*. The Android application can be accessed at LarvaZZa/LMR-State-R3.

The script captures the drone's current location as a LocationCoordinate3D object that provides the coordinates and altitude, which would act as the future target location. The Attitude from the FlightControllerState object is retrieved to get the drone's current yaw position relative to the north and to use it for the AircraftYawAction. The Attitude from the Gimbal object is retrieved to get the gimbal's current pitch position to use it for the GimbalAttitudeAction. And lastly, to capture the drone operator's action of either shooting a photo or recording a video, two buttons are made that respectively save the operator's action. The duration of the video recording is captured with the user's press on the recording button that starts a timer, and pressing it again stops the timer, which would be used for the RecordVideoAction.

The results were satisfactory. All of the data is captured successfully and stored in the Android applications runtime.

*4.5.4 Requirement 4 (**R4**)*

The best way to validate R4, is to implement the entire functionality of the LMR feature in the Android application. However, since we are focusing on the technology's implementation and testing by parts, this R4 has been validated as a combination of all the aforementioned requirements of R1, R2, and R3. R3 facilitates the executions of R1 and R2. R3 can be executed any number of times, thus creating new values for R1 and R2, which are the new points and actions the drone will eventually have to repeat. To ensure a consecutive execution of each R3 states, an ArrayList could be used, which is an ordered list that can store multiple R3 values as custom objects.

4.6 Limitations

Given that the LMR feature implementation in this research was primarily designed for affordable drones, it is important to consider the limitations that this feature may impose when compared to more expensive drones.

4.6.1 Location Accuracy

GPS is the primary means of identifying a location on the horizontal plane within a three-dimensional environment. However, its accuracy can be affected by various environmental factors. This inconsistency in GPS accuracy is evident in the specifications[5] of the DJI Mini 2, with DJI guaranteeing an accuracy of ±1.5 meters on the horizontal plane. Such inaccuracy could pose a significant challenge during inspections, as the drone might potentially collide with the infrastructure being inspected, or the camera's positioning could be compromised, failing to consistently capture the points of interest.

Technologies like Real-Time Kinematic (RTK) positioning can significantly enhance the precision of a drone's location compared to a basic GPS setup [12]. DJI has incorporated RTK into their higher-end drones, including the Matrice 300 RTK drone[6], which boasts a horizontal accuracy of ±0.1 meters. The distinction between GPS and RTK becomes particularly apparent when their implementations are compared side-by-side [13].

---

[5] https://www.dji.com/nl/mini-2/specs

[6] https://enterprise.dji.com/matrice-300/specs

4.6.2 Camera Quality

Camera quality is crucial in any inspection activity. The effectiveness of a camera can influence whether a drone needs to fly closer to the infrastructure or if it can maintain a safe distance while preserving the quality of the footage. For this reason, a highly capable camera can potentially eliminate the need for highly precise location-determining tools such as RTK while still maintaining the footage quality.

Affordable DJI drones, such as DJI Mini 2, often come equipped with 4K and even 8K single image resolution cameras, which could potentially capture detailed information about the inspected object from a safe distance. Conversely, a lower camera resolution would necessitate the drone to fly closer to the object, thus increasing the risk of collisions.

4.6.3 Gimbal Capabilities

Limited gimbal mobility may hinder inspections of some critical areas, for instance, floor beams under the bridge deck, which are hard to reach. With more affordable drones, such as the DJI Mini 2, the gimbal can rotate up to +20 degrees relative to the drone's body. However, this might not be adequate to inspect locations directly above the drone. The advantage of higher-end drones is their capability to accommodate top-mounted gimbals, like on the Matrice 300 RTK. These gimbals can rotate in all directions, making them an excellent option for conducting inspections from below and inspecting hard-to-inspect areas, such as the floor beams of the bridge.

4.6.4 Autonomous Navigation Path

Autonomous navigation of drones from one place to another can result in challenges related to obstacle collisions. Such complications can arise when the inspection mission is autonomously repeated and the drone's path planning around potential obstacles hasn't been considered. The LMR feature overlooks the route the drone operator took to maneuver around the environment to reach the subsequent state, including circumventing potential obstacles. This constraint limits the LMR feature's ability to account for the path it takes to reach the next state.

A potential solution could involve incorporating a failsafe feature, like the one in the Matrice 300 RTK. By integrating additional obstacle detection sensors, collisions of the drone during its path could be prevented.

5 FUTURE WORK

This research does not only discover unique insights into the implementation and integration capabilities of the LMR technology for starter drones, but also opens up opportunities for a new way of using drones for observation and inspection missions. This can lead to more research opportunities that would uncover the LMR feature potential and applicability in various domains.

TNO, an independent research organization in The Netherlands, took part in this research. Their Department of Digital Built Environment has shed an interest in spacial requirements of drones and their potential for general infrastructure inspections in The Netherlands. Close cooperation with TNO has opened the opportunity for more in-depth research of the LMR feature's technical implications, value, and interoperability in existing systems.

5.1 LMR Technical Implications

During the research of the LMR feature implementation and integration for a starter drone, a list of limitations have been made aware that should be considered. These limitations open up new research possibilities about finding the balance between the drone's navigational accuracy and camera quality for the best image/video results, or looking into new ways of creating an autonomous navigation system that would help the drone avoid obstacles and prevent costly collisions.

Further research could be conducted in collaboration with TNO to integrate this technology, in relation to the data processed through drones. This would include keeping the consistency of images collected throughout the years, which requires the images to be taken from the same physical place, which the LMR feature is great for. In addition to this, other considerations could be made about the physical feasibility of the drone being in a suitable position to capture the image of the inspected object, since the surrounding area might not be suitable for the drone to reach or even the camera quality could influence the data consistency. This research would be beneficial for the asset managers for which TNO conducts research, where maintaining the historical consistency of data collection is crucial.

5.2 Value of the Live Mission Recording

Further research can be done about the exact value the technology brings to the table for inspection missions. Potentially, the technology could be used to either inspect the entire bridge with video recordings with the purpose of detecting any emergence of cracks/corrosion or other types of damage, or to monitor the development of existing damages on the object via images. The detection of damages could be achieved by using Object Detection, and the analysis and assessment of the damages could be done by Image Processing. Various issues such as facilitating the processes of Object Detection and Image Processing should be considered. In addition to that, depending on the context, future research should look into the value Image Processing can bring when analyzing an image of a certain context. The exact implementation of these technologies in tandem with the LMR feature offers a wide possibility of outcomes and a great research focus.

Accurate and structured Image Processing and Object Detection are of particular interest to TNO. The data emerging from Image Processing might be useful for infrastructure assessment. In regards to Object Detection, it can help to find and classify damages easier than what is currently done with manual labor. The value of the technology may be seen not only for inspection companies, but also for TNO, who conduct research for assessing the service life of civil infrastructure for public asset managers, and even create models for portfolio asset analysis to prioritize renovations and replacements. TNO's research might also optimize the effectiveness and efficiency of the LMR technology, which could reduce the necessity of complicated manual labor of data processing, and the usage of structured data to link relevant properties of civil infrastructure. The Live Mission Recording feature can facilitate a more controlled environment for data collection and consistency, for example in periodic inspection missions where the progress of damages, such as cracks or corrosion, is monitored. With TNO's collaboration, expert opinions on the technology could be gathered, which could help to acquire very useful information from the people that would use the technology in the first place.

5.3 Interoperability

The Smart Applications REFerence (SAREF) [17] ontology could be used for the specific LMR technology to contribute to the Internet of Things (IoT). SAREF is a shared model of consensus that facilitates the matching of existing assets in the smart applications domain. A set of models have already been prepared that are related to this specific LMR feature implementation with the DJI Mini 2 drone. The models are focused on the data that is produced and gathered to acquire the state of the drone. The models also provide a firm foundation for further research of the technology's properties and their relations.

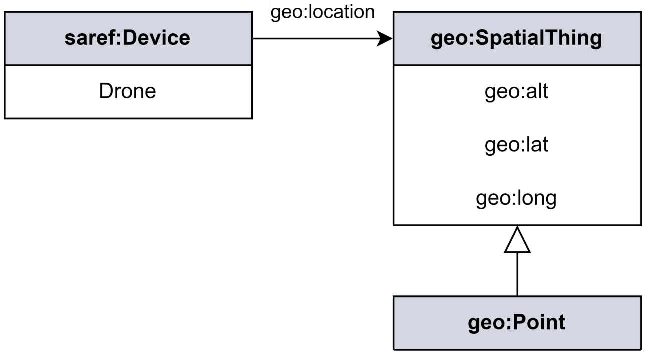


Figure 7 - Geolocation of the DJI Mini 2 drone.

Two ontologies, namely SAREF and GEO [18], have been combined to create a geolocation model of the DJI Mini 2 drone (See Figure 7). This model informs the altitude, latitude, and longitude data that is retrieved and used to position the drone at a specific point in the three-dimensional environment.

The following two models (See Appendix A.1 and Appendix A.2) inform about the data gathered and used to position the camera towards the inspected object using SAREF. Since the DJI Mini 2 drone has limited gimbal capabilities, allowing the camera to only move along the pitch axis, the drone's body has to compensate for the lack of yaw movement of the gimbal. Both of the models are relevant for the camera positioning during the state capture. The compass as a sensor is not made explicitly clear in any official DJI Mini 2 specification documentation, however, it has been assumed that it is implemented as an actual sensor according to the official DJI mobile SDK documentation of the Compass class, which measures the angle between the true north and drone's orientation thus providing us with the yaw value [15].

And lastly, another model using SAREF has been constructed about the battery's current charge (See Appendix A.3). Monitoring the battery's current charge level is essential in order to prevent the drone from crashing during mission execution. Of course, there are preventative measures already in place by DJI that make the drone autonomously fly back from the launching position by ascending to a certain height [16]. However, these measures are not always the best approach, since the drone may collide with an infrastructure while accending autonomously, for example when the drone is under the bridge deck. Thus, monitoring the battery level and implementing other types of safety features could prove to be more useful.

6 CONCLUSION

This paper presents the core concepts behind the Live Mission Recording feature and its implementation for the DJI Mini 2 drone. The limited functionalities present in the DJI Mini 2 forced us to discover another way of implementing autonomous navigation to reach a certain point in the physical environment, which is a crucial element of a functional LMR feature. Manual calculations of the bearing angle and distance between two coordinate points combined with the Virtual Sticks feature facilitated the implementation of the autonomous navigation system for the drone. This discovery combined with other ControlMission action elements lead to the integration and testing-by-parts of the LMR feature using the DJI Mini 2. The integration and testing yielded a successful result of the LMR feature's functionality for starter drones. The aftermath of the feature's integration for starter drones led to a list of limitations that should be considered when using the technology in the field.

In addition to the LMR feature's implementation and integration, this research opened new research opportunities for the LMR's feature usage in the field of inspection missions. TNO's interest in the technology's contribution to infrastructure inspection missions in The Netherlands offers a great collaboration opportunity to engage in a practical appliance of the technology's promising benefits.

REFERENCES

[1] van der Merwe, D., Burchfield, D. R., Witt, T. D., Price, K. P., & Sharda, A. (2020). Drones in agriculture. Advances in agronomy, 162, 1-30.

[2] Seo, J., Duque, L., & Wacker, J. (2018). Drone-enabled bridge inspection methodology and application. Automation in construction, 94, 112-126.

[3] Xu, F., & Muneyoshi, H. (2017). A case study of DJI, the top drone maker in the world. Kindai Manag. Rev, 5, 97-104.

[4] DJI Enterprise. (2020, December 18). Smart Inspection (Waypoint 2.0, Live Mission Recording, AI Spot-Check), Matrice 300RTK & H20 Series [Video]. Youtube. https://youtu.be/XYoBx1As6Sg

[5] DJI Enterprise. (n.d.). Matrice 300 RTK - Industrial grade mapping inspection drones https://enterprise.dji.com/matrice-300#:~:text=Live%20Mission%20Recording

[6] DJI Official. (n.d.). DJI Pilot - Download Center. https://www.dji.com/nl/downloads/djiapp/dji-pilot

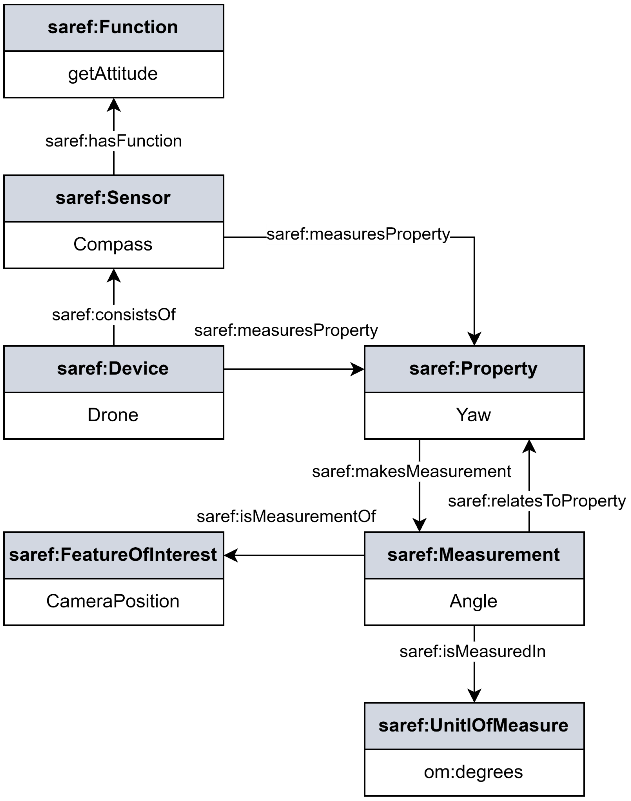
[7] Besada, J. A., Bergesio, L., Campana, I., Vaquero-Melchor, D., López-Araquistain, J., Bernardos, A. M., & Casar, J. R. (2018). Drone Mission Definition and Implementation for Automated Infrastructure Inspection Using Airborne Sensors. *Sensors*, *18*(4), 1170. https://doi.org/10.3390/s18041170

[8] J. A. Besada, A. M. Bernardos, L. Bergesio, D. Vaquero, I. Campaña and J. R. Casar, "Drones-as-a-service: A management architecture to provide mission planning, resource brokerage and operation support for fleets of drones," 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kyoto, Japan, 2019, pp. 931-936, doi: 10.1109/PERCOMW.2019.8730838.

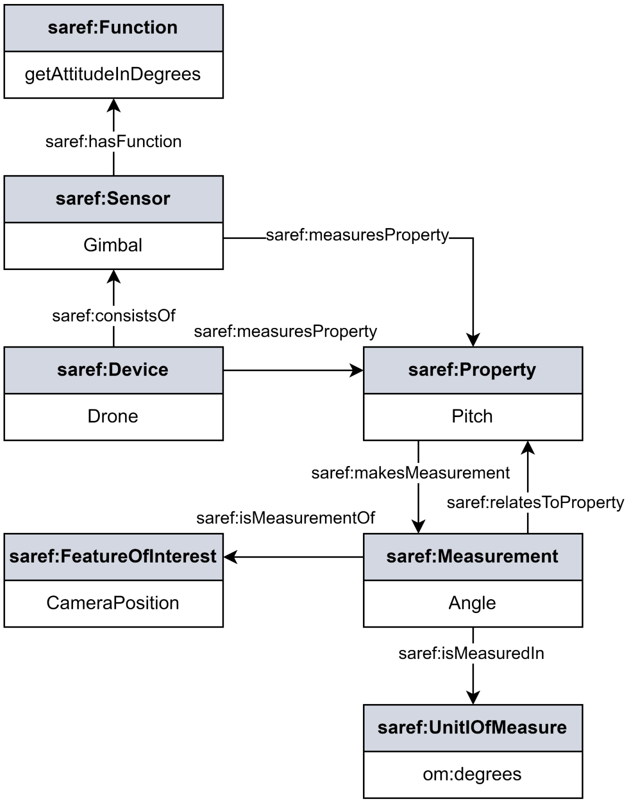
[9] Dji-Sdk. (n.d.). *Mavic mini does not support MissionAction's GoToAction action. Issue #606 - dji-sdk/Mobile-SDK-Android*. GitHub. https://github.com/dji-sdk/Mobile-SDK-Android/issues/606
[10] Dji-Sdk. (n.d.). *gotoAction Mini2. Issue #1091 - dji-sdk/Mobile-SDK-Android*. GitHub. https://github.com/dji-sdk/Mobile-SDK-Android/issues/1091
[11] Help - Litchi. (n.d.). https://flylitchi.com/help#waypoints-p3
[12] Nordin, N. A., Mustapa, N., & Satar, A. A. (2021). Ability of RTK-Based GPS Measurement Method in High Accuracy Work in Geomatics Study. Asian Journal of University Education, 17(4), 60-70.
[13] D1. (2021, February 17). RTK vs GPS Positioning. YouTube. Retrieved May 31, 2023, from https://youtu.be/1pM6FnCmfSk
[14] Chris Veness. (n.d.). *Calculate distance, bearing and more between Latitude/Longitude points*. Movable Type Scripts. http://www.movable-type.co.uk/scripts/latlong.html
[15] DJI Mobile SDK Documentation. (n.d.). https://developer.dji.com/api-reference/android-api/Components/Compass/DJICompass.html#djicompass_heading_inline
[16] DJI Forum, D. (n.d.). Unsafe low battery behavior! https://forum.dji.com/forum.php?mod=viewthread&tid=277793#:~:text=Low%20Battery%20RTH%20is%20triggered,battery%20level%20is%20extremely%20low.
[17] Ontology Documentation. (n.d.). https://ontology.tno.nl/logico
[18] SAREF: the Smart Applications REFerence ontology. (n.d.). https://saref.etsi.org/core/v3.1.1/

APPENDIX

A.1 Appendix A.1



A.2 Appendix A.2



A.3 Appendix A.3