INVESTIGACIÓN E INNOVACIÓN EN INGENIERÍAS

Software Tool for Drone Flight Management in Supervising Building Extern Façades

Herramienta Software para la Gestión de Vuelo de un Dron en la Supervisión de Fachadas Externas de Edificios

Mateo Barrera, **Bladimir Bacca Cortes**
Universidad del Valle, Colombia
Jaiver Castillo García
Universidad Autónoma de Occidente, Colombia

OPEN ACCESS

Recibido:
22/01/2024
Aceptado:
12/04/2024
Publicado:
04/05/2024

Correspondencia:
mateo.barrera@correounivalle.edu.co

DOI:
https://doi.org/10.17081/invinno.12.1.6667

Abstract
Objective: The main goal is to develop a software called DroneFaçade to manage the drone flight and generate an image mosaic of a building façade, allowing the continuous monitoring and registering the state of the façade through time. **Methodology:** The software DroneFaçade was developed considering three operational phases: GPS-based acquisition of waypoints, image-data acquisition flying the drone through the computed trajectory, and the offline mosaic computation. The waypoints are captured using GPS Exchange Format. Then, these waypoints are introduced into DroneFaçade, proper coverage of the façade is computed using the intrinsic camera parameters, the drone trajectory is computed, and the battery changing points are defined. Afterwards, the flight mission starts capturing image data the in the on-board Jetson-Nano development system running ROS. Next, the telemetry and image data are imported into a SQL-database system, and the mosaic computation is performed. At the end, DroneFaçade generates a report in PDF about the façade inspection. **Results:** To validate DronFaçade two different tests were performed: integration tests according with RUP methodology to check the functional requirements fulfillment, and a quantitative field test to check que quality of the mosaic computed. As a result of the quantitative test, the mosaic image computed by DronFaçade has low structural errors, preserves phase congruency and the magnitude of the image gradients, keeps high correlation and an appearance similarity of up 92.6% with respect to the real scene. **Conclusions:** DroneFaçade is a software tool which can be an option to register through time the building façades state and performing corrective actions to avoid high building deterioration thanks to the high quality image mosaics generated.

Keywords: Façade deterioration, UAV, mosaic image, ROS, image processing.

Resumen
Objetivo: Desarrollo de una herramienta software llamada DroneFaçade para la gestión de vuelo de drones y generar un mosaico de la fachada del edificio, permitiendo así la supervisión continua y el registro el estado de la fachada a lo largo del tiempo. **Metodología:** El software DroneFaçade fue desarrollado considerando tres fases operacionales: adquisición de way-points usando GPS, adquisición de imágenes volando el dron a lo largo de la trayectoria calculada, y cálculo del mosaico fuera de línea. Los way-pooints son capturados usando el formato GPS Exchange. Luego, estos way-points son introducidos en DroneFaçade, una adecuada cobertura es calculada usando los parámetros intrínsecos de la cámara, se calcula la trayectoria del dron y se definen los puntos de cambio de batería. Luego, la misión de vuelo inicia capturando las imágenes en el sistema de desarrollo Jetson-Nano a bordo que corre ROS. Luego, los datos de telemetría e imágenes son importados en una base de datos SQL, y se calcula el mosaico de la fachada inspeccionada. Al final, DroneFaçade genera un reporte en PDF acerca de la inspección de la fachada. **Resultados:** Para validar DroneFaçade dos diferentes pruebas fueron desarrolladas: pruebas de integración de acuerdo con la metodología RUP para chequear el cumplimiento de los requerimientos funcionales, y una prueba cuantitativa en campo para chequear la calidad del mosaico calculado. Como resultado de la prueba cuantitativa, la imagen de mosaico calculada por DroneFaçade tiene bajos errores estructurales, preserva la congruencia de fase y los gradientes de la imagen, mantiene una alta correlación y una similitud de apariencia de hasta el 92.6% con respecto a la escena real. **Conclusiones:** DroneFaçade es una herramienta software que puede ser una opción para registrar a lo largo del tiempo el estado de las fachadas de los edificios y realizar acciones correctivas para evitar altos deterioros en éstos gracias a los mosaicos generados de alta calidad.

Palabras claves: Deterioro de fachadas, UAV, imagen mosaico, ROS, procesamiento de imágenes.

Como citar (IEEE): M. Barrera, B. Bacca Cortes, J. Casrukki García, "Software Tool for Drone Flight Management in Supervising Building Extern Façades", Investigación e Innovación en Ingenierías, vol. 12, no. 1, pp. 83–101, 2024, doi: https://doi.org/10.17081/invinno.12.1.6667

Introduction

The façade is a good sign of the building deterioration according with the infrastructure norm NSR-10 [1]. The deterioration process commonly is due to aging, weather, and earthquakes. These phenomena cause façade and structural damage. The latter in most cases have effects on the building façade. In this context, early warnings about failures in the building façade can be used to trigger repairment a rehabilitation process in the building, increasing in this way the security and integrity of the building.

The façade inspection traditionally is performed using tachymetry in conjunction with manual aerial inspection [2]. These methods are expensive, take long periods of time, and they have a subjective component introduced by human-based concepts. In most cases, these methods do not handle historical data. Then, in case this feature is needed, façade inspection at regular periods of time could be used to implement preventive maintenance plans. It is also common to perform manual inspection of façades, which does not guarantee access to all the available surfaces due safety reasons. As a result, several façade damages are not detected.

Nowadays, unmanned aerial vehicles (UAV) are high maneuverable, they are available in different sizes, they are more affordable, they have many sensors on-board and due these features they can access to all the façade surfaces [2, 3]. These properties are desirable to be used as a proper tool in supervising extern building façades, and they were considered in this work to propose a suitable software application that includes important aspects such as: the mission planning, the mission configuration, the mission execution, the data persistence, and data processing can be performed within one compact environment.

This paper is structured as follows: related works are described in the first section, the two following sections will describe the design and implementation of DronFaçade, results and discussion are presented in section four, and section five shows the conclusions.

Related Works

In contemporary times, Unmanned Aerial Vehicles (UAVs) have become widely employed as versatile tools across various applications. In this work, their application focuses on the supervision of building facades. A comprehensive comparison of the relevant literature, as outlined in Table 1, is performed, considering several criteria. These criteria include the system application, UAV platform, path planning methodology, inspection sensors utilized, graphical user interface (GUI) employed, type of data analysis conducted, data processing methods employed, and whether the respective study generates an inspection report. Further analysis of the table is also provided subsequently.

Table 1. Related works.

Ref.	App.	UAV	Path P.	Sensors	GUI	Analysis	Proc. Data	Report
[4]	Remote sensing, Energy efficiency	*AscTec Pelican*	Radio-controlled	*Thermoteknix MicroCAM 384M.*	*AscTec Atom board* and ROS	Multispectral (RGB e IR)	*FLIR tool,* ENVI for ITT	No
[5]	2D reconstruction	*AscTec NEO*	GPS Navigation	Monocular camera 640×480	ROS + Matlab	Mosaics	OpenCV	No

			controled by ROS					
[6]	Inspection of civil structures	*Parrot AR Drone*	Semi-autonomous	Monocular camara	Linux + *Aerostack*	Surface imperfection recognition	Frequency Histogram of Connected Elements (FHCE)	Yes
[7]	Visual inspection and 3D reconstruction	*DJI Phantom 4 pro V2*	RC assisted by DJI	Monocular camera 20 MP	*DJI flight control*	Ortho-mosaics and 3D reconstruction	*Agisoft PhotoScan*	No
[8]	Façades visual inspection	*DJI Spark*	RC assisted by DJI	Monocular camera 3968×2976	*QGIS*	GIS 2D	*2D Reconstruction with OpenCV*	No
[9]	BIM model of external façades	*DJI Phathom 4 RTK*	GPS Navigation + DJI RTK	Monocular camera FC6310R	*DJI Flight control*	BIM Model	*Crack detection with RCNN and Python*	No
[10]	BIM models + time (4D)	*3DR IRIS+* Custom drone	*Ardupilot*	*Sony alpha 5100*	*Drone code*	BIM model	*BIM model to path planning*	No
[11]	Automatic building façade exploration	*DJI Phathom 4 RTK*	GPS Navigation using BIM model	Monocular camera	Custom Python based path planning	Path following and façade coverage	*DJI GS Pro*	No
[12]	Topographic survey	*DJI Phantom 3* and Mavic	GPS navigation	Monocular camera	*DJI, ArGIS, PIX4D*	Orthomosaics and planimetry	ArcGIS	Yes
[13]	UAV navigation and mapping	*Quadropter*	Assisted RC (Pixhawk and	Monocular camera *SJ4000 WIFI*	PC Linux, *Mission*	Orthophotography	*OpenPano*	No

			Raspberry)		*Planner and Hermes*			
[14]	UAV navigation and mapping	*Albris, Phantom 3*	Assisted RC with Pixhawk	Monocular camera	DJI	3D models	Agisoft Photoscan, Pix4D, Photomodeler Micmac by IGN	No
[15]	Inspection of towers, bridges and wide surfaces	Mavic Pro	Assisted RC with GPS	Monocular camera	DJI	No	ArcGis, Pix4D	Yes
[16]	Inspection of wide surfaces and façades	Mavic 3	Assisted RC	Monocular camera	DJI	No	ArcGis, Pix4D	Yes

Source: Own elaboration

Observing Table 1, the works reviewed are focused on applications such as remote sensing [4], 2D or 3D reconstruction [5, 7] , civil structures inspection [6, 8, 11, 15, 16] navigation and mapping [13, 14] and building BIM (Building Information Modeling) models [9 10] . It is worth noting that the inspection applications as well as the 2D/3D reconstruction and building BIM models are the most usual, and at the end these applications give results focused on offering a traceability of the infrastructure state.

Other important aspects to consider in these applications are the UAV platform and the path planning method used. Table 1 shows that all the reviewed works, including [13] and [15], have high performance drones to increase the autonomy and the payload to carry on more sensors. In addition, most of works in Table 1 have assisted navigation which means the presence of a human pilot. This is understandable since flying drones is a task subject to local regulations, and it could be mandatory around civil structures. However, works such as [5, 6] and [9-11] propose autonomous and semi-autonomous navigation methods based on previously extracted geometric constraints, i.e. BIM models.

Table 1 also considers the sensor type used. Due payload reasons monocular RGB camera is the most popular sensor, since using this unique sensor and combining the data from the drone telemetry data analysis such as building mosaics [5], orthomosaics [7, 12] orthophography [13], 2D/3D reconstructions [7, 8, 14] and planimetry [12] can be performed offline using specialized software. However, in [4] multispectral analysis is performed including thermal data to the building inspection. This is required since the goal of [4] is to evaluate building energy efficiency. Most of these generated data by the drones are processed using commercial software as shows Table 1. But, specific data processing can also found, for instance in [6] authors propose the FHCE method to detect imperfections in civil structures using RGB cameras; in [8] authors use OpenCV and GIS tools to generate

georeferenced image mosaics; in [9] authors detect cracks in façades using deep learning methods, and map them into BIM models to guarantee traceability of the building information model; and, in [10] the BIM model is processed automatically to generate obstacle-free path for the drone and define the waypoints where to perform data acquisition.

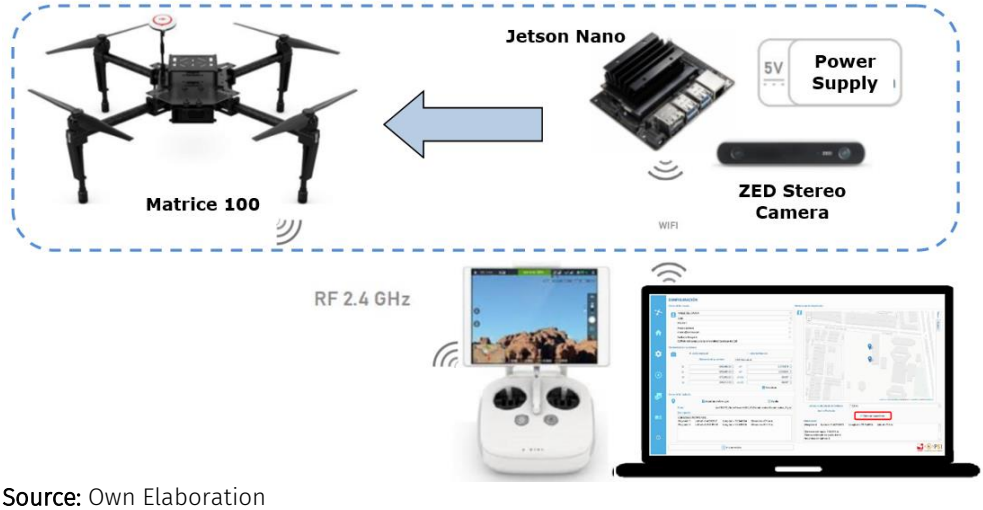
Human interaction with the drone and the application functionality is very important in inspection tasks. This interaction is performed through a GUI, but most works in Table 1 uses the UAV platform manufacturer GUI which limits users doing additional tasks such as data preprocessing, autonomous flight, sensor integration and data communication with the base station. However, in [4] and [5] ROS based GUI are developed which allows to upgrade the application functionality using a frameworks as ROS; in [6] and [13] mission planners based on Linux PCs are developed; and, in [10] and [11] authors have developed custom GUIs to embed the data processing in those works. In this way, these authors maintain software independence with respect to the manufacturer, and it is more feasible to develop future functionalities to applications developed.

In the context of civil structures inspection, it is important to have a report of this inspection that includes all relevant data of the mission, sensors, and data processing results. Table 1 shows that commercial solutions such as [12, 15, 16] include this feature. Also, in [6] authors that have developed a custom GUI have considered this feature. Report generation is an important feature to consider in inspection software tools, and better if this feature is integrated in the same working environment.

Material and methods

Figure 1 shows the proposed system, where a Matrice100 drone was used to carrying on the image acquisition and preprocessing system (Jetson Nano + ZED stereo camera + power supply). The Matrice100 flight controller has a RF communication link with the pilot handheld, and the Jetson Nano is connected to the DronFaçade software tool using WiFI. In addition, the Jetson Nano has a serial link with the flight controller to handle the mission planning and execution. It is worth noting that the Jetson Nano middleware was developed in ROS (Robotic Operating System) [17].

Figure 1. System structure proposed in DronFaçade.



Source: Own Elaboration

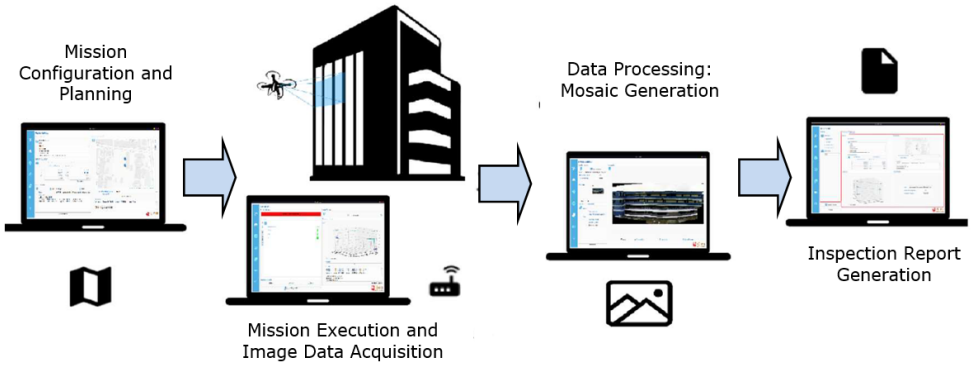
Design of DroneFaçade

DroneFaçade is a software tool which was developed using RUP (Rational Unified Process) software engineering methodology [18]. Basically, it includes the following outcomes: functional and non-functional requirements, conceptual diagram, real use cases, sequence diagrams, class diagram, relational diagram, and test reports. However, due space reasons the DroneFaçade design will include functional and non-functional requirements, conceptual, class, and relational diagrams.

To start with, considering the analysis of Table 1 described above the following functional requirements were considered in the DroneFaçade design phase: the GUI must allow the mission and sensors configuration parameters; the GUI has to show the drone trajectory and telemetry; DroneFaçade has to handle the communication with the UAV flight controller; DroneFaçade has to plan the drone trajectory considering the drone autonomy; DroneFaçade has to implement commands such as pausing, stopping, resuming and executing the mission; DroneFaçade has to coordinate the data acquisition to store the imaging record; DroneFaçade has to handle the data storage into a SQL database; DroneFaçade has to coordinate the data download from the drone into the SQL database; DroneFaçade has to compute the resulting mosaic and be able of exporting it; DroneFaçade has to capture all inspection data needed to generate a report; and DroneFaçade has to export the report document in a PDF file.

In addition, DroneFaçade was developed considering the following non-functional requirements: it uses OpenCV 4.1 to analyze the input images and generate the resulting mosaic, ROS melodic is needed to be installed into the Jetson Nano embedded system to ensure the data acquisition; PyQT 5 is used to build the GUI; the DJI_SDK ROS package to communicate the Jetson Nano with the drone flight controller; and Python 3.6.9 was used to develop the DroneFaçade logic.

Figure 2. DroneFaçade concept diagram.

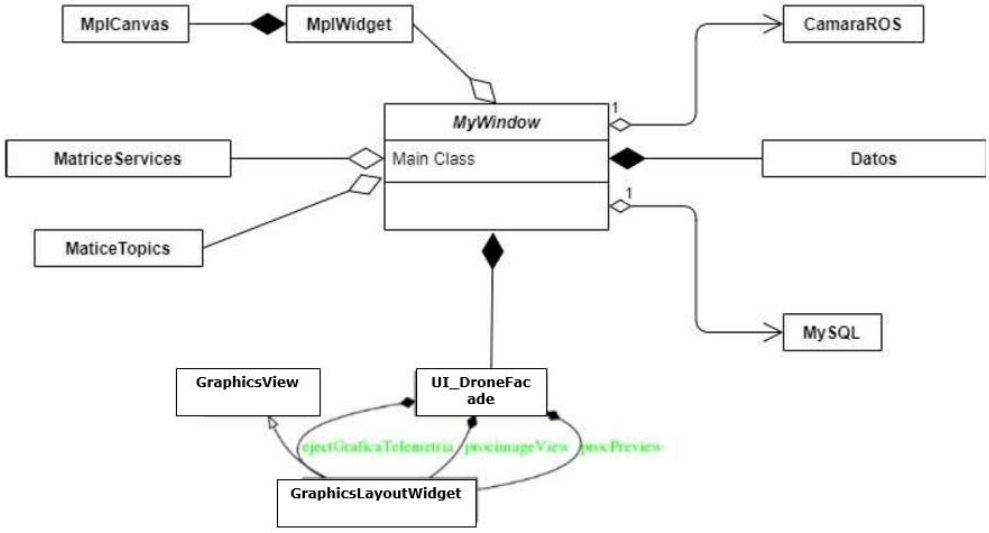


Source: Own Elaboration

From the user point of view and considering the DroneFaçade functional requirements, Figure 2 shows the concept diagram of how DroneFaçade works over all the mission phases. First, user must configure and plan the mission, this includes user identification, sensor configuration, introducing trajectory waypoints and trajectory computation. Second, mission execution and data acquisition which means a continuous communication with the flight controller, telemetry data reading and plotting, control of the flight mission, and image acquisition. Third, data processing which consists of importing the image data into the SQL database, configuring the mosaic generation module, computing the mosaic and visualize it into the DroneFaçade GUI. Finally, if user is satisfied with the mosaic result, gathering the data from the SQL database and generate the PDF file where the mission report is stored.

The simplified class diagram of DroneFaçade is shown in Figure 3. The class MyWindow starts the DroneFaçade application. All GUI components and logic of DroneFaçade is in charge of UI_DroneFace, GraphicsView and GraphicsLayoutWidget classes. The communication with the drone flight controller is performed by MatriceServices and MatriceTopics classes. The CameraROS class handles the image data acquisition. All data gathered along the mission is handled by the Datos class. These generated data is stored in a SQL database using the MySQL class. And, the MplWidget and MplCanvas handle all the plotting resources in DroneFaçade.

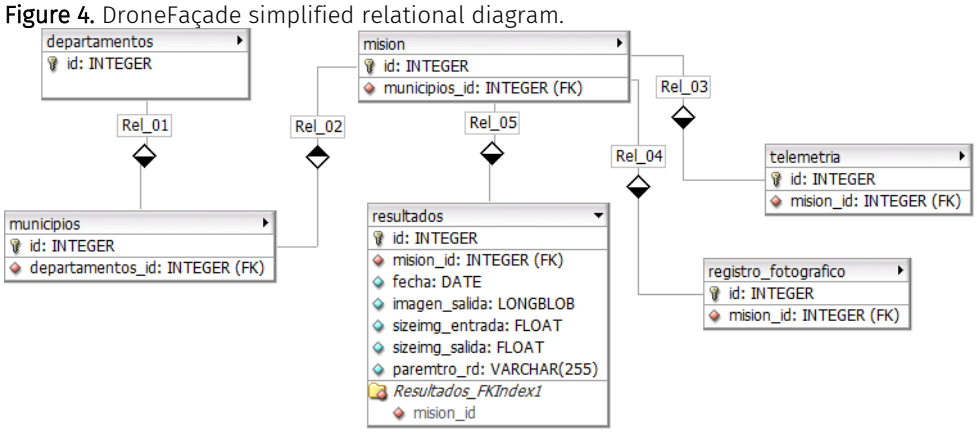
Figure 3. DroneFaçade class diagram.



Source: Own Elaboration

DroneFaçade uses a SQL database to store all data generated by the mission and to guarantee the traceability of inspections. Figure 4 shows the simplified relational model, where the main entity is the table mision. It stores the user identification, sensor calibration, waypoints, façade dimensions and estimated flying time. The coarse location of the façade is handled by entities departamentos and municipios. All sensor data generated in the mission is stored in entities telemetria and registro_fotografico. Finally, the entity resultados summarizes the data processing results to build the report document.

Figure 4. DroneFaçade simplified relational diagram.

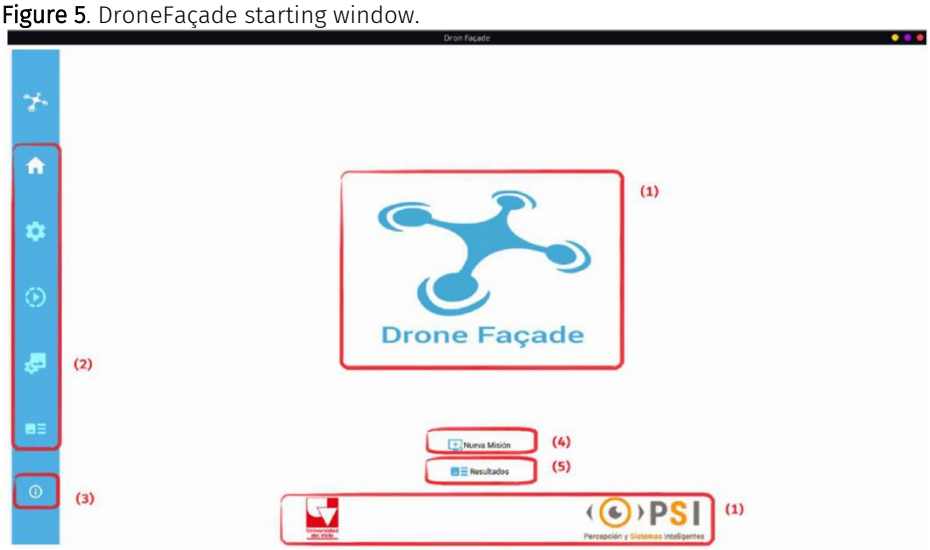


Source: Own Elaboration

Implementation of DroneFaçade

DroneFaçade was implemented considering 4 modules namely configuration, execution, data processing and report generation modules. Figure 5 shows the starting GUI window of DroneFaçade, where users can access to the modules (2), getting help (3), start a new mission (4) and generate a report of already processed missions (5).

Figure 5. DroneFaçade starting window.

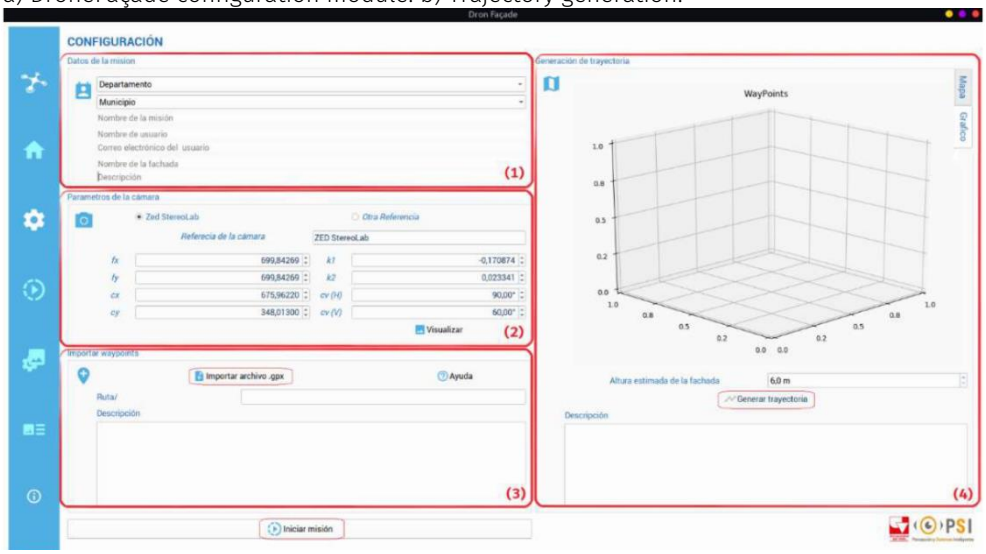


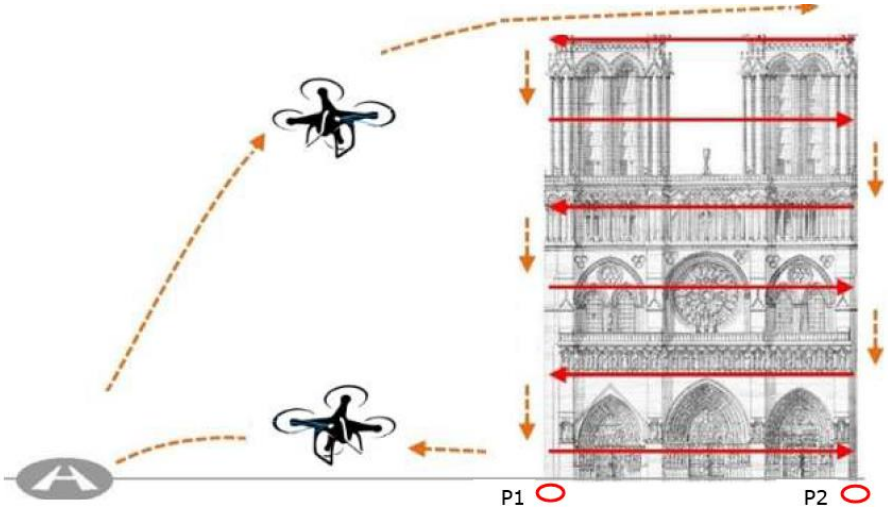
Source: Own Elaboration

To start with, users must go to the configuration module shown in Figure 6a when starting a new mission. Here, the user identification information, and address of the façade to inspect are introduced in (1). The main sensor is a ZED camera which intrinsic and extrinsic parameters are captured from the device itself, or in case of using other camera these parameters must be introduced manually (2). In (3) the main waypoints are imported using GPX exchange format. These waypoints must be taken on the floor, facing the façade, and at each façade corner if any (i.e. P1 and P2 points in Figure 6b). In this way, these waypoints can be considered as the limits of the façade and use them to get the total coverage area.

Figure 6. a) DroneFaçade configuration module. b) Trajectory generation.

a)





b)
Source: Own Elaboration

Then, the next step is computing the vision sensor coverage projected on the façade. Using the camera calibration parameters and Equations (1) and (2) the width and height of the camera coverage can be computed.

$$X_{width} = 2 * z * tan\left(\frac{FOV_H}{2}\right) * Of \quad (1)$$

$$Y_{height} = 2 * z * tan\left(\frac{FOV_V}{2}\right) * Of \quad (2)$$

Where, z is the distance between the drone and the façade, FOVH and FOVV are the camera horizontal and vertical field of view, and Of is the overlap factor which in this case is set to 50%. Using these horizontal and vertical distances, a point cloud of waypoints is computed, and an initial drone trajectory is generated starting in the façade upper-right corner. Then, going to the left and performing an up to bottom motion as shown in Figure 6b. Now, this trajectory does not include mandatory pauses in the mission execution due battery rechanges. To do so, considering a constant velocity and Equation (3), the autonomy traveled distance can be estimated over the computed trajectory, and mandatory waypoints to rechange the better are introduced [19].

$$T_{flight} = \left(\frac{Bc * Bsf}{1000 * \frac{UAV_W * PbKg}{V_{Bat}}}\right)\frac{60}{Sf} \quad (3)$$

Where, Bc is the battery capacity, Bsf is the battery safety factor, UAVW is the drone weight, PbKg is the drone power by Kg, VBat is the battery voltage, and Sf is the time safety factor. It is common to setting up Bsf between 0.4 and 0.7, and Sf = 1.5 to ensure enough time to returning the drone to a safe position.

The next step is to execute the flight mission, Figure 7 shows the GUI for the execution module. Here, four parts are shown: the mission status (1), the drone status (2), the trajectory performed by the drone (3) and the mission controls (4). Once all waypoints of the drone trajectory are computed, the following parameters are sent using a serial connection to the DJI flight controller: the velocity range, the linear drone velocity, the action to perform when the trajectory is finished, the yaw mode, the trace mode between waypoints, the action in case of loss the communication, and the list of waypoints. Afterwards, the mission status (1) is enabled to show the following states: DJI flight controller connected, taking off, landing, on land, finished and error. Each one of these states are shown with different colors.

Figure 7. DroneFaçade execution module.

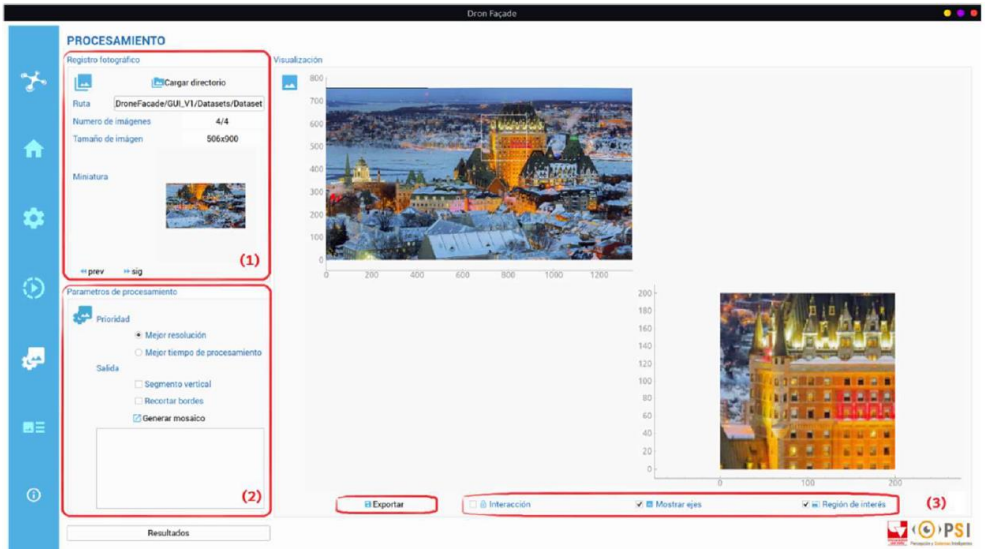
Source: Own Elaboration.

At the moment of starting the mission using the control panel (4), two main processes are initiated in the main on-board controller (Jetson Nano): first, a security procedure in case of dangerous loss of satellite communications for the GPS signal, and second the telemetry and image data acquisition. The latter is shown in the drone status (2) in Figure 7. Also, the current drone trajectory and status messages are shown in (3).

In the control panel (4), users also can pause and stop the mission execution. When the mission is paused, the drone hovers and the image data acquisition is suspended. In this state, the drone can also be manually controlled if it is needed.

When the flight mission finishes, the drone is on the land and the next step is downloading the telemetry and image data, storing them in the local database and computing the image mosaic. This is done in the processing module shown in Figure 8. Here, the first step (1) is downloading the images from the Jetson nano, registering in the database, and storing the local hard drive. Then, parameters as the destination path are needed to do this. Afterwards, to compute the image mosaic (2), the following parameters are needed to set up: optimize the image mosaic computation considering processing time or image resolution. Then, the resulting image mosaic is shown in (3) where the user can navigate over the big picture using the mouse and zoom-in / zoom-out controls. Also, this image mosaic can be exported to the user local disc.

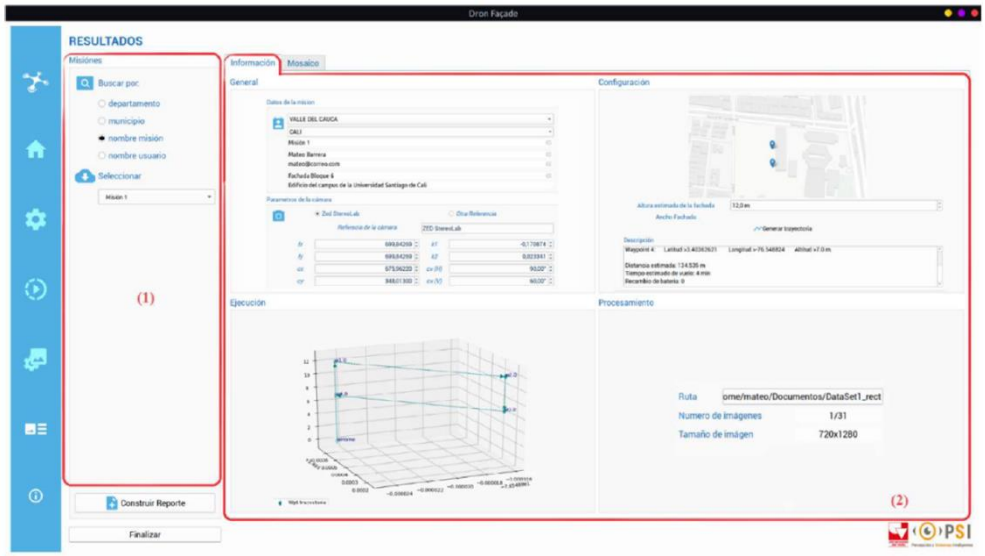
Figure 8. DroneFaçade data processing.



Source: Own Elaboration

Finally, the Figure 9 shows the GUI of the report module. Where, users can generate a PDF file that includes the mission information. To do so, users can search in the database by the mission location or name as shown in (1). Then, DroneFaçade loads all relevant data of the mission selected such as location, name of the user, contact information, camera calibration parameters, waypoints placed in a map, final trajectory executed, number of images captured and path where they are stored as shown in (2). Then, the report can be built using a PDF template and this file is stored in the user local hard drive.

Figure 9. DroneFaçade report module.



Source: Own Elaboration

Performance evaluation

The main goal of DroneFaçade is to perform drone flight management, building a façade mosaic on regular basis and registering the state of the façade through time. Then, it is important to measure the accuracy of mosaic generated by DroneFaçade, since this outcome will be stored in the database for further façade evaluations. For these reasons, in this work, the mosaic generated by DroneFaçade was compared with common known mosaic building tools such as Image Composite

Editor of Microsoft [20] and PTGui [21]. The comparison with these tools were performed in the context of the following metrics:

- RMSE (Root Mean Square Error): It measures the amount of change in a pixel given an image processing method. Equation 4 shows how it is computed. Where, M and N are the width and height of the images, I and K are the predicted and reference images.

$$RMSE = \sqrt{\frac{1}{M*N}\sum_{i=0,j=0}^{M-1,N-1}[I(i,j) - K(i,j)]^2} \quad (4)$$

- SSIM (Structural Similar Index Measure) [22]: It measures the amount of degradation caused by the image processing method. Equation 5 shows how it is computed. Where, μx and μy are the average of a window centered in x, y; σx and σy are the variances of a window centered in x, y; σxy is the covariance of xx and yy, c1 and c2 are two variables that stabilize the division with a weak denominator.

$$SSSIM(x,y) = \frac{(2\mu_x\mu_y+c_1)*(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu y^2+c_1)*(\sigma_x^2+\sigma_y^2+c_2)} \quad (5)$$

- UIQ (Universal Image Quality index) [23]: It considers three factors to measure the image distortion such as loss of correlation, luminance distortion and contrast distortion. Equation 6 shows how it is computed. Where, the first term measures the linear correlation between x and y images; the second term measures the similarity of the mean luminance between x and y images; and the third term measures the contrast similarity between images.

$$Q = \frac{\sigma_{xy}}{\sigma_x\sigma_y} * \frac{2\overline{xy}}{\bar{x}^2+\bar{y}^2} * \frac{2\sigma_x\sigma_y}{\sigma_x^2+\sigma_y^2} \quad (6)$$

- OpenAI CLIP [24]: It is an AI (artificial intelligence) method which can be used to measure the similarity between images. This method extracts deep features in both images. Afterwards, these features are compared considering contrast and luminance conditions.

Results

DroneFaçade software tool was tested in two different scenarios: integration tests considering the RUP software engineering methodology, and field tests.

The integration tests have the aim of validating the functional requirements of the software tool. These tests were performed considering the RUP software engineering methodology and the functionalities of the DroneFaçade modules such as the configuration module functionalities, the execution module functionalities, and the reports module functionalities. At each test, the following aspects were considered: functional requirements involved in the test, hardware and software required, test goal, input data for the test, procedure performed, expected results, and results achieved. As a result, all the functional requirements were satisfied. For space reasons, these tests are not included in this paper.

The quantitative field test was performed using the DJI UAV Matrice 100 as experimentation platform observed in the Figure 10. This platform also includes: the flight control board, the on-board GPS device, the Jetson Nano embedded system and the ZED stereo camera. It is worth noting that the Jetson Nano is running Ubuntu 18.04 and ROS Melodic. Considering Figure 1, the Jetson Nano handles a wireless data link with DroneFaçade to monitor and control the UAV

Matrice 100. It is important to remember that a redundant manual control is always present as depicted in Figure 1.

Figure 10. Experimentation platform.

Source: Own Elaboration.

The quantitative field test was performed in the Universidad Santiago de Cali campus, specifically at building No. 6 as depicted in Figure 11. At the moment of the tests, the weather conditions were evaluated using the UAV Forecast application [25], a sample of the most important weather conditions evaluated at each flight are listed as follows:

- Temperature: 28.3°C
- Wind speed: 0.9 m/s
- Wind gust speed: 4 m/s
- Chance of precipitation: 12%
- Cloudiness: 75%
- Visibility: 15 Km
- Satellites in range: 17
- Kp (solar activity): 1
- Evaluation of weather conditions: Good to flight.

Figure 11. Building No. 6, Universidad de Santiago de Cali, Valle del Cauca.

Source: Own Elaboration

Then, having the suitable weather conditions to fly, five (5) drone missions were performed of the façade shown in Figure 11. At each mission, a set of images was captured, and the corresponding mosaic image was computed using the

DroneFaçade software tool. Also, these sets of images were introduced in the Composite Editor of Microsoft and PTGui software tools. Afterwards, the corresponding mosaics were obtained using these software tools. Furthermore, the similarity image measures described above were computed the Python package called image-similarity-measures 0.3.6 [26, 27]. Therefore, Table 2 shows the mean results of applying these similarity measures.

Table 2. Similarity measures of mosaics acquired.

Similarity Measures			DroneFaçade Vs. Comp. Editor	DroneFaçade Vs. PTGui
Metric	Min. Value	Max. Value		
SSIM	-1	1	0.778	0.679
RMSE	0	> 0	0.01892	0.0239
UIQ	-1	1	0.0351	0.0014
OpenAI	0%	100%	92.62%	86.81%

Source: Own Elaboration

Table 2 shows two columns called minimum and maximum value. These columns show the range of values in which the metric vary. In case of SSIM a desirable value is 1 showing perfect structural similarity; the RMSE desirable value is close to zero; in case of UIQ the ideal value is 1, which is achieved in case the images are identical, values close to zero are typical; and, desirable values for OpenAI are close to 100%.

Considering Table 2, DroneFaçade software tool computes a mosaic image which has low RMSE values; it has low structural errors in the reconstruction considering the SSIM values; it has a correlation with respect to the real scene bigger than the average (UIQ values), it has luminance and contrast distortion are less than the average (UIQ values); and it has high appearance similarity with respect to the real scene considering the OpenAI values.

Considering the related works reviewed and the results presented in this section, it can be said that DroneFaçade is software tool that can be used in building appearance-based BIM models of civil structures inspection. In this context, and given the Colombia local regulations a human pilot is mandatory; however, DroneFaçade performed a semi-autonomous navigation in the field, it means that at the very beginning the Matrix 100 drone was previously placed in the initial pose to begin the mission. Afterwards, the drone was commanded by the DroneFaçade software tool until finish the mission. Which is a common behavior found in the works reviewed.

At this moment, DroneFaçade software tool can capture images, compute a mosaic given those images and store them in a SQL database for further revision by experts. The related works use similar sensor set or a combination of sensors to implement other interesting applications doing an online or offline data processing. In this work, a ZED stereo camera was used which can be used to acquire visual and 3D data.

Finally, DroneFaçade GUI was custom developed as most of the related works reviewed. Since, it is wise to separate the human interaction of the framework at the moment of developing this kind of applications. Then, DroneFaçade software tool can be improved adding image processing modules for crack detection, performing multispectral analysis, or merging visual data with structural information in BIM models.

Conclusion

This paper presented the development of DroneFaçade, a software tool to manage the flight of a drone, capturing images from a building façade, computing its mosaic image and provide persistence of all mission data into SQL database to register the state of the façade through time. DroneFaçade was developed in four main modules namely configuration, mission execution, data processing and reports. The configuration module includes the parameterization of the software tool and mission, and the trajectory generation considering the UAV autonomy. The mission execution module connects with the Jetson Nano in the UAV using a WIFI link; then, mission plan is executed considering the temporal stops for battery changes. At the end of the UAV flight, the data processing module saves the images into the SQL database and start computing the image mosaic. Finally, the reports module generates a PDF file including all the mission data.

DroneFaçade software tool was validated performing integration tests to check if the functional requirements were satisfied, and a quantitative field test to measure the accuracy of the mosaic image generated by DroneFaçade. After checking the weather conditions of the flight mission, 5 different flights were accomplished in the Universidad Santiago de Cali campus, specifically in front of the building No. 6. The corresponding images were stored in the SQL database and they were used to compute the image mosaics using the DroneFaçade software tool. Additional image mosaics were computed using Microsoft Composite Editor and PTGui popular software tools. Then, using the Python package called image-similarity-measures 0.3.6 the following image similarity measures were computed RMSE, SSIM, UIQ and OpenAI.

As a result, DroneFaçade image mosaics had low RMSE values up 0.0239; low structural errors in the image mosaic since the SSIM value was up 0.778; observing the UIQ values, DroneFaçade image mosaics have correlation with respect to the real scene, and the luminance and contrast distortion are less than average; in addition, DroneFaçade image mosaics have high appearance similarity with respect to the real scene considering the OpenAI value of 92.6%. Then, DroneFaçade software tool has bee proved as an interesting option to perform façades inspection of civil structures using UAV drones. Future works include adding image processing capabilities for 3D reconstruction, and since DroneFaçade uses a SQL database to register data through time, 4D BIM models can be implemented.

Acknowledgements

This work was funded by the research project “Multi-Sensory Integration System for Classification of Surface Faults in Flexible Flooring” by the Universidad del Valle with CI 21034. Also, this work appreciates the institutional collaboration with the Universidad Santiago de Cali, specifically the UAV robotics lab.

Bibliographic References

[1] Ministerio-Ambiente-Vivienda-Desarrollo-Territorial, "REGLAMENTO COLOMBIANO DE CONSTRUCCIÓN SISMO RESISTENTE NSR-10," Bogotá D.C., 2010.

[2] S. Emelianov, A. Bulgakow, and D. Sayfeddine, "Aerial laser inspection of buildings facades using quadrotor," Procedia Eng., vol. 85, pp. 140–146, 2014.

[3] R. R. S. de Melo, D. B. Costa, J. S. Álvares, and J. Irizarry, "Applicability of unmanned aerial system (UAS) for safety inspection on construction sites," Saf. Sci., vol. 98, pp. 174–185, 2017.

[4] A. Carrio et al., "UBRISTES: UAV-Based Building Rehabilitation with Visible and Thermal Infrared Remote Sensing," Springer, Cham, 2016, pp. 245–256.

[5] S. S. Mansouri, C. Kanellakis, G. Georgoulas, D. Kominiak, T. Gustafsson, and G. Nikolakopoulos, "2D visual area coverage and path planning coupled with camera footprints," Control Eng. Pract., vol. 75, no. January, pp. 1–16, 2018.

[6] P. F. Amar, "A Human-Robot Cooperation System For Surface Inspection Aerial Missions," E.T.S. de Ingenieros Informáticos (UPM), 2019.

[7] R. D. Ballesteros Ruiz, A. C. Lordsleem Júnior, and J. H. Aquino Rocha, "Inspeção de fachadas com Veículos Aéreos Não Tripulados (VANT): estudo exploratório," Rev. ALCONPAT, vol. 11, no. 1, pp. 88–104, 2021.

[8] K. Chen, G. Reichard, A. Akanmu, and X. Xu, "Geo-registering UAV-captured close-range images to GIS-based spatial model for building façade inspections," Autom. Constr., vol. 122, p. 103503, Feb. 2021.

[9] Y. Tan, G. Li, R. Cai, J. Ma, and M. Wang, "Mapping and modelling defect data from UAV captured images to BIM for building external wall inspection," Autom. Constr., vol. 139, p. 104284, Jul. 2022.

[10] H. Freimuth and M. König, "Planning and executing construction inspections with unmanned aerial vehicles," Autom. Constr., vol. 96, pp. 540–553, Dec. 2018.

[11] Y. Tan, S. Li, H. Liu, P. Chen, and Z. Zhou, "Automatic inspection data collection of building surface based on BIM and UAV," Autom. Constr., vol. 131, p. 103881, Nov. 2021.

[12] Visual Drone, "Visual Drone - Topografía y fotogrametría," 2022. [Online]. Available: https://visualdrone.co/topografia-y-fotogrametria/. [Accessed: 26-Dec-2019].

[13] C. mauricio Gualán llanos and C. D. Campodónico Durango, "Implementación de un sistema de navegación y mapeo en vehículos aéreos no tripulados utilizando dispositivos embebidos y software libre," Escuela superior politécnica del litoral, 2017.

[14] A. Murtiyoso, M. Koehl, P. Grussenmeyer, and T. Freville, "Acquisition and processing protocols for uav images: 3d modeling of historical buildings using photogrammetry," ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci., vol. 4, no. 2W2, pp. 163–170, 2017.

[15] Air Photo Colombia S.A.S, “Air Photo Colombia,” 2022. [Online]. Available: https://www.airphotocolombia.com. [Accessed: 29-Aug-2022].

[16] Drones Sky Zoom, “Sky Zoom fotografía y video aéreo,” 2022. [Online]. Available: https://www.dronesskyzoom.com/. [Accessed: 29-Aug-2022].

[17] Open-Robotics, “ROS: Home,” 2022. [Online]. Available: https://www.ros.org/. [Accessed: 06-Sep-2022].

[18] P. Kruchten, The Rational Unified Process: An Introduction, 3rd ed. Addison-Wesley Professional, 2003.

[19] W. Richardson, “How to estimate the maximum and recommended flight times of a UAS, UAV, or Drone System,” Louisiana, 2015.

[20] Microsoft, “Image Composite Editor,” 2023. [Online]. Available: https://www.microsoft.com/en-us/research/project/image-composite-editor/. [Accessed: 01-Jun-2023].

[21] New-House-Internet-Services, “PTGui,” 2023. [Online]. Available: https://ptgui.com/about.html. [Accessed: 01-Jun-2023].

[22] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image Quality Assessment: From Error Visibility to Structural Similarity,” IEEE Trans. Image Process., vol. 13, no. 4, pp. 600–612, Apr. 2004.

[23] Zhou Wang and A. C. Bovik, “A universal image quality index,” IEEE Signal Process. Lett., vol. 9, no. 3, pp. 81–84, Mar. 2002.

[24] A. Radford et al., “Learning Transferable Visual Models From Natural Language Supervision,” Feb. 2021.

[25] M. Lloyd, “UAV Forecast,” 2023. [Online]. Available: https://www.uavforecast.com/. [Accessed: 01-Jun-2023].

[26] A. Maluf Blanco y E. S. Michelena Fernández, "Estudio de factores que afectan la integración de los sistemas de gestión en el Centro Nacional de Biopreparados, Cuba", Investigación e Innovación en Ingenierías, vol. 8, n.º 1, pp. 37–53, ene. 2020. DOI: https://doi.org/10.17081/invinno.8.1.3592

[27] M. U. Müller, N. Ekhtiari, R. M. Almeida, and C. Rieke, “SUPER-RESOLUTION OF MULTISPECTRAL SATELLITE IMAGES USING CONVOLUTIONAL NEURAL NETWORKS,” ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci., vol. V-1–2020, pp. 33–40, Aug. 2020.