Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 58-9 (2024) 281–286

Using Visual Odometry to Determine the Position of a UAV

Marco Pintér * **Jiří Janoušek** * **Jan Klouda** * **Petr Marcoň** *

* *Brno University of Technology, Dept. of Theoretical and Experimental Electrical Engineering, Faculty of Electrical Engineering, Brno University of Technology, Technicka 3082/12, Brno, 616 00 (e-mail: xpinte06@vut.cz, Jiri.Janousek@vut.cz, 214380@vut.cz, marcon@vut.cz)*

Abstract: We discuss the main concepts and problems related to navigating an unmanned aerial vehicle (UAV) in a three-dimensional space via visual odometry. In the nearest future, GMSS-free UAV navigation will embody a critical part of autonomous navigation systems, with information from on-board cameras enabling the user to estimate the UAV's movement and position. In the given context, this paper presents different types of visual odometry, sensors for visual odometry, implementation components, and application scenarios. To allow the development and innovations, we utilize the widely used Robotic Operating System (ROS).

Keywords: UAV, visual-odometry, automation, navigation, ROS

This paper focuses on visual odometry (VO), a process for estimating position and camera motion from a frame sequence. Presently, the procedure has become a key component of navigational systems in autonomous cars, UAVs, and robots. Visual odometry compares the currently captured image with the previous one, seeking differences in the displacement of selected features or reference points via the optical flow method. The new pose is obtained by adding the estimated position vector relative to the previous pose. The VO localization field includes two main approaches, namely, frame-to-frame and frame-to-reference localization, each having specific use and applicability Bai et al. (2023), Couturier and Akhloufi (2021), Xiu et al. (2022).

1. VISUAL ODOMETRY

The methods include Relative Visual Odometry (RVO), Absolute Visual Odometry (AVO), and Inertial Odometry (IO).

1.1 Relative visual odometry

This technique exploits the optical flow procedure for estimating the position change. Various algorithms to detect the edges, corners, pixel brightness alterations, and other visually distinctive points are employed in obtaining the key points in the image. Between two frames, feature matching is applied to allow identifying these points. Between two frames is applied feature matching to allow the identification of key points in both frames. The change in position is then calculated based on the displacement of these key points in the image. After computing the change in position, the current camera pose is updated. Over time is the change of position accumulated and stored as a trajectory.

The main problem of the method lies in the drift over time. The pose deviation actually embodies a product of the recursive use of the estimated change of pose for calculating a new position variation. The central issue rests in the fact that an error in the previous calculation affects the accuracy of the current one. This deficiency can be solved through applying Loop Closure Detection (LCD), Simultaneous Localization and Mapping (SLAM), and fusion with an Inertial Measurement Unit (IMU) or, potentially, reinitialization.

By contrast, the main advantage is that, to be functional, the method does not require previous knowledge of or additional data about the environment. Further, the technique is suitable for use in indoor applications, being independent from the Global navigation satellite system (GNSS).

In Figure 1, all the steps involved in the RVO-based pose estimation are illustrated. The system detects the key features within the visual data; these features embody distinctive points that can be reliably detected and tracked across consecutive frames. After the features have been detected, the system will match corresponding features between the current and the previously observed frames. This process provides essential information in estimating the motion. Using the matched features, the system estimates the relative motion between the consecutive frames. This involves determining how the camera has moved between the frames in terms of translation and rotation. Based on the estimated motion, the system updates the pose

⋆ The research was funded from the Ministry of the Interior of the Czech Republic (MVCR) grant no. VB02000053 (An UAV carrying a multi-sensor head with an artificial intelligence) and the assistance provided by the general student development project being executed at Brno University of Technology.

⋆⋆[1]Faculty of Electrical Engineering and Communication, Brno University of Technology, 61600 Brno, Czech Republic marcon@vut.cz

10.1016/j.ifacol.2024.07.410

(position and orientation) of the camera relative to the previous pose Bai et al. (2023), Couturier and Akhloufi (2021).

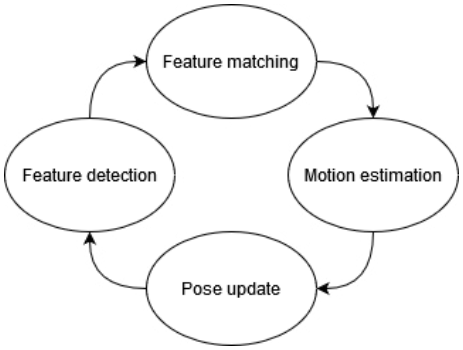

Fig. 1. Steps of RVO pose estimation

1.2 Absolute visual odometry

The actual approach fundamentally differs from the RVO, the main difference being in the detection and assignment of the features. The method utilizes pre-collected reference data, assuming that these must be accurately georeferenced before their potential use in the localization. The data may comprise aerial images captured in advance by the UAV itself and georeferenced via the GNSS on the UAV. The georeferenced data may be sourced from, for instance, platforms such as Google Earth.

An advantage of the method over the RVO rests in its immunity to error accumulation over time, as georeferenced data are exploited. However, the functionality and accuracy depend on the prepared dataset the technique interacts with. A common issue accompanying freely available data is that not all the snapshots are captured under uniform lighting conditions. Another problem arises from the dynamic character of the terrain: an effect such as the movement of vehicles on the roads, for instance, can deliver significant errors.

Yet another key difference from the RVO is that the AVO does not operate with consecutive frames, comparing the captured image during the localization with a database and seeking matches within the image. An option rests in using template matching, where the captured image is directly compared with a known database and the similarity of the images is evaluated. Further, we can employ feature matching, similarly to the RVO; by contrast, however, the search for distinctive points in the captured image will be compared with the points in the dataset.

In Figure 2, all the steps involved in the pose estimation using the AVO are illustrated. The main difference from the RVO is that during the feature matching process the current frame is compared with the reference features in the map. The AVO estimates the absolute position and the pose of the camera relative to the global coordinate system referenced to the images in the map. The last step of the AVO process involves the option to update the images within the map if necessary Bai et al. (2023)Couturier and Akhloufi (2021).

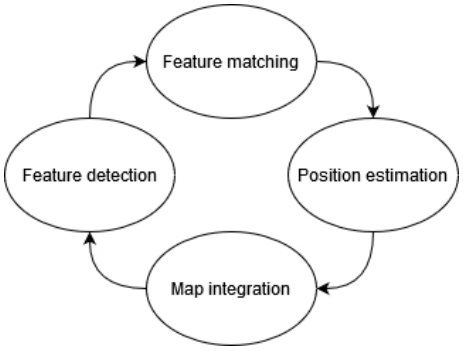

Fig. 2. Steps of AVO pose estimation

1.3 Inertial odometry

This procedure for determining the position utilizes motion sensors such as accelerometers and rotational sensors (gyroscopes) to calculate the current position, orientation, and speed of a moving object without the need of an external reference. Calculating the current position involves dead reckoning, a technique that computes the current position of the moving object based on the previously determined position, estimated speed, direction, and elapsed time.

All inertial navigation systems encounter the issue known as integration error, which arises because the slightest error in measuring the acceleration or speed will gradually accumulate over time. Even the best accelerometers accumulate an average deviation of around 50 meters over 17 minutes. This error also manifests itself in situations where the device remains stationary, as the system will, after a certain period, declare a change relative to the initial pose.

Inertial odometry is primarily designed for measuring short distances. To measure longer distances, the technology has to be combined with other methods for position determination, such as the GPS, Light Detection and Ranging (LiDAR), or another source of location tracking. When measuring greater distances, we need to eliminate the integration error of the method by resetting the position from a different source. This combination allows for a system capable of determining the position at a higher precision Acharya (2014).

2. SENSORS FOR ACQUIRING IMAGE DATA

In visual odometry applications, we can employ various types of cameras, such as monocular, stereo, or depth ones. Further, distinctive points in the image and their positions in space have to be determined.

2.1 Monocular cameras

Monocular cameras embody the first choice for various automation and robotics applications, such as image capturing, motion detection in images, and photogrammetry. However, these devices are generally unsuitable in estimating the distance of an unknown object in an image: The distance of the object is estimable only if the relevant dimensions are known. Concerning the advantages of the cameras, one of the main aspects lies in the simple design

and flexibility, compact dimensions, cost-effective character, and ability to be employed in capturing various types of terrain.

TSelecting between a monocular camera and the AVO or the RVO depends on the specific requirements and constraints of the application. Monocular cameras in the AVO can provide a global context by estimating the absolute position of a device or vehicle in relation to the global reference frame. For real-time applications where the incremental updates are crucial, the RVO is preferable. In general, visual odometry with monocular cameras may be more susceptible to drift over time due to the accumulated errors in the feature tracking. This effect can then compromise the long-term position estimation accuracy Xiu et al. (2022).

2.2 Stereo cameras

Stereo cameras are essentially an extension of monocular cameras. In practical terms, a stereo camera consists of two horizontally aligned and vertically displaced monocular cameras. The main difference is that a stereo camera allows calculating the distance of an object in the image without knowing the dimensions; this principle exploits parallax, which involves a change in the observation angle to the captured object.

When used with visual odometry, the first step relies on detecting the features in the image from the left camera. Subsequently, the features are extracted from the right camera image, and the distances between the features for the given image are computed. This process is then repeated with a new image. At the next stage, the features are matched across the different images, and the camera's position change in space is estimated. The system's accuracy heavily depends on the lens calibration and lighting conditions, as the differences in the images can lead to undesired errors.

In general, stereo visual odometry algorithms are computationally more intensive compared to those of depth cameras, especially as regards high-resolution images, due to the complex processing required for the stereo matching and depth estimation from multiple camera viewpoints Jiang et al. (2013).

2.3 Depth cameras

Depth-sensing cameras utilize diverse technologies to determine the distance between the camera and the objects in the scene. Options such as the Time of Flight (ToF) or structured light are used for the calculation. The main advantage is that the output of a depth camera is an image where each pixel is assigned a color corresponding to the distance of the point in the image. Each type of depth camera relies on a known parameter when acquiring the images; with stereo cameras, for example, the distance between the sensors is known. When using structured light, the light pattern is known. Depth cameras based on the Time of Flight (ToF) rely on the known speed of light. Similarly to the lidar, this type of camera utilizes laser light to measure the distance. Subsequently, the distance is calculated using the known speed of light propagation and the time of flight. The maximum measured distance depends on the power and wavelength of the light applied. The main disadvantage of the camera is the potential interference from another camera or light source. Stereo depth cameras employ two horizontally aligned and vertically displaced sensors. Similarly to stereo cameras, they compare the captured image between the two sensors. As the distance between the sensors is known, the depth can be determined based on a comparison of changes in the image. Unlike basic stereo cameras, where the distance of points in the image needs to be computed through stereo matching algorithms, advanced stereo depth cameras eliminate such a necessity. These cameras can utilize any visual elements for the depth measurement, and thus they perform well in most lighting conditions, including outdoor environments. With the addition of an infrared projector, it also becomes possible to capture a depth detail even in low-light conditions. Unlike monocular and basic stereo cameras, the depth ones provide accurate depth information, enabling more robust spatial perception and tracking. Using depth cameras, visual odometry systems can better handle challenging scenarios, such as low-light conditions, occlusions, and dynamic environments, resulting in an enhanced navigation and localization accuracy. Presently, the Intel RealSense camera family is readily available, offering advanced depth-sensing capabilities in a variety of applications and industries Arafat et al. (2023), IntelRealsense (2019).

3. ALGORITHMS FOR THE IMAGE PROCESSING

Currently, multiple visual odometry algorithms are available, including the ORB-SLAM, OpenVINS, and RTAB-Map. In the experiment described herein, the RTABMap algorithm will be utilized.

3.1 RTAB-Map

The RTAB-Map (Real-Time Appearance-Based Mapping) is a SLAM algorithm for trajectory acquisition, based on processing RGB-D, stereo, or lidar data. This algorithm relies on an incremental image matching detector that facilitates the detection of previous features. This principle practically exploits relative visual odometry, the difference being the detection of matches with the previous frames. Thus, the problem of accumulating position estimation errors over time is eliminated. The loop closure detection procedure utilizes the bag-of-words (BoW) method to determine the probability of a match between the new image and the previous location and to define whether a new location is represented. In loop closure detection, which involves recognizing the patterns in visual elements to identify the previous locations, the algorithm retroactively corrects the accumulated trajectory errors. To perform such a detection, a larger number of the previous frames need to be stored. Without limiting the quantity of stored frames, the computational complexity would increase over time, as processing a larger amount of data would be required. To address this issue, the RTAB-Map also implements memory management, restricting the number of locations designated for the loop closure detection. This ability guarantees that the computational performance will remain sufficient for a real-time comparison. Additionally, the RTAB-Map allows a fusion with inertial odometry.

Another advantage lies in its compatibility with the ROS 1 and the ROS 2.

The RTAB-Map implements both the AVO and the RVO techniques in its visual odometry process. While the AVO focuses on determining the camera's absolute position by referencing maps or landmarks via loop closure detection, the RVO estimates its position relative to the previous locations by utilizing incremental visual changes. Through integrating the AVO and the RVO approaches, the RTAB-Map enhances its ability to achieve robust localization and mapping results, particularly in dynamic environments Labbe and Michaud (2018), Labbé and Michaud (2019).

4. HARDWARE SELECTION

In this chapter, companion computer platforms and cameras suitable for processing the visual and inertial data on a UAV are described.

4.1 Companion computer

When selecting, we need to consider the dimensions, weight, and computational power. In the majority of current image processing applications, platforms like the Raspberry Pi or NVIDIA Jetson Nano are commonly employed. These options offer a balance of affordability, suitable dimensions, and adequate computational power for the regular image processing tasks. However, to achieve the maximum frame rate of the odometry, we decided to use the Intel NUC (Next Unit of Computing) as the companion computer.

The Intel NUC is a series of mini PCs developed by Intel. These compact devices offer a high performance in a small and elegant design, making them ideal for use in various applications. One of the significant benefits is the small size and the ability to expand the RAM and storage capacity. Additionally, a benefit rests in connecting an SSD disk via the M.2 port. Similarly to the competing boards, the Intel NUC features a wide range of ports, including the USB, HDMI, Ethernet, and many others. Yet another advantage is the ability to install the standard Ubuntu operating system. .

4.2 Camera

A multitude of stereo depth cameras are offered by Intel. Each model differs in the sensor, supported resolution, and frame rate. Some of the models also carry an integrated IMU unit. An additional benefit lies in the support of the ROS. Further, Intel delivers a ROS wrapper to process the visual and inertial data.

A most commonly used depth camera is the D400 series product. The device utilizes infrared projectors and sensors to capture the depth the image, enabling 3D mapping of the space. Additionally, the cameras feature a high-resolution color sensor and a wide field of view. Thanks to the latter factor, it is possible to capture a large area in a single frame. In the experiments, we decided to use the Intel RealSense D455 depth camera.

In Figure 3, the connection between the UAV and the Intel NUC companion computer is shown. To connect the RealSense camera to the companion PC, we employed a high-speed USB 3.0 bus to achieve the fastest data transfer. The flight controller was connected to a computer via a USB TTL converter.

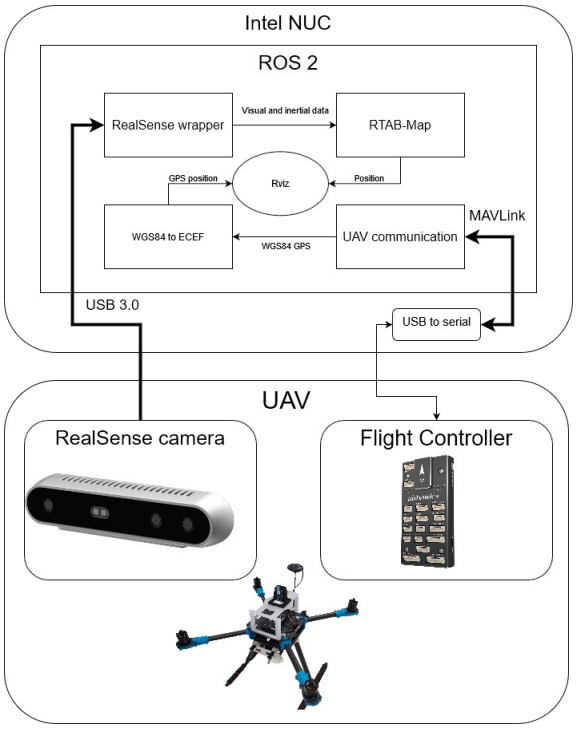

Fig. 3. Hardware connection between UAV and companion computer

5. SOFTWARE IMPLEMENTATION

To facilitate the fast development, the widely used ROS (Roboting operating system), Humble version, was utilized. We decided to use the Ubuntu 22.04 as the companion computer operating system, especially due to its compatibility with the selected version of the ROS. To handle the visual and inertial data from the RealSense camera, we applied an Intel-developed ROS wrapper. From the RTABmap library we used appropriate nodes to process the visual and inertial data delivered by the RealSense camera. We utilized Rviz to visualize the data, also employing the RTAB-Map Viz in visualizing the odometry process. To compute the relative position from the WGS84 GPS format, we implemented custom nodes in our system.

In Figure 4, the connection between the ROS nodes and topics is shown. The camera node consists of topics, and these include the RGB and the depth image data, as well as inertial data. The data from the depth and the RGB cameras are processed in the nodes **/point_cloud_xyz** and **/pointcloud_to_depthimage**. The inertial data are filtered in the **/imu_filter_madwick** node, which ensures the orientation estimation for the IMU. The visual and inertial data are proccessed in the **/rgbd_odometry** node; this node delivered the visual odometry position estimation. Additionally, the **/rtabmap_viz** node is employed to visualize the odometry position estimation.

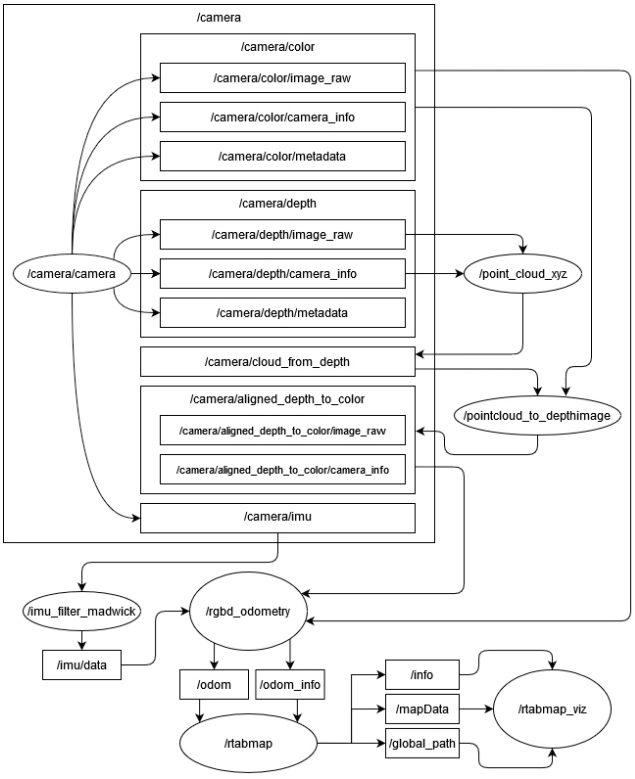

Fig. 4. Software infrastructure of ROS nodes and topics connection

5.1 Converting GPS to relative coordinates implementation

A node has been implemented for computing the relative coordinates from the GPS WGS84 format. The first received coordinates are taken as the reference ones. To allow visualizing the data in Rviz, the output of the node is in the **nav_msgs/msg/Path** format; this format can be visualized in the Rviz environment using **Path**.

5.2 Error computation

To measure the error between the visual odometry pose and the relative position from the GPS, we implemented a ROS node. This node subscribes to the current positions of both trajectories. Subsequently, the difference between the current positions of the two trajectories is evaluated. The image in Figure 5 illustrates the connection of nodes to the
/error_path node, which computes the difference error between trajectories. The output from the node can be visualized afterwards using visualization tools.

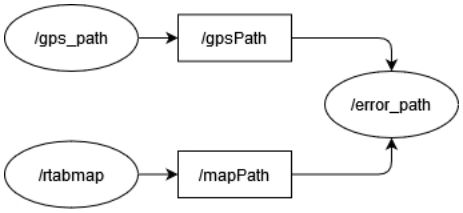

Fig. 5. Connection of nodes for error computation

6. EXPERIMENT

The experiment was performed by using the created ROS nodes; in the process, all of the data on the UAV were recorded. A rosbag was recorded in the course of the experiment, allowing a subsequent flight reconstruction. Using the rosbag facilitates developing various other applications with the UAV data and evaluating the measured data. In the experiment, we focused on comparing the positioning precision between the odometry and the GPS. The image below exposes the UAV to fly the experiment, equipped with the Intel NUC as the companion computer and the Intel Realsense camera D455.

After launching the UAV, we had to wait for a sufficient number of GPS satellites to determine the GPS position. Subsequently, all the nodes to communicate with the flight controller, RealSense camera, and visual-inertial odometry needed to be started. When all of the the necessary nodes have been launched, we started recording the published topics into the rosbag. Upon initializing the system, the experimental flight was taken. During the experiment, the GPS system had an access to approximately 8 satellites, and the RTK (Real Time Kinematic) GPS was not utilized. The data stored in the rosbag were visualized afterward, allowing a comprehensive analysis and evaluation of the recorded information.

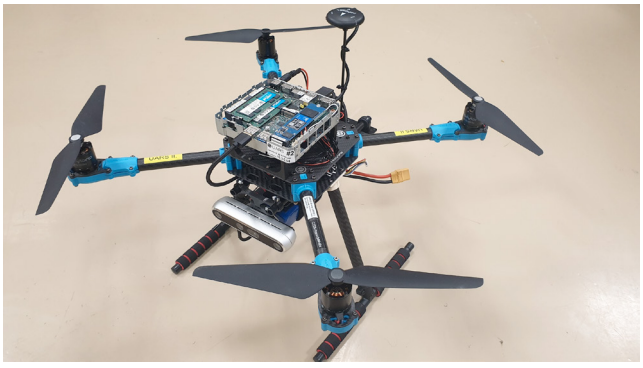

Fig. 6. UAV ready for the experiment

In Figure 7, the trajectory of the UAV estimated by the odometry is shown in the RTAB-Map viz environment. The image also exposes the created 3D point cloud map of the road and sidewalk. The trajectory and map creation are observable from the recorded rosbag after the actual step.

6.1 Experiment evaluation

The measured data were processed using the created ROS nodes **/gps_path** and **/error_path**. The resulting trajectory from odometry and the GPS trajectory can be visualized via the rviz environment. In figure 8, the visualization of the trajectories in the rviz environment is delivered: The blue line represents the GPS trajectory, and the green one stands for the visual odometry trajectory.

As is obvious from the trajectory image, the shapes are approximately identical, with certain deviations. During the experiment, a straight flight over a sidewalk was performed, followed by a return to the starting point. The

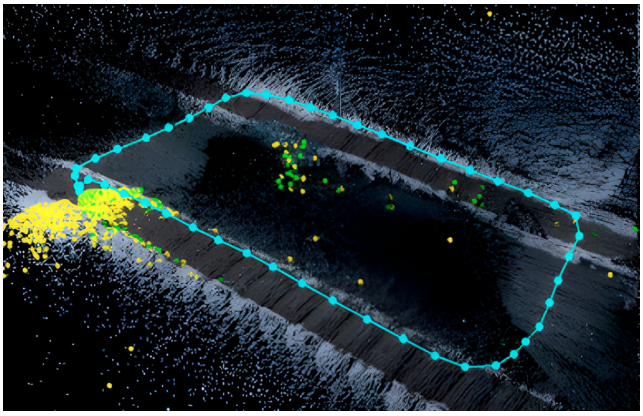

Fig. 7. Visualization of visual odometry trajectory in RTAB-Map viz

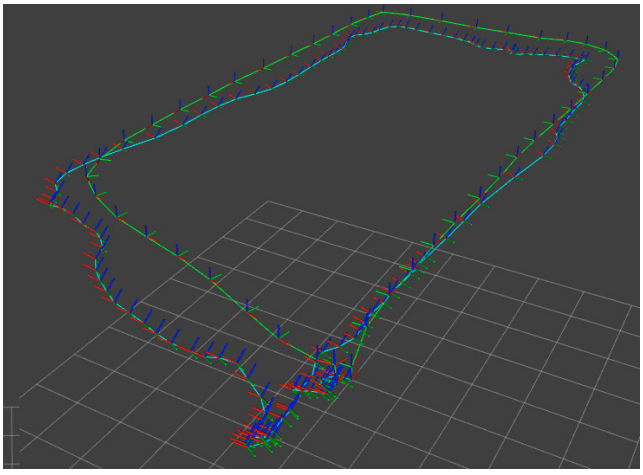

Fig. 8. Visualizing the trajectories in the rviz environment

output data from the **/error_path** node were processed in the spreadsheet editor. The image in Figure 9 exposes the deviation between both trajectories; the greatest deviation was apparently recorded in the y-axis. The average deviation in the x-, y- and z-axes equaled 0.294m, -0.431m, and 0.357m, respectively.

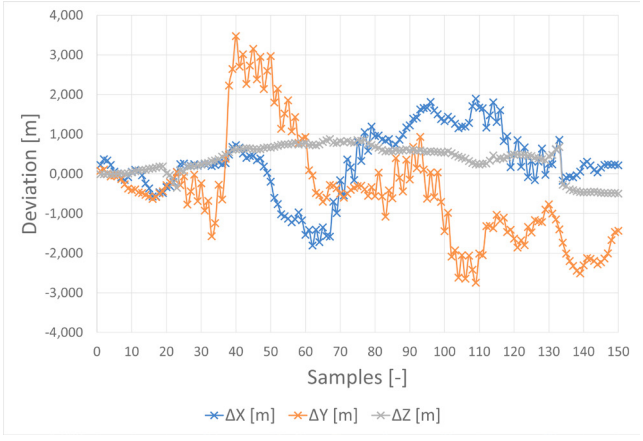

Fig. 9. Graph of deviations

In the evaluating the data, no consideration was assigned to the displacement of the camera and the GPS module. This fact may have caused a distortion in the y-axis deviation. As the average accuracy of the GPS typically ranges from 2 to 5 meters, our experiment demonstrated a minimum deviation between the odometry and the GPS.

7. CONCLUSION

Based on the experiment, we can conclude that visual inertial odometry achieves very good results compared to the GPS. The dataset shows that, in estimating the position, visual odometry exhibits less oscillations than the applied GPS system. The maximum absolute deviation was observed in the Y-axis, attaining 3.474 m; by contrast, the same quantity in the x-axis equaled 1.895m. The experiments have proved that visual odometry can replace the GPS in monitoring the position of UAVs. In the future, the technique will also be employed to control UAV positions. The main advantage over the GPS lies in the independence from satellites and/or other positioning signal sources; this property is crucial in most current applications, as the GPS signal can interfere with others or not be available at all.

Prospectively, it appears interesting to measure the precision of both technologies against ground-based reference points because our current experiment evaluates the accuracy of visual odometry against the GPS.

REFERENCES

Acharya, R. (2014). *Understanding Satellite Navigation*. Academic Press, 1st edition.

Arafat, M.Y., Alam, M.M., and Moh, S. (2023). Vision-based navigation techniques for unmanned aerial vehicles. *Drones*, 7(2). doi:10.3390/drones7020089.

Bai, Y., Zhang, B., Xu, N., Zhou, J., Shi, J., and Diao, Z. (2023). Vision-based navigation and guidance for agricultural autonomous vehicles and robots. *Computers and Electronics in Agriculture*, 205. doi:10.1016/j.compag.2022.107584.

Couturier, A. and Akhloufi, M.A. (2021). A review on absolute visual localization for uav. *Robotics and Autonomous Systems*, 135. doi:10.1016/j.robot.2020.103666.

IntelRealsense (2019). Beginners guide to depth (updated). *Robotics and Autonomous Systems*.

Jiang, Y., Xu, Y., and Liu, Y. (2013). Performance evaluation of feature detection and matching in stereo visual odometry. *Neurocomputing*, 120, 380–390. doi:10.1016/j.neucom.2012.06.055.

Labbe, M. and Michaud, F. (2018). Long-term online multi-session graph-based splam with memory management. *Autonomous Robots*, 42(6), 1133–1150. doi:10.1007/s10514-017-9682-5.

Labbé, M. and Michaud, F. (2019). Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of Field Robotics*, 36(2), 416–446. doi:10.1002/rob.21831.

Xiu, H., Liang, Y., Zeng, H., Li, Q., Liu, H., Fan, B., and Li, C. (2022). Robust self-supervised monocular visual odometry based on prediction-update pose estimation network. *Engineering Applications of Artificial Intelligence*, 116. doi:10.1016/j.engappai.2022.105481.