Development of a Visual Inertial Odometry Positioning System for Autonomous Interplanetary Drone State Estimation

A project presented to
The Faculty of the Department of Aerospace Engineering
San Jose State University

In partial fulfillment of the requirements for the degree
Master of Science in Aerospace Engineering

by

Steven Rispoli

May 2021

Approved by

Dr. Periklis Papadopoulous
Faculty Advisor

3.5 VIO Algorithm

Abstract

Autonomous vehicles such as interplanetary drones are an expanding topic of interest. Autonomous vehicles such as the Ingenuity Helicopter are making history and inspiring a new era of space exploration. Traditional autonomous vehicles such as the Mars Exploration Rovers used Visual Odometry to know their own position. However, visual Odometry is very computationally intensive producing motion estimates every few minutes. Interplanetary drones need Visual Odometry systems that produce motion estimates at a much faster rate. This paper will develop a Visual Odometry system for an autonomous drone to extrapolate the performance of these systems. These estimates will be able to help predict the functionality of similar systems on Ingenuity and other interplanetary drones in development.

Acknowledgements

I would like to thank my parents for supporting me from the beginnings of my college journey to the end of Graduate school. Thank you for the sacrifices that you made as I had to prioritize my schooling over spending time with both of you. I would also like to thank my friends who have shared in my journey in and out of school. I would also like to thank my Advisor Dr. Periklis Papadopoulos for believing in me and pushing me to go the extra mile to succeed. Without all of you I wouldn’t be where I am today.

	Normalized Camera Y Coordinate	Pixels

EKF	Extended Kalman Filter	--------
EOM	Equations of Motion	--------
FAA	Federal Aviation Administration	--------
FPGA	Field-Programmable Gate Array	--------
IMU	Inertial Measurement Unit	--------
FPS	Frames Per Second	
GPS		

SBC	Single Board Computer	--------
SGBM	Semi-Global Block Matching	--------
SLAM	Simultaneous Localization and Mapping	--------
VIO	Visual Inertial Odometry	--------
VO	Visual Odometry	--------

List of Figures

Figure 1: Ingenuity Engineering Model Design 21

Figure 2 – Pinhole Camera Model Visualization 32

Figure 3 – The Effect of Distortion k_1 on an Undistorted Image 34

Figure 4 – Stereoscopic Camera Model Visualization 35

Figure 5 – Average Execution Time of the Tested Devices. There are 2 data points at 1.43GHz and 2.26GHz for standalone and HITL. 41

Figure 6 – VO Estimate Rate of the Tested Devices. There are 2 data points at 1.43GHz and 2.26GHz for standalone and HITL.

1 Introduction

1.1 Motivation

Interplanetary drones are an exciting field of research. With the successful flights of Ingenuity, JPL's Mars helicopter, a new era of aviation is emerging. However, this new era of aviation brings new challenges with it. These drones cannot be flown by a human pilot due to the delay in the commands reaching the drone on these far away planets. Flight on other planets will require these new aircraft to be autonomous.

A large challenge for autonomous drones is the measurement of accurate state data such as its position and velocity. Earthbound autonomous drones face the same problem as interplanetary drones in that they both do not have access to GPS. Autonomous drones on Earth must fly indoors where GPS signals are weak, and interplanetary drones will not have access to GPS at all. Linear acceleration sensors from IMU's can be used to extrapolate velocity and position of the drone, but they are not very reliable. Error in measurement causes sensor drift to accumulate quickly over time, rendering the data useless. Traditionally, GPS would correct the IMU drift for the drone's control system, but without GPS other methods must be used to correct the IMU data.

Visual odometry, or VO, is one such way to correct IMU drift. VO uses an optical sensor systems to accurately measure the 3D motion that a singular camera setup cannot. VIO will give position and velocity data that has much less drift then the IMU alone.

1.2 Literature Review

One the first times VO was done was back in 1976 by PhD students at Stanford. Hans Moravec and Donald Gennery developed a visual system to track the movement of an electric vehicle. The system worked by sending the camera images wirelessly to a server which did the VO calculations. There are 4 steps to their VO system. First an interest operator determines areas in the image that will be easily identifiable from image to image to track. Then the binary search correlator attempts to find the features from the interest operator. Matches of the features in consecutive frames are sent to the high-resolution correlator to improve upon the matches accuracy. Finally, the camera solver uses the improved matches from the high-resolution correlator to calculate the camera position in both frames to find how far the vehicle has travelled (Moravec, 1976).

1.2.1 Improvement and Applications of VO

Jiawei Mo and Junaed Sattar point out a downfall of monocular VO, such as that mentioned above. Monocular VO can track the movement in only 2 directions. It lacks the depth perception necessary for the tracking of full 3D motion. To gain movement information in the third dimension, they proposed a stereoscopic approach. They use one camera for traditional VO while the other allows the system to fuse the third dimensions motion with that of the monocular VO. In previous work the fusion of monocular VO with a second camera is done by stereoscopic matching. Stereoscopic matching is computationally expensive however, so Mo and Sattar replaced stereoscopic matching with a scale optimizer which performs calculations on a single pixel location instead of the region around the projected pixel. This allows scale optimization to be more computationally efficient. The scale improvement saw a decrease in computational time versus the stereoscopic matching implementation during testing with the KITTI computer vision dataset transformations in the VO algorithm does not add up as quickly with lower framerates. However, Howard notes that there is a lower bound for this decrease in framerate to improve accuracy due to the travel of the features from frame to frame. The second test was conducted on DARPA's LAGR vehicle. The LAGR vehicle is a platform to assess the slip during ground traversing. VO is used to check the wheel odometry where a discrepante to it0W*nBT/F1 11.04 Tf1 0 0 1 540.1 745.56 Tm/GS7 camera moves. During testing versus IMU coupling that is fused with a standalone VO system, Usenko et al.'s VIO system is superior. With a dataset collected by a drone, The VIO system outperformed the fused VIO, particularly when there was significant motion blur. This held true for both rotation and translation over a course of 40 meters, as well as over time on a course of over 1140 meters. Another test on a dataset taken from a car was also done to test the VIO systems qualitative results, as it was able to successfully track the movement of the car in a challenging optical environment (Usenko, 2016).

1.2.1.1 Drone Use Cases

was observed that the translation error averaged 0.021 meters with a peak under 0.05 meters, while the orientation error averaged 2.183 degrees with a peak below 5 degrees. The second flight showed similar results, however there was a large spike in the orientation error due to an error in the ground truth capture. Two additional flights were conducted outside, which compared the VIO position data to GPS data taken during the flight. Both flights showed drift in the VIO data compared to the GPS data. However, the GPS data was not reliable in the second outdoor flight and aid by the VIO system gave better results than the GPS alone. Overall, Dornellas et al. quantitatively showed that VIO could be used on a drone to fill in the gaps of poor GPS signal (Dornellas, 2019).

1.2.1.2 Interplanetary VO

Visual Odometry does have some precedence in interplanetary use as well. Both MER utilized VO for portions of the autonomous driving that was instructing. Wheel encoders were used successfully on level rigid surfaces but were found to exceed the 10% per 100 meters drift required. A VO system had been developed for use on the MER but had not gone through the

As well, the long computation time necessitated that the human driver decides what to image for the VO calculation, so that the feature detector has enough data to track. Due to these challenges VO was only used in very specific situations consisting of short steep slippery terrain. Regardless of these limitations, VO proved to be an effective tool for the MER to safely and accurately traverse the Martian landscape (Maimone, 2007).

Building upon the success of VO on the MER, VO was implemented on the MSL Curiosity rover as well. Curiosity saw a large improvement on its VO computation time over the MER. This was not only due to a newer generation CPU, but the introduction of a FPGA as a vision co-processor. During tests in Curiosity's development, Howard et al. used a Xylinx Virtex 5 FPGA to assess its performance over MER's VO implementation. The results of the tests showed a vast improvement over MER's implementation. The stereoscopic calculations took about 24 to 30 seconds on the RAD6000 CPU found on the MER compared to 0.005 seconds on the FPGA accelerated setup. The performance gains continued to be stable as increasing the resolution of the images from a width of 256 to 1024 pixels only saw the FPGA take 0.082 seconds to do the stereoscopic calculation. Additionally, a test of the whole VO system was done. The MER analog completed the visual odometry calculations in about 160 seconds while the FPGA as a co-processor took 0.016. However, only the feature detection and matching were run on the FPGA and following calculations would have to be done on a separate CPU (Howard, 2012).

The Mars 2020 Perseverance rover see's additional improvements over Curiosity's VO hardware and programming. Perseverance adds an additional RAD750 CPU and Vertex 5 FPGA. As well as this, the calculations have been parallelized. This results in the VO that took 65 seconds on Curiosity to take a total of 9.8 seconds on Perseverance. This increase in VO calculation, among other penalizations, increased Perseverance speed to about 100 meters per

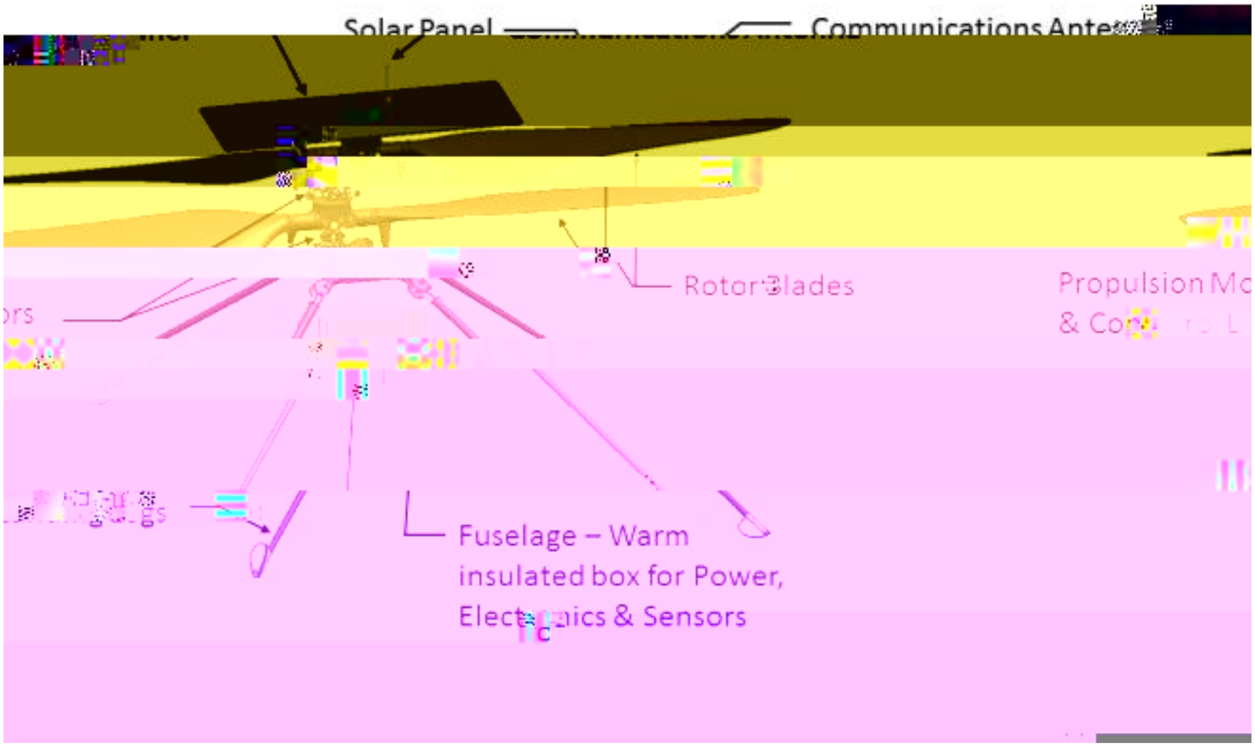

Figure 1: Ingenuity Engineering Model Design

It has a complex avionics system that ensures safe flight for the helicopter. Its primary processor is a Qualcomm Snapdragon™ 801 processor at 2.26 GHz with 2 GB of RAM and 32 GB of Flash memory. It has 2 identical Texas Instruments Hercules TMS570LC43x automotive processors at 300 MHz with their own 512 KB of RAM and 4 MB of flash memory. These are redundant processors that receive and process identical. These perform flight control functions that are critical to the operation of the drone. During flight one of the two will be the primary processor while the other is waiting to be hot-swapped in case a fault causes the primary to restart.

To perform the mission critical tasks, Ingenuity utilizes a military grade radiation hardened ProASIC3L FPGA from MicroSemi. It performs all the I/O to the sensors, actuators,

including the MER, which used the radiation hardened RAD6600 (Cheng, 2006). The RAD series of CPUs is the premiere line of radiation hardened CPUs. Currently the most capable RAD series CPU is the RAD750. Even though it is the cutting edge for space processors it is an order of magnitude behind cutting edge of contemporary processors in terms of processor speed.

GPUs are a processor that is starting to gain popularity for heavy computational tasks. Generally its individual clock speed is slower than a CPU, however, they have many more cores than CPUs. This large amount of cores is the main draw for GPUs. This allows them to parallelize many small processes, and run them concurrently instead of serially like with CPUs. GPUs are generally more difficult to program, but the performance benefits justify the extra effort. Unlike the RAD series of CPUs, there are no commercially available radiation hardened GPUs. This could become a problem for interplanetary use as they will be much more susceptible to faults.

FPGAs are a unique type of processor that can have its hardware reprogrammed to perform a specific task at the hardware level. Like CPUs, FPGAs als

Harris corner detector. For the edge detector, the FPGA ran the algorithm 1.92 – 0.93 times faster than the GPU and 27.3 – 23.3 times faster than the CPU, as the image size increased from 512x512 to 3936x3936 pixels. Additionally, the energy consumption of the FPGA was 154 – 167 times more energy efficient than the GPU, and 94 – 102 times more energy efficient than the CPU, as the image size increased from 512x512 to 3936x3936 pixels. For the corner detector, the FPGA ran the algorithm 2.02 – 0.96 times faster than the GPU and 17.4 – 20.9 times faster than the CPU, as the image size increased from 512x512 to 3936x3936 pixels. Additionally, the energy consumption of the FPGA was 154 – 166 times more energy efficient than the GPU, and 94 – 101 times more energy efficient than the CPU, as the image size increased from 512x512 to 3936x3936 pixels. However, none of these processors tested were radiation hardened so the comparison isn't a direct comparison for an interplanetary drone use case. That said, Ingenuity has a non-radiation hardened CPUs and was able to fly multiple successful flights.

1.3 Proposal

The goal of this project is to extrapolate the performance of Earth based VIO systems to an interplanetary drone. A VIO system will be developed on a more conventional embedded system rather than with an FPGA. Additionally, this project does not look to improve on existing VIO but to implement an existing VIO system for benchmarking. Both performance and efficiency of the implemented VIO will be conducted, then based on a variety of factors, will be expanded to make an approximation of the performance that can be expected on an interplanetary drone.

There are three success criteria for this project as follows.

1. Complete Success Criteria:

a. A VIO system will be implemented and flown on the Theia drone and in the virtual simulator. Tests will be performed to assess the execution time of the VIO system on the actual drone during flight. Groundtruth from the simulator will be used to assess the accuracy of the algorithm. From these tests a feasibility study will extrapolate these results to hardware that is likely to appear on other interplanetary drones.

2. Partial Success Criteria:

   a. A VIO system will be implemented and tested in the virtual simulator. Tests will be performed to assess the execution time of the VIO system on the actual drone during flight. Groundtruth from the simulator will be used to assess the accuracy of the algorithm. From these tests a feasibility study will extrapolate these results to hardware that is likely to appear on other interplanetary drones.

3. Minimum Success Criteria:

   a. A VO system will be implemented and tested with a stereoscopic camera setup. Tests will be performed to assess the execution time of the VO system. A feasibility study will extrapolate these results to hardware that is likely to appear on other interplanetary drones.

1.4 Methodology

There are two major parts to the completion of this project. The first part is to implement a VIO system that will allow a testing drone to achieve autonomous flight without receiving state data from a GPS. The second part is to take the results from the first part to approximate expected VIO results.

Part I: Validation of VIO on an Autonomous Drone

1. Specify test drone embedded flight computer.
2. Identify VO/VIO algorithms that are best suited for autonomous flight.
3. Implement the identified VO/VIO algorithm in C++. Add IMU fusion in the case the identified algorithm is VO only.
4. Deploy the C++ VIO code to the test drone/simulator.
5. Test the VIO under manual control to compare with GPS data in flight, or groundtruth data from the virtual simulator.
6. Make adjustments to the VIO system if it does not achieve a reasonable error.
7. Test the VIO on the test drone for a predetermined autonomous flight.

Part II: Extrapolate Autonomous VIO Performance to Interplanetary Performance

1. Determine hardware that will likely be used for interplanetary drones.
2. Compare performance criteria of selected hardware with the hardware used on the test drone.
3. Investigate performance tradeoffs between different processor architectures.
4. Use the performance tradeoffs study to scale test drone VIO performance to an interplanetary drone.
5. Discuss the viability of VIO as a real time system on interplanetary drones

2.3 Supporting Drone Systems

(3.1)

projective transformation scaling . However, finding can be difficult so another method to find the image scale must be used.

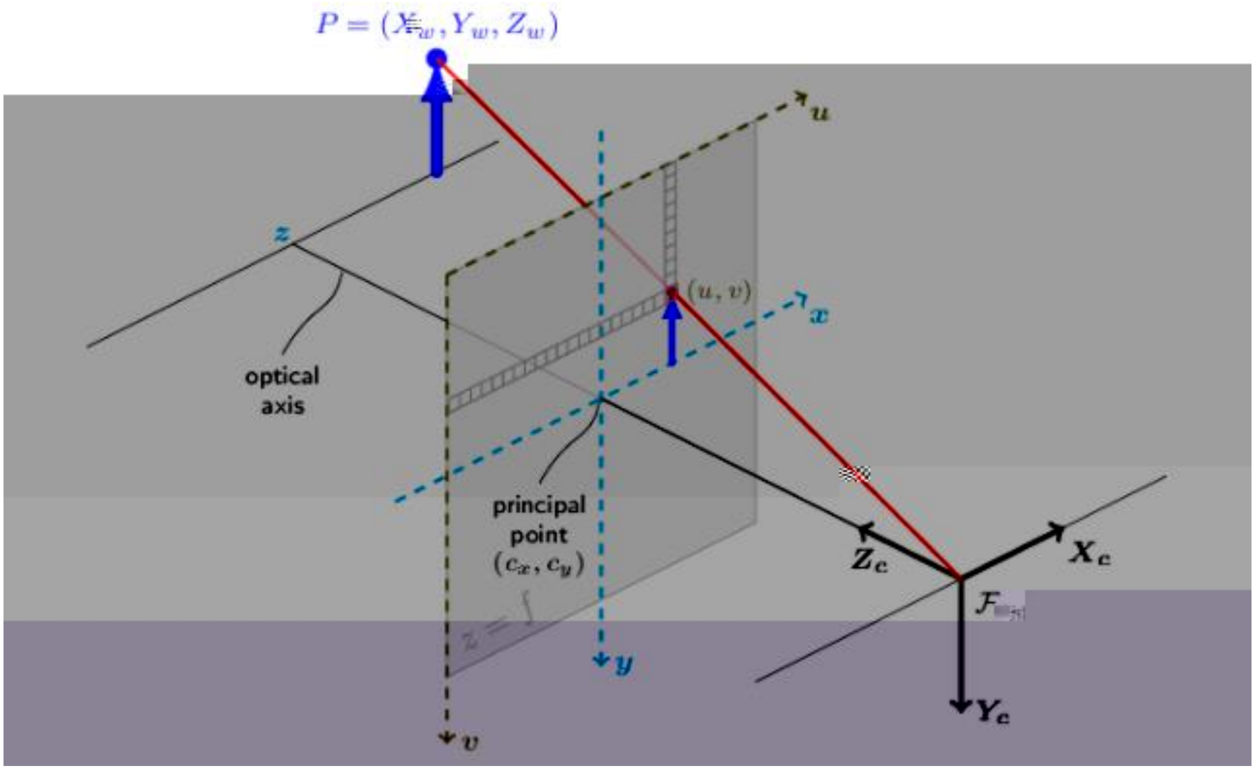

Figure 2 – Pinhole Camera Model Visualization

3.1.1 Extension to Stereoscopic Calibration

Zhang (Zhang, 2009) extended his model for camera calibration to stereoscopic cameras to support other projects he was working on at the time. In his stereo model the second camera would be denoted with a . The transform between cameras is such that . This relationship is shown more precisely in equations 3.11 and 3.12.

(3.11)

(3.12)

To find the cost function in equation 3.13 is minimized.

(3.13)

In equation 3.13 and are Booleans denoting if the point is visible to the camera.

3.2 Distortion Model

The distortion model used in OpenCV is a combined version of the Brown-Conrady model (Brown, 1966) and the Fitzgibbon division model (Fitzgibbon, 2001). The combined model includes the radial and tangential distortion terms as well as the thin prism model as seen in Wang (Wang, 1992). The full OpenCV model is as follows

(3.14)

2Where

å

After some simplification we get

(3.19)

The disparity in the images is given by . However, since depth is tied to one pixel, the stereoscopic model projects the depth to only one of the camera views. Additionally, This model assumes that the images from the two camera are rectified so that there is no vertical translation or any rotations between the images.

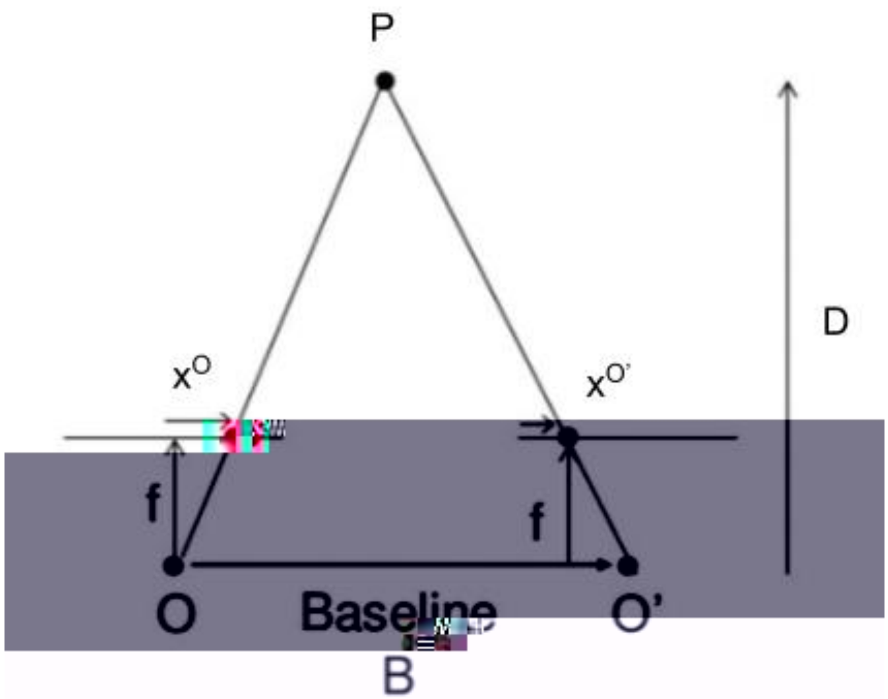

Figure 4 – Stereoscopic Camera Model Visualization

3.4 Stereoscopic Correspondence

The stereoscopic correspondence algorithm used in OpenCV is based on Heiko Hirschmuller

Where is the cost along each path calculated by where is the direction of the pixel and and are constant penalties:

between frames is found using the Levenberg-Marquardt least-squares algorithm. The Error equation is given as

(4.1)

Where

is the reprojection error

is a feature in the image

is the clique of the matched feature inliers

is the homogenous image coordinates

is the homogenous world coordinates

is the camera projection matrix

is the transformation between image a and b.

If there are enough points in the clique, the co-linearity is close to one, and the reprojection error is below a certain threshold the transformation is valid. The transformation is calculating the egomotion of the stereoscopic camera, which in turn measures the movement of the drone.

4 Results

Overall, successful trajectory estimates were not able to be achieved. There are bugs in the that it can be used by the RGBD Odometry object. It is possible that the assignment of the depth

Desktop based AMD Ryzen 9 3950X @ 3.5GHz.

The devices included are both desktop level and embedded system level. Additionally, the 2 Jetson devices were run distributed in a HITL mode with another device handling the supporting systems, and as a standalone device running the VO as well as the communication server.

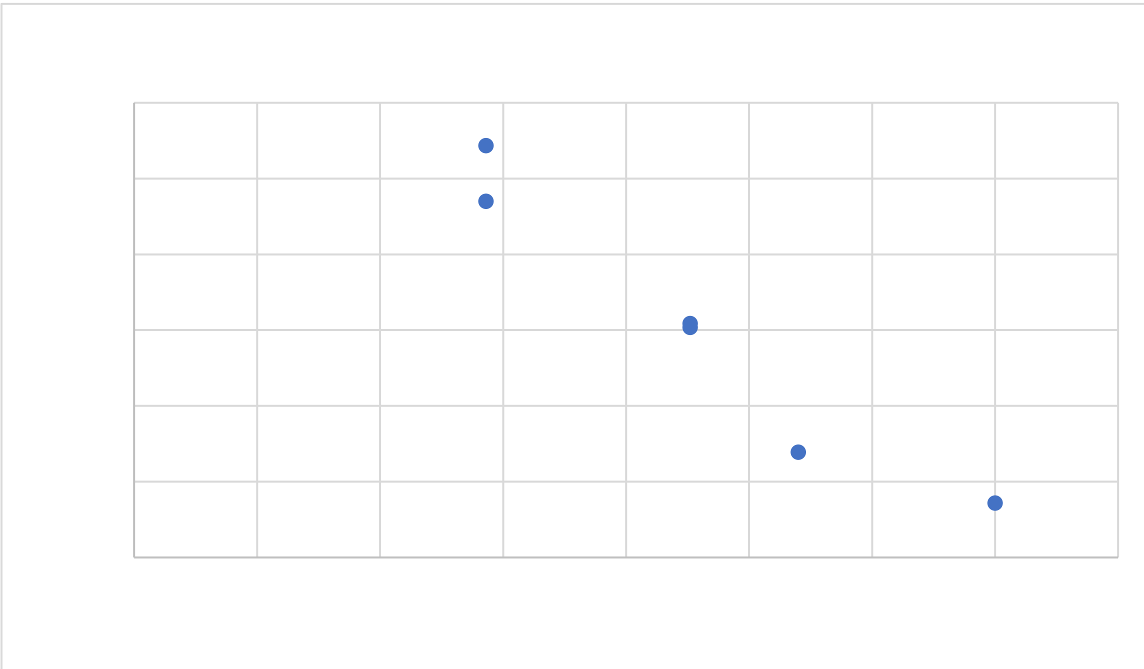

Figure 5 – Average Execution Time of the Tested Devices. There are 2 data points at 1.43GHz and 2.26GHz for standalone and HITL.

References

Usenko, V., Engel, J., Stuckler, J., & Cremers, D. (2016). Direct visual-inertial odometry with stereoscopic cameras. *2016 IEEE International Conference on Robotics and Automation (ICRA)*. doi:10.1109/icra.2016.7487335

Wang, J., Cohen, P., & Herniou, M. (1992). Camera Calibration with Distortion Models and Accuracy Evaluation. *IEEE Transactions On Pattern Analysis and Machine Intelligence, 14*(10).

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(11), 1330-1334. doi:10.1109/34.888718

Zhang, Z. (2009). CAMERA CALIBRATION.