*Preprints* (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 May 2020 doi:10.20944/preprints202005.0221.v1

Peer-reviewed version available at *Robotics* **2020**, *9*, 56; doi:10.3390/robotics9030056

*Article*

# Real-world Comparison of Visual Odometry Methods

**Alexandre Alapetite [1,*], Zhongyu Wang [1], John Paulin Hansen [2], Marcin Zajączkowski [2] and Mikolaj Patalan [2]**

[1] Alexandra Institute, Njalsgade 76, 2300 Copenhagen S, Denmark; alexandre.alapetite@alexandra.dk, zhongyu.wang@alexandra.dk
[2] Technical University of Denmark, Diplomvej 371, 2800 Kongens Lyngby; jpha@dtu.dk, marcin.zajaczkowski95@gmail.com, patalan.m@gmail.com
* Correspondence: alexandre.alapetite@alexandra.dk

**Abstract:** Positioning is an essential aspect of robot navigation, and visual odometry an important technique for continuous updating the internal information about robot position, especially indoors without GPS. Visual odometry is using one or more cameras to find visual clues and estimate robot movements in 3D relatively. Recent progress has been made, especially with fully integrated systems such as the RealSense T265 from Intel, which is the focus of this article. We compare between each other three visual odometry systems and one wheel odometry, on a ground robot. We do so in 8 scenarios, varying the speed, the number of visual features, and with or without humans walking in the field of view. We continuously measure the position error in translation and rotation thanks to a ground truth positioning system. Our result show that all odometry systems are challenged, but in different ways. In average, ORB-SLAM2 has the poorer results, while the RealSense T265 and the Zed Mini have comparable performance. In conclusion, a single odometry system might still not be sufficient, so using multiple instances and sensor fusion approaches are necessary while waiting for additional research and further improved products.

**Keywords:** odometry; camera; positioning; navigation; indoor; robot

## 1. Introduction

Robot localization within its environment is one of the fundamental problems in the field of mobile robotics. One way of tracking this problem is to use vision-based odometry (VO), that is capable of accurately localizing robots' position with low drift over long trajectories even in challenging conditions. Many VO algorithms were developed that are categorized into direct, semi-direct and feature-based on what image information is used in order to estimate egomotion. The hardware setup varies, as camera images can be captured in monocular or stereo vision. Many augmentations of VO are available, which perform sensor fusion of computed egomotion with other sensors that can refine the trajectory such as IMU and depth sensors.

There are many benchmark comparisons of VO algorithms, usually focusing on one of VO applications. Benchmarks compare various VO algorithms in terms of translation and rotation error, memory usage, computation time and CPU consumption. [1] compared monocular visual-inertial odometry for six degree of freedom (6DoF) trajectories of flying robots. [2] assessed performance of VO algorithms using image and depth sensors (RGB-D) for an application of mobile devices. [3] evaluated VO techniques in challenging underwater conditions. [4] KITTI dataset features benchmark for visual odometry where researchers can evaluate their algorithms on 11 unknown beforehand trajectories and the best performing VO in terms of rotation and translation error are

© 2020 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 May 2020 doi:10.20944/preprints202005.0221.v1

Peer-reviewed version available at Robotics 2020, 9, 56; doi:10.3390/robotics9030056

listed online. [5] compared filtering-based methods and optimization-based methods of VI-SLAM through experiments, it also foresees the trend of running SLAM system on dedicated hardware, for example Intel RealSense.

The purpose of this research is mainly to assess the accuracy of a new commercial hardware-software technology – the RealSense T265 from Intel – and compare it with a few other alternatives. The conducted evaluation of VO solutions is especially meant for practitioners, serving as a guideline in choosing the right VO solution. Zed Mini and RealSense provide out-of-a-box hardware that contain dedicated software solutions, bringing the best of its hardware-software synergy. The RealSense and Zed Mini performance will be compared to well-established ORB-SLAM2 algorithm. ORB-SLAM2 was chosen for this purpose, since it is one of the most widely used SLAM algorithms for VO purpose, therefore it is easier for readers to establish common frame of reference regarding accuracy performance. Moreover, Zed Mini and RealSense T265 will be compared to wheel odometry and evaluated against ground truth obtained by motion capture system OptiTrack.

## 2. Materials and Methods

### *2.1. Test environment*

The experiments were conducted indoors, in a controlled area, on a flat, non-slippery surface. As visible in Figure 1, we used some pieces of dark textile to make the scene more, or less, feature-rich, i.e. to adjust the quantify of visual clues in the robot's field of view. Indeed, visual odometry systems are especially challenged when facing uniform surfaces such as a long white wall. Another important parameter affecting the quality of visual odometry is whether those visual clues are static (not moving) or whether some of them might be moving (dynamic). In order to compare the robustness of the different odometry systems over moving visual elements, we asked 3 persons to walk repeatedly along the walls of the experiment area.

It is important to note that in order to ensure that we are testing the different systems in a fair way, all visual odometry systems were running in parallel, meaning that there were exposed to exactly the same environment. We believe this is an interesting approach to compare VO systems.

### *2.2. Ground truth*

An OptiTrack[1] system was used as a ground truth system. It is a motion capture system that is capable of tracking objects with positional error less than 0.3mm and rotational error less than 0.05°, using seven Prime 13 cameras[2] (cf. Figure 1), which can detect passive markers placed on the tracked object. Five markers placed on the top of the robot were used to track robots 6DoF position. The pivot point was marker location where the final position was calculated for. In the experiment, pivot point was located in the centre of the camera, which is was ~25cm in front of the centre of the robot.

[1] https://optitrack.com/motion-capture-robotics/

[2] https://optitrack.com/products/prime-13/

*Preprints* (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 May 2020 doi:10.20944/preprints202005.0221.v1

Peer-reviewed version available at *Robotics* **2020**, *9*, 56; doi:10.3390/robotics9030056

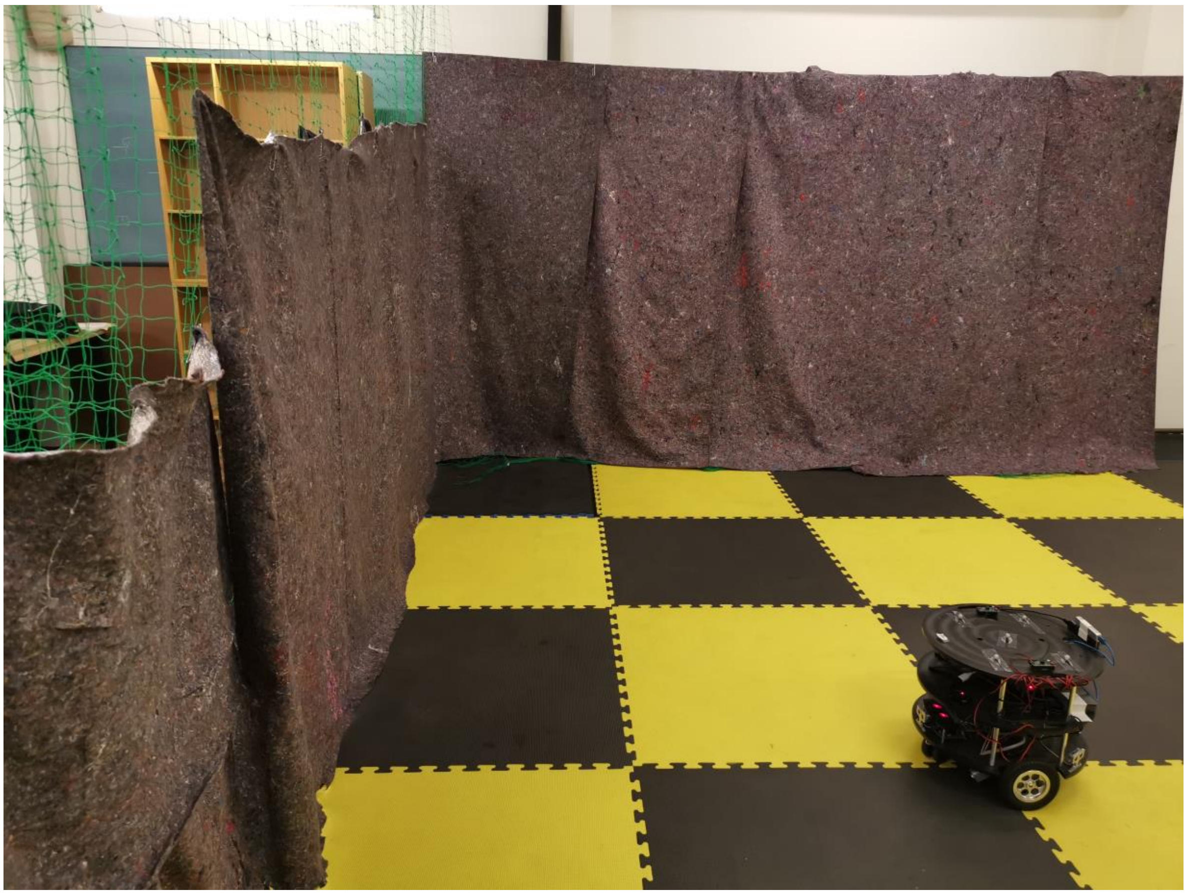


**Figure 1.** A photo of the test environment, with the robot, and illustrating how we reduce the number of visible features.

**Figure 2.** A photo of one of the seven OptiTrack cameras.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 May 2020 doi:10.20944/preprints202005.0221.v1

Peer-reviewed version available at Robotics 2020, 9, 56; doi:10.3390/robotics9030056

### 2.3. Robot setup

The platform on which the tests were performed is a Parallax Arlo[3] Robot Platform System (cf. Figure 1), commercialised by Parallax Inc.[4]. This platform was utilized as the physical framework for visual odometry research. Two standard wheels with motors on the sides of the robot and two castor wheels in front and back, cause the platform to be nonholonomic. The platform has two battery packs (12V, 7Ah) connected to DHB-10 Motor Controller and Propeller Activity board. It also supplied the Nvidia Jetson TX2 and Raspberry Pi 3. The Activity board was connected to the Raspberry Pi which delivered the control signals for the motors.

To improve the efficiency of the visual odometry computations, one has divided the vision systems into two independent parts, running in parallel. First one running under the Raspberry Pi 3, having Arlobot's Activity board and Intel RealSense T265 connected to. The other system was running on Nvidia Jetson TX2, which has significantly higher computation possibilities. The board was powered by the 12V output from the battery pack mounted on the Arlobot's platform. The Nvidia Jetson had connected only one external camera – ZED Mini delivered by Stereolabs. Apart from holding the zed-mini computations, this station held the ORB-SLAM2 algorithm on itself as well.

Both cameras (Intel RealSense T265 and ZED Mini) were mounted on top of each other in the front of the Arlobot platform. The mounting position was shifted from the robot's rotation axis to the front of it by 25cm.

### 2.4. Software setup

The robot software packages operate on ROS[5] (Robot Operating System, from the Open Source Robotics Foundation), more precisely ROS Kinetic under the GNU/Linux distribution Ubuntu 16.04 LTS. One has used modified "ROS packages for ArloBot" on Raspberry Pi to obtain communication with the "Parallax Activity board[6]" (microcontroller) on the robot.

### 2.5. Intel RealSense Tracking Camera T265

The RealSense T265 camera is a tracking camera that was released by Intel in March 2019 at a price of 199 USD. It includes two fisheye lens sensors as well as an Inertial Measurement Unit (IMU). The Visual SLAM algorithm runs directly on built-in Intel Movidius Myriad 2 VPU. This gives very low latency between movement and its reflection in the pose, as well as low power consumption that stays around 1.5W. Since all the computations are performed in real-time onboard it does not require any computations to be held on the master computer.

[3] https://www.parallax.com/product/28966

[4] https://www.parallax.com/product/arlo-robotic-platform-system

[5] https://www.ros.org

[6] https://www.parallax.com/product/32912

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 May 2020 doi:10.20944/preprints202005.0221.v1

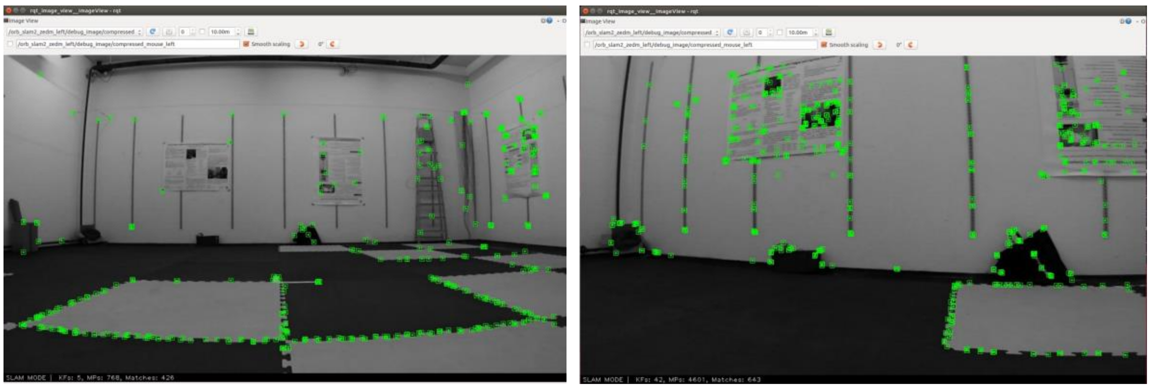
Peer-reviewed version available at *Robotics* **2020**, *9*, 56; doi:10.3390/robotics9030056

### 2.6. *ZED Mini*

The ZED Mini[7] is a visual-inertial depth camera, that features dual high-speed 2K image sensors and a 110° field of view. With an eye separation of 63 mm, the camera senses depth from 0.1 meters to 12 meters with improved accuracy and fewer occlusions in the near range. Using visual-inertial odometry technology, inertial measurements are fused at 800Hz with visual data from the stereo camera. Sensor fusion allows for accurate tracking even when visual odometry gets lost due to insufficient amount of feature matches.

### 2.7. *ORB-SLAM2*

ORB-SLAM2 is a complete SLAM system [6] for monocular, stereo and RGB-D cameras that achieve state-of-the-art accuracy in many environments. In this study the monocular setup was used. The main three components that are executed in parallel are: 1) tracking thread which estimates pose of current frame and optimizes its position minimizing the reprojection error applying motion-only Bundle Adjustment (BA). 2) local mapping thread that saves new keyframes, performs local BA and saves Visual Words for later BoW place recognition. 3) the loop closure to detect large loops using BoW approach and refine trajectory first performing pose-graph optimization and lastly performing full BA in order to obtain optimal structure and motion solution.


rqt_image_view__ImageView - rqt
Image View
/orb_slam2_zedm_left/debug_image/compressed
10.00m
/orb_slam2_zedm_left/debug_image/compressed_mouse_left
Smooth scaling
0°
rqt_image_view__ImageView - rqt
Image View
/orb_slam2_zedm_left/debug_image/compressed
10.00m
/orb_slam2_zedm_left/debug_image/compressed_mouse_left
Smooth scaling
0°

```
SLAM MODE |  KFs: 5, MPs: 768, Matches: 426
```

```
SLAM MODE |  KFs: 42, MPs: 4601, Matches: 643
```

**Figure 2.** Two screenshots of ORB-SLAM2 representation of the detected features, far from walls (left) and close to a wall (right), in a scenario with many features (e.g. additional items, posters), visualised via the ROS tool rqt_image_view. The full camera data can be downloaded (cf. *Supplementary materials*).

[7] https://www.stereolabs.com/blog/introducing-zed-mini/

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 May 2020 doi:10.20944/preprints202005.0221.v1

Peer-reviewed version available at *Robotics* **2020**, *9*, 56; doi:10.3390/robotics9030056

### *2.8. Scenarios*

For each scenario, the robot starts by driving forward for 3 meters. And then it makes three full turns plus 180° (1260° in total) around its own spot. The process is repeated four times during each scenario.

We repeated the experiments for 3 different parameters, giving a total of 8 combinations (cf. Table 1):

- *Quantity of visual features*: We changed the number of visual features in the field of view: either "*Many*" with several paper posters on the walls to increase the number of visual clues, or "*Few*" with mostly grey walls. The floor is unchanged between conditions.
- *Robot speed*: We made the robot drive at two different reference speeds: either "*Fast*" with ~1.07m/s linear speed for ~2.52rad/s angular speed (when the robot turns), or "*Slow*" with ~0.36m/s linear (i.e. ~3 times slower) speed for ~0.35rad/s angular speed (i.e. ~7 times slower).
- *Moving visual elements*: We made the visual environment more, or less stable: either "*Static*" with nothing moving, or "*Dynamic*" with some persons constantly walking along the walls around the room.

**Table 1.** List of scenarios to test the 8 combinations of 3 conditions.

| Quantity of visual features | Robot speed | With moving visual elements (Dynamic) or without (Static) |
|---|---|---|
| Many | Slow | Static |
| Many | Slow | Dynamic |
| Few | Slow | Static |
| Few | Slow | Dynamic |
| Many | Fast | Static |
| Many | Fast | Dynamic |
| Few | Fast | Static |
| Few | Fast | Dynamic |

## 3. Data Analysis

The datasets come from three different sources, namely OptiTrack system, Raspberry Pi and Jetson TX2. The first thing to do is to transform the data into same format. Because the robot runs only in a 2D plane, the position of different methods can be transformed into robot position (x, y) and robot orientation theta. Afterwards, the three datasets are synchronized and merged into one. In order to analyse the performance of different visual odometry systems relative to the OptiTrack, some columns of the dataset such as velocity and OptiTrack are interpolated (filled with previous values if the cells are empty) since the OptiTrack data does not come at the same timestamp as the others. Before calculating the errors, the ORB-SLAM2 data is scaled and the scale coefficient is found by gradient decent. Besides, the robot wheel odometry data also needs to be transformed to the camera centre so that all measurements are in the same coordinate system. An example of how the data looks like at this stage can be seen in Figure 3.

*Preprints* (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 May 2020 doi:10.20944/preprints202005.0221.v1

Peer-reviewed version available at *Robotics* **2020**, *9*, 56; doi:10.3390/robotics9030056

See details in the "supplementary material" section for the source code and the data.

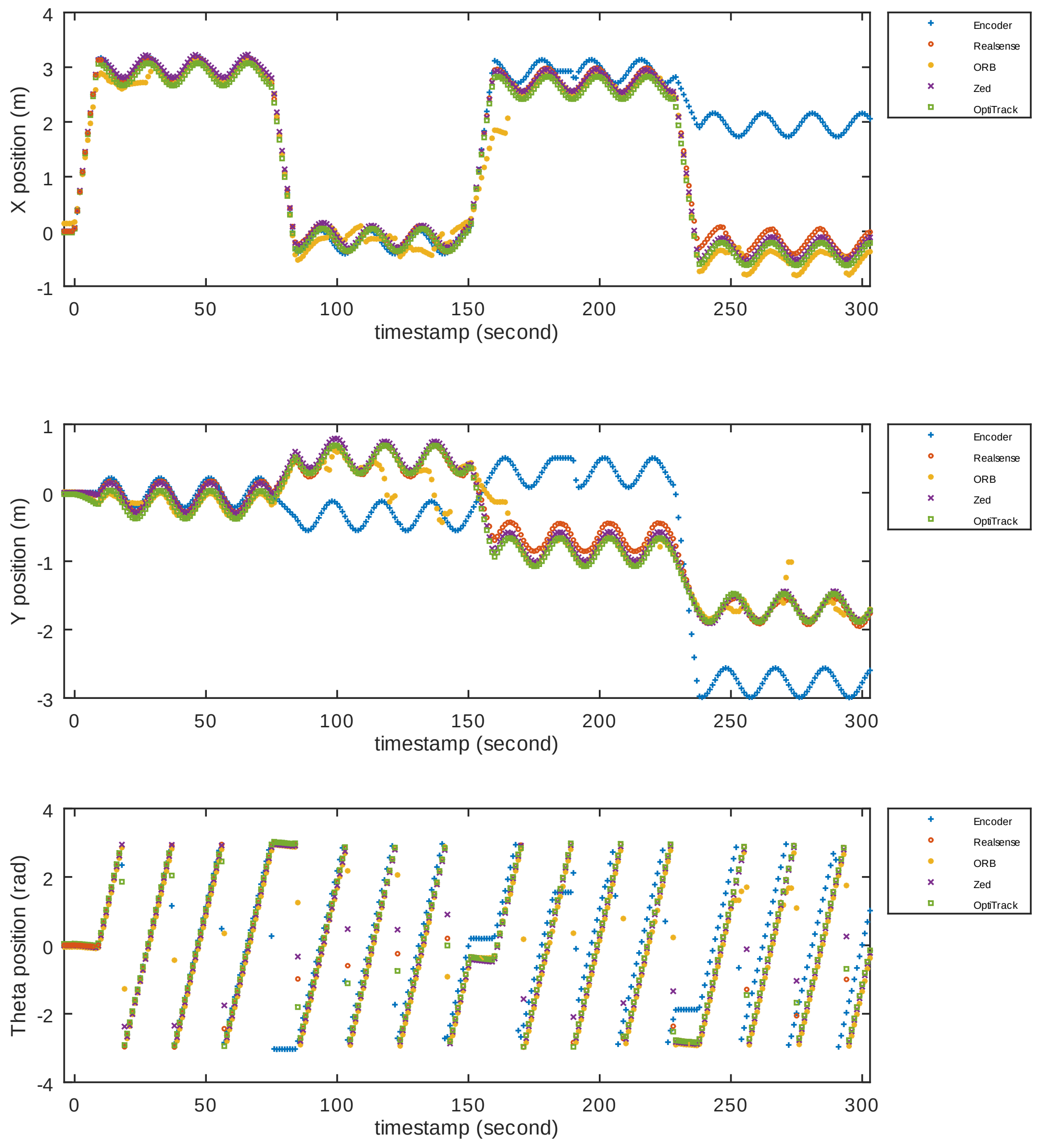


**Figure 3.** Example of raw data during a single experiment run. The scenario was "many features, slow, static environment). The upper graph reports the x-position of the robot over time (5 minutes in this example) according to each odometry, which can be compared with the ground truth (OptiTrack series, green squares). The middle graph if for the y-position of the robot, while the bottom graph is for the rotation (orientation) of the robot.

The error of the visual odometry system is evaluated from translation error and the rotation error, where the translation error is calculated by the distance offset relative to the ground truth and the rotation error is calculated by the angle offset. In addition, the incremental of the errors over time is also computed.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 May 2020 doi:10.20944/preprints202005.0221.v1

Peer-reviewed version available at Robotics 2020, 9, 56; doi:10.3390/robotics9030056

## 4. Results

### 4.1 Descriptive statistics

In order to get a better understanding of the data, a first round of descriptive statistics is performed. The two most informative visualisations are reported in Figure 4 and Figure 5, respectively for translation error (i.e. robot {x, y} position estimation error) and for rotation error (i.e. robot orientation error).

We observe that wheel odometry always provide a poor translation (Figure 4) and rotation (Figure 5) estimation but does so in a quite consistent manner: wheel odometry is indeed not much affected by the scenarios – not even speed – which is not surprising in non-sliding condition. The measurements are more consistent during translations than during rotations.

Expectedly, we observe that the visual odometry systems are much affected by the challenging scenarios, with sometimes big errors for all of them, especially ORB-SLAM2, and especially during rotations.

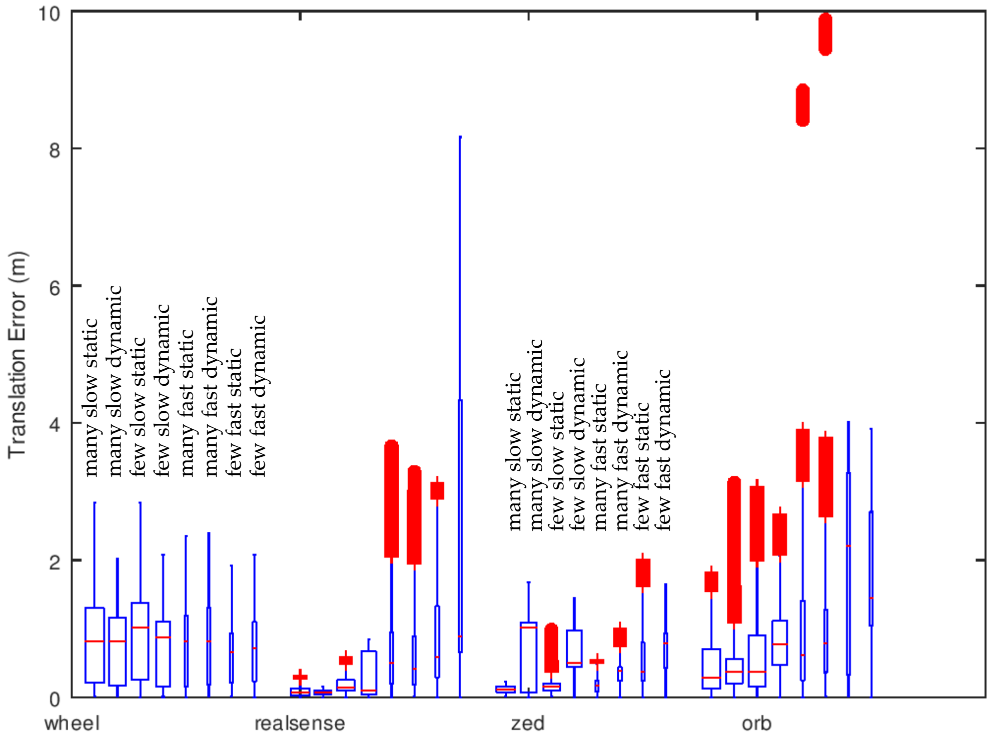


**Figure 4.** For each of the odometries (Wheel encoders, RealSense T265, Zed-Mini, ORB-SLAM2), we report the median translation error (red horizontal line), the 75% observed translation errors (blue rectangle box), the 95% observed translation errors (blue error lines), as well as outliers (red crosses above the rest). For each odometries, from left to right, are reported the scenarios: “many slow static”, “many slow dynamic”, “few slow static”, “few slow dynamic”, “many fast static”, “many fast dynamic”, “few fast static”, “few fast dynamic”.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 May 2020 doi:10.20944/preprints202005.0221.v1

Peer-reviewed version available at Robotics 2020, 9, 56; doi:10.3390/robotics9030056

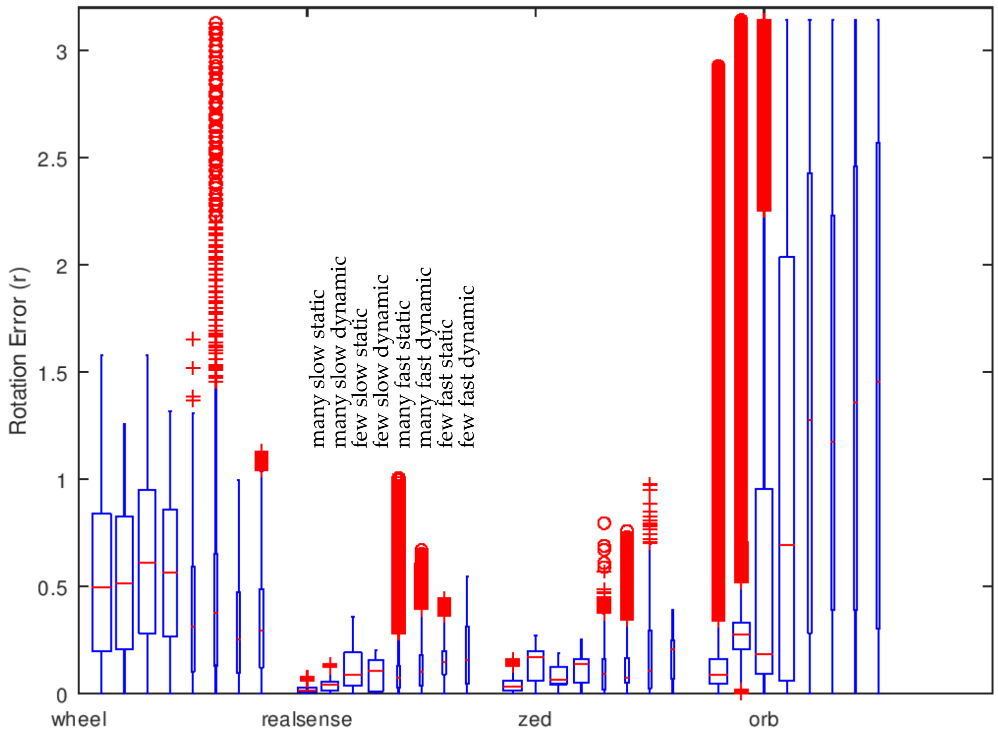


**Figure 5.** For each of the odometries (Wheel encoders, RealSense T265, Zed-Mini, ORB-SLAM2), we report the median rotation error (red horizontal line), the 75% observed rotation errors (blue rectangle box), the 95% observed rotation errors (blue error lines), as well as outliers (red crosses above the rest). For each odometries, from left to right, are reported the scenarios: “many slow static”, “many slow dynamic”, “few slow static”, “few slow dynamic”, “many fast static”, “many fast dynamic”, “few fast static”, “few fast dynamic”.

## *4.2 Statistical analysis*

In order to help identifying relevant differences, we did a light statistical analysis, with a series of t-Tests of the type “*Two-Sample Assuming Unequal Variances*” from the “Analysis ToolPak” of Excel (Microsoft Office 365 version 1910). Table 2 contains results for the average translation error, while Table 3 contains the average rotation error, across all scenarios.

**Table 2.** Summary of the t-Tests of the average translation error (in meters), between odometries.

| | | Wheel encoders | RealSense T265 | Zed-Mini | ORB-SLAM2 |
|---|---|---|---|---|---|
| Mean | | 0,8432m | 0,6706m | 0,4249m | 1,1710m |
| Variance | | 0,0144 | 1,3356 | 0,1120 | 0,3683 |
| Observations | | 24 | 24 | 24 | 24 |
| P-value | Wheel encoders | | 2,37E-01 | 1,54E-06** | 7,79E-03** |
| | RealSense T265 | | | 1,63E-01 | 3,44E-02* |
| | Zed-Mini | | | | 3,25E-06** |

P-values marked with one asterisk are better than 0,05; two asterisks when better than 0,01.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 May 2020 doi:10.20944/preprints202005.0221.v1

Peer-reviewed version available at Robotics 2020, 9, 56; doi:10.3390/robotics9030056

**Table 3.** Summary of the t-Tests of the average rotation error (in radians), between odometries.

| | | Wheel encoders | RealSense T265 | Zed-Mini | ORB-SLAM2 |
|---|---|---|---|---|---|
| Mean | | 0,4682 rad | 0,1057 rad | 0,1061 rad | 1,0409 rad |
| Variance | | 0,0167 | 0,0092 | 0,0060 | 0,3314 |
| Observations | | 24 | 24 | 24 | 24 |
| P-value | Wheel encoders | | 2,80E-14** | 1,63E-14** | 3,52E-5** |
| | RealSense T265 | | | 4,93E-01 | 2,21E-08** |
| | Zed-Mini | | | | 2,05E-08** |

P-values marked with one asterisk are better than 0,05; two asterisks when better than 0,01.

The statistical analysis confirmed the main trends observed in the descriptive statistics.

### *4.3. Main findings*

Wheel odometry is not much affected by the different scenarios, not even by the change of speed, leading to more consistent values, especially during translation. This is not surprising because the floor surface remained identical. However, the standard deviation of wheel odometry is typically higher than for the visual odometries, making it generally less precise, especially during the easy scenarios (i.e. one or more of: low speed, many features, static environment).

But the scenarios do have a significant effect on visual odometries. In our tests, speed had the greatest effect (in the "Fast" scenarios, linear speed was ~3 times higher and angular speed ~7 times higher), followed by the number of features, while the static vs. dynamic environment had the smallest effect.

Among the visual odometries, ORB-SLAM2 has the poorer results in our experiments, both in translation ($p < 4E-2$) and rotation ($p < 4E-5$), and for all scenarios. This materialises in a higher imprecision, a higher standard deviation, and more outliers than other methods.

Except for a few outliers, the RealSense T265 and the Zed Mini have comparable results in average ($p > 0.1$). RealSense is a bit more negatively affected by speed than Zed, especially during translation.

## 4. Discussion

### *4.1. Camera lens types*

The RealSense T265 has a wide field of few, making it able to potentially spot many more visual features than the ZED Mini, but the drawback is in principle a poorer image quality. In our experiments, in the end, it did not seem to make a significant difference, although we cannot tell which part of the results is due to the lens and which part is due to a difference of processing. The RealSense T265 would arguably have an advantage in scenarios where the interesting visual features are in the periphery of the camera field of view, i.e. not visible by cameras with narrower field of views.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 May 2020 doi:10.20944/preprints202005.0221.v1

Peer-reviewed version available at Robotics 2020, 9, 56; doi:10.3390/robotics9030056

### 4.2. *Processing power*

On a robot or drone, aspects such as total weight, price, and power consumption are essential factors. On those factors, the RealSense T265 globally wins over the Zed Mini and ORB-SLAM2, as it comes with built-in data processing, while the other visual odometries require an additional powerful computer board such as an NVIDIA Jetson or similar.

### 4.3. *Multiple sensors & Sensor fusion*

In our experiments, we compared the different method with one single sensor for each of them, but it would be possible to combine several cameras for potentially better quality. This is especially doable for larger robots.

A similar approach is to combine different types of sensors with a sensor fusion approach. Outside, such a sensor fusion could be done with e.g. GPS, and indoor for instance with fiducial markers such as ArUco markers or other 2D-barcodes. Noticeably, the RealSense T265 offers a built-in sensor fusion mechanism that can be fed with wheel odometry, but this was outside the scope of those experiments.

## 5. Conclusions

As shown by the experiments, the Intel RealSense T265 compares well with the state of the art, especially when accounting for price, built-in processing power, and sensor fusion abilities. However, a single RealSense T265 does not solve the visual odometry challenge fully. Therefore, even for basic indoor navigation needs, several sensors or techniques must be combined. For the time being, visual odometry remains a domain with room for additional research and improvements.

**Supplementary Materials:** The source code of the data analysis, as well as the raw data (in ROS Bag format) for the different odometry systems is available online from:
https://github.com/DTU-R3/visual_odometry_comparison

**Author Contributions:** Conceptualization, Alexandre Alapetite and John Paulin Hansen; Data curation, Zhongyu Wang; Formal analysis, Alexandre Alapetite and Zhongyu Wang; Funding acquisition, Alexandre Alapetite and John Paulin Hansen; Investigation, Marcin Zajączkowski and Mikolaj Patalan; Methodology, Alexandre Alapetite; Project administration, Alexandre Alapetite and John Paulin Hansen; Resources, John Paulin Hansen; Software, Zhongyu Wang, Marcin Zajączkowski and Mikolaj Patalan; Supervision, Alexandre Alapetite; Validation, Alexandre Alapetite; Visualization, Alexandre Alapetite and Zhongyu Wang; Writing – original draft, Alexandre Alapetite, Marcin Zajączkowski and Mikolaj Patalan; Writing – review & editing, Alexandre Alapetite and John Paulin Hansen.

**Funding:** This research was partially funded by InfinIT (the innovation network for IT in Denmark) and by the Bevica Foundation, Denmark.

**Conflicts of Interest:** The authors declare no conflict of interest.

*Preprints* (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 May 2020 doi:10.20944/preprints202005.0221.v1

Peer-reviewed version available at *Robotics* **2020**, *9*, 56; doi:10.3390/robotics9030056

## References

1. Delmerico, J; Scaramuzza, D; A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots. In 2018 IEEE International Conference on Robotics and Automation (ICRA) 2018, pp. 2502-2509. doi:10.1109/ICRA.2018.8460664
2. Angladon, V; Gasparini, S; Charvillat, V; Pribanić, T; Petković, T; Đonlić, M; Ahsan, B; Bruel, F; An evaluation of real-time RGB-D visual odometry algorithms on mobile devices. Journal of Real-Time Image Processing 2019, 16(5), pp. 1643-1660. doi:10.1007/s11554-017-0670-y
3. Joshi, B; Rahman, S; Kalaitzakis, M; Cain, B; Johnson, J; Xanthidis, M; Karapetyan, N; Hernandez, A; Li, AQ; Vitzilaios, N; Rekleitis, I; Experimental Comparison of Open Source Visual-Inertial-Based State Estimation Algorithms in the Underwater Domain. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). arXiv preprint arXiv 2019, 1904.02215.
4. Geiger, A; Lenz, P; Stiller, C; Urtasun, R; Vision meets robotics: The KITTI dataset. The International Journal of Robotics Research 2013, 32(11), pp. 1231-1237. doi:10.1177/0278364913491297
5. Chen, C.; Zhu, H.; Li, M.; You, S. A Review of Visual-Inertial Simultaneous Localization and Mapping from Filtering-Based and Optimization-Based Perspectives. Robotics 2018, 7, 45. doi:10.3390/robotics7030045
6. Mur-Artal, R; Tardós, JD; Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Transactions on Robotics 2017, 33(5), pp. 1255-1262. doi:10.1109/TRO.2017.2705103