Visual-inertial sensor fusion via machine learning

Habib Boloorchi[1][a], He Bai[1][b] and Christopher Crick[2][c]

[1]*Department of Computer Science, Oklahoma State University, My Street, MyTown, MyCountry*

[2]*Department of Computing, Main University, MySecondTown, MyCountry*

{f_author, s_author}@ips.xyz.edu, t_author@dc.mu.edu

Keywords: The paper must have at least one keyword. The text must be set to 9-point font size and without the use of bold or italic font style. For more than one keyword, please use a comma as a separator. Keywords must be titlecased.

Abstract: Robot localization is a well-studied problem, with many competing solutions, from odometry to inertial measurement to SLAM to GPS and other beacon-based approaches. Some methods work only with certain modalities: measurement of wheel odometry is useless for legged or flying locomotion, for instance. Visual landmark detection and dead reckoning are useful to robots with many different structures and means of locomotion, and can be implemented with very low-cost cameras and inertial measurement units (IMUs). However, existing approaches require fairly sophisticated hardware with precise, rapid, real-time control loops in order to fuse the visual inertial odometry data adequately. We present algorithms to better handle the uncertainty which stems from noisy, inconsistent IMU and machine vision data, using a machine learning approach to provide a robot better awareness of its location and the effect of its self-motion on visual cues. This approach yields robust localization performance which generalizes across very different robot platforms using low-cost sensing and computation.

1 INTRODUCTION

Data fusion in Robotics has a huge and wide spectral of fields. The aim of most scientists in this area is helping robots to be more autonomous than before. One of the puzzles in this area is Navigation and localization is its bottleneck because a robot needs to have an estimation its state. In other words, having consciousness on where it is, will be a must for decision on next steps . In our research, we found flexibility in data fusion for the process of localization baffling.Not only, Concise timing is matter but we also found lack of a Software that help us handle devices that can do localization which does not required expert, cost and energy. This necessity to consistency and having experienced in mechanical and electronic engineers and an expensive device lead us to create a device that can work better with less consistency in data fusion. The demand to having data fusion stems from the role of Visual-inertial odometry. Visual-inertial odometry needs to gather data from both perception and Inertial Measurement Unit(IMU). This Odometry can estimate Coordinate in 3 Dimensional world,which Opticounters and GPS can only do it in 2 Dimension, when corresponding data is received in a consistent latency.

The Perceptio part is added to inertial odometry to be more Robust to uncertainty. Vision of localization can handle the situation in a variety of environments such as GPS unfriendly places and inertial can overcome situations such as dark or low textured, which could not let the camera has good perception. These challenges shows up when we use visual inertial odometry:

- Combining data that comes from IMU and camera should offer the least latency and be so accurate. and if not algorithms could not work well.
- Even if we could create a device that works accurately calibration each time we use it in order to have good results would cost so much time.
- Cost of stable Visual inertial odometry sensor is so high
- Most of the Visual inertial Odometry applications are not enough user-friendly for calibration and regular use. you need to be an expert in Control to know the terms.

[a] https://orcid.org/0000-0000-0000-0000

[b] https://orcid.org/0000-0000-0000-0000

[c] https://orcid.org/0000-0000-0000-0000

In order to have more sense of these deficiencies, we can come up with an example. In Agriculture Departments, we have a big population of plants that we need to know the effect of each variable on plants. These kinds of research need to observe every plant and this can be tested by taking Photo of each plant. Visual inertial odometry can help us to localize a self-announcement mobile vehicle that can pull the trigger of shutter. Now it comes more baffling when we need an expert in control who is also a scientist in agriculture. We offered a machine learning approach that can resolve the combining data, expenses, and requirement of an expert. We get data to do the preprocessing that does not need the information of the camera or cameras for the perception. It can work with every visual-inertial sensors. These sensors can be cheap because the learning procedure is robust to noises. In our method we does perception using essential matrix then we take advantage of Random Forest Regression method to estimate the Position of the robot.

2 RELATED WORKS

When it comes to reduce the uncertainty, localization has a vital role (**?**). In other words, Odometry is using motion data comes from sensors in order to reduce the uncertainty of robots position in environment(Huang et al., 2017; Valencia and Andrade-Cetto, 2018).This data can come from rotation of wheels, GPS, IMU(Inertial Measurement Units), Cameras. In large-scale Domains GPS (Global Positioning System) is one of the most popular for localization but it has some draw backs in smaller space. For example it does not have enough accuracy and also there is some places it is not feasible to use it. For instance, under water, as a GPS-denied circumstance, we can realize that we need a robocentric approach for other ways for odometry (Saska et al., 2017). IMU(Inertial Measurement Units) is one of the tools that can be helpful in approximation of the trajectory (Gui et al., 2015). However, when we use IMU some problems as a matter of noisy data and motor noises on devices can be appeared. Furthermore, we need to have trajectory in circumstances as an example Mars Rovers specifically we cannot use it alone.(**?**) Rotary encoders that can estimate wheels rotation with getting pulse from opto-counters is a technique for wheeled robots while this instrument cannot be useful for highly dynamic cases such as bipedal robots or in three dimensional spaces for flying robots. (Bloesch et al., 2015). The human inspired visual ego-motion can be counted as a liable approach to extract trajectory(Engel et al., 2018). This Motion estimation, which apply machine vision, is called Visual Odometry. One of the most application of Visual Odometry is SLAM (Simultaneous Localization and Mapping) (Mur-Artal et al., 2015; Mur-Artal and Tardós, 2017; Forster et al., 2017; Mueggler et al., 2017). One of the problems that ORB-SLAM1(Mur-Artal et al., 2015) ,as an example of best Visual Odometry shows is drifting stems from lack of having closed loop algorithms.(Mur-Artal and Tards, 2017). Another, disadvantage that can lead to get lost as a robot is having fast angular movements that can lead to miss of landmarks. Since these landmarks are critical sources to estimate the position of our path, robot can lose its position. Thus, vision need to be aided by IMU. In addition, the integration of IMU can offers some benefits(Leutenegger et al., 2015). accuracy can be named as one of these advantages(Bloesch et al., 2017; Schneider et al., 2018; Sun et al., 2018).

We can integrate images and Inertial data via variety of algorithms. Using probability to estimate the better likelihood of location(Bowman et al., 2017) is one of the approaches to estimate the position. However, It is more popular to do this fusion can be done by extended Kalman filter(Lynen et al., 2013). The advantage of Extended Kalman Filter and Uncented Kalman Filter from Kalman filter is using Jacobian to extract the position of the robot in nonlinear system (Julier and Uhlmann, 2004; Wan and Van Der Merwe, 2000).

The major difference between UKF and EKF is their ability to estimate in the nonlinear system. In other words EKF is more for weak nonlinearity system while UKF is for high nonlinearity system. (St-Pierre and Gingras, 2004)

Bloesch et al (Bloesch et al., 2015; Bloesch et al., 2017)mars-rover-slam offers Iterated Extended Kalman Filter in order to robust the ability of estimation of trajectory. Iterations can robust the ability to find the maximum likelihood to estimate the position of a robot by robocentric formula (Bloesch et al., 2015; Bloesch et al., 2017).

To do landmark detection, Bloesch et al. employed Shi-Tomasi algorithm (Shi and Tomasi, 1993). In this method, landmarks is extracted by utilization of three times down sampling and more probable landmarks between candidates will be chosen. ROVIO (Robust Visual Inertial odometry) uses these landmarks and by warping of land marks the movement in perception can be estimated.

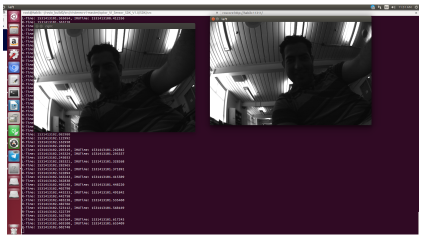
Other innovations can conceive the ego-motion (Zhou et al., 2017) using unsupervised learning. To understand the motion. This algorithm uses Convolutional Neural Network to find out the warping.

As we have three dimensional views using RGB-d or Stereo cameras we can both interpret angular and linear movements.

Although, There are several datasets that contains ground truth in 6 degree of freedom that shows angles and also position of the a flying robot and a get data from inertial sensors and stereo cameras(**?**),in this article, we want to make sure that we process data real-time and we also want to use our algorithm on flying robots. so we suggested a creation of device that able to get inertial and perception data relatively. we also suggested a creation of anew device to collect a data to have relative image and IMU data and having less delay with getting th data.

our perception is based on the aforementioned Rovio (Bloesch et al., 2017), and integrate it to IMU data. Instead of using iterated Extended Kalman Filter, we are going to take advantage of supervised Learning to create trajectory. Although, our method cannot show the result as accurately as Rovio, we offer benefit of having more robustness to latency inconsistency in devices such as the one show in Figure 8 which has received in a progress without pattern that we see in Figure 1.

In our approach, we create reliable algorithm which can give function better than other approach when we have latency.



```
L-Time: 1531413101.363654, IMUTime: 1531413100.412336
R-Time: 1531413101.363218
R-Time: 1531413102.082980
R-Time: 1531413102.122992
R-Time: 1531413102.162950
R-Time: 1531413102.202918
L-Time: 1531413102.203319, IMUTime: 1531413101.262842
L-Time: 1531413102.243324, IMUTime: 1531413101.295557
R-Time: 1531413102.243033
L-Time: 1531413102.283321, IMUTime: 1531413101.328268
R-Time: 1531413102.282965
L-Time: 1531413102.323214, IMUTime: 1531413101.371891
R-Time: 1531413102.322894
L-Time: 1531413102.363243, IMUTime: 1531413101.415509
R-Time: 1531413102.362838
L-Time: 1531413102.403248, IMUTime: 1531413101.448220
R-Time: 1531413102.402790
L-Time: 1531413102.443233, IMUTime: 1531413101.491842
R-Time: 1531413102.442758
L-Time: 1531413102.483238, IMUTime: 1531413101.535460
R-Time: 1531413102.482766
L-Time: 1531413102.523112, IMUTime: 1531413101.568169
R-Time: 1531413102.522739
R-Time: 1531413102.562760
L-Time: 1531413102.563164, IMUTime: 1531413101.617243
L-Time: 1531413102.603100, IMUTime: 1531413101.655409
R-Time: 1531413102.602748
```

root@habib: ~/rovio_build0/src/vi-stereo-v1-master/optor_VI_Sensor_SDK_V1.0/SDK/src
roscore http://habib:11311/
right
left

Figure 1: Data that received by our Visual inertial sensor.

3 Method

In order to have a flexible algorithm which can neglect inconsistency of delivered data, we create an algorithm that receives data in an uncertain environment. Our method divided to four steps:finding landmarks, tracking best matches of sequenced frames, finding robocenteric differentiation in distance, Fusion Data, and learning them.

3.1 Finding Landmarks

In every dataset, it is extremely hard and complex to use all data. In addition, It is too time consuming to have all the pixels as data for video or sequence images.Thus, we need to extract features. Harris is one of the most popular feature extractors that finds corners (Harris et al., 1988). We use Shi-Tomasi (Shi and Tomasi, 1993) which down-sample the region of interest to have more robustness for noisy images(Bloesch et al., 2017). Figure 2 shows the extracte features that are circumscribed by patches.

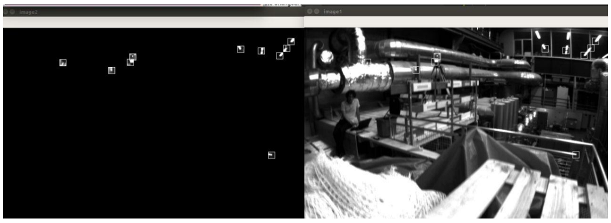


Figure 2: The right image has the features that has rectangle around it and the left is the features which are extracted

3.2 Tracking The Best Matching Features

Based on the fact that the Features are unique in each image, it is worth thinking about the challenge of finding associated landmarks in two consecutive frames. In the approach that we pursue, we create a patch around all of points of interest and flatten them to see them as vectors. Manhattan distance was our choice to find the similar matches. These matches are sorted and the difference between coordinates of best points of interests. To Normalize data, we insert both the difference of coordinates in each row as well as distance in position of tracked features in a data-frame.

3.3 Computing Rotation and Transformation

We do the Visual Odometry ignoring the presence of Inertial Measurement Unit. in this section we create a Fundamental matrix (**?**) based on the tracked points. The fundamental matrix extract the key points that can give us all information about matching point. This process needs Extrinsics and Intrinsics of the camera in order to give us Epipolar Geometry.(**?**) In the following equation we can compute Fundamental Matrix:

$$\mathbf{x}'^{\top}\mathbf{F}\mathbf{x} = 0 \tag{1}$$

x and x' are the vector of landmarks and F is fundamental matrix. To find out the distance of the landmarks and Epipolar Geometry is not enough. we need to have Essential Matrix to decompose Rotation and transformation based on that. This is the equation that tells us how to compute Essential Matrix(**?**) :

$$\mathbf{E} = (\mathbf{K}')^{\top}\,\mathbf{F}\,\mathbf{K} \tag{2}$$

K and K' is intrinsic of each camera. Essential Matrix consist of Rotation and Translation.Singular Value Decomposition can compute rotation and translation our stereo visual sensor.

3.4 Fusion Data

To do visual inertial odometry, we add linear acceleration and angular velocity from Inertial Measurement unit. These data should come together at once. However, there is some latency or difference in frequency in receiving data. To overcome this issue, we proposed a blackboard semi design pattern. In other words, we put all data in global variables and those data that should be used for process will be grabbed by an specific function at the same time.Using this approach we will not have any missing value due to having multi thread callback functions that update the data from stream which can be come from a robot or a Ros-bag.

In other words, as shown in Figure 3 we have an algorithm that collect data from Ros-bag topic, and extract features and important data from images and IMU data. In the next step these valuable information will be shared with machine learning function.

3.5 Supervised learning approach

We defined a laser data that put on devices as a ground truth (Target variable). Our plan is to estimate the position of Visual Inertial Odometry Sensor via learning. As we see in 1 the linear regression can offer less Root Mean Square error.

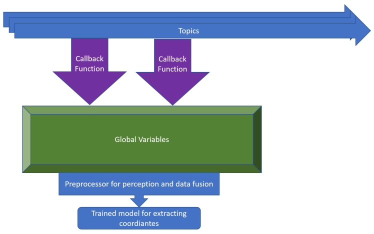
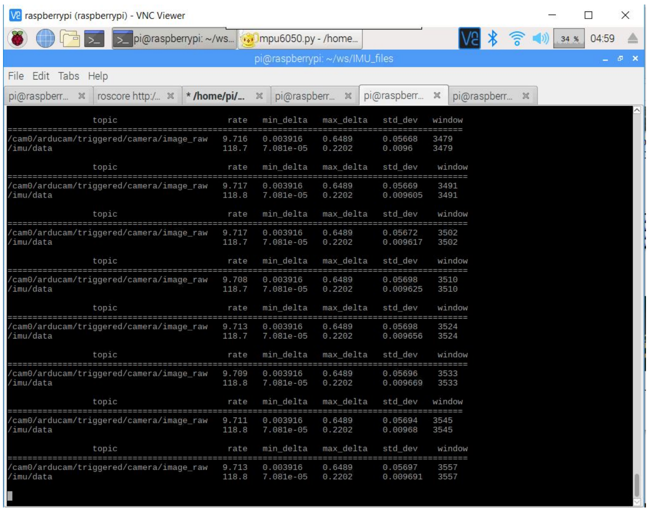

Topics
Callback Function
Callback Function
Global Variables
Preprocessor for perception and data fusion
Trained model for extracting coordiantes

Figure 3: ROS Topics are streams that convey messages. These messages can be images, texts, or data-structures such as Dictionaries. Callback functions will be run in a loop automatically and put messages from topics in global variables. data will be collected by a preprocessor function. Then Processed data will be shared with the machine learning method.



```
                topic                      rate   min_delta   max_delta   std_dev   window
==========================================================================================
/cam0/arducam/triggered/camera/image_raw   9.716  0.003916    0.6489      0.05668   3479
/imu/data                                  118.7  7.081e-05   0.2202      0.0096    3479

                topic                      rate   min_delta   max_delta   std_dev    window
===========================================================================================
/cam0/arducam/triggered/camera/image_raw   9.717  0.003916    0.6489      0.05669    3491
/imu/data                                  118.8  7.081e-05   0.2202      0.009605   3491

                topic                      rate   min_delta   max_delta   std_dev    window
===========================================================================================
/cam0/arducam/triggered/camera/image_raw   9.717  0.003916    0.6489      0.05672    3502
/imu/data                                  118.7  7.081e-05   0.2202      0.009617   3502

                topic                      rate   min_delta   max_delta   std_dev    window
===========================================================================================
/cam0/arducam/triggered/camera/image_raw   9.708  0.003916    0.6489      0.05698    3510
/imu/data                                  118.7  7.081e-05   0.2202      0.009625   3510

                topic                      rate   min_delta   max_delta   std_dev    window
===========================================================================================
/cam0/arducam/triggered/camera/image_raw   9.713  0.003916    0.6489      0.05698    3524
/imu/data                                  118.7  7.081e-05   0.2202      0.009656   3524

                topic                      rate   min_delta   max_delta   std_dev    window
===========================================================================================
/cam0/arducam/triggered/camera/image_raw   9.709  0.003916    0.6489      0.05696    3533
/imu/data                                  118.8  7.081e-05   0.2202      0.009669   3533

                topic                      rate   min_delta   max_delta   std_dev   window
==========================================================================================
/cam0/arducam/triggered/camera/image_raw   9.711  0.003916    0.6489      0.05694   3545
/imu/data                                  118.8  7.081e-05   0.2202      0.00968   3545

                topic                      rate   min_delta   max_delta   std_dev    window
===========================================================================================
/cam0/arducam/triggered/camera/image_raw   9.713  0.003916    0.6489      0.05697    3557
/imu/data                                  118.8  7.081e-05   0.2202      0.009691   3557
```

Figure 4: Rate of receiving data by Visual inertial Odometry Sensor

4 Experimental Result

The purpose of this research is finding the coordinate of the robot regardless of the accuracy and consistency of the latency in Visual Inertial Odometry Sensor. Figure 4 shows the captured data received by the device that we have built ourselves (Figure 8). As we can see that the latency is not consistent. We used Robotic Operating System to make sure that Callback functions can work parallel with main function. In addition, we used KNIME in order to create plots and extract errors. The whole algorithm can work in a regular laptop. All codes has been written in Python 2.7. The image processing algorithms are using OpenCV2 libraries. We tried several regression algorithms:

- Simple Regression Tree (SRT)
- Gradient Boosted Trees Regression (GBT)
- Random Forest Regression (RFR)
- Linear Regression (LR)
- Polynomial Regression (PR)

All of these algorithms resulted in approximately similar error. Root measure square error was our manifest measurement we had tow candidates , Linear Regression and Polynomial Regression. Between these two linear regression had less Mean absolute error. We tested linear regression with our data and its ground truth for three columns that we can see in Figure 5 - 7.

Figure 5 - 7, we can see variety of estimations compared to ground truth in position, rotation Angular-velocity and linear-velocity. These estimations are obtained by linear Regression algorithm and Dataset for this experiment is gained from EuroC Dataset (**?**).

Table 1: These Errors are obtained from Different regressions which can define best algorithms for Learning the position

	SRT	GBT	RFR	LR	PR
MAE	0.085	0.076	0.073	0.071	0.072
MSE	0.02	0.017	0.016	0.014	0.014
RMSE	0.02	0.017	0.016	0.014	0.014

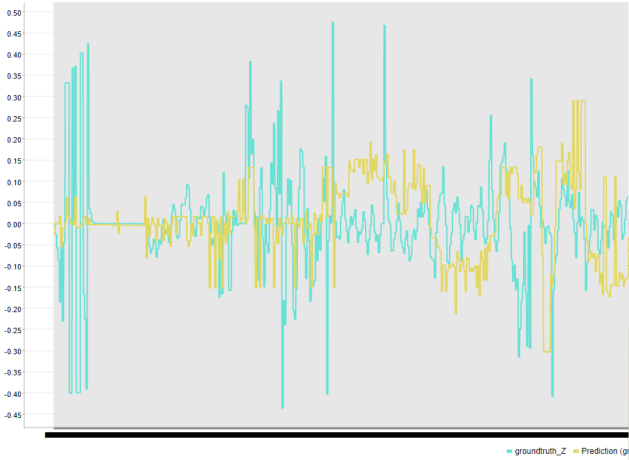


Figure 5: Estimation error of our methods: difference of Machine learning output compared to ground truth for target X.

5 CONCLUSIONS

One of the biggest challenges for our proposed approach was to create a method that can work with every devices without a need for calibration as well as handling cheap and high latency devices. Our results shows that machine learning can manipulate data regardless inconsistency and other unknown hardship of uncertainty in data collection. In Other words we obtain visual inertial models better than other approaches in a way that it is robust to latency in recieving data from Inertial Measurment Unit(IMU).

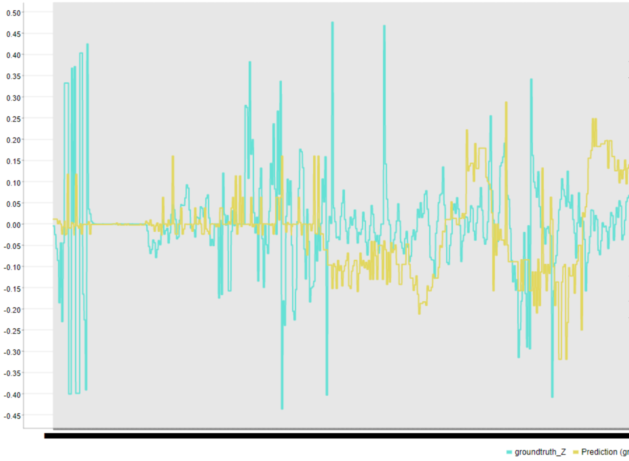


Figure 6: Estimation error of our methods: difference of Machine learning output compared to ground truth for target Y.

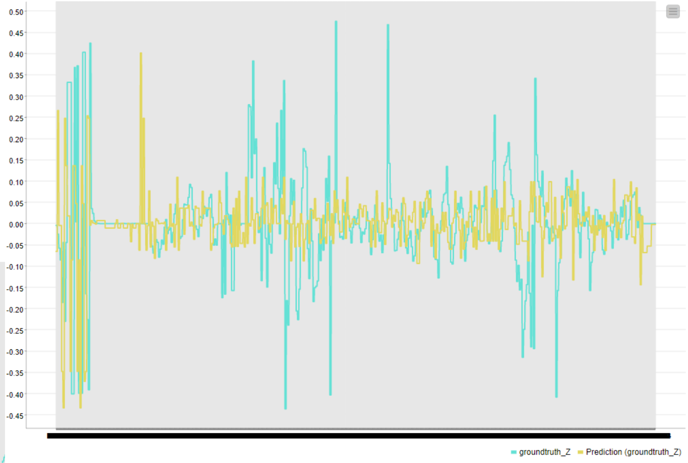


Figure 7: Estimation error of our methods: difference of Machine learning output compared to ground truth for target X.

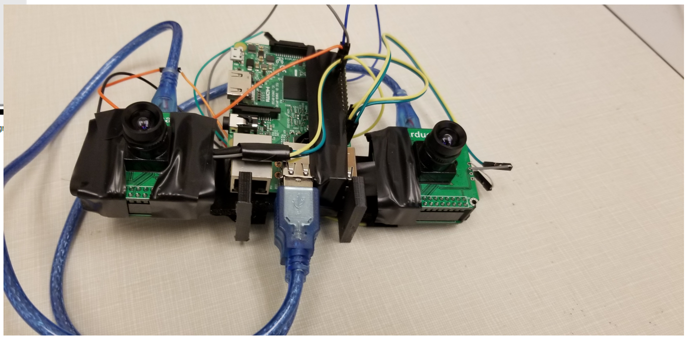


Figure 8: Visual Inertial Odometry Sensor. This sensor has Global Shutter Cameras which take each frame after receiving 10 message from Inertial Measurement Unit by having external trigger. This device is 50 times cheaper than the high precision identical device. and also has the ability to process data since it has a Raspberry pi 3 on it.

In future, We are planning to reduce the error and also use our algorithm on new unknown devices. We will also use it on mobile phones in order to have less dependency on GPS when it comes to be in GPS-free places. Next step will be using the distance of the landmarks to the robot in order to normalize and have better results from perception module of our algorithm.

REFERENCES

Bloesch, M., Burri, M., Omari, S., Hutter, M., and Siegwart, R. (2017). Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback. *The International Journal of Robotics Research*, 36(10):1053–1072.

Bloesch, M., Omari, S., Hutter, M., and Siegwart, R. (2015). Robust visual inertial odometry using a direct ekf-based approach. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 298–304. IEEE.

Bowman, S. L., Atanasov, N., Daniilidis, K., and Pappas, G. J. (2017). Probabilistic data association for semantic slam. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 1722–1729. IEEE.

Engel, J., Koltun, V., and Cremers, D. (2018). Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625.

Forster, C., Zhang, Z., Gassner, M., Werlberger, M., and Scaramuzza, D. (2017). Svo: Semidirect visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics*, 33(2):249–265.

Gui, P., Tang, L., and Mukhopadhyay, S. (2015). Mems based imu for tilting measurement: Comparison of complementary and kalman filter based data fusion. In *Industrial Electronics and Applications (ICIEA), 2015 IEEE 10th Conference on*, pages 2004–2009. IEEE.

Harris, C. G., Stephens, M., et al. (1988). A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer.

Huang, A. S., Bachrach, A., Henry, P., Krainin, M., Maturana, D., Fox, D., and Roy, N. (2017). Visual odometry and mapping for autonomous flight using an rgb-d camera. In *Robotics Research*, pages 235–252. Springer.

Julier, S. J. and Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422.

Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., and Furgale, P. (2015). Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334.

Lynen, S., Achtelik, M. W., Weiss, S., Chli, M., and Siegwart, R. (2013). A robust and modular multi-sensor fusion approach applied to mav navigation. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3923–3929. IEEE.

Mueggler, E., Rebecq, H., Gallego, G., Delbruck, T., and Scaramuzza, D. (2017). The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam. *The International Journal of Robotics Research*, 36(2):142–149.

Mur-Artal, R., Montiel, J. M. M., and Tardos, J. D. (2015). Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163.

Mur-Artal, R. and Tardós, J. D. (2017). Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262.

Mur-Artal, R. and Tards, J. D. (2017). Visual-inertial monocular slam with map reuse. *IEEE Robotics and Automation Letters*, 2(2):796–803.

Saska, M., Baca, T., Thomas, J., Chudoba, J., Preucil, L., Krajnik, T., Faigl, J., Loianno, G., and Kumar, V. (2017). System for deployment of groups of unmanned micro aerial vehicles in gps-denied environments using onboard visual relative localization. *Autonomous Robots*, 41(4):919–944.

Schneider, T., Dymczyk, M., Fehr, M., Egger, K., Lynen, S., Gilitschenski, I., and Siegwart, R. (2018). maplab: An open framework for research in visual-inertial mapping and localization. *IEEE Robotics and Automation Letters*, 3(3):1418–1425.

Shi, J. and Tomasi, C. (1993). Good features to track. Technical report, Cornell University.

St-Pierre, M. and Gingras, D. (2004). Comparison between the unscented kalman filter and the extended kalman filter for the position estimation module of an integrated navigation information system. In *IEEE Intelligent Vehicles Symposium*, pages 831–835. Citeseer.

Sun, K., Mohta, K., Pfrommer, B., Watterson, M., Liu, S., Mulgaonkar, Y., Taylor, C. J., and Kumar, V. (2018). Robust stereo visual inertial odometry for fast autonomous flight. *IEEE Robotics and Automation Letters*, 3(2):965–972.

Valencia, R. and Andrade-Cetto, J. (2018). *Mapping, planning and exploration with Pose SLAM*. Springer.

Wan, E. A. and Van Der Merwe, R. (2000). The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158. Ieee.

Zhou, T., Brown, M., Snavely, N., and Lowe, D. G. (2017). Unsupervised learning of depth and ego-motion from video. In *CVPR*, volume 2, page 7.