Adaptive Sensor Fusion using Deep Learning

Caio Fischer Silva[1,2], Paulo V K Borges[1], and Jose E C Castanho[2]

***Abstract*— A reliable perception pipeline is crucial to the operation of a safe and efficient autonomous vehicle. Given that different types of sensors have diverse sensing characteristics, fusing information from multiple sensors has become a common practice to increase robustness. Most systems rely on a rigid sensor fusion strategy which considers the sensors input (e.g., signal and corresponding covariances), without however incorporating characteristics of the environment. This often causes poor performance in mixed scenarios. In our approach, we adapt the sensor fusion strategy according to a classification of the scene around the vehicle. A convolutional neural network were employed to classify the environment and this classification used to select the best sensor configuration accordingly. We present experiments with a full size autonomous vehicle operating in a heterogeneous environment. The results illustrate the applicability of the method with enhanced odometry estimation when compared to a rigid sensor fusion scheme.**

I. Introduction

With the recent advances in robotics, autonomous mobile robots are now operating in a broad range of domains. Some well-known examples include industrial plants [1], urban traffic [2] and agriculture [3]. The navigation system required to perform efficiently in such scenarios needs to be robust to a number of operational challenges such as obstacle avoidance and reliable localization. When the same vehicle/platform navigates through significantly different environments as part of its route, navigation can be even more challenging.

The site shown in Figure 1 is representative of this situation. The highlighted routes represent operation paths on which an autonomous vehicle should navigate to perform a given task. The image shows that different regions of the routes present distinct structural characteristics. In the middle path, for instance, the vehicle must travel through a heavily built-up area, with large structures and metallic sheds. In contrast, in other sections, the path is unstructured and mostly surrounded by vegetation, including off-road terrain.

Sensors are required in robotic navigation to obtain information about the robot's surroundings. Since each sensor has advantages and drawbacks, a single sensor is often not sufficient to reliably represent the world, and hence fusing data from multiple sensors has become a common practice. Probabilistic techniques, such as the Kalman filter [4] and the Particle filter [5], enable sensor fusion by explicitly modeling the uncertainty of each sensor.

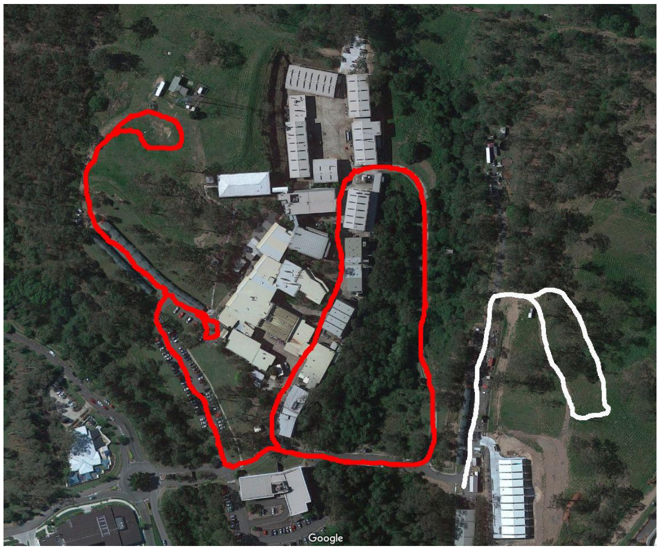


Fig. 1: Satellite view illustrating a heterogeneous operation space. The red path were used to train the CNN to classify the environment, while the white was used to validate its performance. Image from Google Maps.

These are well-known approaches that work well when the navigation takes place in quite homogeneous environments. However, in challenging and mixed scenarios, as described above, employing rigid statistical models of sensor noise may provide a sub-optimal solution. An environment-aware sensor fusion, which dynamically adapts to each different environment, can allow a better sensor fusion performance.

Previous work using teach and repeat approach illustrated the effectiveness of such adaptive scheme [6], but it is limited to a previously defined path. To overcome this constraint, we propose applying convolutional neural networks and camera images to recognize typical navigation environments (indoor, off-road, industrial, urban, etc.) and intelligently associate that information to the best sensor-fusion strategy. This is the gist of the method proposed in this paper.

The proposed method is implemented and evaluated in an autonomous ground vehicle, a full size utility vehicle shown in Figure 4. To validate the performance of the adaptive sensor fusion scheme, we employed it to odometry estimation. In the presence of ground truth, provided by an reliable localization system, the error estimation in odometry becomes trivial, which makes the performance evaluation of the proposed method simple and accurate. Figure 2 shows a block diagram of the proposed method. Experimental results show a reduction in the errors in comparison to using a rigid

[1] Robotics and Autonomous Systems Group, CSIRO, Australia paulo.borges@csiro.au, vini@ieee.org

[2] Universidade Estadual Paulista (Unesp), Faculdade de Engenharia, campus Bauru, SP, Brazil, castanho@unesp.br

This work was partly supported by the program Internship Program Abroad - BEPE-IC of São Paulo Research Foundation (FAPESP) under Grant #2018/02122-0.

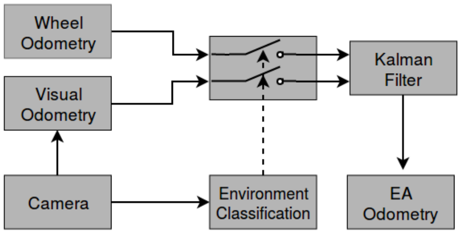


Fig. 2: Architecture overview of the Environment-Aware sensor fusion applied in odometry. This caption does not explain adequately the contents of the figure. I would be fine if the main text did, but there is also no clear explanation there

combination of the sensors.

A. Related Work

Dividing the navigation map to considerate different domain features has been used earlier to enhance robotics performance [6]–[9]. In this approach, usually known as the "teach and repeat" paradigm, the navigation map is visited in an initial phase in which the environment is learned. Then, it is divided in sub-maps which are used later to adapt the behavior of the robot in each sub-map. This paradigm was employed in [10] to enhance navigation of longrange, autonomous operation of a mobile robot in outdoor, GPS denied, and unstructured environments. However, using submaps to change system behavior can only be used in locations previously visited. That is, the behavior of the system is not defined in unknown places, even if they are similar to those previously visited.

In Romero et al. [6] an adaptive sensor fusion technique is applied for obstacle detection. It performs better than using each sensor alone or a covariance-based weighted combination of them. The authors have driven an automated vehicle through a heterogeneous operation environment and quantified the performance of each obstacle-detecting sensor along the trajectory. This information is used by an environment-aware sensor fusion (EASF) strategy that provides different confidence levels to each sensor based on its location along the path. The method uses a look-up table that relates the vehicle's location with the best sensor configuration.

Burgard et al. [11] also proposed the use of an adaptive approach to obstacle detection for mobile robots. A random forest classifier was trained to identify each environment using local geometrical descriptors from a point cloud, so it can classify places not visited during the training. The work presents a classification metric, but does not elaborate on how the obstacle detection was improved by the adaptive strategy.

An adaptive scheme for robot localization was used by Guilherme et al. [12]. The robot is equipped with a short-range laser scanner and a Global Positioning System (GPS) module. A histogram of the distances between occupied cells on an occupancy grid from the laser scanner was used to classify the environment in outdoor or indoor using the k-nearest neighbor algorithm. In outdoor environments the localization system would rely only on the GPS module, while it uses the laser scanner and a previously built map indoors.

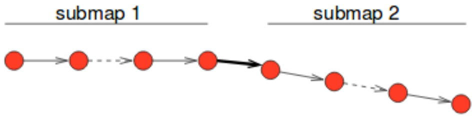


Fig. 3: Example of why the absolute difference is suboptimal. [14]

In 2016, NVIDIA's researchers [2] have trained a convolutional neural network to map the raw pixels from a front-facing camera to steering angles to control a self-driving car on roads and highways (with or without lane markings). According to the authors, the end-to-end learning leads to better performance than optimizing human-selected intermediate criteria, like the lane detection. This work has shown the great potential of convolutional neural networks to face the highly challenging tasks of autonomous driving. Zhou et al. [13] have shown that a properly trained convolutional neural network can identify different environments based only on visual information, using the so-called "deep features".

These related works show that adding information about the environment can lead to more robust systems, able to operate on mixed scenarios. We combine the teach and repeat paradigm with visual scene classification trying to optimize the sensor fusion performance. As result our system assign the optimal sensor configuration to each scene based only on visual information.

II. Environment-Aware Sensor Fusion

This section describes the framework for adaptive sensor fusion, exploiting visual information of the operation environment.

We also provide a performance metric used to compare the each odometry method employed and to define the best sensor configuration to each environment.

A. Performance metric

We chose the metric proposed by Burgard et al. [14] to compare the employed odometry methods. This metric was proposed as an objective benchmark for comparison of SLAM approaches. Since it uses only relative geometric relations between poses along the robot's trajectory, one can use it to compare odometry methods without loss of generalization.

In the presence of a ground truth trajectory, it is usual to get the odometry error by the *absolute difference* between the estimated poses and the ground truth. Burgard et al. [14] claims that this metric is suboptimal because an error on the estimation of a single transition between poses could increase the error in all future poses. To illustrate this behavior, suppose a robot moving in a straight line and an perfect pose estimation, but with a single a rotation error somewhere, let us say on the middle of the trajectory, as shown on Figure 3.

Using the *absolute difference* would assign a zero error to all the poses in the *submap 1*, as expected considering an error-free pose estimator. But it would assign a non-zero error to all the poses in the *submap 2*, even if the error is present only in the transition between two particular poses, shown as a bold arrow in the figure.

The proposed metric is based on the *relative* displacement between the poses. Given two poses x_i and x_j in a trajectory, $\delta_{i,j}$ is defined as the relative transformation that moves from pose x_i to x_j. Given $x_{1:T}$, the set of estimated poses, and $x^*_{1:T}$, the ground truth ones. The *relative difference* is defined in (1) as the squared difference between the estimated and the ground truth transformations, respectively δ and δ^*. In the example from Figure 3, the relative error is non-zero only for the transformation represented by the bold arrow.

$$\epsilon(\delta) = \frac{1}{N}\sum_{i,j}(\delta_{i,j} \ominus \delta^*_{i,j})^2 \qquad (1)$$

By selecting the relative displacement $\delta_{i,j}$, one can highlight certain properties. For instance, by computing the relative displacement between nearby poses the local consistency is highlighted. In contrast, the relative displacement between far away poses enforces the overall geometry of the trajectory. In the experiments we used a mid-range displacement, big enough to include some big scale geometry information while highlighting local consistency.

We used a 10 seconds time interval to compute the relative transformations, which resulted in a 25 meters average distance between each pose when the vehicle was moving in a straight line. The ground truth trajectory was provided by a 3D LIDAR localization system. It is based on the SLAM algorithm proposed by Bosse and Zlot [15] operating in a previously mapped area.

B. Informed sensor fusion strategy

We define a sensor configuration λ as the combination of weights describing the reliability of each sensor. Considering a system equipped with n different sensors, the sensor configuration would be a vector of n elements as follows.

$$\lambda = [\alpha_1, \alpha_2, ..., \alpha_n]^T \in \mathbb{R}^n,\ with\ \alpha_i \in [0,1] \qquad (2)$$

Where α_i represents the reliability of the sensor i. If it is equal to zero the sensor will not be used in the fusion and if it is equal to one the sensor will be fused using the provided error model. Intermediate values should proportionally increase the uncertainty of the sensor.

Appropriately changing the sensor configuration can prevent the hazardous situation where the system is very confident about a bad estimation or the suboptimal situation where the system defines an unrealistic high uncertainty to a sensor in all scenarios to compensate for its high error in some domains.

As described in Section I-A, the teach and repeat paradigm can be used to select a suitable sensor configuration, but in this work we propose the use of visual scene classification.

III. Overall System Description

In this section we describe the experimental set-up and some implemented methods.

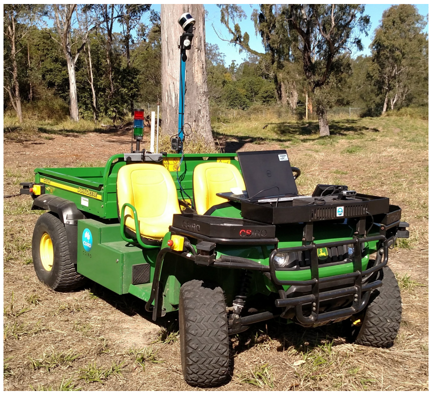


Fig. 4: The robot used, a John Deere Gator holding multiple sensors.

A. Vehicle Description

The robot is built upon a John Deere Gator, an electric medium-size utility vehicle (see Figure 4). The vehicle has been fully automated at CSIRO [16], [17].

The vehicle is equipped with a Velodyne VLP-16 Puck LIDAR, that provides a 360 degrees 3D point cloud, which is used for localization and obstacle avoidance. Besides that, the vehicle has four safety 2D LIDARs (one on each corner). Anytime an object is detected by the lasers inside a safety zone, an emergency stop signal is triggered.

As usual in wheeled robots, the Gator has a wheel odometer, made of a metal disc pressed onto the brake drum and an inductive sensor. In addition to that, a visual odometry algorithm was implemented using as input images from an Intel RealSense D435 [18] mounted front facing in the vehicle, details are provided in Section III-B.

The vehicle holds two computers, one of them used for the low-level hardware control and the other one for high-level tasks, such as localization and path planning. The integration between the computers and the sensors is done using the Robotics Operating System (ROS) [19].

B. Visual Odometry

Visual Odometry(VO) is the process of estimating the movement of a robot given a sequence of images from a camera attached to it. The idea was first introduced in 1980 for planetary rovers operating on Mars [20].

The classical approach to VO relies on extracting and tracking visual features, and then combine the relative motion of this features in sequential images with the camera model to estimates it's movement. The process of simultaneously localize the robot and map the environment using visual information is called Visual SLAM.

A popular Visual SLAM implementation is the ORB_SLAM2 [21], which uses the ORB feature detector. ORB_SLAM2 is an open-source library for Monocular, Stereo and RGB-D cameras, that includes loop closure and relocalization capabilities. We disabled the loop closure and relocalization treads to get a pure visual odometry behavior.

The ORB_SLAM2 classifies the detected features into close and far key points applying a distance threshold. The closest

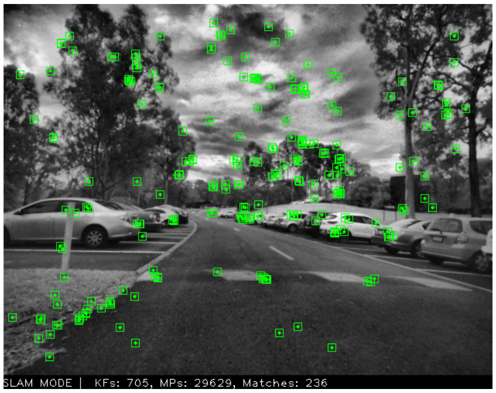


Fig. 5: Sample image after the Adaptive Histogram Equalization, the green dots stands for the detected ORB features.

key points can be safely triangulated between consecutive frames, providing a reliable translation inference. On the other hand, the farthest points tend to give a more accurate rotation inference, since they are supported by multiple views.

We modified the library to provide a ROS friendly interface. In addition to a standard RGB sensor, the Intel RealSense D435 presents a stereo pair of infrared (IR) cameras and an IR pattern projector used for RGB-D imaging. The stereo IR image pair was used for the visual odometry, since it performed better than the RGB-D sensors while outdoors.

The images were equalized before the feature extraction. The histogram equalization is a popular technique in image processing, used to enhance the image's contrast. It often performs poorly when the image has a bi-modal histogram, images that have both dark and bright areas. This effect was minimized using the Contrast Limited Adaptive Histogram Equalization (CLAHE) algorithm.

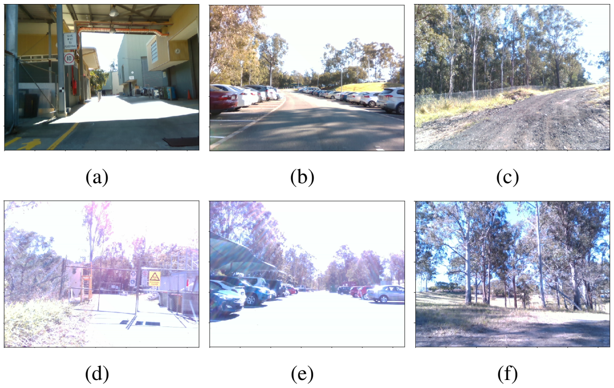
Enhancing the contrast made it easier to find the visual features, making the system more robust to challenging light conditions, inherent of the outdoor operation. The Figure 5 shows an image after the CLAHE, with green dots indicating the extracted ORB features. The features spread over the image, with some key points close to the camera, enhancing the translation estimation.

A demo video of the visual odometry running on the Gator vehicle is available. [1]

C. ROS robot_localization package

The Extended Kalman Filter (EKF) [22] is probably the most popular algorithm for sensor fusion in robotics. Fusing wheel and visual sensors is a classic combination for odometry [23], but there are other options, such as Inertial Measurement Unit (IMU), LIDAR, RADAR, and Global Positioning System (GPS).

The ROS package *robot_localization* [24] provides an implementation of a nonlinear pose estimator (EKF) for robots moving in 3D space. The package can fuse an arbitrary number of sensors. It gets as parameter a binary vector indicating which sensor should be fused and which one should be ignored. This vector can be seen as a particular case of sensor configuration as defined in Section II-B.


(a) (b) (c)

(d) (e) (f)

Fig. 6: First row shows images used to train the CNN and the second images used to validate the performance.

IV. Visual Environment Classification

The environment classification was treated as a classical supervised image classification problem. The operation area shown in Figure 1 was divided into three classes named 'industrial', 'parking lot' and 'off-road.'

In the industrial and the parking lot, the surface is even, made of asphalt or concrete. In this scenario the wheel slippage is low and as consequence the wheel odometry presents low error. Even if the ground does not show many visual features, the visual odometry performs properly relying in the far key points. Hence, both sensors were fused to estimate the odometry.

On the other hand, the off-road environment the flatness assumption of man-made environments does not hold, which allied with the increase in the wheel slipped results in poor wheel odometry performance. In contrast, the visual odometry can benefit from the feature richness of the uneven terrain. So, only the VO is used in this scenario.

We divided the campus in two closed loops: the first one is used to collect the image to train the CNN and the second one to validate the network's classification performance, respectively illustrated by the red and white paths in Figure 1. Both paths present segments on the three classes, but the second path was not exposed to the CNN during the training phase.

The Figure 6 shows samples of images used in the training and testing sets. One can see the challenging lighting conditions, inherent of the outdoor operation.

A pre-trained implementation of the VGG16 [25] was used to classify the images. The VGG16 is a 16-layer network used by the VGG team in the ILSVRC-2014 competition. The network was originally trained using 244×244 images assigned to one of the thousand labels present in the Imagenet Dataset [26]. By the process of transfer learning, we freezed the convolutional layers to train our classifier using a custom built dataset of the three classes described above.

The transfer learning relies on the assumption that the features learned to solve a particular problem on computer

[1] https://youtu.be/I2bq0zsCuME

vision might useful to solve similar ones. The main advantages of the transfer learning are the smaller training time and data requirements.

We used ten thousand images of each label. Since the camera generates around thirty images per second, it is not difficult to collect this many images. The images were collected in different days and times of the day, increasing the statistical significance of the dataset.

At the beginning of the training, the network struggled in the transition between each scenario and some segments on the off-road environment. By inspecting of the classification errors looking for hard-negatives, we detected that the errors were mostly pictures taken off-road but showing buildings, cases that the network classified as industrial. After collecting more data in this circumstance, the network was able to yield a good generalization.

The classification using the neural network was made at 15Hz on the same computer used to run the visual odometry. Assuming that the environment does not change at a high frequency, the real-time execution is not a requirement of the classifier. Thus the prediction could be made less often to reduce the computational burden.

V. Experiments

The experimental site was divided into two closed loop paths. The Figure 1 shows path one in red and path two in white. The path one was visited during the data collection to train the classifier, as described in section IV. Considering the high accuracy achieved on the network validation, we expect a near optimum classification and as consequence the same sensor fusion behavior on both paths. We collected six and four samples from the path one and two respectively.

The raw measurements of all the sensors were saved during the data collection. After that, we estimated offline the vehicle's trajectory using each sensor alone, a rigid fusion of the wheel and visual odometry and the environment-aware sensor fusion strategy described before. These estimated trajectories were compared with the ground truth poses to get the *relative error* as described in section II-A.

VI. Results

A. Scene Classification Accuracy

After the data collection and the training described in Section IV, the network achieved 98.7% classification accuracy on the training set (red path) and 97.2% on the validation set (white path). This high accuracy might be seen as a overfitting since both the training and validation set were collected in the same campus. The accuracy on an extremely different landscape would probably be much lower. But that is also a limitation on the teach and repeat approach. By using convolutional neural networks we introduce to the system the ability operate in places never visited before, the white path was not visited during the training phase, and to adapt to new domains by the exposing it to new data.

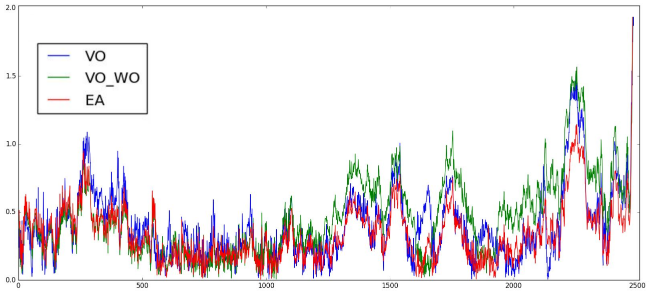


Fig. 7: Relative error in the second path.

Sensor	Mean Relative Error (m)
Wheel Odometry	3.010(±0.568)
Visual Odometry (ORB_SLAM2)	0.541(±0.088)
Wheel Odometry + Visual Odometry (EKF)	0.344(±0.104)
Environment-aware Sensor Fusion (EASF)	0.348(±0.097)

TABLE I: Mean relative error in the training path.

B. Odometry accuracy

Figure 7 shows the *relative error* for each odometry method on a particular sample from second path. The error in the wheel odometer is not on the plot to improve the visualization since it is an order of magnitude bigger. As expected, the error in the environment-aware approach follows the trend of the approach with the smaller error on each time interval.

The Tables I and II presents the average relative error in the paths one and two respectively. Using only the wheel odometry is the worst option on both. In the red path, the EKF (rigid sensor fusion) improved the odometry estimation in 57.2% when compared with the visual odometry, while the EASF approach improved only 55.4%. So the EASF as 1.8% less accurate than the rigid sensor fusion scheme.

However, on the white path, the rigid fusion resulted in a 34.2% increase in the error due to the bad performance of the wheel odometry on this scenario. This noise does not affected the EASF scheme, that reduced the error by 31.1% in relation to the VO. So, the error in the EASF is more the 50% smaller than the error in the EKF.

This difference in the average performance might be caused by the low presence of the off-road scenario in the first path. It is just a small section in a big loop. On the other hand, the second path has near equally distributed sections of both off-road and on-road domains.

Since the covariance on the wheel odometry was measured on the asphalt and concrete, it is not a good representation of the error while driving off-road. This overconfidence leads the rigid fusion to bad estimations.

Sensor	Mean Relative Error (m)
Wheel Odometry	2.722(±0.370)
Visual Odometry (ORB_SLAM2)	0.492(±0.090)
Wheel Odometry + Visual Odometry (EKF)	0.650(±0.247)
Environment-aware Sensor Fusion	0.361(±0.104)

TABLE II: Mean relative error in the testing path.

The results in the second path proved that using the visual information to switch between odometry sources according to the environment might lead to a better performance than always fusing all available sensors. In more challenging operational spaces, for instance paths including mud and gravel, our approach might perform even better.

VII. Conclusions

A new approach to dynamically adapt a sensor fusion strategy for robot autonomous navigation based on the surrounding environment features were presented. Convolutional neural networks were trained to recognize images of the environment on which the robot navigates and based on this information the system adapts its sensor fusion strategy.

To validate the concepts, we also presented a practical implementation of the system on an autonomous vehicle. It is shown that, in environments where the sensor behavior changes, it is possible to select a more suitable sensor configuration using visual information to improve the odometry capabilities of the system. Experimental results have shown an improvement in performance when compared to a rigid sensor fusion approach.

The results presented here only consider the use of two sensors, so future work will add more sensors to the current framework. Further, the methodology can also be directly extended to localization and mapping. Creating "informed" mapping strategies for long-term localization.

Acknowledgements

The authors would like to thank Russell Buchaman, Jiadong Guo and the rest of the CSIRO team for their assistance during this work. This work was partially funded by the Sao Paulo Reseach Foundation (FAPESP).

References

[1] P. V. K. Borges, R. Zlot, and A. Tews, "Integrating off-board cameras and vehicle on-board localization for pedestrian safety," IEEE Transactions on Intelligent Transportation Systems, vol. 14, pp. 720–730, June 2013.

[2] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," arXiv preprint arXiv:1604.07316, 2016.

[3] P. Lottes, J. Behley, A. Milioto, and C. Stachniss, "Fully convolutional networks with sequential information for robust crop and weed detection in precision farming," IEEE Robotics and Automation Letters (RA-L), vol. 3, pp. 3097–3104, 2018.

[4] P. S. Maybeck, "The Kalman Filter: An Introduction to Concepts," in Autonomous Robot Vehicles, pp. 194–204, New York, NY: Springer New York, 1990.

[5] P. S. Maybeck, "The Kalman Filter: An Introduction to Concepts," in Autonomous Robot Vehicles, pp. 194–204, New York, NY: Springer New York, 1990.

[6] A. Rechy Romero, P. V. Koerich Borges, A. Elfes, and A. Pfrunder, "Environment-aware sensor fusion for obstacle detection," in 2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), pp. 114–121, IEEE, sep 2016.

[7] S. Lowry, N. Snderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual place recognition: A survey," IEEE Transactions on Robotics, vol. 32, pp. 1–19, Feb 2016.

[8] W. Churchill and P. Newman, "Practice makes perfect? managing and leveraging visual experiences for lifelong navigation," in Robotics and Automation (ICRA), 2012 IEEE International Conference on, pp. 4525–4532, IEEE, 2012.

[9] C. McManus, P. Furgale, B. Stenning, and T. D. Barfoot, "Visual teach and repeat using appearance-based lidar," in 2012 IEEE International Conference on Robotics and Automation, pp. 389–396, May 2012.

[10] P. Furgale and T. D. Barfoot, "Visual teach and repeat for long-range rover autonomy," Journal of Field Robotics, 2010.

[11] B. Suger, B. Steder, and W. Burgard, "Terrain-adaptive obstacle detection," in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3608–3613, Oct 2016.

[12] R. Guilherme, F. Marques, A. Lourenco, R. Mendonca, P. Santana, and J. Barata, "Context-aware switching between localisation methods for robust robot navigation: A self-supervised learning approach," in 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 004356–004361, IEEE, oct 2016.

[13] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning Deep Features for Scene Recognition using Places Database," 2014.

[14] W. Burgard, C. Stachniss, G. Grisetti, B. Steder, R. Kmmerle, C. Dornhege, M. Ruhnke, A. Kleiner, and J. D. Tards, "Trajectory-based comparison of slam algorithms," in In Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots & Systems (IROS, 2009.

[15] M. Bosse and R. Zlot, "Continuous 3d scan-matching with a spinning 2d laser," in 2009 IEEE International Conference on Robotics and Automation, pp. 4312–4319, May 2009.

[16] P. Egger, P. V. Borges, G. Catt, A. Pfrunder, R. Siegwart, and R. Dubé, "Posemap: Lifelong, multi-environment 3d lidar localization," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3430–3437, IEEE, 2018.

[17] A. Pfrunder, P. V. Borges, A. R. Romero, G. Catt, and A. Elfes, "Real-time autonomous ground vehicle navigation in heterogeneous environments using a 3d lidar," in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2601–2608, IEEE, 2017.

[18] L. Keselman, J. Iselin Woodfill, A. Grunnet-Jepsen, and A. Bhowmik, "Intel RealSense Stereoscopic Depth Cameras," ArXiv e-prints, May 2017.

[19] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in ICRA Workshop on Open Source Software, 2009.

[20] H. P. Moravec, Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover. PhD thesis, Stanford, CA, USA, 1980. AAI8024717.

[21] R. Mur-Artal and J. D. Tards, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," IEEE Transactions on Robotics, vol. 33, pp. 1255–1262, Oct 2017.

[22] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," Proceedings of the IEEE, vol. 92, pp. 401–422, March 2004.

[23] P. Corke, J. Lobo, and J. Dias, "An introduction to inertial and visual sensing," The International Journal of Robotics Research, vol. 26, no. 6, pp. 519–535, 2007.

[24] T. Moore and D. Stouch, "A generalized extended kalman filter implementation for the robot operating system," in Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13), Springer, July 2014.

[25] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv e-prints, p. arXiv:1409.1556, Sep 2014.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in CVPR09, 2009.