Dynamic representations for autonomous driving

Juan Sebastian Olier[1, 2, *], Pablo Marín-Plaza[3], David Martín[3], Lucio Marcenaro[2], Emilia Barakova[1], Matthias Rauterberg[1], and Carlo Regazzoni[2, 3]

[1] Department of Industrial Design, Eindhoven University of Technology, Eindhoven, The Netherlands
[2] Department of Electrical, Electronics and Telecommunication Engineering and Naval Architecture, University of Genoa, Genoa, Italy
[3]Intelligent Systems Lab, University Carlos III de Madrid, Madrid, Leganés, Spain
[*]J.S.Olier.Jauregui@tue.nl, sebastian.olier@ginevra.dibe.unige.it

Abstract

This paper presents a method for observational learning in autonomous agents. A formalism based on deep learning implementations of variational methods and Bayesian filtering theory is presented. It is explained how the proposed method is capable of modeling the environment to mimic behaviors in an observed interaction by building internal representations and discovering temporal and causal relations. The method is evaluated in a typical surveillance scenario, i.e., perimeter monitoring. It is shown that the vehicle learns how to drive itself by simultaneously observing its surroundings and the actions taken by a human driver for a given task. That is achieved by embedding knowledge regarding perception-action couplings in dynamic representational states used to produce action flows. Thereby, representations link sensory data to control signals. In particular, the representational states associate visual features to stable action concepts such as turning or going straight.

1. Introduction

Efficient and robust protection of critical infrastructure is a major need of the modern society. Diverse research activities and projects have been carried out recently for designing and studying novel solutions that improve the protection of critical assets [1]. That includes numerous issues, such as crowd control [2], intrusion detection, person recognition, and perimeter protection [3]. That last functionality is essential for a variety of applications including forest fire detection, oil leakage tracking, and border protection [4]. Recently, the evolution of autonomous vehicles starts to allow their usage for effective perimeter monitoring. For example, autonomous underwater vehicles have been used for detecting the boundaries of benthic zones or finding polluted regions [5]. Swarms of unmanned ground vehicles are utilized for perimeter surveillance and spill detection [6].

However, in those examples, the autonomous entities are programmed through rule-based approaches. The possibility of an agent to automatically replicating behaviors by observing examples of tasks executed by another artificial/natural agent can be a desirable feature to propagate best practices easily. If the observed agent is a human performing a given task, and the observer is an autonomous artificial agent provided with sensors and actuators, the issue is how the observing entity can use its sensors and computational abilities for a series of functionalities, namely:

- The observing agent obtains necessary information to learn representations that can be used later on.
- The acquired representations should be appropriate not only to describe the observed dynamic interaction between the agent and its environment but also as an inference tool from which the observing agent can predict world states when facing similar situations.
- More interestingly, if the actuators in the observing agent allow so, (i.e. if its actuators can replicate the actions carried out by the observed agent), the observing agent should be able to use the obtained representations to monitor if the actions it selects on-line are coherent with the previously observed interaction.

Considering the accomplishment of such functionalities, one could say that the representations acquired should be enough for assessing similarities between observed and reproduced interactions. That implies a self-awareness capability useful for the agent to self-evaluate its performance when replicating the target interaction as taught by another

978-1-5386-2939-0/17/$31.00

entity (e.g. a human). That can be achieved through a process that learns to actively update the internal representations throughout the interaction. Such updating process should be partially conformed by the control generation, since that would favor context-dependent perception-action couplings that allow the agent to adapt to evolving situations also through its actions.

Moreover, given that the knowledge being acquired is centered on how to act in the world, representational abilities should depend on the couplings between the agent, its actions, and the environment as all its sources of information. In [7] it has been developed how representations, and particularly conceptual ones, should be understood as dynamic and emergent during the interaction, and not as isolated symbol-like structures. Those ideas are central to this work as the kind of data from which an agent is to learn in the scenario described is not structured in such a symbolic way. Instead, knowledge should be acquired to embed the action-perception relationships inherent to an interaction, permitting the agent to adapt and interpret its environment dynamically from its sensory data. That approach is related to the ideas of embodied cognition (EC), in which symbolic representations of behavior and control are replaced with bodies perceptually coupled with their environments. In particular, in [8] it is argued that, in general, abstract representations of objects are not needed, but appropriate representational structures arise from the dynamics of actions related to the objects in a given situation. That can be translated to going from segmenting the scene semantically in, for example, pedestrians, vehicles, or empty spaces, to learn dynamic representations relating the scene to adequate actions.

The objective here is to learn models enabling an agent to autonomously perform a perimeter monitoring task previously perceived while conducted by a person. The vehicle is equipped with sensors (cameras) and controlled actuators that allow autonomous driving.

The rest of the article is divided as follows: Section 2 analyses state of the art regarding control generation from artificial vision in autonomous vehicles; sections 3 and 4 describe the general approach and the proposed method. In section 5 the scenario on which the system is evaluated, and the platform used for the data acquisition are explained. Results are reported in section 6 and conclusion and future work are discussed in the last section.

2. Artificial vision for control generation in autonomous vehicles

For autonomous vehicles to understand the environment, vision-based approaches have been proved to be highly cost-effective while enabling close-to-production assemblies given their size and integrability. Stereo-vision systems are commonly used due to the visual characteristics and depth information they provide [9]. Stereo-vision algorithms usually make assumptions about the ground or expected free and occupied spaces, thus providing rich geometry information about the scene, and enabling descriptions of the environment like probabilistic occupancy maps [10], elevation maps [11] or full 3D models [12].

From video data, vast amounts of information can be extracted to understand a scene and thus generate control. In particular, accurate obstacle detection and classification has been seen as an essential feature for self-driving tasks. Consequently, a significant number of algorithms have been developed for obstacle avoidance. Efforts often focus on vehicle [13] and pedestrian detection [14] as the most frequent obstacles in road scenes. Lately, different approaches have focused on deep learnings techniques for understanding the environment. In particular, Convolutional Neural Networks (CNN) are used to detect and segment objects based on methods as the one in [15], which are used to define objects of interest for further processing [16]. Those approaches assume a semantic segmentation as the basis for control decisions. However, there are other approaches, like the one in [17] which uses an end-to-end learning approach to control a self-driving car through a regular CNN. That work shows satisfying results, and notably, the task is achieved without any given specification about the relations between actions and environment as it learns to produce control signals directly from raw images.

3. Approach

In section 1, the relevance of adapting to situations by dynamically updating internal representations has been stressed. We argue that such feature should emerge from a process integrating sensory data, previous states, and the prediction of future inputs. As a consequence, representations are to acquire meaning for the particular action context, thus representing world states based on their relation to expected actions.

In [7] such features are linked to predictive coding[18], and the proposals by Friston [19], which suggest that the core of neural processing is maximizing Bayesian evidence or reducing variational free energy. Under such view, the capability of predicting sensory inputs is linked to the creation of adequate representations. Moreover, those postulates see actions as automatic reflexes maximizing expectations. The presented architecture links to such ideas as is based on a contextual updating of internal states, inspired by Bayesian filtering, and seeing actions as reflexes to fulfill expectations. That is relevant since, contrary to end-to-end approaches based on CNNs like the one in [17], here explicit world state representations are created. Such representations are important to predict world states, learn temporal relations to keep track of the interaction, or to inte-

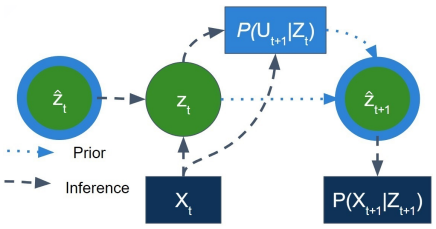

Figure 1. Schema of the architecture. Z_t are the representations, while U_t is the control. X_t is the input.

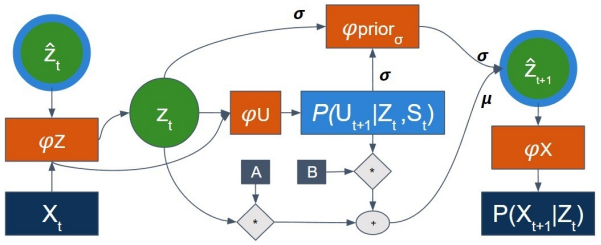

Figure 2. Implementation schematic. circles denote the variational representations (Z), or prior ($\widehat{Z}$). Orange boxes represent NNs (φ), dark blue real and predicted inputs (X), and grey boxes the matrices A and B. The light blue box indicates the control (U).

grate more complex capabilities.

4. Architecture

For a schematic of the proposed model see figure 1. The architecture is based on variational deep generative models. As shown in [20], variational representations can be learned end-to-end through stochastic gradient descent (SGD). Moreover, in [21] the possibility of including time dependencies and filtering capabilities has been shown. In particular, the architecture here proposed relates to the framework in [7] and part of the architecture in [22], but the recursion here is addressed taking into account a system inspired by Bayesian filtering using a linear transition, which explicitly separates the influence of the control on the transition model. The different parts of the architecture are described as follows:

To infer Z_t, incoming sensory data (X_t) and the prediction at $t-1$ are used to estimate the current status of the world as $Z_t \sim \mathcal{N}\left(\mu_{z,t}, diag(\sigma^2_{z,t})\right)$ where $[\mu_{z,t}, \sigma_{z,t}] = \varphi^Z(X_t, \widehat{Z}_t)$. The input $\widehat{Z}_t$ to φ^Z corresponds to the prior calculated for Z_t at $t-1$. That can be interpreted as updating the beliefs about the state of the world when the sensory information arrives. Thus, the update includes information at $t-1$, and the corresponding beliefs about how the world would change. As in [20], during training the value of Z_t is sampled from the distribution, while for testing μ_z is used.

The corresponding control (U_{t+1} in figure 1) is also considered as Gaussian and produced from Z_t and the input X_t in order to capture the noise from the measurements. That is given by $[\mu_{u,t+1}, \sigma_{u,t+1}] = \varphi^U(Z_t, X_t)$. In $P(U_{t+1}|Z_t, X_t)$ the sub index is $t+1$ instead of t since U_t corresponds to the action taken to get from Z_{t-1} to Z_t.

Afterwards, $\widehat{Z}_{t+1}$ and the expected input $(X_{t+1}|\widehat{Z}_{t+1})$ are estimated. Z_t and U_{t+1} are used to predict the next state as $\widehat{Z}_{t+1} \sim \mathcal{N}\left(\mu_{o_t}, diag(\sigma^2_{o_t})\right)$, where $\mu_{o_t} = A * Z_t + B * \mu_{u,t+1}$ and $\sigma_{o,t} = \varphi^{prior_\sigma}(\sigma_{u,t+1}, \sigma_{z,t})$; A and B are learned matrices, and $*$ indicates the inner product.

Finally, expected inputs are estimated as the distribution $X_{t+1} \sim \mathcal{N}\left(\mu_{x_{t+1}}, diag(\sigma^2_{x_{t+1}})\right)$. with $\left[\mu_{x_{t+1}}, \sigma_{x_{t+1}}\right] = \varphi^X(\widehat{Z}_{t+1})$.

All the functions, φ^Z, φ^U, φ^{prior_σ} and φ^X are approximated by neural networks (NN).

The objective maximized through SGD, in batches of T time steps sequences, is given by:

$$\mathcal{L}_\mathcal{T} = \frac{1}{T}\sum_{t=1}^{T}[-KL\left(P(Z_t|X_t, \widehat{Z}_t)||P(\widehat{Z}_t|U_t, Z_{t-1})\right) + log(P\left(X_{t+1}|\widehat{Z}_{t+1}\right)) + log(P\left(U_{t+1}|Z_t, X_t\right))$$

4.1. Implementation details

The architecture is implemented using NNs (figure 2). φ^Z is implemented via a CNN composed of 4 convolutional layers, with filters of size 3 by 3 and stacks sizes of 32, 64, 128 and 128; the first three convolutions are followed by max-pooling (2 by 2). The output of the last convolution is followed by a fully connected layer (FC) with 64 units and output denoted ϱ^X. The output of φ^Z, is calculated by feeding ϱ^X stacked together with $\widehat{Z}_t$ to what we call a $\mu\sigma Coder$. A $\mu\sigma Coder$ is formed by two FC of 64 units, followed by two separated FC of size Z_t, one yielding the mean as a linear output, and the second, the variance with softplus activations.

φ^U is implemented by a $\mu\sigma Coder$ structure, with input size given by the sum of sizes of Z_t and ϱ^X. φ^{prior_σ} is implemented by three FC, the first two with 64 units, and the last the size of Z_t.

For φ^X a deconvolution [23] is used; its input is constructed by an FC generating an output with the same size as the last convolution in φ^{Z_v} from the prior $\widehat{Z}_{t+1}$. That output is reshaped to a tensor form. Filters sizes are the same as in φ^Z. Up-sampling is achieved by a stride of 2 where needed. The output of φ^X is generated with two different filter banks for the means and the variances as in a $\mu\sigma Coder$. Where not mentioned, activations are leaky ReLU. Z_t and U_t are of size 2.

5. Evaluation

A perimeter monitoring task is considered. The scenario is a square, where the task is to perimeter the area by maintaining the vehicle close to the border, or as far from the center as possible. Thus, one round contains four turns, in

Figure 3. The iCab platform.

this case to the left as completed anticlockwise. This scenario is simplified to test the method. The infrastructure is static in time and space as only the vehicle moves. Thus, the causal relations between the environment and the human driving can be assumed to be space and time invariant. During training, the system observes its sensory data while a human performs the task, and constructs a sufficient representational mechanism to reproduce the observed behavior.

5.1. Platform description

The vehicle used to compose the data set is the iCab (Intelligent Campus Automobile). It is a golf cart model EZGO TxT. The steering wheel has been removed and substituted by a motor encoder system. A micro-controller replaced the electronics, and the throttle has been disconnect to allow movement to be controlled by PWM commands directly to the motor driver. The brake pedal is still working along with an electrical motor to decelerate. Figure 3 shows the sensors in the iCab. To process the information collected, two computers connected by Ethernet wire through a 4G router are employed. The software used is Robot Operating System (ROS) Kinetic Kame version under Ubuntu 16.04 LST. The architecture description and the communication scheme are described in [24] and [25].

5.2. Data set

The data set of the experiment is composed by gathering sensor and odometry information. The messages of raw data from the sensors are stored in bag files. Sampling rates of the used data are 10 Hz for velocity and steering, and 20 Hz for the images. The raw images are of size 640 by 480. Two data sets are taken (training and testing); they are different rounds in the same space and with the same goal.

5.3. Training data preprocessing

We use the video stream form the left camera, and the control signals (steering and velocity). The grey images are resized initially to 120 by 160 pixels. Then, data augmentation techniques are used. For the images, random translations are applied by randomly selecting an initial point for cropping, with which a final size of 90 by 120 is reached. The same cropping is applied to all the images in a single batch. For the control signal, an independent offset is added to both the steering and the velocity. The offset is randomly drawn from a uniform distribution from 0 to 5% of the maximum of each signal. The same offset is applied to all signals in a batch.

Figure 4. Samples of the images. It can be seen, from left to right and top to bottom, a situation in which the car approaches a corner and takes a curve.

6. Results

6.1. Control prediction

The control generation is tested by comparing the signal produced by the system and the actual signal from the human driver. Two different comparisons are performed (figures 5 and 6), one on the training set and a second on the test set. The main differences are found in the velocity signal, which is accurately reproduced for the training data, though for the test set some differences emerge. Such differences mainly arise because there is no particular policy for the human driver regarding the velocity, and thus in the two sets they are different. However, it is clear that during curves the velocity is reduced, while after them it is increased, which is a behavior generated by the system as well. Moreover, the average values are coherent with the driving task.

Regarding the steering, the accuracy is better. For the training data, the actions are reproduced accurately. For the test data, some small differences are found in the turning actions, though the behavior is reproduced. The differences mainly correspond to changes in the driving style of the human driver between the two sets. For example, in the test scenario, around the time step 1000 in figure 6, the last curve is started closer and faster that in the other cases. That can be seen in the velocity signals, where the produced control diminishes the magnitude before the one by the human, who has to break more sharply later on. Such motion is not the behavior the network learned, and the control generated by the system shows a better performance as proposes a smoother curve in that particular case.

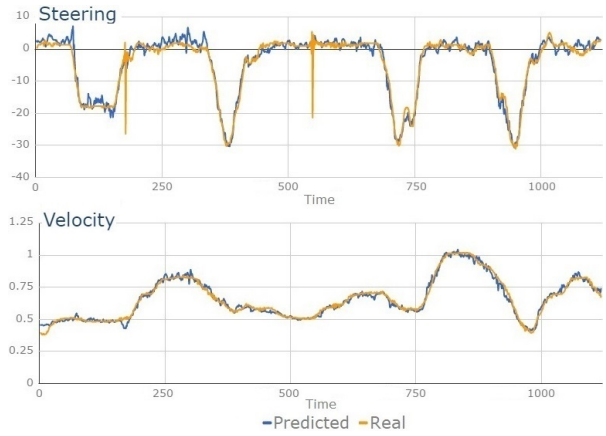

Figure 5. Control comparison on the training data, between the real signal from the human driver and the predicted one.

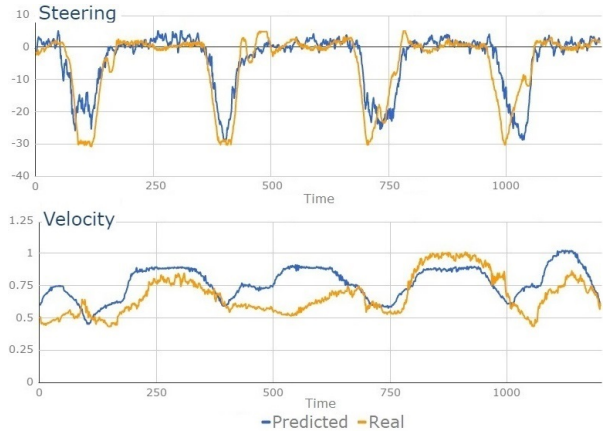

Figure 6. Control comparison on the test data, between the real signal from the human driver and the predicted one.

6.2. Representations

In figure 7 the representations created for the training data are shown over time, and are compared to the ones created by a modified VAE [20], encoding images and reconstructing images and control. The most prominent differences are the oscillations in the representations from the VAE, which are not present in the results with our method. The oscillations are correlated with visual characteristics of the scenario, particularly periodic patterns on the floor (e.g. the dark squares, see figure 4). The VAE focusses on visual features of such kind to create the representations. On the contrary, our method ignores such features as are uninformative regarding the action-environment coupling.

7. Conclusions and future work

The proposed approach generates a representational mechanism that internalizes relations between environment and control in a given task. With such representations human driving behavior in real environments is mimicked.

Figure 7. Representations over time for the VAE (top) and our approach (bottom).

The intelligent agent provided with sensors and actuators has learned solely by observing the naturalistic human-driving, that is, an agent has learned smart models to perform an autonomous navigation task. The architecture has been tested in a real autonomous vehicle performing a perimeter monitoring task in a close-loop square trajectory.

The initial results show that the dynamic representations embed action concepts that allow to reproduce the human behavior in the proposed task. So, the presented comparisons between real human-driving data, which acts as ground truth, and predicted data from dynamic representations, has shown how the system mimics human behaviors by models such as: at curves, the vehicle velocity is reduced, and after them, it is increased like humans do. Finally, a comparison has been presented to show how the dynamic representations created with the architecture extract the features needed for the navigation task while ignoring others, capturing so the action-environment couplings and the dynamics involved in the navigation task.

The future work is focused on closing the perception-control loop based on dynamic representations in the autonomous vehicle. Thus, control commands will be generated by the intelligent system and executed in the real vehicle. The autonomous navigation and behavior of the real vehicle will be analyzed to understand the decision-making capabilities of the presented dynamic representations in a real-time self-driving task.

Acknowledgments

-This work was supported in part by the Erasmus Mundus Joint Doctorate in Interactive and Cognitive Environments, which is funded by the EACEA Agency of the European Commission under EMJD ICE FPA n 2010-0012.

-Carlo Regazzoni has contributed to this work partially under the program UC3M-Santander Chairs of Excellence.

References

[1] Adam M. Lewis, David Ward, Lukasz Cyra, and Naouma Kourti, "European reference network for critical infrastructure protection," *IJCIP*, vol. 6, no. 1, pp. 51–60, 2013.

[2] S. Chiappino, L. Marcenaro, and C. Regazzoni, "Selective attention automatic focus for cognitive crowd monitoring," 2013, pp. 13–18.

[3] S. Rogotis, C. Palaskas, D. Ioannidis, D. Tzovaras, and S. Likothanassis, "Recognizing suspicious activities in infrared imagery using appearance-based features and the theory of hidden conditional random fields for outdoor perimeter surveillance," *Journal of Electronic Imaging*, vol. 24, no. 6, 2015.

[4] R.K. Baggett, C.S. Foster, and B.K. Simpkins, *Homeland Security Technologies for the 21st Century*, Praeger Security International. ABC-CLIO, 2017.

[5] M. Kemp, A. L. Bertozzi, and D. Marthaler, "Multi-uuv perimeter surveillance," in *2004 IEEE/OES Autonomous Underwater Vehicles (IEEE Cat. No.04CH37578)*, June 2004, pp. 102–107.

[6] G. Zhang, G. K. Fricke, and D. P. Garg, "Spill detection and perimeter surveillance via distributed swarming agents," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 1, pp. 121–129, Feb 2013.

[7] Juan Sebastian Olier, Emilia Barakova, Carlo Regazzoni, and Matthias Rauterberg, "Re-framing the characteristics of concepts and their relation to learning and cognition in artificial agents," *Cognitive Systems Research*, vol. 44, pp. 50–68, 2017.

[8] Andrew Wilson and Sabrina Golonka, "Embodied cognition is not what you think it is," *Frontiers in Psychology*, vol. 4, pp. 58, 2013.

[9] Uwe Franke, David Pfeiffer, Clemens Rabe, Carsten Knoeppel, Markus Enzweiler, Fridtjof Stein, and Ralf Herrtwich, "Making bertha see," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013, pp. 214–221.

[10] Hernán Badino, Uwe Franke, and Rudolf Mester, "Free space computation using stochastic occupancy grids and dynamic programming," in *Workshop on Dynamical Vision, ICCV, Rio de Janeiro, Brazil*, 2007, vol. 20.

[11] Florin Oniga and Sergiu Nedevschi, "Processing dense stereo data using elevation maps: Road surface, traffic isle, and obstacle detection," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 3, pp. 1172–1182, 2010.

[12] Alberto Broggi, Stefano Cattani, Marco Patander, Mario Sabbatelli, and Paolo Zani, "A full-3d voxel-based dynamic obstacle detection for urban scenario using stereo vision," in *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*. IEEE, 2013, pp. 71–76.

[13] Sayanan Sivaraman and Mohan Manubhai Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1773–1795, 2013.

[14] Tarak Gandhi and Mohan Manubhai Trivedi, "Pedestrian protection systems: Issues, survey, and challenges," *IEEE Transactions on intelligent Transportation systems*, vol. 8, no. 3, pp. 413–430, 2007.

[15] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[16] Ziyu Zhang, Sanja Fidler, and Raquel Urtasun, "Instance-level segmentation for autonomous driving with deep densely connected mrfs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 669–677.

[17] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al., "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[18] Rajesh PN Rao and Dana H Ballard, "Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects," *Nature neuroscience*, vol. 2, no. 1, pp. 79–87, 1999.

[19] Karl Friston, Thomas FitzGerald, Francesco Rigoli, Philipp Schwartenbeck, and Giovanni Pezzulo, "Active inference: A process theory," *Neural Computation*, vol. 29, no. 1, pp. 1–49, 2017.

[20] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2013.

[21] Maximilian Karl, Maximilian Soelch, Justin Bayer, and Patrick van der Smagt, "Deep variational bayes filters: Unsupervised learning of state space models from raw data," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[22] Juan Sebastian Olier, Emilia Barakova, Matthias Rauterberg, Lucio Marcenaro, and Carlo Regazzoni, "Grounded representations through deep variational inference and dynamic programming," *Proceedings of the 7th ICDL-EPIROB conference*, 2017.

[23] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus, "Deconvolutional networks," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2528–2535.

[24] Ahmed Hussein, Pablo Marín-Plaza, David Martín, Arturo de la Escalera, and José María Armingol, "Autonomous off-road navigation using stereo-vision and laser-rangefinder fusion for outdoor obstacles detection," in *Intelligent Vehicles Symposium (IV), 2016 IEEE*. IEEE, 2016, pp. 104–109.

[25] Pablo Marn-Plaza, Jorge Beltrn, Ahmed Hussein, Basam Musleh, David Martn, Arturo de la Escalera, and Jos Mara Armingol, "Stereo vision-based local occupancy grid map for autonomous navigation in ros," *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2016)*, pp. 701–706, 2016.