https://doi.org/10.2352/ISSN.2470-1173.2019.15.AVM-042

DriveSpace: Towards context-aware drivable area detection

***Ciarán Hughes*[1]*, Sunil Chandra*[1]*, Ganesh Sistu*[1]*, Jonathan Horgan*[1]*, Brian Deegan*[1]*, Sumanth Chennupati*[2] *and Senthil Yogamani*[1]**
[1] ***Computer Vision Platform, Valeo Vision Systems, Tuam, Ireland***
[2] ***Department of Vision Systems, Valeo North America Inc, Troy, United States***

Abstract

Free space is an essential component of any autonomous driving system. It describes the region, which is typically the road surface, around the vehicle which is free from obstacles. However, in practice, free space should not solely describe the area where a vehicle can plan a trajectory. For instance, in a single lane road with two way traffic the opposite lane should not be included as an area where the vehicle can plan a driving path although it will be detected as free space. In this paper, we introduce a new conceptual representation called DriveSpace which corresponds to semantic understanding and context of the scene. We formulate it based on combination of dense 3d reconstruction and semantic segmentation. We use a graphical model approach to fuse and learn the drivable area. As the drivable region is highly dependent on the situation and dynamics of other objects, it remains a bit subjective. We analyze various scenarios of DriveSpace and propose a general method to detect all scenarios. As it is a new concept, there are no datasets available for development and test, however, we are working on creating the same to show quantitative results of the proposed method.

INTRODUCTION

In this paper, we propose an iterative upgrade to the sensing and control architecture for autonomous vehicle. In Figure 1, we show a high-level conceptual diagram of the traditional and the proposed architecture. The traditional[1] architecture is similar to that which is proposed by Lin et al. [1] and Pendleton et al. [2]. However, in such architectures, the context of the vehicle's environment is implicit in the other components (e.g. perception and planning). Here, we argue that as contextualisation is an extremely important component in automated driving, it deserves to be an explicit component of autonomous vehicle architectures. With that in mind, we introduce the concept of a contextual drivable area, known as DriveSpace.

Related to this are two well known and important concepts: free space and drivable area. The relationship of DriveSpace to free space and drivable area is demonstrated in Figure 2. Free space aims to detect an area that is free of obstacles around the host vehicle. This is not the logical inverse of detecting an obstacle, as an object detector has a limited detection rate i.e. there are sometimes missed objects within the expected detection range. Therefore, object detection alone cannot be used to define free space, as missed objects could be present in the free space. This has been extensively studied for autonomous driving by Schreier et al. [3] and Neumann et al. [4] over the past few years and more

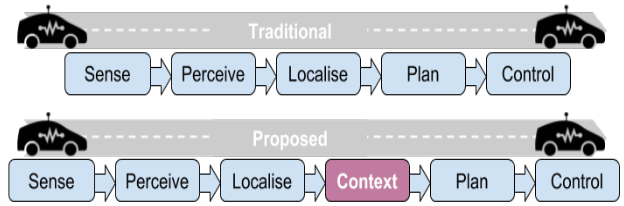


Figure 1. *Traditional vs Proposed Autonomous Driving Pipeline*

recently by Hanish et al. [5] for fisheye cameras that are widely used in automotive.

Free space is used to maintain environmental maps. It is used to erase both static and dynamic obstacles that are no longer present at the observed location while also being utilized when multiple sensors are being combined for a single environmental representation in a fusion map. A good free space model will erase dynamic obstacles from their previously known positions instantaneously without erasing correct static obstacle information. Furthermore free space shall also erase falsely detected obstacles. An additional advantage of such a free space model is to partially correct the error in ego vehicle odometry calculations, when re-detection of a static obstacle is not at the same position. Recent advances in the System on Chip (SoC) hardware offers feasibility to utilize dense optical flow and dedicated accelerators for Convolution Neural Networks (CNNs).

Drivable area is perhaps a little less well defined in the literature. However, this can broadly be seen as the free space around the vehicle in which the vehicle is allowed to travel. For example, some research considers road detection [6], lane marking and kerb detection [7], flat traversable areas [8], and even image-map fusion approaches [9]. There has been some work done on general drivable area detection [10, 11] but there is no systematic definition for this problem and there is no public dataset for evaluation as this is a highly contextual and subjective problem.

In this paper, we prefer to define drivable area as the geometric constraints of where the vehicle can traverse. That is, it is limited by the objects in the scene, be they above the ground or holes in the traversable area, and the space around and between objects. Drivable area does not typically consider context of the scene, such as markings, stop lights, etc. This is a good example of what we said at the beginning of this section: context of the scene is absorbed into other components of the traditional automated driving sensing architecture, whereas in our proposal we have a specific component to handle the contextual information, i.e. DriveSpace. Equally, we can think of DriveSpace as being a complete contextualisation to free space and drivable area detection.

[1] The authors note that despite the fact that autonomous driving is an area of new and active research, we can refer to "traditional" approaches in autonomous driving

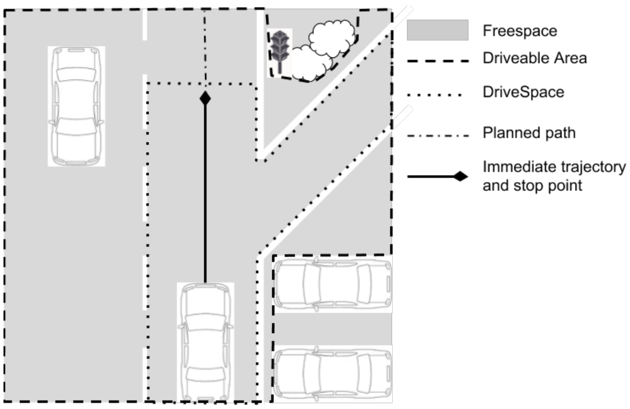


Figure 2. DriveSpace in the context of other concepts such as free space, drivable area and trajectory planning

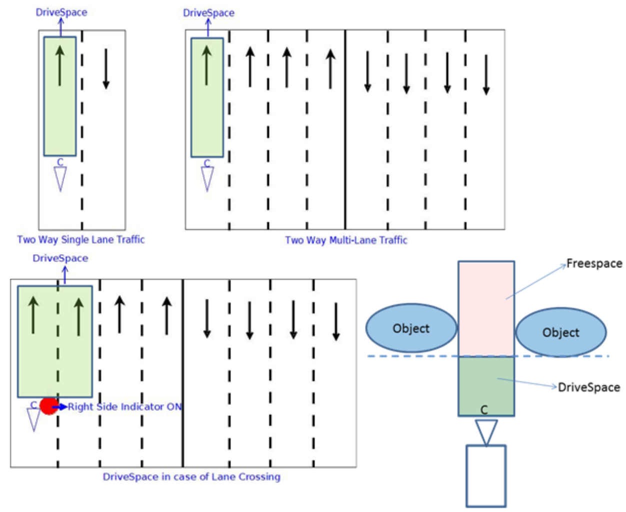


Figure 3. Lane semantics for DriveSpace

Thus, in this paper, we provide initial design concepts into a dense and contextualised free space and drivable area solution, DriveSpace, which fully leverages dense depth and semantics to resolve the issues encountered in the previous free space estimation algorithms.

OVERVIEW OF DRIVESPACE

In the previous section, we provided a brief motivation for DriveSpace. In this section, we provide a more concrete definition, while providing further motivation. DriveSpace is a region around the host vehicle that is obstacle free, safe and legal to drive. It is also dynamic, adapting based on the changing driving environment. In Figure 2, DriveSpace is presented in the context of free space and drivable area. It is also given in the context of the immediate and the planned trajectories. The immediate trajectory is limited by the DriveSpace, as shown in Figure 2. For example, if the traffic light in the diagram is red, then the DriveSpace ends at the traffic light, and therefore the immediate trajectory is limited. On light change to green, the DriveSpace would expand tothe maximum detection range of the given sensors.

Let $\mathbf{S}_{fs} \subset \mathbf{R}^3$ refer to the free space around the vehicle, and $\mathbf{S}_{da} \subset \mathbf{R}^3$ refer to the drivable area. Furthermore, let DriveSpace be represented by $\mathbf{D}$, the restriction exists that $\mathbf{D} \subset \mathbf{S}_{da} \subset \mathbf{S}_{fs}$. This makes sense according to the following examples:

- The ego-vehicle cannot get arbitrarily close to another object (zero distance), and must always stay a "safe" distance from other obstacles. So while we may have drivable area right up to another obstacle, depending on the context of the situation (e.g. ego-vehicle velocity, whether the other obstacle is a pedestrian, vehicle or other object type, etc.), the DriveSpace area is adapted.
- An appropriate lane marking will limit the DriveSpace, but not drivable area (by our definition) nor free space. However, this is malleable. For example, in Figure 3, we describe a scenario where the DriveSpace adapts based on the ego vehicle status.
- Road signs, traffic lights, etc., all provide further contextual information to limit the DriveSpace
- The behaviour of other road users can adapt the DriveSpace. For example, a pedestrian near the road that possibly has the intention of crossing can shrink the DriveSpace in their surrounding region, causing the autonomous vehicle to move away from the kerb. Thus, very advanced contextual algorithms, such as pedestrian intent [12], driver intent [13], and other user intention in general [14], can provide significant cues.

In general, one could achieve robust DriveSpace by incorporating vital cues from free space, semantic segmentation, lane detection, kerb detection, object detection, High Definition (HD) Maps, etc. that can incorporated through multiple sensing modalities. For the purposes of this paper the authors focus on camera information because of the ability of image processing to provide structural, semantic and navigational information about the environment of the vehicle. Few examples are provided below to aid the reader in fully understanding DriveSpace.

In Figure 3, a DriveSpace scenario is shown for both cases of two way single lanes and multi-lane traffic. DriveSpace will adapt as per driving context. In case of a multi-lane traffic scenario, if the host vehicle provides indicator left or right, depending on which lane the vehicle will move, the DriveSpace will be able to cover the region of interested lane. Without this context, the DriveSpace would be limited to the current lane. In case of dynamic object moving towards the vehicle or otherwise, DriveSpace region will adapt it's size automatically.

The simplest case of lane separation is two way lanes. There are more complex scenarios especially in case of junctions and four way crossings. A complex scenario is shown in Figure 4, where the red and yellow regions demonstrate DriveSpace of cars going in opposite direction. This is complex to infer from standard free space and lane detection. Higher level semantic perception is needed to understand the context and validity of DriveSpace.

In lower speed scenarios (e.g. parking), two objects can be detected by the sensors on the host vehicle. Free space detection will give the area between the obstacles as free space. However, if the obstacles are close together (closer than the width of the host vehicle plus an error margin) then the area between the obstacles should not be considered as DriveSpace, as the vehicle cannot manoeuvre between the obstacles. Thus, DriveSpace can

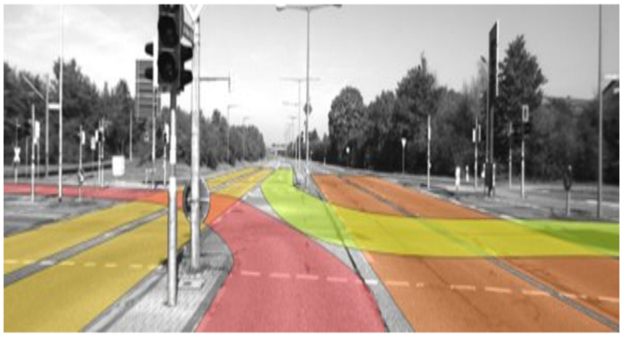


Figure 4. Example of a complex lane crossing scenario. Figure is reproduced from KITTI. The red demosntrates the DriveSpace of a car turning left, the orange shows the drivespace of a vehicle driving forward. Yellow would be the DriveSpace of other vehicle.

take additional contextual information about the obstacles in the host vehicles environment to determine the drivable area.

The requirement for legal manoeuvring in DriveSpace must be considered malleable, as otherwise the autonomous vehicle will become stuck in many scenarios. For example, consider the situation that there is a road in which it is illegal to cross the central lane marking (often denoted by a solid marking). If there is an obstruction (e.g. another parked vehicle) that prevents the ego-vehicle from proceeding within it's own lane, then the ego-vehicle must be able to contextualise and cross the marking safely, even if it is technically considered breaking the law.

Ultimately, we head towards a definition of DriveSpace as follows: DriveSpace is the area in which the vehicle can safely and legally (within reason) manoeuvre considering the immediate context of its environment. It is formed by the following relations:

$$\mathbf{T}_{imm} \subset \mathbf{D} \subset \mathbf{S}_{da} \subset \mathbf{S}_{fs} \subset \mathbf{R}^3$$

where $\mathbf{T}_{imm}$ is the set of immediately possible trajectories for the vehicle, and the other terms have been previously defined. As in the traditional autonomous vehicle architecture as show in Figure 1 the DriveSpace information is continuously fed to a planning module for usage in updating the optimal vehicle trajectories.

BUILDING BLOCKS OF DRIVESPACE

In this section, we discuss the building blocks of DriveSpace. Firstly, we make use of existing surround-view cameras which provide full 360°perception around the vehicle. We are also interested in commercial deployment and leverage the hardware accelerators in automotive SOCs. Real-time design is another important criteria for the proposed algorithm as emphasized in [15] and [16].

Surround view cameras

Surround view cameras are best suited for near field sensing. These systems consist of four fisheye cameras with $> 180°$ Field of view (FOV). Usually the front camera is mounted in front grill, left and right mirror cameras are mounted on wing mirrors and rear camera mounted under boot lip and are connected to a single Electronic Control Unit (ECU). Commercially, this setup has been extensively used for surround viewing applications. As a result of increasing computational power, they can additionally support a diverse set of applications. More details of surround view cameras are provided by Yu et al. [17] and Heimberger et al. [18].

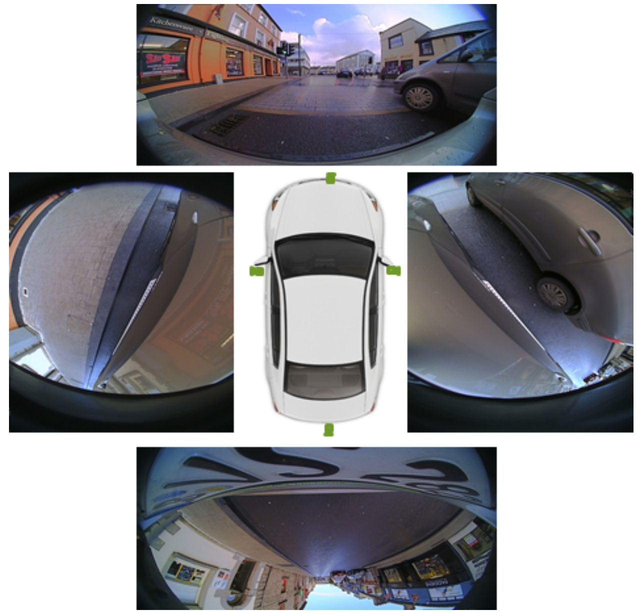


Figure 5. Surround view camera cocoon

From the perspective of parking systems, the cues and layout are setup according to the human visual system. Because of this, cameras are impossible to replace, even with expensive LIDAR sensors. Cameras capture dense semantics, sometimes not available in other sensors and they are relatively inexpensive sensors, with low power consumption as a result of their passive nature. For surround view systems, wide-angle fish-eye lenses are used in which the horizontal field of view can exceed beyond 180°. This means that it is not possible to create a single undistorted view. Most of the academic literature in computer vision, is focused on rectilinear images and with fisheye distortion there are many challenges to re-target algorithms. Typically, these consist of four cameras which form a camera network with a small overlap between them. Four cameras are usually sufficient to cover the near field environment around a vehicle. Figure 5 shows the four views of a typical camera network such as this. It is important to note that they use wide-angle lenses to cover a larger FOV and hence the fisheye distortion.

Semantic Segmentation

Semantic image segmentation has witnessed tremendous progress recently with deep learning. Semantic segmentation is targeted towards partitioning the image into semantically meaningful parts as shown in Figure 6 with various applications. Semantic segmentation for automated driving has many a priori constraints relative to a general version. In this section, we discuss the various aspects which brings a simplifying structure to the problem. For more details on semantic segmentation for automated driving, please refer to work by Siam et al. [19]. Prior information could simplify model complexity greatly. There are different types of prior information that can be used. Spatial priors such as the fact that lanes lie on a ground plane, or that road segmented is mostly in the bottom half of the images. Geometric priors on the

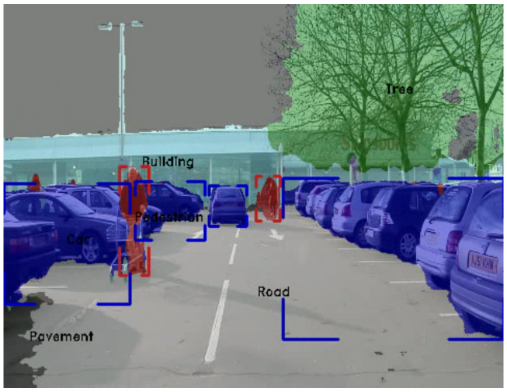


Figure 6. *Segmentation of an automotive scenes*

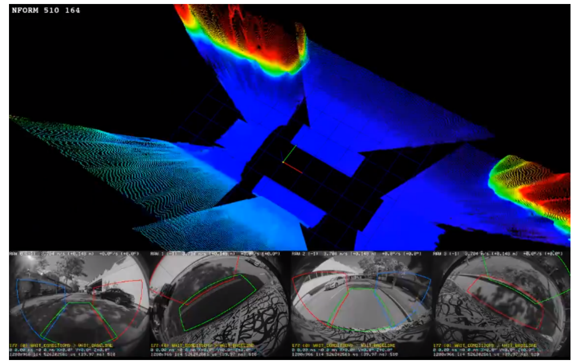


Figure 7. *Dense point cloud around the vehicle*

shapes of objects, for examples lanes are thick lines that are all converging into a vanishing point. Examples of color priors are the color of traffic lights or white lanes. Finally, Location priors, for example the lane, road or buildings locations based on high definition maps or aerial maps.

Typically automotive systems uses a multi-camera network. Current systems have at least four cameras and it is increasing to more than ten cameras for future generation systems. It covers the entire 360° field of view surrounding the car. The geometric structure of the four cameras and the motion of the car induces a spatio-temporal structure across the four images. For example, when the car is turning left, the region imaged by the front camera will be imaged by the right-mirror camera after a delay. There is also similarity in the near-field road surface in all the four cameras as they belong to the same road surface.

Dense SFM

Depth estimation refers to the set of algorithms aimed at obtaining a representation of the spatial structure of the environment within the sensors FOV. In the context of automated parking, it is the primary mechanism by which computer vision can be used to build a map. This is important for all parking use cases: it enables better estimation of the depth of parking spaces over the existing ultrasonic-based parking systems, and thus better trajectory planning for both forward and backward perpendicular and fishbone park manoeuvring; it increases the reliability of kerb detection, improving the parallel parking manoeuvre; and, it provides an additional detection of obstacles, which, in fusion, reduces significantly the number of false positives in auto emergency braking.

Depth estimation is the primary focus of many active sensor systems, such as Time of Flight (TOF) cameras, lidar and radar, this remains a complex topic for passive sensors such as cameras. There are two main types of depth perception techniques for cameras: namely stereo and monocular [20]. The primary advantage of stereo cameras over monocular systems is improved ability to sense depth even when the camera system is static. It works by solving the correspondence problem for each pixel, allowing for mapping of pixel locations from the left camera image to the right camera image. The map showing these distances between pixels is called a disparity map, and these distances are proportional to the physical distance of the corresponding world point from the camera. Using the known camera calibrations and baseline, the rays forming the pixel pairs between both cameras can be projected and triangulated to solve for a 3D position in the world for each pixel. Figure 7 shows an example of dense 3D reconstruction.

Monocular systems are also able to sense depth [21], but motion of the camera is required to create the baseline for reconstruction of the scene. This method of scene reconstruction is referred to as structure from motion (SFM). Pixels in the image are tracked or matched from one frame to the next using either sparse or dense techniques. The known motion of the camera between the processed frames as well as the camera calibration, are used to project and triangulate the world positions of the point correspondences. Bundle adjustment [22] is a commonly used approach to simultaneously refine the 3D positions estimated in the scene and the relative motion of the camera, according to an optimality criterion, involving the corresponding image projections of all points.

After the Optical Flow computation step, the flow vectors are separated to static and dynamic vectors. These are calculated using 3 parts 1) epipolor geometry constraint, 2) spatial consistency constraint and 3) temporal propagation constraint. The dynamic points are separated out and passed through a lattice based clustering algorithm. In contrast to a regular clustering method, here we have the points distributed on a regular lattice and the partial ordering of the points in the lattice can be exploited to produce better clustering algorithms. The other points which are static go through a structure from motion (SFM) pipeline. Firstly, the relative pose of the camera is calculated by a combination of planar homography based on points on the ground plane and essential matrix computation for non-ground points. The complementary combination provides robustness to the estimate which is key to the accuracy of the next steps in the SFM pipeline. 3D reconstruction is computed using re-projection error metric and iterative least squares.

PROPOSED SOLUTION FOR DRIVESPACE

DriveSpace modeling around the ego vehicle involves consideration of many variables such as the road surface, ground markings, vulnerable user behavior, dynamic vehicles' motion direction and static objects characteristics etc. DriveSpace can be modeled as a supervised learning problem, reinforcement learning problem or even as a graphical model inference. Though supervised approaches involve the collection of semantically annotated data, it is proven to be an efficient solution in a number of relative

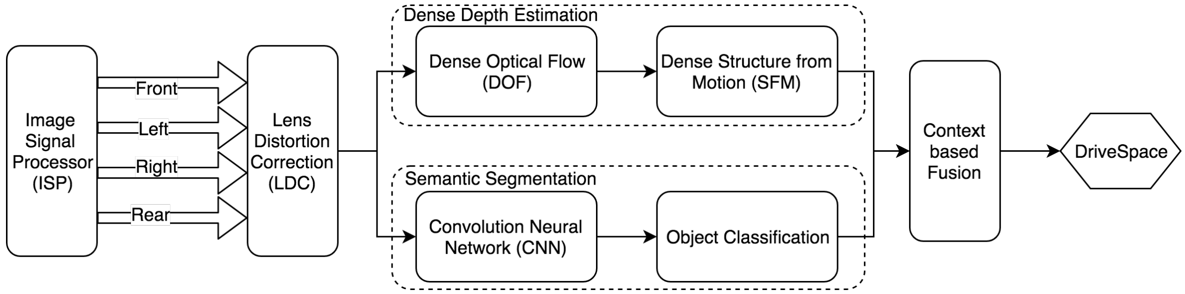


Figure 8. *Block diagram of DriveSpace algorithm.*

tasks. In [23], authors modelled drivable road area as two stage problem, first collecting the labels via OpenStreetMaps, vehicle pose sensors and camera parameters and later using the driven paths as labels to train a CNN. In [24], the images from a vehicle mounted camera are fused with its corresponding local route maps to obtain map-fusion image. The map-fusion image having complementary features is used to train a CNN, namely FCN-VGG16, to extract the drivable road regions. Though the proposed method models the drivable region in an unsupervised fashion it is limited to fixed route autonomous vehicles. In this paper we discuss how DriveSpace can be posed as a supervised learning problem and show how it can be modeled as a late fusion of traditional computer vision and deep learning algorithms. Figure 8 shows the proposed architecture. In this work we argue that using only camera sensors and perception techniques we can efficiently model drivable regions around an autonomous vehicle. The key stages involved are, estimating geometric clues via dense reconstruction, semantic clues via convolutional neural networks and drivepspace extraction via probabilistic fusion of geometric and semantic information.

Data Augmentation

The labeling data for DriveSpace is a subset of semantically labeled data for free space. Below we show how the internal and external heuristics and priors help in building efficient labeling data for DriveSpace.

1. **Free Space Segmentation:** Free space is generally not semantically labelled directly, however, road segmentation is part of a wide variety of public datasets for autonomous driving applications and is available with pixel level accuracy. While road segmentation is not the same as free space for certain driving environments there is complete overlap between the two e.g. highway environment.
2. **Lane Information:** The initial labeled data for free space can be modified based on the lane in which the ego vehicle is traveling, as shown in Figure 9,
3. **Static Object Detection:** Static objects annotated near the free space boundary regions are considered as those are the potential candidates for DriveSpace divergence. Also they play crucial role in differentiating the free space from DriveSpace during modeling. An instance based labelling of objects is more suitable for modelling individual behaviours.
4. **Dynamic Object Detection:** Dynamic object annotation, be it vulnerable road users or vehicles, helps in estimating the dynamic variations in the DriveSpace regions. Hence the annotations for the same helps in modeling the DriveSpace.
5. **HD Map Localisation:** Knowledge of the accurate position of the vehicle within a HD map, gained through camera landmark localisation, provides access to a vast amount of geometric, semantic and real-time data around the vehicle that can be captured and utilized without a great deal of extra annotation effort.

DriveSpace Modeling

The recent trend is to model all these prior semantic clues as an end to end learnable system as an ample amount of data is available for this type of dense parametric learning. Another efficient way is to model this as a fusion system, where a number of computer vision or deep learning algorithm inferences are fused for better estimation. Both the methodologies have their own advantages and disadvantages. The unified learning framework needs vast and varied data to be captured and annotated at significant cost. However, this approach can suffer if the data is weakly labeled or noisy, whereas the fusion systems are efficient at handling these issues as human engineered priors are induced. An optimal way is to make the system end to end learnable with strong geometric and structured priors included inherently.

CNNs, fusion and hybrid models are further explained below:

1. **Deep Convolutional Neural Networks:** CNNs have shown remarkable performance in pixel level segmentation tasks such as road segmentation. Unlike the free space detection, DriveSpace detection can't be handled in as trivial a manner in terms of labelling. To pose the problem as pixel level loss function, the annotations related to prior structures specified in the previous section should be considered while labeling as shown in Figure 9.
2. **Fusion based Methodology:** In this approach the DriveSpace is modeled as a hierarchical model. At each stage the information is extracted such that it forms the basis for the next stage. The information is modeled at pixel level which forms basis for object and geometric stages. Further this object level information is passed to scene level processing. As it involves carefully designed feature engineering blocks, it makes the method robust to noisy data samples.
3. **Hybrid Method:** In this method the geometric clues can be induced into the learnable framework in an explicit or implicit fashion. As shown in the Figure 8, the geometric and object level clues are provided as external inputs to the CNN along with the image information. This modeling not

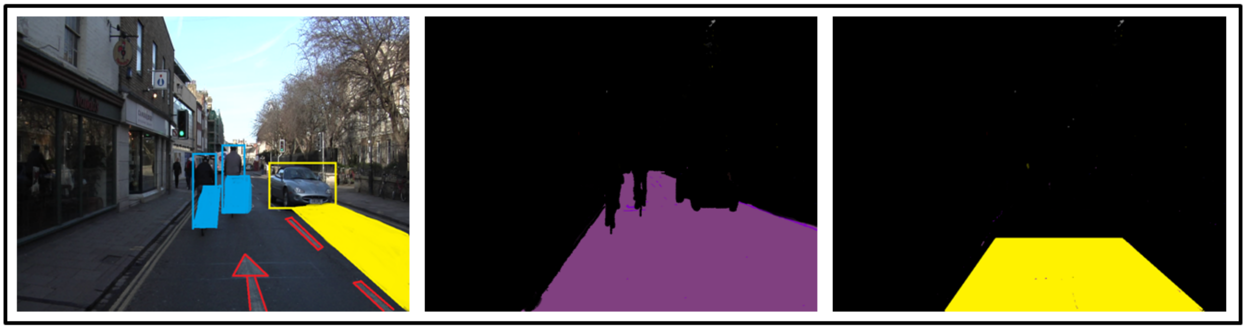


Figure 9. Data Annotation Strategy for DriveSpace

only encapsulates the visual information but also the spatial correlations between the objects of interest.

In the case of autonomous vehicles, multiple camera architectures can be leveraged to fuse more semantic information into the models. The same is shown in the proposed hybrid architecture. The proposed architecture involves two pipelines joined by a fusion algorithm.

Geometric Vision Pipeline: The geometric pipeline extracts dense depth information that helps in encoding the semantic information in an directly applicable environment related dimension compared to the image dimension. The major building blocks of proposed geometric pipeline are Dense Optical Flow (DOF) and Structure From Motion (SFM). Flow information can be used to estimate the six degrees of freedom camera motion for stereo rectification in addition to differentiating the dynamic and static objects. The depth estimation helps in clustering semantic information in the temporal domain.

Deep Semantic Pipeline: As stated earlier modeling DriveSpace as a pixel level semantic segmentation problem may not be an efficient solution because of the spatial and temporal correlations between the objects in the scene may not be captured. Conversely, the semantic information inferred by a well trained CNN can help the fusion algorithm to efficiently infer the free space region.

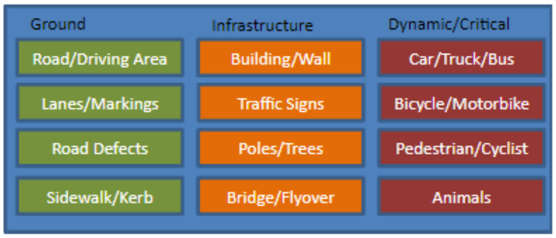
The major blocks of deep semantic pipeline are an encoder/decoder architecture based CNN for segmentation at image level and an object extraction algorithm to extract out the possible object like structures in the semantic map. The loss function for the semantic segmentation is mean softmax cross entropy over all the pixels. The number of classes here are limited to road, lanes, pedestrians and vehicles. While the road segmentation acts as a pixel level super set of DriveSpace, the lanes, pedestrians and vehicles brings the contextual information for extracting the subset of pixels from the road superset.

Fusion: Here we propose fusion and tracking in the image domain leveraging all image feature information which provides further semantic information. Image-level fusion of two complementary cues is from reconstruction and recognition where epistemic uncertainty provides confidence of what it is (e.g.: Edge has gradient score which provides confidence) and spatial uncertainty provides confidence of where it is on the image. Fusion is performed by combination of the two epistemic uncertainty heat maps. We propose a spatio-temporal object tracking using Probabilistic Graphical Models (PGM) after image level fusion. PGM can capture the global structure of the scene. Global context helps in improving robustness of detection. Figure 10 illustrates how human beings can easily detect the car because of the external context of road and buildings even though the image features of the car are very blurred. This can also help handle occlusion reasoning and handling of missing foot points.

Figure 10. Illustration of context based detection of car [26] (left) and occlusion handling of pedestrians from car (right)

To enable fusion, we obtain uncertainty heat maps from the estimations of the two blocks. For semantic segmentation, uncertainty can be obtained by Bayesian modeling as discussed in [25]. For geometric vision pipeline, uncertainties can be heuristically obtained by using confidence metric and prior noise variances. We model the uncertainties as random variables and do a weighted fusion to get a single object map. Once we get the objects, we categorize them into 3 parts based on their nature and criticality as shown in Figure 11 (top). Each object is then modelled as a random state vector as shown in Figure 11 (bottom). We use spatio-temporal PGMs to connect all these objects and find weights across object relationships. This way it can exploit spatial context to improve estimation (e.g.: pedestrian on curb/road) and can handle occlusion via spatial reasoning (e.g.: occluded pedestrian inferred from other pedestrians or occluding object reasoning). In order to meet real-time constraints, we propose to use approximate inference through Conditional Independence Assumptions (Markov), Sampling (Monte Carlo) and belief propagation.

One of the main advantages of PGM is the ability to encode prior knowledge in a Bayesian formalism. Automotive scenes have plenty of known rigid structure/rules which can be incorporated. There are varying levels of model complexity and uncertainty. For large uncertainty, generic model with least assumptions can be used (e.g.: Pedestrian Detection). For less uncertainty, assumptions can be encoded by experience/observation




```
Node[
        Spatial {
            X,Y,Z
            Bounding Box
            }
        Semantics {
            Class1: Pedestrian, posterior
            Class2: Vehicles, posterior
            Class3: Bicycles, posterior
            }
            Context {
            On road
            Near wall
            }
            Temporal {
            Velocity Direction
            Interaction
            }
    ]
```

Figure 11. *Classification of objects into 3 parts (top) and representation of each object (bottom)*

(e.g.: Lane Detection). Prior information could simplify model complexity greatly (e.g.: Rotation Matrix with less degrees of freedom). We propose incorporation of the following types of priors

- **Spatial**, e.g. lanes lie on ground plane.
- **Geometric**, e.g. lanes have predefined shape and thickness.
- **Color**, e.g. lanes are typically white.
- **Location**, e.g. lane location based on HD maps.

CONCLUSION

Free space and drivable area are important concepts for autonomous driving. In this paper, we generalize and contextualise these notions to DriveSpace. DriveSpace the provides the basis for context aware autonomous vehicles. We first motivated the need for defining DriveSpace via concrete use cases which require semantic understanding of the environment and making a context based decision. We proposed an architecture and discussed steps to achieve the solution. As it is a newly proposed concept, there is no dataset available and this remains the main bottleneck. Future work is to develop a DriveSpace dataset and systematically compare the various proposed methods.

ACKNOWLEDGEMENTS

The authors would like to thank their employer for the opportunity to perform research. Thank you to Etienne Perot for providing Figure 6 and Pantelis Ermilios for providing Figure 7. Thanks to Bruno Arsenali for reviewing the paper for us.

References

[1] S.-C. Lin, Y. Zhang, C.-H. Hsu, M. Skach, M. E. Haque, L. Tang, and J. Mars, "The architectural implications of autonomous driving: Constraints and acceleration," in *International Conference on Architectural Support for Programming Languages and Operating Systems*, 2018.

[2] S. D. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. H. Eng, D. Rus, and M. H. Ang, "Perception, planning, control, and coordination for autonomous vehicles," *Machines*, 2017.

[3] M. Schreier and V. Willert, "Robust free space detection in occupancy grid maps by methods of image analysis and dynamic b-spline contour tracking," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pp. 514–521, IEEE, 2012.

[4] L. Neumann, B. Vanholme, M. Gressmann, A. Bachmann, L. Kählke, and F. Schüle, "Free space detection: A corner stone of automated driving," in *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, pp. 1280–1285, IEEE, 2015.

[5] S. Hänisch, R. H. Evangelio, H. H. Tadjine, and M. Pätzold, "Free-space detection with fish-eye cameras," in *Intelligent Vehicles Symposium (IV), 2017 IEEE*, pp. 135–140, IEEE, 2017.

[6] A. B. Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: a survey," *Machine Vision and Applications*, 2012.

[7] C. Fernández, R. Izquierdo, D. F. Llorca, and M. A. Sotelo, "Road curb and lanes detection for autonomous driving on urban scenarios," in *IEEE 17th International Conference on Intelligent Transportation Systems (ITSC)*, 2012.

[8] Z. Liu, S. Yu, and N. Zheng, "A co-point mapping-based approach to drivable area detection for self-driving cars," *Engineering*, 2018.

[9] Y. Cai, D. Li, X. Zhou, and X. Mou, "Robust Drivable Road Region Detection for Fixed-Route Autonomous Vehicles Using Map-Fusion Images," *Sensors*, 2018.

[10] Z. Liu, S. Yu, X. Wang, and N. Zheng, "Detecting drivable area for self-driving cars: An unsupervised approach," *arXiv preprint arXiv:1705.00451*, 2017.

[11] S. Söntges and M. Althoff, "Computing the drivable area of autonomous road vehicles in dynamic road scenes," *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[12] R. Q. Mnguez, I. P. Alonso, D. Fernndez-Llorca, and M. A. Sotelo, "Pedestrian path, pose, and intention prediction through gaussian process dynamical models and pedestrian activity recognition," *IEEE Transactions on Intelligent Transportation Systems*, 2018.

[13] D. Tran, W. Sheng, L. Liu, and M. Liu, "A Hidden Markov Model based driver intention prediction system," in *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, 2015.

[14] S. Casas, W. Luo, and R. Urtasun, "IntentNet: Learning to Predict Intention from Raw Sensor Data," in *The 2nd Conference on Robot Learning*, 2018.

[15] A. Chavan and S. K. Yogamani, "Real-time dsp implementation of pedestrian detection algorithm using hog features," in *2012 12th International Conference on ITS Telecommunications*, pp. 352–355, IEEE, 2012.

[16] M. Siam, M. Gamal, M. Abdel-Razek, S. Yogamani, and M. Jagersand, "Rtseg: Real-time semantic segmentation comparative study," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 1603–1607, IEEE, 2018.

[17] M. Yu and G. Ma, "360 surround view system with parking guidance," *SAE International Journal of Commercial Vehicles*, vol. 7, no. 2014-01-0157, pp. 19–24, 2014.

[18] M. Heimberger, J. Horgan, C. Hughes, J. McDonald, and S. Yoga-

mani, “Computer vision in automated parking systems: Design, implementation and challenges,” *Image and Vision Computing*, vol. 68, pp. 88–101, 2017.

[19] M. Siam, S. Elkerdawy, M. Jagersand, and S. Yogamani, “Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges,” in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–8, Oct 2017.

[20] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[21] A. Gurram, O. Urfalioglu, I. Halfaoui, F. Bouzaraa, and A. M. Lopez, “Monocular depth estimation by learning from heterogeneous datasets,” in *2018 IEEE Intelligent Vehicles Symposium, IV 2018, Changshu, Suzhou, China, June 26-30, 2018*, pp. 2176–2181, 2018.

[22] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, “Bundle adjustment — a modern synthesis,” in *Vision Algorithms: Theory and Practice* (B. Triggs, A. Zisserman, and R. Szeliski, eds.), (Berlin, Heidelberg), pp. 298–372, Springer Berlin Heidelberg, 2000.

[23] A. Laddha, M. K. Kocamaz, L. E. Navarro-Serment, and M. Hebert, “Map-supervised road detection,” in *Intelligent Vehicles Symposium (IV), 2016 IEEE*, pp. 118–123, IEEE, 2016.

[24] Y. Cai, D. Li, X. Zhou, and X. Mou, “Robust drivable road region detection for fixed-route autonomous vehicles using map-fusion images,” *Sensors*, vol. 18, no. 12, p. 4158, 2018.

[25] A. Kendall and Y. Gal, “What uncertainties do we need in bayesian deep learning for computer vision?,” in *Advances in neural information processing systems*, pp. 5574–5584, 2017.

[26] A. Torralba, “Contextual priming for object detection,” *International journal of computer vision*, vol. 53, no. 2, pp. 169–191, 2003.

Author Biography

Ciarán Hughes is a computer vision architect at Valeo, focusing on algorithms for surround-view cameras. He has 15 years of experience in automotive computer vision and camera design. He has a Ph.D. in Electronics from NUI Galway.

Sunil Chandra is a computer vision algorithm lead and software architect at Valeo Vision system, Ireland. He has an MSc in Mathematics, MPhil in Applied Mathematics. He received his PhD in computer vision from Centre for Vision Speech and Signal Processing, University of Surrey, Guildford UK. He is having over 16 years experience in Computer Vision, spread across both Industrial and academic fields. He is working with Valeo from last 6 years on automotive computer vision for Advanced Driver Assistance Systems (ADAS). He co-invented 23 patents in the various ADAS applications.

Ganesh Sistu is a deep learning research engineer at Valeo Vision Systems, Ireland. His areas of interest includes Computer Vision, Image Analysis and Machine Learning.

Jonathan Horgan is a Computer Vision & Deep Learning Architecture Manager and Expert at Valeo Vision Systems. He has worked in the field of computer vision for over 14 years with a focus over the last 8 years on automotive computer vision for Advanced Driver Assistance Systems (ADAS) and automated parking & driving. He is now working on next generation advanced computer vision and deep learning with the ultimate goal of achieving fully autonomous driving and parking. He has 17 publications in peer reviewed conferences & journals and over 70 patents filed in the field of computer vision for ADAS and Autonomous driving.

Brian Deegan is a Senior Research Engineer and Senior Expert at Valeo Vision Systems. He received a Bachelor of Engineering in Computer Engineering in 2004, and a Masters in Science in Biomedical Engineering in 2005. He received a PhD in Biomedical Engineering in 2011. His main research interests are in automotive image quality for both viewing and machine vision applications.

Sumanth Chennupati is a Computer Vision Systems Engineer at R&D department of Valeo North America Inc. He has been with Valeo since 2016 and currently responsible for development of Advanced Driving Assistance Systems. He holds extensive academic and industrial research experience in Computer Vision, Machine Learning and Deep Learning. Sumanth is an author for several academic publications and serves as a reviewer for various IEEE conferences. Sumanth has a Master's degree in Computer Engineering with Specialization in Deep Learning from Rochester Institute of Technology, New York.

Senthil Yogamani is a computer vision architect and technical leader at Valeo Vision systems. He is currently focused on research and design of the overall computer vision algorithm architecture for surround-view camera visual perception in autonomous driving systems. He has over 13 years of experience in computer vision and machine learning including 10 years of experience in industrial automotive systems. He is an author of 50 peer reviewed publications and 33 patents. He serves in the editorial board of various leading IEEE automotive conferences including ITSC and ICVES and advisory board of various industry consortia including Khronos, Cognitive Vehicles and IS Auto. He is a recipient of best associate editor award at ITSC 2015 and best paper award at ITST 2012.