Near-field Sensing Architecture for Low-Speed Vehicle Automation using a Surround-view Fisheye Camera System

Ciarán Eising[1,2], Jonathan Horgan[1] and Senthil Yogamani[1]
[1]Computer Vision Platform, Valeo Vision Systems, Ireland
[2]Dept. of Electronic and Computer Engineering, University of Limerick, Ireland

Cameras are the primary sensor in automated driving systems. They provide high information density and are optimal for detecting road infrastructure cues laid out for human vision. Surround view cameras typically comprise of four fisheye cameras with 190° field-of-view covering the entire 360° around the vehicle focused on near field sensing. They are the principal sensor for low-speed, high accuracy and close-range sensing applications, such as automated parking, traffic jam assistance and low-speed emergency braking. In this work, we describe our visual perception architecture on surround view cameras designed for a system deployed in commercial vehicles, provide a functional review of the different stages of such a computer vision system, and discuss some of the current technological challenges. We have designed our system into four modular components namely Recognition, Reconstruction, Relocalization and Reorganization. We jointly call this the *4R Architecture*. We discuss how each component accomplishes a specific aspect and how they are synergized to form a complete system. Qualitative results are presented in the video at https://youtu.be/ae8bCOF77uY.

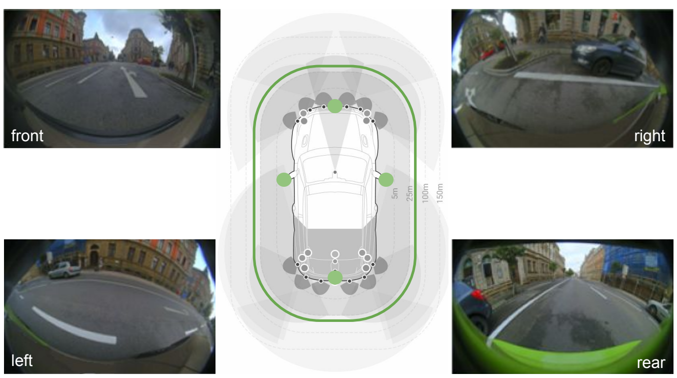
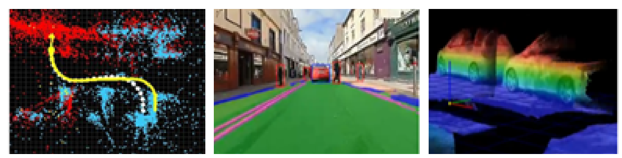


Fig. 1: Images from the surround-view camera network. Green perimeter shows 360° near-field sensing around the vehicle.



(a) (b) (c)

Fig. 2: Examples of (a) *Relocalization*, (b) *Recognition* and (c) *Reconstruction* in action on a surround-view camera system.

I. Introduction

Recently, Autonomous Driving (AD) gained huge attention with significant progress in deep learning and computer vision algorithms [1], where globally it is one of the highest trending technologies. Within the next 5-10 years, AD is expected to be deployed commercially [2], [3], [4], with widespread deployment in the coming decades. Currently, most automotive original equipment manufacturers (OEMs) around the world are working on development projects focusing on autonomous driving technology [5], [6], with computer vision having high importance [7], [8]. However, as more is asked from computer vision systems deployed for vehicle autonomy, the architectures of such become ever more complex. Thus, it is of advantage to take a step back, and consider the architectures at the highest level. While what we propose should be considered a general discussion on the structure of automotive computer vision systems, we will use specific examples of computer vision applied to Fisheye camera networks, such as surround view/visual cocoon (see Figure 1).

The work of this paper is inspired, in part, by the work of Malik et al. in [9]. The authors of that work propose that the core problems of computer vision are reconstruction, recognition and reorganization, what they dub as the 3Rs of Computer Vision. In this work, we propose to extend and specialise the 3Rs of Computer Vision to the 4Rs of Automotive Computer Vision: *Reconstruction*, *Recognition*, *Reorganisation* and *Relocalization*. We very much see this paper as a specialisation to the field of automotive computer vision of the work of Malik et al. Figure 2 shows examples of the first three Rs.

As with [9], *Reconstruction* means inferring scene geometry from a video sequence, including the position of the vehicle within the scene [10]. The importance of this should be obvious, as it is central to problems in scene mapping, obstacle avoidance, maneuvering and vehicle control. Malik et al. extend this beyond just geometric inference to include properties such as reflectance and illumination. However, these additional properties are not (currently, at least) significant in the context of automotive computer vision, and so we define *Reconstruction* in the more traditional sense of meaning 3D geometry recovery.

Recognition is the term used for attaching semantic labels to aspects of a video image or scene. As in [9], hierarchies are included in recognition. For example, a cyclist has a spatial hierarchy, as it can be divided into the subsets of bicycle and rider, and a vehicle category can have taxonomic subcategories of car, lorry, bicycle, etc. This can continue as far as is useful for an autonomous driving system. Lights can be categorised by the type (vehicle light, street lights, stop lights, etc.), colour

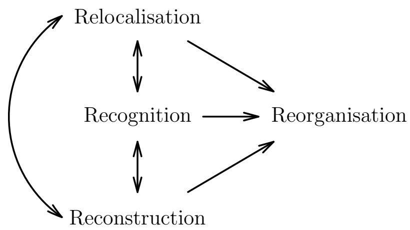


Fig. 3: The 4Rs of automotive vision, and their information flow paths.

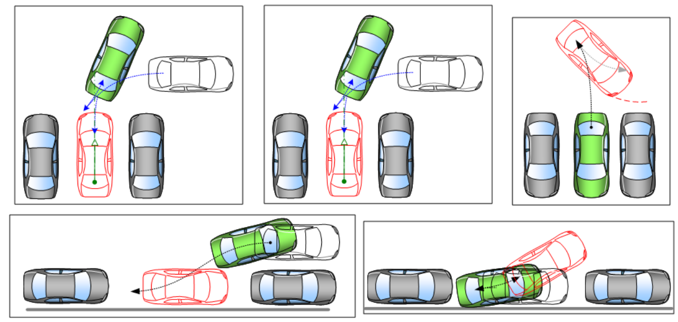


Fig. 4: Typical Parking use cases. Top row from left to right: Perpendicular Backward Parking In, Perpendicular Forward Parking In and Perpendicular Backward Park Out. Bottom row from left to right: Parallel Backward Parking In and Park Out.

Fig. 5: Parking with cameras can be guided by road markings and not just objects unlike Ultrasonic and Radar.

(red, yellow, green), and their importance to the autonomous vehicle (need to respond, can ignore), which infers higher level reasoning of the system.

Relocalization is place recognition and metric localization of a vehicle relative to its surroundings. Relocalization can happen against a pre-recorded trajectory in the host vehicle, for example, for trained parking [11], or against a map that is transferred from the infrastructure, for example, HD Maps [12]. It is highly related to loop closure in SLAM [13], though rather than consider just the problem of loop closure, we consider the broader problem of the localization of the vehicle against one or many pre-defined maps.

Reorganization is the approach of combining information from the previous three components of computer vision into a unified representation. In the work of Malik et al., reorganisation is derived from the term "perceptual organisation", and roughly equates it with segmentation. Image segmentation approaches are now dominated by CNN-based semantic segmentation and instance segmentation and therefore fall into the domain of recognition. In this paper, we use the term to equate with "late fusion", which is the manipulation, filtering and reorganisation of inputs into a unified output. This is an important step in the context of vehicle automation, as a unified representation of the sensor outputs is required for a vehicle control response. This also admits the fusion of the outputs of multiple cameras at a late stage and can be a pre-filter for automotive sensor fusion [14].

The useful information flow paths are shown in Figure 3. As we shall discuss, each of the three of Relocalization, Recognition and Reconstruction have useful information for the other, and all three feed the Reorganisation component. It could be argued that even Reorganisation has useful information for the other three components. While recurrent systems have seen some traction in specific automotive vision problems [15], [16], this is not commonplace at an architecture level in automotive visual systems.

Malik et al. [9] describe, with reference to early work of Marr [17], that in early years of computer vision, the concept of advantageous division of computer vision into low, medium and high level tasks was popular, but later became redundant. Later in this paper, we will argue that there is still merit to that mode of thinking, particularly when considering real-time embedded computer vision systems, such as those common in the automotive industry. For example, accessing the pixel information from an image is memory bandwidth intensive, and so doing all intensive pixel processing in a single step is logical. Therefore, we also propose a sub-division of computer vision into processing stage, the stages being *Pixel Processing*, *Intermediate Processing* and *Object* stages, wrapped by *pre-* and *post-processing* stages. We will generally refer to this as the computer vision *pipeline* or *pipeline stages*.

Additionally, while what we discuss can, in the general sense, be applied to the entire field of automotive computer vision, we must acknowledge that convolutional neural networks (CNNs) have become the standard building block for many visual perception tasks in vehicle autonomy. Thus, many of our examples will focus on the role of neural networks. Bounding boxes for object detection is one of the first successful applications of CNNs for detecting not only pedestrians and vehicles, but also their positions. Recently semantic segmentation is becoming more mature [18], [19], starting with detection of roadway composition like road surface, lanes, road markings, curbs, etc. CNNs are also becoming competitive for geometric vision tasks like depth estimation [20] and Visual SLAM [21]. However, the reader should keep in mind that the discussions in this paper are not be strictly restricted to fisheye cameras nor to CNN-based processing. These should simply be considered as pertinent examples to further the argument being made.

The rest of the paper is structured as follows. Section II discusses pertinent low-speed, near-field sensing use cases, we will introduce the concept of the 4Rs of computer vision, and we provide a brief overview of the *WoodScape* dataset [22], which is the dataset we use to provide some results later in the paper. Section III provides an overview of a surround view sensing system architecture and how the visual perception is structured. Section IV discusses the components of 4R in detail individually and Section V discusses the interactions between

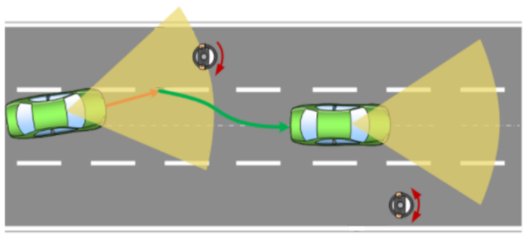


Fig. 6: In traffic jam assist systems, the vehicle can assume both lateral and longitudinal control, including stopping and starting the motion of the vehicle.

Fig. 7: Surround view cameras can be used for traffic jam assistance systems by detection leading vehicles and lane markings, for example. In addition, vulnerable road users (e.g. crossing pedestrians) that are outside the angular range of other sensors can be detected.

them. Finally, Section VI summarizes the paper and provides potential future directions.

II. Low speed vehicle automation framework

Fisheye cameras offer a distinct advantage for automotive applications. Given their extremely wide field of view, they can observe the full surrounding of a vehicle with a minimal number of sensors. Typically four cameras is all that is required for full 360° coverage of a car (Figure 1). However, this advantage comes with a cost given the significantly more complex projection geometry exhibited by fisheye cameras. The reader is referred to [23] for a discussion on aspects of automotive camera networks, and to [11] for an introduction to the use of fisheye camera networks for automated parking systems.

A. *Near field sensing use cases*

Here we will discuss a few of the most pertinent use cases in vehicle autonomy for surround view computer vision systems.

1) *Automated Parking Systems*

Automated parking systems are one of the primary use cases for short range sensing [11], with some typical parking use cases described in Figure 4. As early as 1992, prototypes of semi-automated parking systems using radar systems were proposed, though not produced commercially [24]. Early commercial partially automated parking systems employed either ultrasonic sensors or radar [25], [26]. However, more recently, surround view cameras are becoming one of the primary sensors for automated parking [11], [27], extending the capabilities of or providing inexpensive alternatives to other sensors. A major limitation of ultrasonic and radar sensors for automated parking is that parking slots can only be identified based on the presence of other obstacles (Figure 5). Extending this, surround view camera systems allow for parking in the presence of visual parking slot markings such as painted line markings while also being seen as a key enabling technology for Valet Parking systems to become a reality [28], [29].

2) *Traffic Jam Assistance Systems*

As a substantial proportion of accidents are rear-end collisions at low speed [30], traffic jam situations are considered to be one of the areas of driving that automation can give benefit in the short term [31], though current systems perhaps lack robustness [32]. In automated traffic jam assistance systems, the vehicle assumes control of the longitudinal and lateral position while in the traffic jam scenario (Figure 6), in contrast to adaptive cruise control, where only the velocity is controlled. Traffic jam assistance functionality is typically used in low speed environments, with maximum speeds of ∼60kph [32], though even lower maximum speeds of 40kph are suggested [33]. While typically highway scenarios are considered for traffic jam assistance [31], there has been investigation into the urban traffic jam assistance systems [34]. Given the low speed nature of traffic jam assist, surround view cameras are an ideal sensor, particularly in urban settings where, for example, pedestrians can attempt to cross from areas that are outside the field of view of traditional forward facing cameras or radar systems. Figure 7 shows examples of using surround view cameras for traffic jam assist. In addition to detecting other road users and markings, features such a depth estimation [20] and SLAM [21] are also important for inferring distances to objects and controlling the vehicle position.

3) *Low Speed Braking*

Protection of vulnerable road users in low speed reversing situations has become a focus of legislation in some jurisdictions [35], with initial efforts to simply display the rearward portion of the vehicle to the driver [36]. More recently, work has gone into the automated braking of the vehicle in such low speed scenarios. This has proven fruitful, as it is shown in one study that automatic rearward braking significantly reduced collision claim rates [37], with vehicles equipped with rear camera, parking assistance and automatic braking showing a 78% reduction of reported collisions. Low speed braking, not just rearward, but forward as well, is important in reducing injury and material damage [30]. Surround view camera systems are extremely useful for low speed braking, as the combinations of depth and object detection are building blocks for this functionality.

B. *WoodScape dataset*

Even though this paper should be considered as part review and part positional, throughout we will offer insights into results to support our arguments. Mostly, though not ubiquitously, we will use the Woodscape Dataset [22]. It is therefore pertinent to give a description of the dataset, which was captured in two distinct geographical locations: USA and Europe. While the majority of data was obtained from saloon vehicles there is a significant subset from a sports utility vehi-

Task	Model	Metric	Value
Recognition			
Segmentation	ENet [38]	IoU	51.4
2D Bounding Box	Faster R-CNN [39]	mAP (IoU>0.5)	31
Soiling Detection	ResNet10 [40]	Category (%)	84.5
Reconstruction			
Depth Estimation	Eigen [41]	RMSE	7.7
Motion Segmentation	MODNet [42]	IoU	45
Visual Odometry	ResNet50 [40]	Translation (<5mm)	51
		Rotation (<0.1°)	71
Relocalization			
Visual SLAM	LSD SLAM [43]	Relocalization (%)	61

TABLE I: Summary of baseline results of 4Rs on our WoodScape dataset.

cle ensuring a strong mix in sensor mechanical configurations. Driving scenarios are divided across highway, urban driving and parking use cases. Intrinsic and extrinsic calibrations are provided for all sensors as well as timestamp files to allow synchronization of the data. Relevant vehicle's mechanical data (e.g. wheel circumference, wheel base) are included. High-quality data is ensured via quality checks at all stages of the data collection process. Annotation data undergoes a rigorous quality assurance by highly skilled reviewers. The sensors recorded for this dataset are listed below:

- 4x 1MPx RGB fisheye cameras (190° horizontal FOV)
- 1x LiDAR rotating at 20Hz (Velodyne HDL-64E)
- 1x GNSS/IMU (NovAtel Propak6 & SPAN-IGM-A1)
- 1x GNSS Positioning with SPS (Garmin 18x)
- Odometry signals from the vehicle bus.

Table I gives an overview of the baseline results for this dataset for three of the four Rs.

III. System Architecture

A significant consideration in the design of automotive computer vision, in particular the pipelining, is the constraints of embedded systems in which multiple cameras and multiple computer vision algorithms must run in parallel. It is therefore useful to give a brief overview in order to understand the constraints better, though readers are referred to [11] for a more detailed review of these considerations. This covers all aspects from "glass-to-glass", i.e. from the camera lens to the display. In a spatially dispersed vehicular camera network, the communication channel from camera to ECU is typically either Ethernet or LVDS.

Perhaps the most important component to consider is the System-on-Chip (SoC), and typically the first step in designing a commercial automotive camera system is the selection of the SoC for embedded systems, based on criteria including performance (Tera Operations Per Second (TOPS), utilisation, bandwidth), cost, power consumption, heat dissipation, high to low end scalability and programmability. The SoC choice provides the computational bounds in the design of algorithms. As computer vision algorithms are compute intensive, Automotive SoCs have a lot of dedicated hardware accelerators for image signal processing, lens distortion correction, dense optical flow, stereo disparity, etc. It is essential to design the computer vision architecture leveraging these HW accelerators.

In computer vision, deep learning is playing a dominant role in various recognition tasks and gradually for geometric tasks, like depth [20] and motion estimation [44]. The progress in CNN has also led to the hardware manufacturers typically including a custom hardware intellectual property core to provide a high throughput of over 10 TOPS [45].

In [11], we provide a detailed discussion on the use of SoCs in automotive computer vision systems, and basically boil down the major factors to system cost and performance. Other factors, such as power consumption, programmability and heat dissipation, while important, aren't typically as critical in the selection process.

A. Pipeline Stages

As mentioned previously, when designing a vision system, particularly an embedded vision system, it is advantageous to think in terms of processing stages, and to consider commonality and shared processing at each processing stage. In this manner, expensive early operations are shared amongst later processing stages that are closer to application layers. This maximizes the performance from any piece of hardware. In Figure 8, we show an example of a 4R architecture that has the pipelines split into the processing stages that we now discuss.

1) Pre-processing:

The pre-processing stage of the pipeline can be thought of as the processing that prepares the data for computer vision. This consists of the Image Signal Processing (ISP). ISP are the steps of processing performed on an image, such as White Balance, Denoise, Color Correction and Color Space Conversion. For a full discussion on ISP and the tuning of ISP for computer vision tasks in an automotive setting, the reader is referred to [46]. ISP is usually done by hardware engines, on the sensor chip itself, as a companion chip, or as part of the primary SoC. It is rarely done in software, as there is a vast amount of pixel level processing to be completed. Methods are being proposed to automatically tune the hyperparameters of ISP pipelines to optimize the performance of computer vision algorithm [46], [47]. It should be noted that methods are being proposed to simplify the ISP pipeline for visual perception [48].

2) Pixel Processing Stage:

Pixel processing can be considered as those parts of a computer vision architecture that *touch the image* directly. In the context of classical computer vision, these would be algorithms such as edge detection, feature detection, descriptors, morphological operations, image registration, stereo disparity and so on. Readers are referred to any one of a number of textbooks for further information on these steps, such as [49]. In the context of neural networks, this would equate with the early layers of a CNN encoder. The processing at this stage is dominated by relatively simple algorithms that must run on potentially millions of pixels many times each second. That is, the computational cost is associated with the fact that these algorithms run many many times, rather than the complexity of the algorithm itself, and when considering embedded systems, often memory constraints can dominate. Processing hardware at this stage is typically dominated by Hardware Accelerators and GPUs, though some elements may be suitable for DSP.

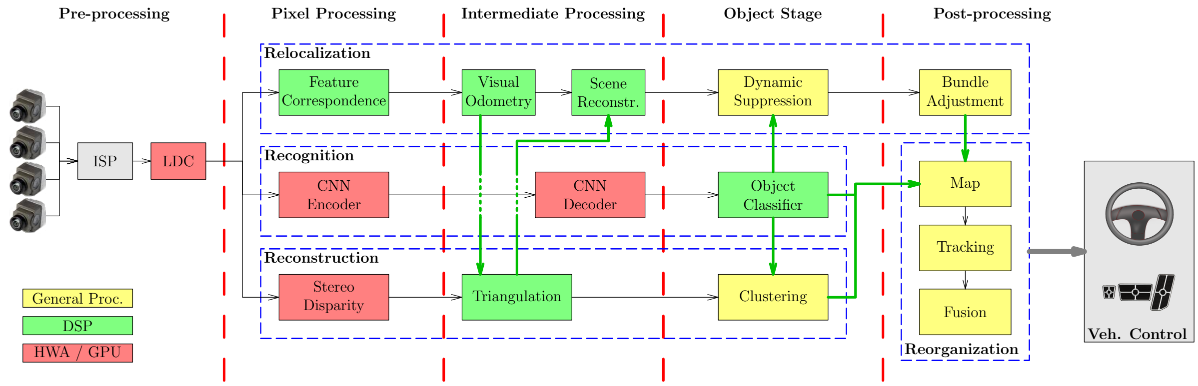


Fig. 8: High level architecture of our 4R framework. The 4Rs are shown with each of the processing stages. Useful communication between each of the 4Rs are shown as green arrows. Each block has an example target processing unit, being Hardware Accelerator (HWA) or GPU, Digital Signal Processor (DSP), or general purpose processing unit, such as ARM. LDC = Lens Distortion Correction, ISP = Image Signal Processor.

3) Intermediate Processing Stage:

As the name suggests, the intermediate processing stage bridges the gap from the pixels to the object detection stage. Here, the amount of data to process is still high, but significantly lower than the pixel processing stage. This may include steps such as, given a set of image correspondences, returning an estimate of vehicle motion through visual odometry, stereo triangulation of a disparity map, and the general feature-wise reconstruction of a scene. We would also include CNN decoders at this stage of the pipeline. The naming of this stage is vague, as it should be seen as preparing data for the Object Stage of the pipeline. Processing hardware at this stage would typically be digital signal processors.

4) Object Processing Stage:

The object processing stage is where higher level reasoning is incorporated. It is here that we may cluster point clouds to create objects, where objects are classified (including things like motion classification), and where, through said reasoning, we can apply algorithms to suppress relocalization on movable objects. The processing at this stage is dominated by more complex algorithms, but operating on fewer data points. In terms of hardware, it is often suitable to run these on general purpose processing units, such as ARM, though digital signal processors would commonly be utilised as well.

5) Post-processing:

Finally, we have the post processing stage, which could also be termed the *global stage* of processing. It is here that we persist data temporally and spatially. As we can have long temporal persistence and large spatial maps, the overall goal of the preceding stages is to minimise the amount of data reaching this stage while maintaining all of the pertinent information that will finally be used for vehicle control, and without this data minimisation, this stage of the processing pipelines would become infeasibly memory and processing intensive. In this stage, we would include steps such as bundle adjustment, map building, high level object tracking and prediction and fusion of the various computer vision inputs (e.g. fusing CNN-based vehicle detection with a stereo-generated point cloud). As we are dealing with the highest level of reasoning in the system, and ideally with the lowest amount of data points, general purpose processing units are typically desirable here.

IV. 4R Components

It is useful to begin the detailed description of the 4Rs with an example, and so, in Figure 8, we show at a high level what a computer vision architecture for autonomous driving might look like. Four video streams from an automotive surround view system (Figure 1) are passed to an SoC. Each algorithmic block is mapped to one of a set of available processing units that are typically available on high-end automotive vision SoCs, being hardware accelerator or graphical processing unit, digital signal processor, and general purpose processing core (e.g. ARM). Each of the 4R pipelines shows one of the standard algorithms, being visual SLAM for relocalization, CNN for object detection and motion stereo for reconstruction. Reorganization shows a typical map, tracking and fusion pipeline.

What is of interest is the possible links between the 4Rs, even in such a standard system. For example, Visual Odometry is a natural part of any SLAM pipeline, but can be reused in a motion stereo [50] context for dense triangulation, and equally, the stereo triangulation can be used for the scene reconstruction component of visual SLAM, which is really a seed for bundle adjustment. Robust SLAM can only be achieved in scenes that are dominated by dynamic objects if motion segmentation is readily available [51]. CNNs provide options on moving object detection [42], [52], potentially as part of a multi-task network [53], [54], and can provide the motion segmentation. Motion segmentation can also be used to suppress issues with incorrect triangulation.

Now that we have discussed briefly a simple example of how the 4Rs architecture might work, we will describe in more detail what the 4Rs are, what the processing stages are, and how the 4Rs of automotive computer vision can provide support for one another to improve overall system performance.

A. Recognition

The *Recognition* task identifies the semantics of the scene via pattern recognition. The first successful application was pedestrian detection which was performed using a combination of hand-designed features like Histogram of Oriented Gradients (HOG) and a machine learning classifier like Support Vector Machines (SVM). Recently, this approach has been superseded by CNNs. Automotive scenes are very diverse and the system is expected to work across countries as well as varying weather and lighting conditions. One of the main challenges is to build an effective dataset which covers diverse aspects [55]. CNNs are computationally intensive and efficient design techniques are critical to be incorporated [56], [57]. CNNs are well studied for rectilinear images, however, the assumption of translation invariance is broken in fisheye images which poses additional challenges [58]. In particular, bounding box object detection representation which is used as a standard representation breaks for fisheye images [59]. Typically the CNN models are single image based but in automotive scenes they can be extended to leverage the temporal structure [60].

CNN (Convolutional Neural Networks) or Deep learning have demonstrated remarkable performance leaps for various computer vision tasks especially for object recognition applications [61]. Our proposed recognition pipeline uses a multi-task deep learning network for identifying objects based on their appearance patterns. It comprises of three tasks namely bounding box objection detection (pedestrians, vehicles and cyclists), semantic segmentation (road, curbs and road markings) and lens soiling detection (opaque, semi-transparent, transparent, clear). Object detection and semantic segmentation are standard tasks and for more implementation details, refer to our FisheyeMultiNet paper [62]. One of the challenges is to balance the three tasks weights during training phase as one task may converge faster than the others [63]. Additional auxiliary tasks like end to end driving which do not have associated annotation costs can aid the training of expensive annotation tasks such as segmentation [64].

Fisheye cameras are mounted relatively low on a vehicle (∼0.5 to 1.2m above ground) and are susceptible to lens soiling due to road spray from other vehicles or water from the road. Thus, it is vital to detect soiling on the camera lens in order to alert the driver to clean the camera or to trigger a cleaning system. The soiling detection task and its usage for cleaning and algorithm degradation is discussed in detail in SoilingNet [65]. This implementation was further optimized for efficiency in TiledSoilingNet [66] where the output was quantized to fewer tiles. It is quite challenging to build a diverse dataset for soiling and thus we explored GAN based techniques to learn a generative model for soiling and use it to generate diverse patterns [67]. A closely related task is desoiling where the soiled areas are restored through inpainting [68], but these desoiling techniques remain in the domain of visualisation improvements rather than usage for perception for now. It is an ill-defined problem as it is not possible to predict behind the occlusion, however this can be improved by leveraging temporal information. As the CNN processing capacity is limited on the low power automotive ECU, we make use of multi-task architecture where majority of the computation is shared in the encoder as illustrated in Figure 9.

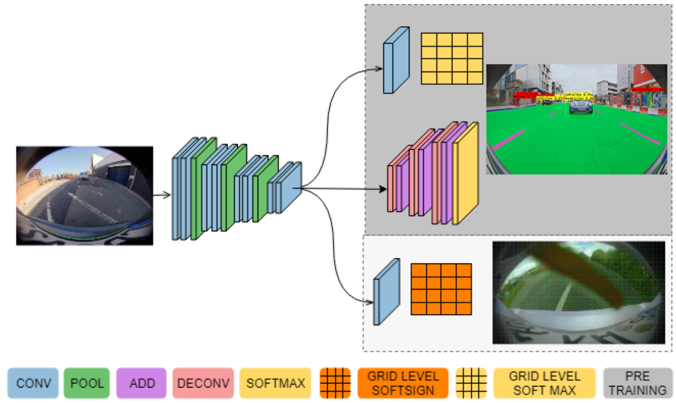


Fig. 9: Illustration of multi-task Recognition architecture comprising of object detection, semantic segmentation and soiling detection tasks.

B. Reconstruction

As mentioned already, *Reconstruction* means inferring scene geometry from a video sequence. This typically means, for example, estimating a point cloud or voxelised representation of a scene. However, we can also consider a temporal aspect of this – if the object is moving, we wish to know it's vector of motion.

The first aspect, the reconstruction of static objects, is traditionally done using approaches such as motion stereo [50] or triangulation in multi-view geometry [69]. In the context of designing a depth estimation algorithm, a brief overview of how humans infer depth is given in [70] with useful further referencing. There are four basic approaches to inferring depth: monocular visual cues, motion-parallax, stereopsis and depth from focus. Each has it's equivalent in computer vision.

Based on earlier theoretical work by Marr & Poggio [71], Grimson provided a computational implementation of stereo vision in the early 1980s [72]. Since then, work has continued on stereo vision (see [73] for an overview of early works). However, stereo systems do not achieve ubiquitous deployment on vehicles, and as such, monocular motion-parallax methods remain popular in automotive research. To any reader, motion-parallax is intuitive – as one moves their head, distant objects appear to move further than close objects in our retinal image. Computationally, depth from motion parallax is traditionally done through feature triangulation [74], but motion stereo has also proven popular [75]. Neural network approaches to extracting pixel-wise depth from moving cameras have also proven successful in recent years [41], [20], [76].

Perhaps one of the more interesting approaches is the use of monocular cues for depth. These are things like texture scale change, occlusions (if A occludes B, then B must be behind A), shading and lighting, object scale (small vs far away), etc. As noted in [70], such monocular cues require contextual interpretation. To work well, they require knowledge of

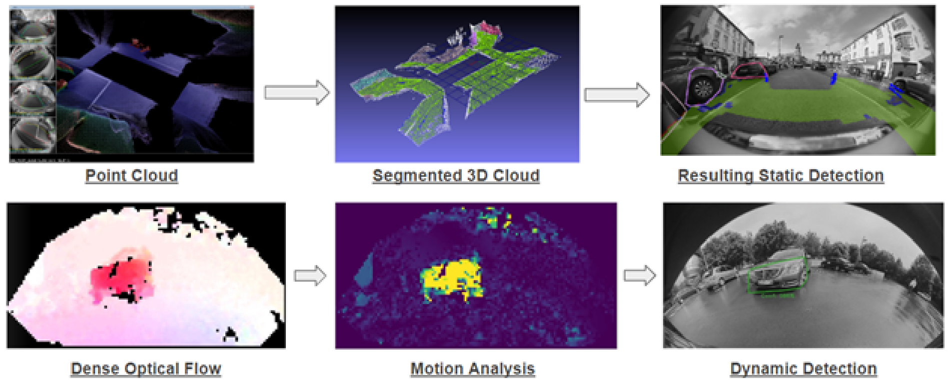


Fig. 10: An example of a reconstruction pipeline with some sample outputs at different processing stages. Top row shows the static pipeline and bottom row shows the dynamic pipeline.

the entire image, and patch-based approaches generally fail. Saxena et al. proposed a hierarchical Markov Random Field approach [70]. Van Dijk and Croon [77] have shown that, for four publicly available mono-depth networks at least, neural networks learn a correlation between vertical position of the object in the image and the depth to the object.

In the automotive context, stereo depth is the most robust depth estimator, though issues with hardware cost and restricted field of view limit ubiquitous deployment. Single-view (one camera, one frame) approaches are extremely desirable, as they remove the need for the camera to move for depth extraction, and solve problems such as the focus-of-expansion that exist with motion-based reconstruction. However, as yet they have proven not to be robust enough [77], though they may be useful still in certain scenarios. For example, having a non-robust depth estimate when the car is still may be better than no estimate. Motion-based depth estimation continues to be popular, particularly for surround view systems where the overlapping field of view of the cameras is negligible.

The second aspect of reconstruction is the extraction of moving objects from the video sequence (motion segmentation). 3D-reconstruction of dynamic objects results in position inaccuracy in the global sense, as triangulation assumptions are broken. Typical attempts to reconstruct the geometry of an object under motion requires image motion segmentation, relative fundamental matrix estimation and reconstruction (with scale/projective ambiguity). Of course, significant advances have been made. For example, using Multi-X [78] the first two steps can essentially be combined, as the segmentation can be done based on the fundamental matrix estimation. However, such approaches tend to be either computationally too expensive or not robust enough for embedded automotive applications. Additionally, scale must be resolved for such reconstruction, and objects (such as pedestrians) can have different fundamental matrices for different parts of the body. Therefore, in automotive, the task of dynamic object detection is usually simply motion segmentation.

Klappstein et al. [79] describe a geometric approach to motion segmentation in the automotive context. This work is significantly extended to the surround view camera case by Mariotti and Hughes [80]. However, in both cases the geometry cannot perfectly distinguish all types of moving feature. That is, there is a class of object motion that makes associated features indistinguishable from static features. Thus, a global or semi-global approach must be taken. In traditional approaches, this is done by grouping optical flow vectors with similar properties to ones that are classed as under motion. CNNs offer globality in a more native way [52]. However, as with the static object reconstruction, the results from [52] seem to indicate that it is performing recognition rather than geometric motion estimation, as still pedestrians are often classed as under motion. It is therefore likely that a much improved overall motion segmentation will be obtained by incorporating the geometric constraints of [80] with the more global CNN approach of [52], though this is certainly non-trivial and remains work in progress.

Figure 10 shows an example of different reconstruction stages, including dense motion stereo, 3D point cloud and a clustering of static obstacle, alongside a dense optical flow base motion segmentation.

C. Relocalization

Visual Simultaneous Localization And Mapping (VSLAM) is a well studied problem in robotics and autonomous driving. There are primarily three types of approaches namely (1) Feature based methods, (2) Direct SLAM methods and (3) CNN approaches. Feature based methods make use of descriptive image features for tracking and depth estimation [81] which results in sparse maps. MonoSLAM [82], Parallel Tracking and Mapping (PTAM) [83] and ORBSLAM [84] are seminal algorithms of this type. Direct SLAM methods work on the entire image instead of sparse features to aid building a dense map. Dense Tracking and Mapping (DTAM) [85] and Large-Scale Semi Dense SLAM (LSD-SLAM) [43], [86] are the popular direct methods which are based on minimization of photometric error. CNN based approaches are relatively less mature for Visual SLAM problems and they are discussed in detail in [87].

Mapping is one of the key pillars of autonomous driving. Many first successful demonstrations of autonomous driving (e.g: by Google) were primarily reliant on localization to pre-mapped areas. HD maps such as TomTom RoadDNA [88]

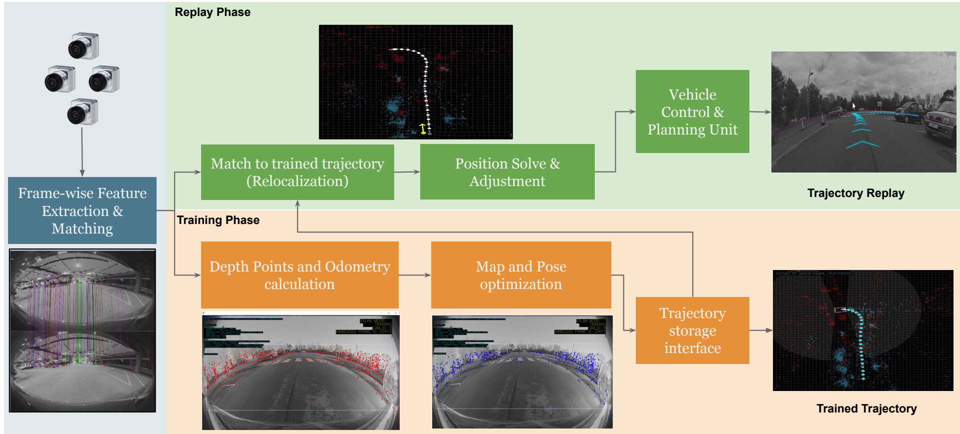


Fig. 11: Relocalization pipeline with intermediate outputs.

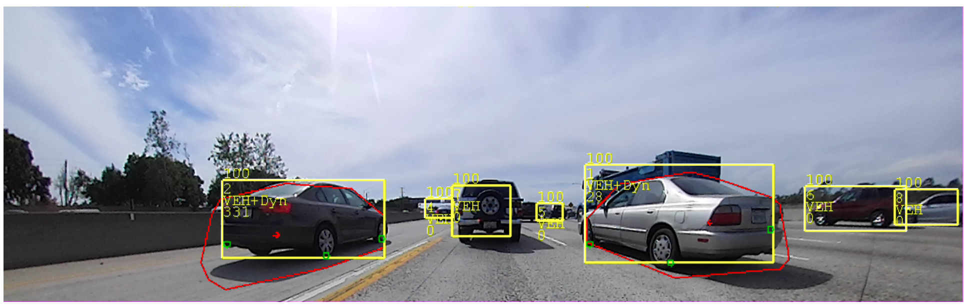


Fig. 12: Reorganization combines Reconstruction's dynamic object detection output (red polygon) and Recognition's object detection (yellow box).

provide a highly dense semantic 3D point cloud map and localization service for majority of European cities with a typical localization accuracy of 10 cm. When there is an accurate localization, HD maps can be treated as a dominant cue, as a strong prior semantic segmentation is already available and it can be refined by an online segmentation algorithm [89]. However, this service is expensive as it requires regular maintenance and upgrades of various regions in the world. Due to privacy laws and accessibility, such a commercial service cannot be used in many situations and a mapping mechanism has to be built within a vehicle's embedded system. For example, a private residential area cannot be mapped legally in many countries, such as Germany [90].

Visual SLAM (VSLAM), in the automotive context, consists of building a map of the environment surrounding the vehicle while simultaneously estimating the current pose of the car within that map [91]. One of the key tasks of VSLAM is the localization of the vehicle against a previously recorded trajectory [92]. A trained trajectory is typically represented by a group of key poses surrounded by landmarks spanned from the vehicle's origin to destination positions. These landmarks are represented using robust image features that are unique in the captured images.

A classical feature-based relocalization pipeline is shown in Figure 11. In feature-based SLAM, the first step is the extraction of salient features. A salient feature in an image could be a region of pixels where the intensity changes in a particular way, such as an edge, a corner or a blob [93], [94], [95]. In order to estimate landmarks in the world, tracking is performed, wherein two or more views of the same features can be matched. Once the vehicle has moved a sufficient amount, VSLAM takes another image and extracts features. The corresponding features are reconstructed to get their coordinates and poses in real world. These detected, described and localised landmarks are then stored in persistent memory in order to describe the relative position of the vehicle for a trajectory. If the vehicle returns to same general location the live feature detections are matched against the stored landmarks in order to recover the vehicle's pose relative to the stored trajectory.

D. Reorganization

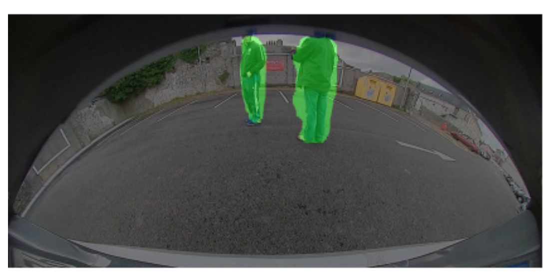


Fig. 13: Learning geometry can have a heavy reliance on recognition - static pedestrians detected as moving [52].

Reorganization performs three functions - 1) Fusion of Recognition and Reorganisation, 2) Mapping of objects in a

Method	Resolution	Abs Rel	Sq Rel	RMSE	$RMSE_{log}$	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
		lower is better				higher is better		
EPC++ [96]	640 x 192	0.141	1.029	5.350	0.216	0.816	0.941	0.976
Monodepth2 [97]	640 x 192	0.115	0.903	4.863	0.193	0.877	0.959	0.981
PackNet-SfM [98]	640 x 192	0.111	**0.829**	4.788	0.199	0.864	0.954	0.980
FisheyeDistanceNet [81]	640 x 192	0.117	0.867	4.739	0.190	0.869	0.960	0.982
SynDistNet [99]	640 x 192	**0.109**	0.843	**4.594**	**0.186**	**0.878**	**0.968**	**0.986**
Monodepth2 [97]	1024 x 320	0.115	0.882	4.701	0.190	0.879	0.961	0.982
FisheyeDistanceNet [81]	1024 x 320	0.109	0.788	4.669	0.185	0.889	0.964	0.982
SynDistNet [99]	1024 x 320	**0.103**	**0.705**	**4.386**	**0.164**	**0.897**	**0.980**	**0.989**

TABLE II: Extract of our results presented in [99]. Comparison of the SynDistNet network, which combines semantic segmentation and depth estimation in a single network, with other monocular methods. KITTI dataset [100] is used in all cases, using raw depth maps as proposed by [101] for evaluation.

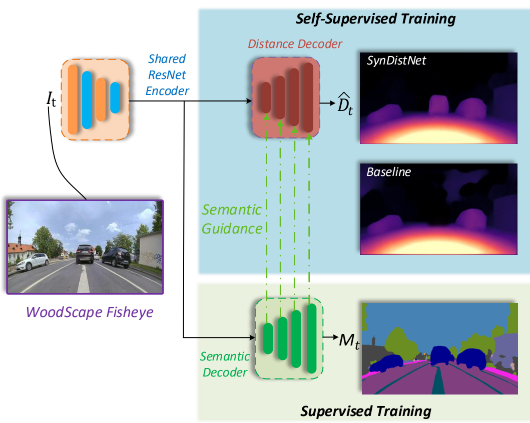


Fig. 14: Overview over the joint prediction of distance and semantic segmentation from a single input image [99].

centralized world co-ordinate system across cameras and 3) Temporal tracking of objects. Although it would be possible for the recognition and reorganisation blocks to feed directly into the environment map we contend there are distinct advantages to implementing some fusion at the vision layer. Typically an environment map operates in the world domain. Projections from the image domain to the world domain are known to be error prone, they are subject to errors from poor calibration, flat ground assumptions, variations in footpoint detection, pixel density and imperfect camera models. Detections in the image domain, prior to projections to the world, are not subject to such error and therefore association of detections from different vision algorithms in the image domain is more robust. Figure 12 shows an example of image-based fusion of vehicles using CNN based bounding box and optical flow based moving object polygon. Tracking can be either be performed in the image co-ordinate system or in the object map. Inconsistencies between the complementary algorithms can be reasoned with where full image data exists. Data can be refined, for example the semantic segmentation from road can be subtracted from the bounding box classification of a vehicle to get obtain a greatly improved object position.

E. Discussion

Overall, we argue that the 4R approach provides a localized semantic-geometric representation of the vehicles environment. By *localized semantic-geometric* representation of the vehicles environment, we mean *localized*: that the 4R processing pipeline provides information about where the vehicle is (can be globally or against pre-learned trajectories), *geometric*: information about the spatial relationship between the vehicle and obstacles in its local environment, and *semantic*: the obstacles will be recognized as belonging to a class of obstacle.

V. System Synergies

In this section, we will discuss system synergies. We will look at how Relocalization, Reconstruction and Recognition tasks can support one another, and we will describe the importance of dual sources of detection in providing redundancy in safety critical applications.

A. Recognition and Reconstruction

As already mentioned, depth estimation is important in geometric perception application. In addition to previous material already discussed, the current state of-the-art are neural network-based methods [102], [103], learnable in a self-supervised manner through reprojection loss [104]. It has been shown that state-of-the-art single frame attempts at monocular depth estimation typically results in a recognition tasks [105], and then using cues such as vertical position in the image to infer depth [77]. Moving object detection appears to have a heavy reliance on recognition as well. This is evidenced by the fact that both [42] and [52] show false positives on static objects that are commonly moving (pedestrians, for example - see Figure 13). This does not, in any way, reduce the importance of such attempts. Rather, it points to a very deep connection between recognition and reconstruction, and that from one, you can infer the other.

When bounding box pedestrian detection was state of the art, before semantic and instance segmentation, most researchers in automotive pedestrian detection will have considered encoding a depth based on the height of the bounding box, or the vertical position of the pedestrian in the image. This is discussed in detail in [77]. However, it is somewhat intuitive that recognition based on deep neural networks can lead to object depth, especially as the accuracy of neural networks improves. Recent work demonstrates the validity of joint learning of semantic labels and depth [106]. For example, in [99], it is shown that, for monocular depth estimation, adding semantic guidance in each of the distance decoder layers (per Figure 14) improves performance at edges of objects, and even returns reasonable distance estimates for

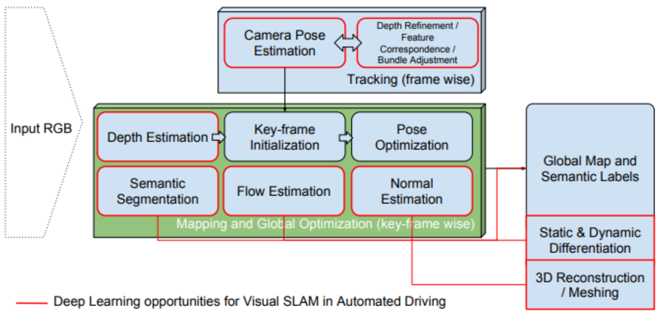


Fig. 15: The Fundamental pipeline of Visual SLAM is composed of multiple geometric vision tasks including depth estimation, optical flow and pose estimation. Those tasks have well known solutions based on CNNs in their individual domain. In contrast, the overall Visual-SLAM is not dominated by Deep Learning. [87]

dynamic objects. Table II shows an extract of the results from our work in [99], in comparison to other mono-depth approaches.

Thus we demonstrate the strong link between recognition and reconstruction. This idea is not particularly new. There was research investigating the potential of joint semantic labelling and depth as early as 2010 [107] (building upon even earlier work in geometric/semantic consistency in images [108]). However, it is fair to say that with the advent of neural networks in the last few years, the true potential of this research is beginning to come to fruition.

B. *Relocalization and Recognition*

Relocalization is the process of a vehicle recognising a previously learned position or path, as discussed. However, in the real automotive world, many things can disturb this. For example, the scene can change due to movable objects - for example, parked vehicles can move between the time the scene is learned and when relocalization is requested. In such a case, semantic segmentation approaches (e.g. [109], [18]), can be used to identify objects that may potentially move (vehicles, bicycles, pedestrians), and remove mapped features associated with such objects that may disturb the ability to relocalize. Further opportunities exist for the support of traditional Visual-SLAM pipelines with deep learning techniques (Figure 15), as described in detail in [87].

Place recognition in Visual-SLAM has a couple of applications. Firstly, it allows loop closure to correct for accumulated drift, and secondly it allows for building and maintaining maps from multiple passes through the same scene. Classical approaches using Bag of Words (e.g. [110]) proved reasonably successful, if perhaps lacking in terms of robustness. CNN-based approaches are proving to be more robust, with appearance-invariant approaches showing promising, if initial, results [111]. The recognition of places when significant time has passed is an important topic. Table III shows a small set of results for a Visual-SLAM pipeline , and demonstrates that errors increase significantly with a six month time difference between training and relocalization.

Finally, view invariant localization can be considered. This is important when the camera viewpoint at the relocalization time is significantly different to the camera viewpoint at training, for example due to a rotation of the vehicle caused by approaching the trained trajectory at a large angle. Traditional Visual-SLAM methods based on feature descriptors fail, as the same surfaces of the landmarks may not even be visible. It has been shown that attaching semantic labels to scene land-marks (via bounding box classification) can significantly improve the performance of viewpoint invariance [112].

C. *Relocalization and Reconstruction*

This is perhaps the most straightforward synergy to discuss. Relocalization, and Visual-SLAM in general, can be considered as the storage of scene reconstruction (i.e. building a map) along with iterative refinement of said map through bundle adjustment (refer to Figure 11). In this way, reconstruction and visual odometry become a seed for the traditional Visual-SLAM approaches. There are direct methods that bypass this seeded approach, for example LSD-SLAM [43] (and it's Omnidirectional camera extension [86]), where photometric error is minimised as opposed to reprojection error. However, if one considers a time-slicing of a bundle-adjusted map, it can also be seen that Visual-SLAM can be used to refine the reconstruction (both scene structure and visual odometry). In addition, it is well known moving objects (as distinct from *moveable* objects discussed in the previous section) can cause significant degradation in the performance of any Visual-SLAM pipeline [113]. Dynamic object detection (e.g. [80], [42], [52]) can therefore be used as an input into a Visual-SLAM pipeline to suppress outliers caused by said moving objects.

D. *Synergies in next generation*

Table IV compares the current 4R architecture with previous and next generation architectures. Previous generation has simplistic features due to limited compute availability. For Recognition, it had only pedestrian and park-slot detection using classical machine learning. Reconstruction was performed using sparse optical flow in software without any hardware accelerators. There was no image level reorganization or relocalization. CNN models have progressed rapidly to provide state-of-the-art results for geometric tasks like reconstruction [19] and relocalization [115]. For the next generation, a unified CNN model with high synergies would be the likely path. We have recently published an initial prototype Omnidet [114] showing joint modelling of reconstruction and recognition. Figure 16 illustrates its high level architecture with cross links shown across the different tasks. It should be noted that some prominent scholars argue that improved performance is obtained by designing principled algorithms for the geometric estimations, and using deep neural networks for the extraction of robust visual features [116], where they argue that taking this approach results in a system capable of redeployment into new scenes without fine-tuning or retraining. That said, in future work, we plan to explore inclusion of relocalization and reorganization in neural network frameworks [117].

Scene		Difference		Average Offset	
Training Date	**Replay Date**	**Time (days)**	**Distance (m)**	**Position (m)**	**Angle (degrees)**
20161208	20161208	0.003	4.723	0.468	4.704
20161208	20161208	0.005	2.483	0.355	5.366
20161208	20161208	0.006	2.692	0.3	5.149
20161208	20170607	181.156	2.49	1.085	8.162
20161208	20170607	181.155	0.066	0.903	9.498
20161208	20170607	181.154	4.96	0.896	10.751

TABLE III: Quantitative results of our relocalization algorithm on selected WoodScape [22] dataset scenes. The time difference is the number of days between training and relocalization. The distance is the starting distance from the trained trajectory. The average offset is given in terms of position and angle.

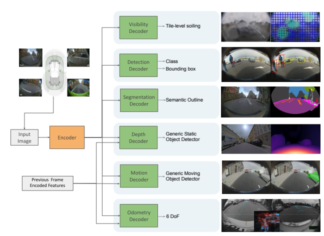
Module	Previous Gen	Current Gen 4R framework	Next Gen (Unified CNN [114])
Recognition	Pedestrian (PD) Park-slot detection (PSD)	Bounding Box - PD, Cyclist, Vehicles Segmentation - Road, curb, road markings	
Reconstruction	Sparse 3D Reconstruction Sparse Flow Visual Odometry(VO)	Dense Depth Dense Flow clustering Multicam VO	
Reorganization	-	Static Obj Fusion - Freespace, Curb Dynamic Obj Fusion - Vehicles, PD, Cyclists Lane Handler - Multicamera 3D lane fit	
Relocalization	-	Sparse feature geometric map	

TABLE IV: Features provided by current generation 4R architecture and comparison with previous and next generation.

E. Dual-sources of detection

We have thus far discussed possible synergies between Reconstruction, Recognition and Relocalization. There is, however, another overarching synergistic consideration: that of redundancy. In automated vehicles, redundancy plays a significant role in the safety of the application. When a system component fails, then another has to be available to ensure that the vehicle remains in a safe state. For example, FuseModNet [118] illustrates a synergistic fusion of cameras which provide dense information and lidar which performs well at low light. In terms of sensing, this would traditionally be achieved by the use of multiple sensor types, such as computer vision systems, radar and laser scanner (Figure 17). In particular for near-field sensing, an array of ultrasonic sensors is a mature low cost sensor which provides robust safety around the vehicle [119].

It is our contention that added safety is achieved through the parallel usage of different computer vision algorithm types. That is, a computer vision system architecture can be configured to maximise redundancy. This is particularly true as the sources are completely different types of processing – for example, statistical processing from the recognition pipeline and geometric processing from the reconstruction pipeline (Figure 8). In addition, such processing will typically run on different silicon components within an SoC. There will generally be a dedicated hardware accelerator for CNN-based processing, whereas geometric computer vision may run on a combination of dedicated hardware accelerators and general purpose processing.

However, one must be aware that if you maximise the other synergies, the potential for redundancy is reduced. For example, if you use a CNN-based depth as a seed for a Visual-SLAM algorithm, you cannot claim the CNN as a redundancy for the Visual-SLAM, as Visual-SLAM is now dependent on the CNN processing. That is, they are no longer parallel processing elements, but are rather sequential. One must also be aware that the two processing elements will likely use the same video feed – and so the safety of the camera itself and associated hardware/software, may also be a limiting factor. However, one should consider the potential for added safety in a system design following the 4R principles.

VI. Conclusions

In this paper, we provided a high level overview of a visual perception architecture on surround view cameras targeting commercial grade automated driving systems. We structure our architecture into modular components namely Recognition, Reconstruction, Relocalization and Reorganization, jointly called 4R architecture, and argue that staging the computer vision pipeline can lead to system efficiencies. We discussed each component in detail and then we discuss how they are synergized to provide a more accurate system. We also provide a system and application context helping understand an industrial system.

The first three of the 4Rs (Recognition, Reconstruction, Relocalization) provide the means for the detection of objects and the extraction of their geometry and location in reference to the autonomous vehicle. However, that is not a complete description of the scene, and the fourth R (Reorganisation) provides a higher level scene understanding that can include the contextual spatial and temporal relationships between objects in the scene and the autonomous vehicle. Though massive advances have been made in the last decade in computer vision, we cannot yet claim to have achieved this complete scene understanding. It is likely that full vehicle autonomy will not be feasible until we have such a high level of visual reasoning deployed on vehicles. However, we propose that the 4R architecture can encapsulate, and provide a framework for, this level of vehicular cognition.

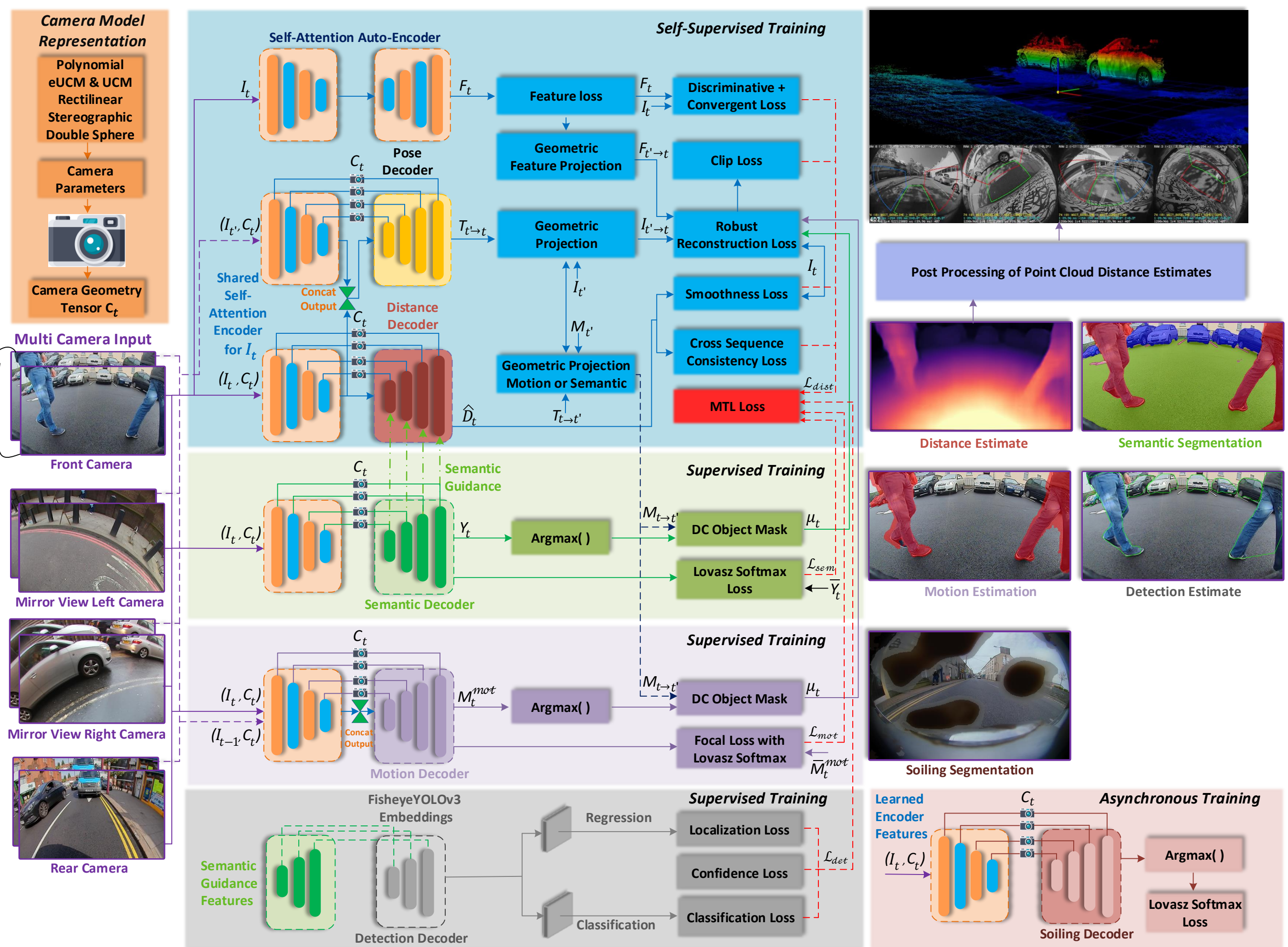


Fig. 16: Overview of our next generation unified multi-task visual perception framework. Refer to our OmniDet paper [114] for more details.

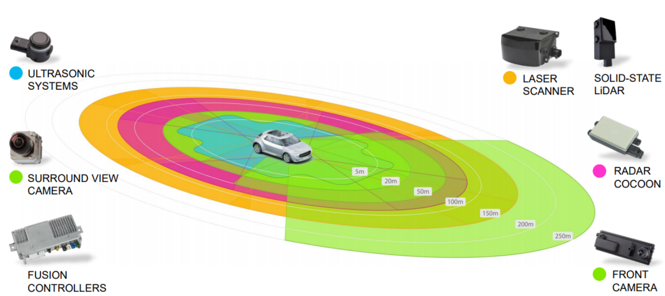


Fig. 17: Perception cocoon of redundant vehicle sensors. Safety is achieved through at least one sensor being available if others become unavailable.

ACKNOWLEDGMENT

We would like to thank our employer Valeo for encouraging advanced research. Many thanks to Edward Jones (NUI Galway) and Matthieu Cord (Sorbonne University and Valeo.ai) for providing a detailed review prior to submission. We would also like to thank our colleagues Fabian Burger, Nagarajan Balmukundan, Pantelis Ermilios and Nivedita Tripathi for supporting the paper.

REFERENCES

[1] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE Access*, vol. 8, pp. 58 443–58 469, 2020.

[2] K. Stricker, T. Wendt, W. Stark, M. Gottfredson, R. Tsang, and M. Schallehn, "Electric and autonomous vehicles: The future is now," *Bain & Company Brief*, 2020, (accessed 30/03/2021). [Online]. Available: https://www.bain.com/insights/electric-and-autonomous-vehicles-the-future-is-now/

[3] Ernst & Young, "Deploying autonomous vehicles: Commercial considerations and urban mobility scenarios," 2014.

[4] OECD International Transport Forum, "Automated and autonomous driving: Regulation under uncertainty," International Transport Forum Policy Papers, Tech. Rep. 7, 2015.

[5] CB Insights, "40+ corporations working on autonomous vehicles," Available at https://www.cbinsights.com/research/autonomous-driverless-vehicles-corporations-list/ (accessed 13/07/2020), 2020.

[6] Y. Ro and Y. Ha, "A factor analysis of consumer expectations for autonomous cars," *Journal of Computer Information Systems*, vol. 59, no. 1, pp. 52–60, 2019.

[7] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. Paixão, F. Mutz, L. Veronese, T. Oliveira-Santos, and A. F. D. Souza, "Self-driving cars: A survey," *Expert Systems with Applications*, vol. 165, p. 113816, 2021.

[8] M. Daily, S. Medasani, R. Behringer, and M. Trivedi, "Self-driving cars," *Computer*, pp. 18–23, 2017.

[9] J. Malik, P. Arbeláez, J. Carreira, K. Fragkiadaki, R. Girshick, G. Gkioxari, S. Gupta, B. Hariharan, A. Kar, and S. Tulsiani, "The three R's of computer vision: Recognition, reconstruction and reorganization," *Pattern Recognitions Letters*, vol. 72, pp. 4–14, 2016.

[10] M. Zollhöfer, P. Stotko, A. Görlitz, C. Theobalt, M. Nießner, R. Klein, and A. Kolb, "State of the art on 3D reconstruction with RGB-D cameras," *Computer Graphics Forum*, vol. 37, no. 2, pp. 625–652, 2018.

[11] M. Heimberger, J. Horgan, C. Hughes, J. McDonald, and S. Yogamani, "Computer vision in automated parking systems: Design, implementation and challenges," *Image and Vision Computing*, vol. 68, pp. 88–101, 2017.

[12] F. Li, P. Bonnifait, and J. Ibanez-Guzman, "Estimating localization uncertainty using multi-hypothesis map-matching on high-definition road maps," in *Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2017.

[13] J. Martinez-Carranza, A. Calway, and W. Mayol-Cuevas, "Enhancing 6D visual relocalisation with depth cameras," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[14] F. Gustafsson, "Automotive safety systems: Replacing costly sensors with software algorithms," *IEEE Signal Processing Magazine*, vol. 26, no. 4, pp. 32–47, 2009.

[15] D. Alvarez-Coello, B. Klotz, D. Wilms, S. Fejji, J. M. Gómez, and R. Troncy, "Modeling dangerous driving events based on in-vehicle data using random forest and recurrent neural network," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, 2019.

[16] G. Borghi, R. Gasparini, R. Vezzani, and R. Cucchiara, "Embedded recurrent network for head pose estimation in car," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, 2017.

[17] D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. San Francisco: WH Freeman, 1982.

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

[19] S. Qiao, Y. Zhu, H. Adam, A. Yuille, and L. Chen, "ViP-DeepLab: Learning visual perception with depth-aware video panoptic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[20] V. R. Kumar, S. Milz, C. Witt, M. Simon, K. Amende, J. Petzold, S. Yogamani, and T. Pech, "Near-field depth estimation using monocular fisheye camera: A semi-supervised learning approach using sparse LiDAR data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018.

[21] R. Li, S. Wang, and D. Gu, "DeepSLAM: a robust monocular SLAM system with unsupervised deep learning," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 4, pp. 3577–3587, 2021.

[22] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, P. Varley, D. O'Dea, M. Uřičář, S. Milz, M. Simon, K. Amende *et al.*, "WoodScape: A multi-task, multi-camera fisheye dataset for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9308–9318.

[23] M. A. Azzopardi, I. Grech, and J. Leconte, "A high speed tri-vision system for automotive applications," *European Transport Research Review*, vol. 2, pp. 31–51, 2010.

[24] P. Walzer and H.-W. Grove, "IRVW Futura - The Volkswagen Research Car," in *Passenger Car Conference and Exposition*. SAE International, 1990.

[25] T. Wu, P. Tsai, N. Hu, and J. Chen, "Research and implementation of auto parking system based on ultrasonic sensors," in *Proceedings of the International Conference on Advanced Materials for Science and Engineering (AMSE)*, 2016, pp. 643–645.

[26] Y. Song and C. Liao, "Analysis and review of state-of-the-art automatic parking assist system," in *Proceedings of the IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, 2016.

[27] C. Wang, H. Zhang, M. Yang, X. Wang, L. Ye, and C. Guo, "Automatic parking based on a bird's eye view vision system," *Advances in Mechanical Engineering*, vol. 6, p. 847406, 2014.

[28] U. Schwesinger, M. Bürki, J. Timpner, S. Rottmann, L. Wolf, L. M. Paz, H. Grimmett, I. Posner, P. Newman, C. Häne, L. Heng, G. H. Lee, T. Sattler, M. Pollefeys, M. Allodi, F. Valenti, K. Mimura, B. Goebelsmann, W. Derendarz, P. Mühlfellner, S. Wonneberger, R. Waldmann, S. Grysczyk, C. Last, S. Brüning, S. Horstmann, M. Bartholomäus, C. Brummer, M. Stellmacher, F. Pucks, M. Nicklas, and R. Siegwart, "Automated valet parking and charging for e-mobility," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, 2016, pp. 157–164.

[29] S. Ma, W. Fang, H. Jiang, M. Han, and C. Li, "Parking space recognition method based on parking space feature construction in the scene of autonomous valet parking," *Applied Sciences*, vol. 11, no. 6, 2021.

[30] I. Isaksson-Hellman and M. Lindman, "The effect of a low-speed automatic brake system estimated from real life data," *Annals of Advances in Automotive Medicine*, vol. 56, pp. 231–240, 2012.

[31] S. Lüke, O. Fochler, T. Schaller, and U. Regensburger, *Traffic-Jam Assistance and Automation*. Springer International Publishing, 2014, pp. 1–13.

[32] L. Ulrich, "Bmw's autonomous "extended traffic jam assistant" needs supervision," *IEEE Spectrum*, 2019.

[33] S. J. Rao and G. J. Forkenbrock, "Test procedures traffic jam assist test development considerations," National Highway Traffic Safety Administration, Tech. Rep. DOT HS 812 757, July 2019.

[34] T. Nothdurft, P. Hecker, S. Ohl, F. Saust, M. Maurer, A. Reschka, and J. R. Böhmer, "Stadtpilot: First fully autonomous test drives in urban traffic," in *Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2011, pp. 919–924.

[35] C. R. Sunstein, "Rear visibility and some unresolved problems for economic analysis (with notes on experience goods)," Harvard Law School, Harvard University, Tech. Rep., July 2019. [Online]. Available: https://dx.doi.org/10.2139/ssrn.3411057

[36] C. Hughes, M. Glavin, E. Jones, and P. Denny, "Wide-angle camera technology for automotive applications: a review," *IET Transactions on Intelligent Transport Systems*, vol. 3, pp. 19–31, 2009.

[37] J. B. Cicchino, "Real-world effects of rear automatic braking and other backing assistance systems," *Journal of Safety Research*, vol. 68, pp. 41–47, 2019.

[38] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A deep neural network architecture for real-time semantic segmentation," 2016. [Online]. Available: https://arxiv.org/pdf/1606.02147.pdf

[39] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 6, pp. 1137–1149, 2017.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[41] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015, pp. 2650–2658.

[42] M. Siam, H. Mahgoub, M. Zahran, S. Yogamani, M. Jagersand, and A. El-Sallab, "MODNet: Motion and appearance based moving object detection network for autonomous driving," in *Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2859–2864.

[43] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014, pp. 834–849.

[44] S. Wang, R. Clark, H. Wen, and N. Trigoni, "Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[45] S. Mittal, "A survey on optimized implementation of deep learning models on the nvidia jetson platform," *Journal of Systems Architecture*, vol. 97, pp. 428–442, 2019.

[46] L. Yahiaoui, J. Horgan, B. Deegan, S. Yogamani, C. Hughes, and P. Denny, "Overview and empirical analysis of isp parameter tuning for visual perception in autonomous driving," *Journal of Imaging*, vol. 5, no. 10, 2019.

[47] A. Mosleh, A. Sharma, E. Onzon, F. Mannan, N. Robidoux, and F. Heide, "Hardware-in-the-loop end-to-end optimization of camera image processing pipelines," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 7529–7538.

[48] M. Buckler, S. Jayasuriya, and A. Sampson, "Reconfiguring the imaging pipeline for computer vision," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[49] R. Szeliski, *Computer Vision: Algorithms and Applications, 1st ed.* Springer, 2010.

[50] R. Szeliski, "A multi-view approach to motion and stereo," Microsoft Research, Tech. Rep. MSR-TR-99-19, 1999.

[51] M. R. U. Saputra, A. Markham, and N. Trigoni, "Visual SLAM and structure from motion in dynamic environments: A survey," *ACM Computing Surveys*, vol. 51, no. 2, 2018.

[52] M. Yahiaoui, H. Rashed, L. Mariotti, G. Sistu, I. Clancy, L. Yahiaoui, and S. Yogamani, “FisheyeMODNet: Moving object detection on surround-view cameras for autonomous driving,” in *Proceedings of the Irish Machine Vision and Image Processing (IMVIP)*, 2019.

[53] G. Sistu, I. Leang, S. Chennupati, S. Yogamani, C. Hughes, S. Milz, and S. Rawashdeh, “NeurAll: Towards a unified visual perception model for automated driving,” in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 796–803.

[54] S. Chennupati, G. Sistu, S. Yogamani, and S A Rawashdeh, “Multinet++: Multi-stream feature aggregation and geometric loss strategy for multi-task learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019.

[55] M. Uřičář., D. Hurych., P. Křížek, and S. Yogamani., “Challenges in designing datasets and validation for autonomous driving,” in *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, 2019, pp. 653–659.

[56] A. Briot, P. Viswanath, and S. Yogamani, “Analysis of efficient cnn design techniques for semantic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018, pp. 663–672.

[57] A. Das., S. Kandan., S. Yogamani., and P. Křížek, “Design of real-time semantic segmentation decoder for automated driving,” in *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, 2019, pp. 393–400.

[58] V. Ravi Kumar, S. Yogamani, M. Bach, C. Witt, S. Milz, and P. Mader, “UnRectDepthNet: Self-supervised monocular depth estimation using a generic framework for handling common camera distortion models,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[59] H. Rashed, E. Mohamed, G. Sistu, V. R. Kumar, C. Eising, A. El-Sallab, and S. Yogamani, “Generalized object detection on fisheye cameras for autonomous driving: Dataset, representations and baseline,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 2272–2280.

[60] M. Ramzy, H. Rashed, A. E. Sallab, and S. Yogamani, “Rst-modnet: Real-time spatio-temporal moving object detection for autonomous driving,” *NeurIPS 2019 Workshop on Machine Learning for Autonomous Driving*, 2019.

[61] M. Hassaballah and A. I. Awad, *Deep learning in computer vision: principles and applications*. CRC Press, 2020.

[62] P. Maddu, W. Doherty, G. Sistu, I. Leang, M. Uřičář, S. Chennupati, H. Rashed, J. Horgan, C. Hughes, and S. Yogamani, “Fisheyemultinet: Real-time multi-task learning architecture for surround-view automated parking system.” in *Proceedings of the Irish Machine Vision and Image Processing Conference (IMVIP)*, 2019.

[63] I. Leang, G. Sistu, F. Bürger, A. Bursuc, and S. Yogamani, “Dynamic task weighting methods for multi-task networks in autonomous driving systems,” in *Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2020.

[64] S. Chennupati, G. Sistu., S. Yogamani., and S. Rawashdeh., “Auxnet: Auxiliary tasks enhanced semantic segmentation for automated driving,” in *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, 2019, pp. 645–652.

[65] M. Uřičář, P. Křížek, G. Sistu, and S. Yogamani, “Soilingnet: Soiling detection on automotive surround-view cameras,” in *Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 67–72.

[66] A. Das, P. Křížek, G. Sistu, F. Bürger, S. Madasamy, M. Uřičář, V. R. Kumar, and S. Yogamani, “TiledSoilingNet: Tile-level soiling detection on automotive surround-view cameras using coverage metric,” in *Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2020.

[67] M. Uřičář, G. Sistu, H. Rashed, A. Vobecky, V. R. Kumar, P. Křížek, F. Burger, and S. Yogamani, “Let’s get dirty: Gan based data augmentation for camera lens soiling detection in autonomous driving,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 766–775.

[68] M. Uřičář, J. Ulicny, G. Sistu, H. Rashed, P. Křížek, D. Hurych, A. Vobecky, and S. Yogamani, “Desoiling dataset: Restoring soiled areas on automotive fisheye cameras,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2019.

[69] Z. Kukelova and V. Larsson, “Radial distortion triangulation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9681–9686.

[70] A. Saxena, S. H. Chung, and A. Y. Ng, “3-D depth reconstruction from a single still image,” *International Journal of Computer Vision*, vol. 76, pp. 53—-69, 2007.

[71] D. Marr and T. Poggio, “A computational theory of human stereo vision,” *Proceedings of the Royal Society of London B*, vol. 292, pp. 301–328, 1979.

[72] W. E. L. Grimson, “A computer implementation of a theory of human stereo vision,” *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, vol. 292, no. 1058, pp. 217–253, 1981.

[73] E. Davies, “The three-dimensional world,” in *Machine Vision (Third Edition) – Theory, Algorithms, Practicalities*. Springer, 2005, ch. 16, pp. 445–485.

[74] Z. Kukelova, J. Heller, M. Bujnak, and T. Pajdla, “Radial distortion homography,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 639–647.

[75] A. Hirata, R. Ishikawa, M. Roxas, and T. Oishi, “Real-time dense depth estimation using semantically-guided lidar data propagation and motion stereo,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3806–3811, 2019.

[76] V. R. Kumar, S. Milz, C. Witt, M. Simon, K. Amende, J. Petzold, S. Yogamani, and T. Pech, “Monocular fisheye camera depth estimation using sparse lidar supervision,” in *Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2853–2858.

[77] T. van Dijk and G. de Croon, “How do neural networks see depth in single images?” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[78] D. Barath and J. Matas, “Multi-class model fitting by energy minimization and mode-seeking,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[79] J. Klappstein, F. Stein and U. Franke, “Detectability of moving objects using correspondences over two and three frames,” in *Proceedings of the Joint Pattern Recognition Symposium*, 2007, pp. 112–121.

[80] L. Mariotti and C. Eising, “Spherical formulation of geometric motion segmentation constraints in fisheye cameras,” *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[81] V. R. Kumar, S. A. Hiremath, M. Bach, S. Milz, C. Witt, C. Pinard, S. Yogamani, and P. Mäder, “FisheyeDistanceNet: Self-supervised scale-aware distance estimation using monocular fisheye camera for autonomous driving,” in *Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA)*, 2020, pp. 574–581.

[82] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, “Monoslam: Real-time single camera slam,” *Transaction Pattern Analysis Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[83] G. Klein and D. Murray, “Parallel tracking and mapping for small AR workspaces,” in *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2007.

[84] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “ORB-SLAM: a versatile and accurate monocular SLAM system,” *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[85] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, “DTAM: Dense tracking and mapping in real-time,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2011, pp. 2320–2327.

[86] D. Caruso, J. Engel, and D. Cremers, “Large-scale direct slam for omnidirectional cameras,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 141–148.

[87] S. Milz, G. Arbeiter, C. Witt, B. Abdallah, and S. Yogamani, “Visual slam for automated driving: Exploring the applications of deep learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018.

[88] TomTom N.V., “TomTomRoadDNA,” https://www.tomtom.com/products/hd-map/, [Online: 12/2020].

[89] B. Ravi Kiran, L. Roldao, B. Irastorza, R. Verastegui, S. Suss, S. Yogamani, V. Talpaert, A. Lepoutre, and G. Trehard, “Real-time dynamic object detection for autonomous driving using prior 3d-maps,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[90] Q. Luo, Y. Cao, J. Liu, and A. Benslimane, “Localization and navigation in autonomous driving: Threats and countermeasures,” *IEEE Wireless Communications*, vol. 26, no. 4, pp. 38–45, 2019.

[91] A. Singandhupe and H. M. La, "A review of slam techniques and security in autonomous driving," in *Proceedings of the IEEE International Conference on Robotic Computing (IRC)*, 2019, pp. 602–607.

[92] A. Kasyanov, F. Engelmann, J. Stückler, and B. Leibe, "Keyframe-based visual-inertial online slam with relocalization," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[93] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 1999, pp. 1150–1157.

[94] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2011, pp. 2564–2571.

[95] P. F. Alcantarilla, J. Nuevo, and A. Bartoli, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 34, no. 7, p. 1281–1298, 2011.

[96] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia, and A. Yuille, "Every Pixel Counts ++: Joint Learning of Geometry and Motion with 3D Holistic Understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[97] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 3828–3838.

[98] V. Guizilini, R. Ambrus, S. Pillai, and A. Gaidon, "3D packing for self-supervised monocular depth estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2485–2494.

[99] V. R. Kumar, M. Klingner, S. Yogamani, S. Milz, T. Fingscheidt, and P. Mader, "Syndistnet: Self-supervised monocular fisheye camera distance estimation synergized with semantic segmentation for autonomous driving," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 61–71.

[100] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[101] D. Eigen, C. Puhrsch, and R. Fergus, "Depth Map Prediction from a Single Image Using a Multi-Scale Deep Network," in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2014, pp. 2366–2374.

[102] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[103] V. P. Feihu Zhang, R. Yang, and P. H. S. Torr, "GA-Net: Guided aggregation net for end-to-end stereo matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[104] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[105] M. Tatarchenko, S. R. Richter, R. Ranftl, Z. Li, V. Koltun, and T. Brox, "What do single-view 3d reconstruction networks learn?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[106] X. Lin, D. Sánchez-Escobedo, J. R. Casas, and M. Pardás, "Depth estimation and semantic segmentation from a single RGB image using a hybrid convolutional neural network," *Sensors*, vol. 19, 2019.

[107] B. Liu, S. Gould, and D. Koller, "Single image depth estimation from predicted semantic labels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[108] S. Gould, R. Fulton, and D. Koller, "Decomposing a scene into geometric and semantically consistent regions," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2009.

[109] M. Siam, S. Elkerdawy, M. Jagersand, and S. Yogamani, "Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges," in *Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–8.

[110] D. Galvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[111] S. Garg, N. Suenderhauf, and M. Milford, "Lost? appearance-invariant place recognition for opposite viewpoints using visual semantics," in *Proceedings of Robotics: Science and Systems (RSS)*, 2018.

[112] J. Li, D. Meger, and G. Dudek, "Semantic mapping for view-invariant relocalization," in *Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA)*, 2019.

[113] S. Wangsiripitak and D. W. Murray, "Avoiding moving outliers in visual SLAM by tracking moving objects," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2009.

[114] V. Ravikumar, S. Yogamani, H. Rashed, G. Sistu, C. Witt, I. Leang, S. Milz, and P. Mader, "Omnidet: Surround view cameras based multi-task visual perception network for autonomous driving," *IEEE Robotics and Automation Letters*, 2021.

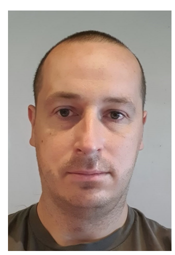
[115] G. L. Oliveira, N. Radwan, W. Burgard, and T. Brox, "Topometric localization with deep learning," in *Robotics Research*, N. M. Amato, G. Hager, S. Thomas, and M. Torres-Torriti, Eds. Cham: Springer International Publishing, 2020, pp. 505–520.

[116] P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, and T. Sattler, "Back to the feature: Learning robust camera localization from pixels to pose," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[117] Y. Cui, R. Chen, W. Chu, L. Chen, D. Tian, Y. Li, and D. Cao, "Deep learning for image and point cloud fusion in autonomous driving: A review," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–18, 2021.

[118] H. Rashed, M. Ramzy, V. Vaquero, A. El Sallab, G. Sistu, and S. Yogamani, "Fusemodnet: Real-time camera and lidar based moving object detection for robust low-light autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2019, pp. 0–0.

[119] M. Pöpperli, R. Gulagundi, S. Yogamani, and S. Milz, "Capsule neural network based height classification using low-cost automotive ultrasonic sensors," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 661–666.



Ciarán Eising obtained his PhD in Electronic and Computer Engineering at the National University of Ireland in 2010. From 2009 to 2020, Ciarán has worked as a computer vision team lead and architect in Valeo Vision Systems, where he also held the title of Senior Expert. In 2016, he was awarded the position of Adjunct Lecturer in the National University of Ireland, Galway. In 2020, Ciarán joined the University of Limerick as a lecturer in Computer Engineering.

Jonathan Horgan is a Computer Vision and Deep Learning Architecture Manager and Senior Expert at Valeo Vision Systems. He has worked in the field of computer vision for over 16 years with a focus over the last 10 years on automotive computer vision for Advanced Driver Assistance Systems (ADAS) and automated parking and 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 25 publications in peer reviewed conferences and journals and over 100 patents published in the field of computer vision for ADAS and Autonomous driving.

Senthil Yogamani is an Artificial Intelligence architect and holds a director level technical leader position at Valeo Ireland. He leads the research and design of AI algorithms for various modules of autonomous driving systems. He has over 14 years of experience in computer vision and machine learning including 12 years of experience in industrial automotive systems. He is an author of 100+ publications with 1500+ citations and 100+ inventions with 70 filed patent families. He serves on the editorial board of various leading IEEE automotive conferences including ITSC and IV 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.