Visual SLAM for Automated Driving:
Exploring the Applications of Deep Learning

Stefan Milz, Georg Arbeiter, Christian Witt
Valeo Schalter und Sensoren GmbH
stefan.milz, georg.arbeiter, christian.witt@valeo.com

Bassam Abdallah
Valeo Vision, Bobigny
bassam.abdallah.ext@valeo.com

Senthil Yogamani
Valeo Vision Systems, Ireland
senthil.yogamani@valeo.com

Abstract

Deep learning has become the standard model for object detection and recognition. Recently, there is progress on using CNN models for geometric vision tasks like depth estimation, optical flow prediction or motion segmentation. However, Visual SLAM remains to be one of the areas of automated driving where CNNs are not mature for deployment in commercial automated driving systems. In this paper, we explore how deep learning can be used to replace parts of the classical Visual SLAM pipeline. Firstly, we describe the building blocks of Visual SLAM pipeline composed of standard geometric vision tasks. Then we provide an overview of Visual SLAM use cases for automated driving based on the authors' experience in commercial deployment. Finally, we discuss the opportunities of using Deep Learning to improve upon state-of-the-art classical methods.

1. Introduction

Automated driving is a rapidly advancing application area with a complex structure (see Fig.1) and lots of progress in Deep Learning. There are two main paradigms in this area:

1. The mediated perception approach which semantically reasons the scene [26, 55] and determines the driving decision based on it.
2. The behavior reflex approach that learns the driving decision end-to-end [5, 66].

The behavior reflex methods can benefit from semantic reasoning of the environment. For example, an auxiliary loss on semantic segmentation [66] was used with end-to-end learning. On the other hand, semantic reasoning is a central task in mediated perception, followed by the control decision separately. Semantic reasoning of the scene includes self-localization, object detection, motion detection, depth estimation, object tracking and others. CNNs (Convolutional Neural Networks) have demonstrated remarkable leaps for various computer vision tasks especially for object recognition. They are computationally intensive and the main challenge is to design efficient regression losses. In contrast Visual-SLAM approaches based on CNN with state-of-the-art results are rare.

Since, the rise of the key-frame based SLAM [16], the standard pipeline of feature-based Visual SLAM mainly consists of the classical steps of a structure from motion (SfM) algorithm [30]. In contrast, more recent approaches like [20] consider the image directly. However, classical approaches for monocular Visual SLAM share a major limitation in map robustness. Indeed, scene changes or varying illumination make the map less efficient if not obsolete for reutilization. In [38] the authors try to learn an illumination-robust feature for place recognition, but it is still limited to some extent and does not face the scene change issue.

The map retraining is a long term subject in the community. Starting with the early approach of [3],[35] builds a schedule to update the map when several sessions are attempted. More recently [13] and [45] proposed two concurrent and promising approaches. In order to compare these methods, the community lacks of a public dataset dedicated to this topic again which authors could compete.

On the hardware side, very few Visual SLAM algorithms in literature [46, 20] are suitable for low computational power constraints of current automotive systems. In contrast, industrial systems such as [44] rely on server-client architecture to carry the heavy computations.

Section 2 provides an overview of existing Visual SLAM approaches. Section 3 discusses the use cases of Visual SLAM in automated driving and the challenges faced by classical approaches. In Section 4 opportunities are pre-

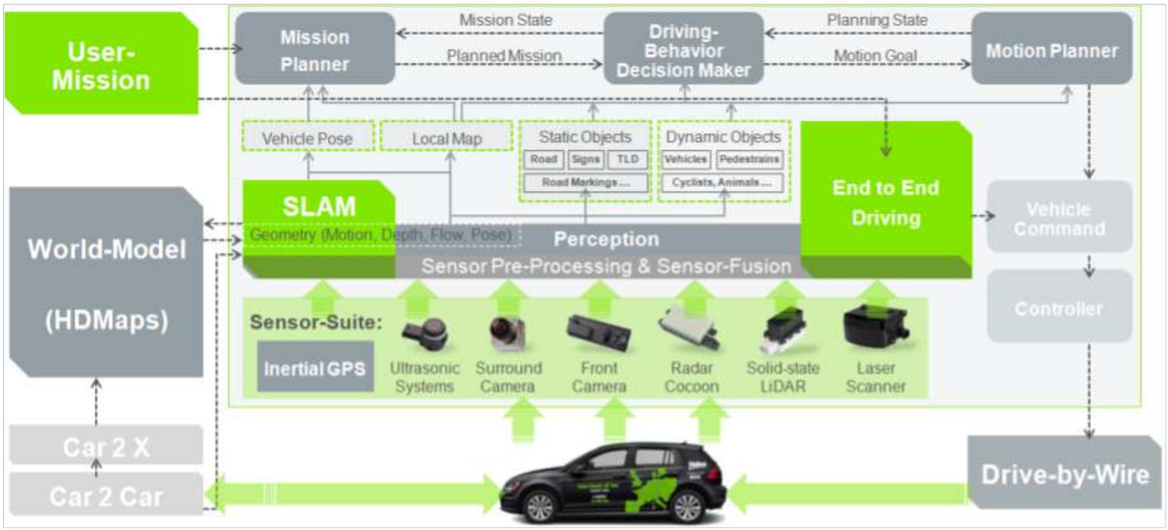

Figure 1: Visual SLAM is inevitable within the complex structure of automated driving. This figure shows how local map generation or vehicle pose estimation are essential for solving tasks within the perception based pipeline of automated driving. SLAM must be used for environmental modeling. In general, SLAM could be done with different sensor types. However, we focus on Visual SLAM, because its able to deal effectively with dense data, the cameras are widely distributed and they have a large field of view with an acceptable range. Compared to all perception algorithms, where best performing methods use CNNs, current state of the art Visual SLAM algorithms are not based on deep learning. We figure out CNN opportunities especially for geometric tasks.

sented where parts of the fundamental pipeline can be replaced using CNN based approaches. Section 5 provides an overview of CNN based pipelines. Finally, Section 6 concludes the paper and provides potential future directions.

2. Visual SLAM approaches

The term Visual SLAM comprises all SLAM approaches that take image-like data as input. Therefore, the main difference to SLAM systems based on other sensors is the need to generate depth information from consecutive camera frames (see Fig. 1).

There are two major state of the art methods, feature based and direct Visual SLAM. The first one relies on descriptive image features like SIFT or ORB whereas the second one uses the image pixels directly. Being different in the aspect of which image information is used, they share the same fundamental processing pipeline, though.

2.1. Fundamental Pipeline

The fundamental pipeline for Visual SLAM is composed of tracking, mapping, global optimization and relocalization.

Tracking between consecutive camera images is utilized in order to generate a local camera trajectory as well as depth information. Usually, this tasks ends up in a nonlinear optimization problem. In most approaches, so called key frames are used as a base for tracking. Once tracking indicates that there is not enough overlap between the current camera frame and the key frame, a new key frame is created.

Mapping is the process of generating a map out of the tracked sensor data. This step is where the main difference between feature based and direct methods is located. The first generates sparse feature maps whereas the second one provides (semi-)dense point maps as output. In some of the approaches, key frames including depth and scale information are stored in a graph with the edges representing the transformation between key frames.

A **Global Optimization** step is needed for correcting the global map as tracking introduces a drift error into the map. As it is computationally expensive, global optimization is usually done from time to time only. The global optimization step relies on recognizing a place that has been seen and mapped before and therefore detecting a loop closure. Based on this detection, all camera poses can be optimized. In some approaches, the 3D information is jointly optimized.

Relocalization is the procedure of placing the sensor at an unknown pose in the map and trying to estimate the pose. This is usually done by comparing the current sensor data with the map. A common approach is to use descriptive image features.

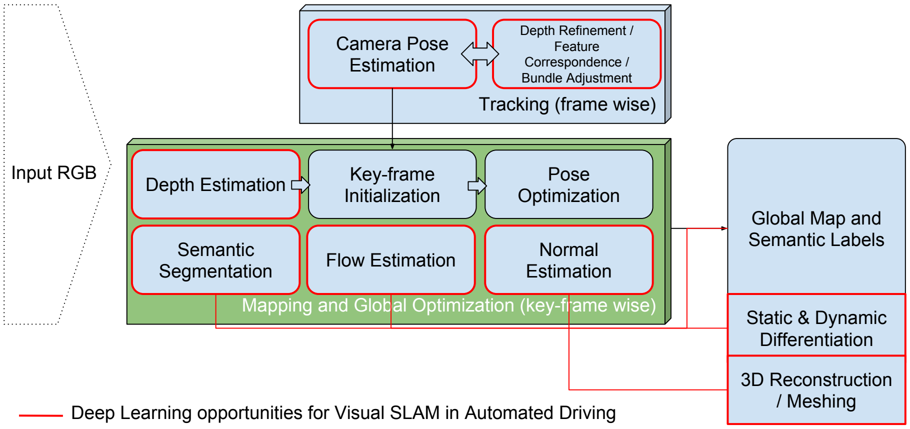
Input RGB
Camera Pose Estimation
Depth Refinement / Feature Correspondence / Bundle Adjustment
Tracking (frame wise)
Depth Estimation
Key-frame Initialization
Pose Optimization
Semantic Segmentation
Flow Estimation
Normal Estimation
Mapping and Global Optimization (key-frame wise)
Global Map and Semantic Labels
Static & Dynamic Differentiation
3D Reconstruction / Meshing
Deep Learning opportunities for Visual SLAM in Automated Driving

Figure 2: 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.

2.2. Feature based SLAM

Feature based Visual SLAM methods utilize descriptive image features for tracking and depth estimation. This results in sparse feature maps. Several approaches will be explained in the following sections.

MonoSLAM by Davison et al. [16] is the first Visual SLAM approach. It uses EKF-based feature tracking. There is no loop closure detection and in order to achieve real-time performance, only few feature points per frame a are considered.

PTAM: The Parallel Tracking and Mapping (PTAM) algorithm [34] extends the approach of Davison by parallelizing the feature point matching part in order to improve real-time performance. For optimization, it uses bundle adjustment (BA). Thus, it can handle many more feature points which increases robustness.

ORB-SLAM [46] extends the functionality of PTAM by adding loop closure detection and global pose graph optimization. It also relies on the ORB feature descriptor which is known to be robust while having low computational cost.

2.3. Direct SLAM

In contrast to feature based approaches, direct methods do not rely on features for tracking but on the whole image. This gives the chance to acquire a dense environment model.

DTAM: Dense Tracking and Mapping (DTAM) [47] is the first direct method published. While lacking features like loop closure detection or global optimization, it introduces tracking on key frames based on minimization of the photometric error

$$\mathbf{C}_r = \frac{1}{\|I(r)\|} \sum_{m \in I(r)} \|\mathbf{I}_r(\mathbf{u}) - \mathbf{I}_m(\mathbf{v})\|.$$

The mapping space is discretized into a 3D grid which limits the maximum size of the map. Real-time performance is achieved by performing computations on a GPU.

LSD-SLAM: Large-Scale Semi Dense SLAM (LSD-SLAM) [20] is also based on the minimization of the photometric error. It extends the functionality to large scale by building a pose frame graph and global optimization including loop closure detection. Computational efficiency is achieved by reducing the number of image pxiels used for tracking to those showing a high intensity gradient.

DSO: In the publication of Direct Sparse Odometry (DSO) [19], the authors extend the minimization model of LSD-SLAM by taking the geometric error into the account as well as exposure time and lens distortion

$$\mathbf{C}_r = \frac{1}{\|I(r)\|} \sum_{m \in I(r)} \|\mathbf{I}_r(\mathbf{u}) - b_r - \frac{t_r e^{a_r}}{t_m e^{a_m}} \mathbf{I}_m(\mathbf{v}) - b_m\|.$$

This leads to a more robust estimation of the trajectory. Although being a direct method, the map generated is sparse in order to achieve real-time performance. Loop closure detection and global optimization is not an explicit part of the approach but can be done in the same way as for LSD-SLAM.

2.4. Benchmarks on KITTI

Table 1 describes the RMSE (degree per 100 m) for rotational r_{rel} and RMSE (%) translational error r_{rel}. The results are taken from Wang et al. [64]. The data refers to the mean taken from all ten sequences (100 m to 800 m). For these automated driving scenarios, DSO yields the most promising results. However, Wang et al. [64] claimed a bigger error for the monocular implementation on what we focus. Fig. 3 shows the comparison for stereo versus monocular SLAM

Table 1: Qualitative results on the KITTI [27] [64].

	DSO		ORB2		LSD	
Sequence	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
00	0.84	0.26	**0.83**	0.29	1.09	0.42
01	1.43	0.09	**1.38**	0.20	2.13	0.37
02	**0.78**	**0.21**	0.81	0.28	1.09	0.37
10	**0.49**	**0.18**	0.58	0.28	0.75	0.34
mean	**0.84**	**0.20**	0.81	0.26	1.14	0.40

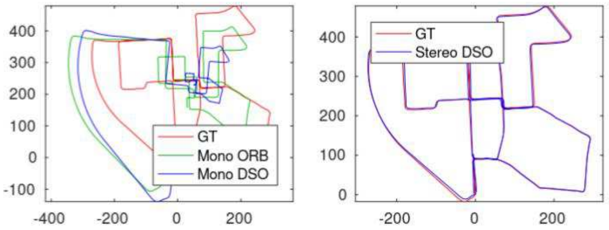

Figure 3: Weakness of traditional monocular Visual SLAM taken from [64]. The figure shows the qualitative results on KITTI [27] trace 00. The left outlines mono SLAM approaches, the right shows stereo DSO.

using DSO and ORB. Although the stereo results are acceptable, the monocular results are weak and unacceptable for automated driving. From that we derive lots of potential using deep learning techniques to improve.

3. Use Cases and Challenges

Use cases for Visual SLAM in automated driving are manifold. A reliable and fast mapping and localization of the car is needed for almost any driving scenario. Due to the high resolution of cameras compared to other sensors like RADAR or LIDAR, situations that require a detailed knowledge about the environment or generate ambiguous signals from other sensors are dedicated for the application of Visual SLAM. Thus, we identify the most relevant use cases to be parking, highway driving and urban driving.

3.1. Driving Scenarios

For certain driving scenarios, application of Visual SLAM is crucial. In the following section we describe parking, highway and urban driving in detail.

3.1.1 Parking

Main requirement of parking is the need of an accurate environment map in the near vicinity of the car while driving at low speed. Most frequent scenarios are parking on a parking deck, on a public parking lot and in the home zone. Whereas the first two require small scale mapping in an unknown environment, parking in an home zone is dedicated for the application of SLAM. First, the car learns a trajectory in the home zone in parallel to recording an initial map. Once, the car returns to the home zone, the map is loaded, relocalization takes place and the car can move on the learned trajectory while updating the map.

Regarding types of maps, both feature maps and dense point maps are suitable for this use case. Depending on the type of features used, feature maps might be more suitable for relocalization whereas dense point maps provide more information about the environment.

3.1.2 Highway Driving

The highway driving scenario is a limited, but important use case for Visual SLAM. Due to the higher speed, compared to parking or urban driving, it gets challenging to run Visual SLAM approaches in real-time, since a high frame rate from at least 30fps is needed. On the other hand, the environment geometry is less complex such as surrounding objects are parallel arranged. Artal et al. [46] (see section 2.2) have shown on the KITTI benchmark suite [27] their highest accuracy on stream four, which is a pure highway scene. The method achieves a RMSE of 1.79 m, which is far below the average over all scenes. It already achieved high accuracy of sparse SLAM techniques for highway driving. There is not much space for improvements using deep learning. However, due to the required high frame rate, an sparse CNN based SLAM technique might be able to outperform state of the art approaches in terms of efficiency.

3.1.3 Urban Driving

Automated driving within the inner city is extreme challenging. Compared to Highway Driving, the environment is much more complex and varying, compared to the parking scenario, the environment includes lots of dynamic objects that have to be detected actively or passively during 3D reconstruction and localization. In the last section, we described the high performance of ORB-SLAM [46] (sparse and direct) on a KITTI highway trace. In contrast, their results on urban scenarios are imprecise for large traces up to an RMSE of 46.36 m (trace 8). This gives a slight imagination how challenging it is and that we may need a dense reconstruction within such an use case. On the other hand, DSO-SLAM [19], a sparse direct method yields much higher performance than ORB-SLAM even on large urban dataset. Stereo-DSO is ranked on 14th for KITTI odometry challenge. Therefore, it ranks higher than the semi-dense direct LSD Stereo SLAM [20] (27th). Hence, not only the number of reconstructed points, even the ability to reconsider static points with stability against lots of dynamic ob-

[0] The ranking refers to the date of submission the 20th of march 2018.

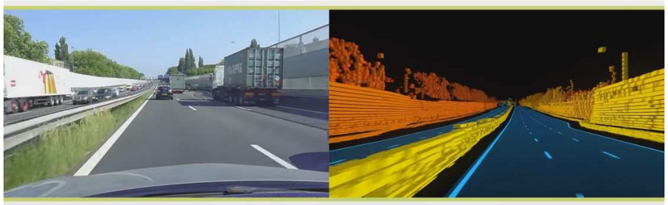

Figure 4: Example of High Definition (HD) map from TomTom RoadDNA (Reproduced with permission of the copyright owner)

jects within the scene is a key strategy. Such intelligent tasks could be improved by CNNs that learn good areas to reconstruct with the aid of a large scaled dataset.

3.2. Types of Maps

Mapping is one of the key pillars of automated driving. The first reliable demonstrations of automated driving by Google were primarily reliant on localization to pre-mapped areas. Because of the scale of the problem, traditional mapping techniques are augmented by semantic object detection for reliable disambiguation. In addition, localized high definition maps (HD maps) can be used as a prior for object detection.

3.2.1 Private Small Scale Maps

There are three primary reasons for the use of customized small scale maps. The first reason is privacy where it is not legally allowed to map the area, for example, private residential area. The second reason is that HD maps still do not cover most of the areas. The third reason is the detection of dynamic structures, that may differ from global measurements. This is typically obtained by classical semi-dense point cloud maps or landmark based maps. Local maps are mainly obtained by methods described in the former section (see Section 2).

3.2.2 Large Scale HD Maps

There are two types of HD maps namely Dense Semantic Point Cloud Maps and Semantic Landmark based Maps. Semantic Landmarked based maps are an intermediate solution to dense semantic point cloud and likely to become redundant.

Dense Semantic Point Cloud Maps: The former is the high end version where all the semantics and dense point cloud are available at high accuracy. Google and TomTom adopt this strategy. As this is high end, it is expensive to cover the entire world and needs large memory requirements. In this case, mapping is treated as a stronger cue than perception. If there is good alignment, all the static objects (road, lanes, curb, traffic signs) are obtained from the map already and dynamic objects are obtained via some sort of background subtraction. TomTom RoadDNA provides an interface to align various sensors like LIDAR, cameras, etc., screenshot below of alignment of dense semantic 3D point cloud to an image. They have mapped majority of European cities and they provide an accuracy of 10 cm assuming a coarse location from GPS.

Landmark based Maps are based on semantic objects instead of generic 3D point clouds. Thus it works primarily for camera data. Mobileye and HERE follow this strategy. In this method, object detection is leveraged to provide an HD map and the accuracy is improved by aggregating over several observations from different cars.

In case of a good localization, HD maps can be treated as a dominant cue and semantic segmentation algorithm greatly simplifies to be a refinement algorithm of priors obtained by HD maps. In Figure 4, the semantic point cloud alignment provides an accurate semantic segmentation for static objects. Note, that it does not cover abstract objects like sky. This would need a good confidence measure for localization accuracy, typically some kind of re-projection error is used. HD maps can also be used for validation or post-processing the semantic segmentation to eliminate false positives.

3.3. Challenges

Despite showing good performance, there are still challenges for Visual SLAM systems to overcome. We identify algorithm and application related challenges.

3.3.1 Algorithm related challenges

- Pure Rotation: If the camera solely rotates, disparity cannot be estimated between consecutive frames.
- Map Initialization: Most approaches start with random initialization and convergence speed depends on the camera movement in the initial phase which makes it unreliable.
- Scale Ambiguity: Visual SLAM system based on a single camera can only estimate the scene and trajectory up to the overall scale. A global reference is needed to solve the scale issue.
- Rolling shutter: Automotive cameras are mainly rolling shutter. If the camera is intended to move at high speed, e.g. for highway driving, rolling shutter distortion occurs. If this is not handled in the algorithm, it will diverge.
- Intelligent Loop-Closure Detection: State of the art approaches use image features to detect loop closures. This is computationally expensive and heavily depends on the robustness of the descriptor.

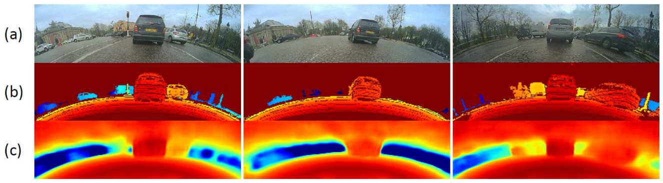

Figure 5: Qualitative results: Example depth map predictions on raw fisheye images. For each image, we show (a) RGB Input (b) LIDAR Ground Truth (c) Predicted Depth Map

3.3.2 Application related challenges

- Self-repairing Maps: Scene structure changes all the time and need to handled by the mapping and localization.
- Mapping on the car: Building a map in the car's embedded processor without having access to cloud infrastructure. It is particularly difficult for CNN based training which needs large compute power.
- Unique signature for large scale areas: Maps for automated driving are very vast and similar structures occur typically which needs to be disambiguated using semantics or global structure.

4. Deep Learning Opportunities

In this section, we explore the replacement of individual blocks of Visual SLAM shown in Figure 2 for performance improvements. Recently, most of the geometric vision tasks are now led by deep learning models [27]. Hence, the following chapter describes their specific capacities and outlines the possibility of using those deep learning solutions within Visual SLAM.

4.1. Depth Estimation

Localization or depth estimation is very critical for automated driving. The genesis of depth estimation using CNN [18] has lead to a wide range of approaches and applications in the depth estimation community. Depth estimation methods [21] mostly stand on architectures that resembles those of semantic segmentation, which are often inspired from classification-based networks. When the depth estimation is supervised, the loss function usually reads as regression loss [18, 39, 41, 59] w/wo regularization terms [70]. Interestingly, [11] uses a ranking loss that penalizes the non relative correspondence between predictions and ground truth while [4] defines depth estimation as a classification problem. In the case of unsupervised depth estimation a projection function between multiple views is carried (using the stereo-rig constraints or estimating a motion between the views) and the consistency of the prediction is assessed based on photometric error [25, 28, 70, 61]. In table 2 we summarize three leading approaches on the KITTI [27] leader-board that all based on CNNs. Hence, Depth estimation using neural networks and inherent applications are promising in the context of Visual SLAM.

Table 2: Raw Depth competition on KITTI [27] from [37].

	Modality	RMSE (0-80m)
Eigen et al. [18]	supervised	7.156
Godard et al. [28]	unsupervised	5.381
Kuznietsov et al. [37]	semi-supervised	**4.621**

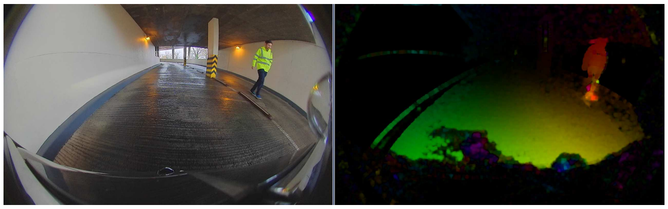

Figure 6: Illustration of dense optical flow from which dense depth for structure from motion can be obtained

Table 3: Flow competition on KITTI [27] from [2]. (background bg, foreground fg)

	bg	fg	bg+fg
Vogel et al. [63]	6.61	20.79	8.97
Menze et al. [42]	7.01	26.34	10.23
Behl et al. [2]	**6.58**	**15.63**	**8.08**

4.2. Optical Flow

CNN based optical flow have produced state of the art results. We verify this with the leaderboard results in Table 3, all based on CNN. Motion detection [31] in particular is a challenging problem because of the continuous camera motion along with the motion of independent objects. Moving objects are the most critical in terms of avoiding fatalities and enabling smooth maneuvering and braking of the car. Motion cues can also enable generic object detection as it is not possible to train for all possible object categories beforehand. Classical approaches in motion detection were focused on geometry based approaches [57, 50, 49, 42, 65]. However, pure geometry based approaches have many limitations, motion parallax issue is one such example. A recent trend [56, 32, 17, 61, 24] for learning motion in videos has emerged. Nonetheless, this trend was focused on pixel-wise motion segmentation. Fragiadaki et. al. suggested a method to segment moving objects [24] that uses a separate proposal generation. However, proposal generation methods are computationally inefficient. Jain et. al. presented a method for appearance and motion fusion in [32]. The work focuses on generic object segmentation. It was not designed

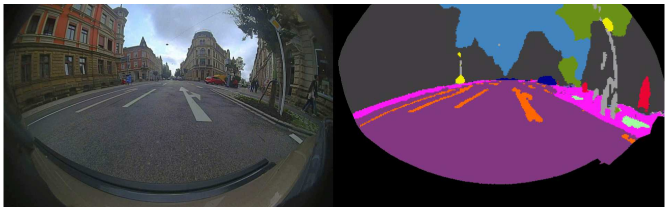

Figure 7: Semantic Segmentation on a fisheye automotive camera

for static/moving vehicles classification. Tokmakov et. al. [56] used a one-stream fully convolutional network with optical flow input to estimate the motion type. The approach works with either optical flow only or concatenated image and flow as input. The concatenated input will not benefit from the available pre-trained weights, as they were trained on RGB only. Drayer et. al. [17] described a video segmentation work that used tracked detections from R-CNN denoted as tubes. This was followed by a spatio-temporal graph to segment objects.

4.3. Feature Correspondence

There are CNN based feature correspondence techniques. For example, a universal correspondence network in [12] by making use of a spatial transformer to normalizer for affine transformations demonstrates state-of-the-art results in various datasets. This is an example of feature correspondence learning independent of the application in which it is used. It is an open problem to learn feature correspondence which is optimal for the later stages like bundle adjustment. For instance, end-to-end learning of feature matching could possibly learn diversity and distribution as well instead of just picking the top high textured features.

4.4. Bundle Adjustment

There is no mature solution for CNN based bundle adjustment. There were a few initial attempts at it last year which were published in CVPR, [58] tries to model projection constraints in a differentiable way. There are techniques to jointly learn a pipeline like Visual SLAM with a learnable part (for feature matching and depth) and a user defined geometric part. For instance, when you jointly learn the feature matching, it could possibly learn diversity and distribution as well instead of just picking the top high textured features.

4.5. Semantic Segmentation

Semantic segmentation is targeted towards partitioning the image into semantically meaningful parts with various applications for that. It has been used in robotics [60, 6, 62, 36], medical applications [14, 71], augmented reality [43], and most prominently automated driving [69, 53, 9, 15]. There were mainly three subcategories of the work that was developed.

Table 4: Semantic Competition on Cityscapes [15].

	IoU Class	IoU Category
Mapillary	82.0	91.2
SR-AIC	81.9	91.3
EFBNET	81.8	90.7

The first [22, 23, 29] used patch-wise training to yield the final classification. In [22] an image is fed into a Laplacian pyramid, each scale is forwarded through a 3-stage network to extract hierarchical features and patch-wise classification is used. The output is post processed with a graph based classical segmentation method. In [29] a deep network was used for the final pixel-wise classification to alleviate any post processing needed.

The second subcategory [40, 48, 1] was focused on end-to-end learning of pixel-wise classification. It started with the work in [40] that developed fully convolutional networks (FCN). The network learned heatmaps that was then upsampled within the network using deconvolution to get dense predictions. Unlike patch-wise trainings this method uses the full image to infer dense predictions. In [48] a deeper deconvolution network was developed, in which stacked deconvolution and unpooling layers are used. In Segnet [1] a similar approach was used where an encoder-decoder architecture was deployed. In Figure 7 an example of the semantic segmentation output of Segnet applied in an automated driving setting is shown.

Finally, the work in [68, 22, 48, 10, 51, 52] focused on multi-scale semantic segmentation. Initially in [22] the scale issue was addressed by introducing multiple rescaled versions of the image to the network. The skip-net architecture in [40] was used to merge heatmaps from different resolutions. Since these architectures rely on downsampling the image, loss of resolution can hurt the final prediction. The work in [52] proposed a u-shaped architecture network where feature maps from different initial layers are upsampled and concatenated for the next layers. Another work in [68] introduced dilated convolutions, which expanded the receptive field without losing resolution based on the dilation factor.

4.6. Camera pose estimation

Localization inside the map is a crucial part of SLAM, where the position can be described by a 6-DOF camera pose. Such poses can be recovered using feature-based pipelines like SfM. Kendall et al. [33] trained a CNN to map a single RGB image directly to a cameras orientation and position in an end-to-end manner. Unlike methods based on image databases, this proposed neural network, PoseNet, does not require memory linearly proportional to the size of the scene. Furthermore PoseNet was shown to be robust to difficult lighting, motion blur and different camera intrinsics where SIFT based registration fails.

Instead of using a direct regression of the 6-DOF camera pose, Brachmann et al. [8] used a sequence of less complex tasks. A first network learns to map local image patches to corresponding positions in 3D scene space. Subsequently a differentiable RANSAC [7] approach is used to get a camera pose that aligns to the predicted scene coordinates. While still being an end-to-end trainable pipeline, this approach exploits geometrical constraints and achieves superior results.

5. CNN Based Pipelines

Due to the nature of deep neural networks, the same network architecture can be jointly learned for different high-dimensional regression tasks. By sharing features for various tasks the efficiency and generalization is increased. This is especially useful for real-time critical application like automated driving.

In section 4 we investigated in detail the building block technology of reconstructing a 3D scene with Visual SLAM using CNN geometric vision tasks. This section unfolds the closed relationship between the 3D scene and the basic geometric tasks.

5.1. Joint Supervised Semantic SLAM

Tateno et al. [54] proposed a CNN to jointly learn semantic segmentation and depth maps. Their approach integrates a CNN based depth prediction with SLAM to overcome traditional limitations of monocular reconstructions. By fusing predicted semantic labels with the dense point cloud, they obtain a semantically coherent scene reconstruction from a monocular view.

This approach combines efficient geometric building blocks like depth estimation and semantic segmentation, to improve the traditional pipeline of Visual SLAM (e.g. PTAM, LSD-SLAM).

5.2. Joint Unsupervised SLAM using Optical Flow

Recently, [67] Yin et al. proposed a joint architecture that simultaneously learns monocular depth, optical flow and egomotion estimation based on video inputs using an unsupervised manner. They achieve state of the art results for each vision task such as odometry using the KITTI benchmark suite [27]. The approach removes the need of data annotation for CNN based SLAM. The key idea is to get use of the strong dependence of each geometric vision task (depth, pose and optical flow) to design a joint loss function that is purely based on consistency checks. Therefore, a rigid decoder for depth and pose such as a non-rigid motion decoder for optical flow is designed. The loss is defined in the following manner:

$$\mathcal{L} = \sum\sum[\mathcal{L}_{rw} + \mathcal{L}_{ds} + \mathcal{L}_{fw} + \mathcal{L}_{fs} + \mathcal{L}_{gc}] \quad (1)$$

$\mathcal{L}_{rw}$ (warping loss) and $\mathcal{L}_{ds}$ (depth smoothness) denote the rigid decoder. $\mathcal{L}_{fw}$, $\mathcal{L}_{fs}$ and $\mathcal{L}_{gc}$ design the non-rigid motion localizer (see Fig. 8). All could be directly derived from the 3D scene purely based on consistency. The results on KITTI for odometry estimation are highlighted in Table 5. The method outperforms ORB-SLAM on an automotive scenario. The short outline emphasize the possibility of using deep learning for SLAM.

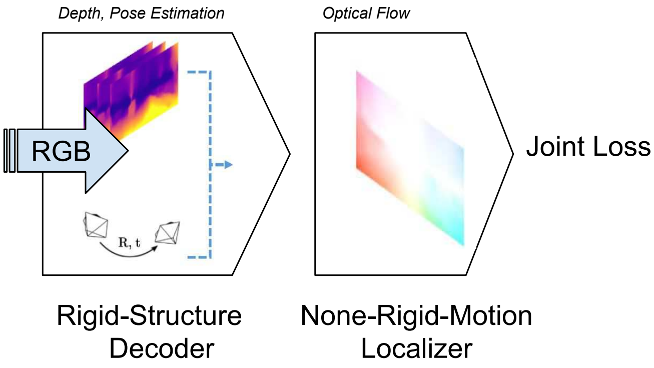

Figure 8: Joint Unsupervised Pipeline based on basic geometric vision tasks: depth estimation, pose estimation and optical flow [67]

Table 5: Absolute Trajectory Error (ATE) on KITTI odometry dataset. The results of other baselines are taken from 8.

	Sequence 09	Sequence 10
ORB-SLAM	0.014 ± 0.008	0.012 ± 0.011
Yin et al.	**0.012±0.007**	**0.012±0.007**

6. Conclusion

CNNs have become the de facto approach for object detection and semantic segmentation in automated driving. They also show promising progress in geometric computer vision algorithms like depth and flow estimation. However, there is slow progress on CNN based Visual SLAM approaches. In this work, we provided an overview of Visual SLAM for automated driving and surveyed possible opportunities for using CNNs in various building blocks. The authors feel that this is an exciting area of research and hope that this work will encourage further progress. Future research is to prototype and evaluate the accuracy of the proposed approaches.

ACKNOWLEDGMENT

The authors would like to thank their employer for the opportunity to work on fundamental research. We would also like to thank B Ravi Kiran (AKKA France) and Gabriel Leivas Oliveira (University of Freiburg) for reviewing the paper and providing feedback.

References

[1] V. Badrinarayanan, A. Handa, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv preprint arXiv:1505.07293*, 2015. 7

[2] A. Behl, O. H. Jafari, S. K. Mustikovela, H. A. Alhaija, C. Rother, and A. Geiger. Bounding boxes, segmentations and object coordinates: How important is recognition for 3d scene flow estimation in autonomous driving scenarios? In *International Conference on Computer Vision (ICCV)*, 2017. 6

[3] P. Biber, T. Duckett, et al. Dynamic maps for long-term operation of mobile service robots. In *Robotics: science and systems*, pages 17–24, 2005. 1

[4] B. Bischke, P. Helber, J. Folz, D. Borth, and A. Dengel. Multi-task learning for segmentation of building footprints with deep neural networks. *arXiv preprint arXiv:1709.05932*, 2017. 6

[5] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016. 1

[6] T. M. Bonanni, A. Pennisi, D. Bloisi, L. Iocchi, and D. Nardi. Human-robot collaboration for semantic labeling of the environment. In *Proceedings of the 3rd Workshop on Semantic Perception, Mapping and Exploration*, 2013. 7

[7] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother. DSAC - differentiable RANSAC for camera localization. *CoRR*, abs/1611.05705, 2016. 8

[8] E. Brachmann and C. Rother. Learning less is more - 6d camera localization via 3d surface regression. *CoRR*, abs/1711.10228, 2017. 8

[9] G. J. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009. 7

[10] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. *arXiv preprint arXiv:1511.03339*, 2015. 7

[11] W. Chen, Z. Fu, D. Yang, and J. Deng. Single-image depth perception in the wild. In *Advances in Neural Information Processing Systems*, pages 730–738, 2016. 6

[12] C. B. Choy, J. Gwak, S. Savarese, and M. Chandraker. Universal correspondence network. In *Advances in Neural Information Processing Systems*, pages 2414–2422, 2016. 7

[13] W. Churchill and P. Newman. Practice makes perfect? managing and leveraging visual experiences for lifelong navigation. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4525–4532. IEEE, 2012. 1

[14] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 424–432. Springer, 2016. 7

[15] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. *arXiv preprint arXiv:1604.01685*, 2016. 7

[16] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6):1052–1067, June 2007. 1, 3

[17] B. Drayer and T. Brox. Object detection, tracking, and motion segmentation for object-level video segmentation. *arXiv preprint arXiv:1608.03066*, 2016. 6, 7

[18] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014. 6

[19] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. In *arXiv:1607.02565*, July 2016. 3, 4

[20] J. Engel, T. Schöps, and D. Cremers. Lsd-slam: Large-scale direct monocular slam. In *European Conference on Computer Vision*, pages 834–849. Springer, 2014. 1, 3, 4

[21] S. R. Fanello, C. Rhemann, V. Tankovich, A. Kowdle, S. Orts-Escolano, D. Kim, and S. Izadi. Hyperdepth: Learning depth from structured light without matching. In *CVPR*, volume 2, page 4, 2016. 6

[22] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1915–1929, 2013. 7

[23] C. Farabet, N. EDU, C. Couprie, L. Najman, and Y. LeCun. Scene parsing with multiscale feature learning, purity trees, and optimal covers. 7

[24] K. Fragkiadaki, P. Arbelaez, P. Felsen, and J. Malik. Learning to segment moving objects in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4083–4090, 2015. 6

[25] R. Garg, V. K. BG, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer, 2016. 6

[26] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun. 3d traffic scene understanding from movable platforms. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):1012–1025, 2014. 1

[27] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 4, 6, 8

[28] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, volume 2, page 7, 2017. 6

[29] D. Grangier, L. Bottou, and R. Collobert. Deep convolutional networks for scene parsing. In *ICML 2009 Deep Learning Workshop*, volume 3. Citeseer, 2009. 7

[30] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1

[31] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017. 6

[32] S. D. Jain, B. Xiong, and K. Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. *arXiv preprint arXiv:1701.05384*, 2017. 6

[33] A. Kendall, M. Grimes, and R. Cipolla. Convolutional networks for real-time 6-dof camera relocalization. *CoRR*, abs/1505.07427, 2015. 7

[34] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, November 2007. 3

[35] K. Konolige and J. Bowman. Towards lifelong visual maps. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 1156–1163. IEEE, 2009. 1

[36] A. Kundu, Y. Li, F. Dellaert, F. Li, and J. M. Rehg. Joint semantic segmentation and 3d reconstruction from monocular video. In *European Conference on Computer Vision*, pages 703–718. Springer, 2014. 7

[37] Y. Kuznietsov, J. Stückler, and B. Leibe. Semi-supervised deep learning for monocular depth map prediction. *CoRR*, abs/1702.02706, 2017. 6

[38] H. Lategahn, J. Beck, B. Kitt, and C. Stiller. How to learn an illumination robust image feature for place recognition. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 285–291. IEEE, 2013. 1

[39] B. Li, C. Shen, Y. Dai, A. van den Hengel, and M. He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1119–1127, 2015. 6

[40] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 7

[41] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016. 6

[42] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3061–3070, 2015. 6

[43] O. Miksik, V. Vineet, M. Lidegaard, R. Prasaath, M. Nießner, S. Golodetz, S. L. Hicks, P. Pérez, S. Izadi, and P. H. Torr. The semantic paintbrush: Interactive 3d mapping and recognition in large outdoor spaces. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3317–3326. ACM, 2015. 7

[44] P. Muehlfellner, P. Furgale, W. Derendarz, and R. Philippsen. Evaluation of fisheye-camera based visual multi-session localization in a real-world scenario. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 57–62. IEEE, 2013. 1

[45] P. Mühlfellner, M. Bürki, M. Bosse, W. Derendarz, R. Philippsen, and P. Furgale. Summary maps for lifelong visual localization. *Journal of Field Robotics*, 33(5):561–590, 2016. 1

[46] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 1, 3, 4

[47] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 2320–2327, Washington, DC, USA, 2011. IEEE Computer Society. 3

[48] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015. 7

[49] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *IEEE transactions on pattern analysis and machine intelligence*, 36(6):1187–1200, 2014. 6

[50] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1777–1784, 2013. 6

[51] G.-J. Qi. Hierarchically gated deep networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 7

[52] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015. 7

[53] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3234–3243, 2016. 7

[54] K. Tateno, F. Tombari, I. Laina, and N. Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction. *arXiv preprint arXiv:1704.03489*, 2017. 8

[55] M. Teichmann, M. Weber, M. Zoellner, R. Cipolla, and R. Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. *arXiv preprint arXiv:1612.07695*, 2016. 1

[56] P. Tokmakov, K. Alahari, and C. Schmid. Learning motion patterns in videos. *arXiv preprint arXiv:1612.07217*, 2016. 6, 7

[57] P. H. Torr. Geometric motion segmentation and model selection. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 356(1740):1321–1340, 1998. 6

[58] S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. 7

[59] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox. Demon: Depth and motion network for learning monocular stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 5, 2017. 6

[60] A. Valada, G. L. Oliveira, T. Brox, and W. Burgard. Deep multispectral semantic scene understanding of forested environments using multimodal fusion. In *The 2016 International Symposium on Experimental Robotics (ISER 2016)*, 2016. 7

[61] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki. Sfm-net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804*, 2017. 6

[62] V. Vineet, O. Miksik, M. Lidegaard, M. Nießner, S. Golodetz, V. A. Prisacariu, O. Kähler, D. W. Murray, S. Izadi, P. Perez, and P. H. S. Torr. Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2015. 7

[63] C. Vogel, K. Schindler, and S. Roth. 3d scene flow estimation with a piecewise rigid scene model. *International Journal of Computer Vision*, 115(1):1–28, 2015. 6

[64] S. Wang, R. Clark, H. Wen, and N. Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 2043–2050. IEEE, 2017. 3, 4

[65] S. Wehrwein and R. Szeliski. Video segmentation with background motion models. In *BMVC*, 2017. 6

[66] H. Xu, Y. Gao, F. Yu, and T. Darrell. End-to-end learning of driving models from large-scale video datasets. *arXiv preprint arXiv:1612.01079*, 2016. 1

[67] Z. Yin and J. Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose, 2018. 8

[68] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 7

[69] H. Zhang, A. Geiger, and R. Urtasun. Understanding high-level semantics by modeling traffic patterns. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3056–3063, 2013. 7

[70] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, volume 2, page 7, 2017. 6

[71] W. Zhu and X. Xie. Adversarial deep structural networks for mammographic mass segmentation. *arXiv preprint arXiv:1612.05970*, 2016. 7