AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY

FIELD OF SCIENCE: ENGINEERING AND TECHNOLOGY

SCIENTIFIC DISCIPLINE: AUTOMATION, ELECTRONICS, ELECTRICAL ENGINEERING AND SPACE TECHNOLOGIES

SUMMARY OF ACCOMPLISHMENTS

Depth Completion for FMCW Radars

Author:	*Mariusz Karol Nowak*
Supervisor:	*Dr hab. inż. Paweł Skruch, prof. AGH*
Completed in:	*AGH University of Science and Technology* *Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering* *Department of Automatic Control and Robotics*

Cracow, Poland, 2023

Abstract

A model of the environment which includes distance to the nearest obstacle is necessary for a system tasked with autonomous movement. Therefore, it is reasonable to say that every autonomously-navigating system must be able to estimate depth (i.e. the distance to the nearest object). However, there is a trade-off between the size, complexity and cost of a depth sensor on one hand, and the density and accuracy of its measurements on the other hand.

The goal of this dissertation is to examine the feasibility of depth completion (i.e. producing a dense depth-map from a sparse depth input) for extremely sparse depth measurements. The bulk of this work deals with depth completion on automotive Frequency Modulated Continuous Wave (FMCW) radars. Additionally, I present a novel way to perform depth completion on lidar point cloud, which I developed for the purpose of creating the training dataset for the radar depth completion task, an algorithm utilizing zero-centered, additive weight perturbations during neural network training and simple simulations showing the capability of a neural network to perform angle finding in the presence of 2 targets.

The first two chapters of the dissertation describe the problem formulation, motivation and automotive sensors, with emphasis put on FMCW radars. The next chapters describe my original contributions.

The first contribution is an algorithm utilizing zero-centered, additive weight perturbations during neural network training. I show why it helps to increase the amount of information the neural network can learn for a given size and demonstrate its usefulness for some commonly used neural network architectures. Weight perturbations were utilized in the WeaveNet training and in the training of the radar angle finding model on simulated data.

The second contribution is the definition of WeaveNet - a neural network whose architecture is designed to perform well in the task of lidar depth completion on variable input sparsity depth measurements (name inspired by the convolutional kernels pattern, which looks like a woven fabric). It was trained and tested utilizing the data from the KITTI Depth Completion challenge. WeaveNet was instrumental in creating a dense depth dataset, that was later used in my radar depth completion solution.

The third contribution consists of a set of idealized simulations, showing that neural networks are capable of finding angles to two targets when radar antennae receive a superposition of reflected waves (a necessary step for a radar depth completion network).

The fourth contribution is the creation of a dataset used to train and validate algorithms for radar depth completion. It consists of over 1,000,000 Radar Data Cubes (RDCs) from a forward-facing radar, together with corresponding camera images and dense depth maps (produced using WeaveNet).

The fifth contribution is the design of a neural network architecture capable of transforming the low-level RDC input into abstract channels in the azimuth-elevation plane. Consequently, it was possible to train this neural network to predict dense depth maps using RDC as input. They were trained and tested on the dataset that was created for the purpose of the work on this PhD dissertation. The output of the radar-only depth completion networks is visually reasonable and much better than the linear interpolation of the point cloud obtained using standard radar processing algorithms. To the best of my knowledge, this is the first solution producing a dense depth map on the basis of automotive FMCW radar RDC.

The final contribution is the design of the neural network architecture capable of fusing the RDC-derived data from the radar and the data derived from RGB camera images for the purpose of radar depth completion (also trained and tested on the same dataset). I have shown that the networks utilizing RDC in addition to visual data obtain results between 3% and 21.5% better than analogous networks trained to use visual data only (during different data collection drives).

1 Problem Formulation

1.1 Motivation

In a very broad sense, the goal of this work is to use neural networks to extract more information from an automotive Frequency Modulated Continuous Wave (FMCW) radar. In order to achieve this goal I decided to use a low-level radar output - specifically I used a Radar Data Cube (RDC) as the network input. RDC is a three dimensional representation of the radio wave reflections detected by the radar at a particular point in time. First dimension represents the distance between the sensor and the reflection source, second dimension represents the relative radial velocity between the sensor and the reflection source and the third dimension represents the signal at the different antennae.

More specifically, my goal was to check, whether it is possible to use a neural network to extract a lidar-like scene representation (a dense depth map) from an automotive FMCW radar. I chose the depth map as the representation which I trained the network to produce for 2 main reasons.

The first reason is that the task is ambitious - the standard automotive radar output is much different than a dense depth map (presented in Figure 1.2). Simple interpolations between radar depth measurements fail to produce a representation of a scene that would be interpretable by a human. Hence, if the model is able to produce a visually recognizable scene representation, we can conclude that it is much better at extracting the information from the signal than the standard algorithms used in radars. In fact, my algorithm achieves this goal, as evidenced by the Figure 1.3. In that figure, I present a camera image, dense depth map obtained using lidar and the output of my radar model. Contrary to the raw reflections interpolation, my depth map reconstruction from low-level radar data produces a visually understandable scene. I consider it to be a good marker of the performance of my solution.

The second reason for choosing to predict a dense depth map is more mundane - the ease of dataset construction. A dataset of dense depth maps can be created without the need for costly human labeling. I created a 1,000,000 frames dataset using a rooftop-mounted lidar and a WeaveNet neural network (that was purpose-designed by me in the preliminary phase of the research described in this dissertation). As

the depth completion datasets can be created in an automated way (hence relatively cheaply), they can be used for pretraining a neural network whose final task is different.

1.2 Inputs and Outputs

As stated in Section 1.1, this dissertation deals with the problem of transforming the output of an automotive FMCW radar into a dense depth map. I achieve it using a custom-designed neural network. I created two versions of the network: a radar-only version and radar+camera version. The radar-only version of the network uses only RDCs to create a dense depth maps, and the radar+camera version utilizes both RDC and camera images. Schematic representation of its inputs and outputs is presented in Figure 1.1.

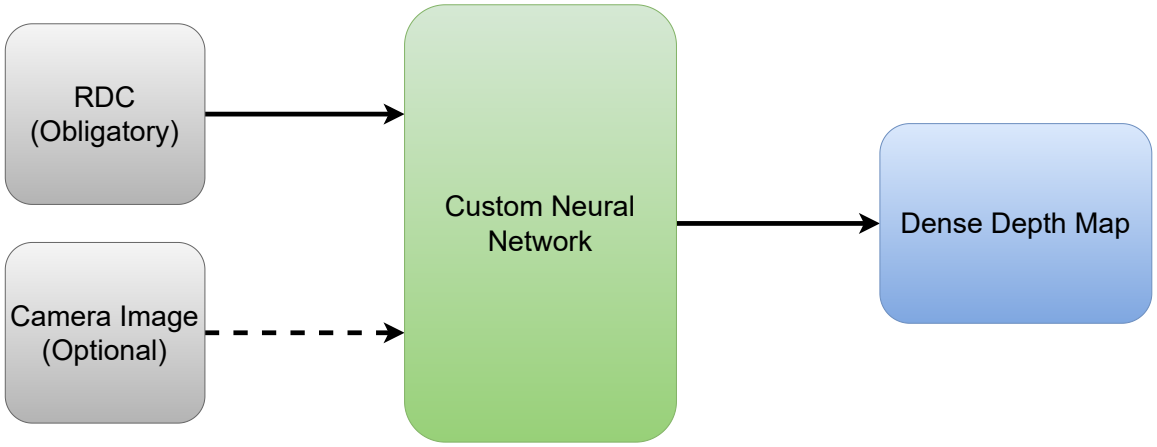

Figure 1.1. Schematic representation of the inputs and outputs of the radar depth completion network. The camera image input is optional.

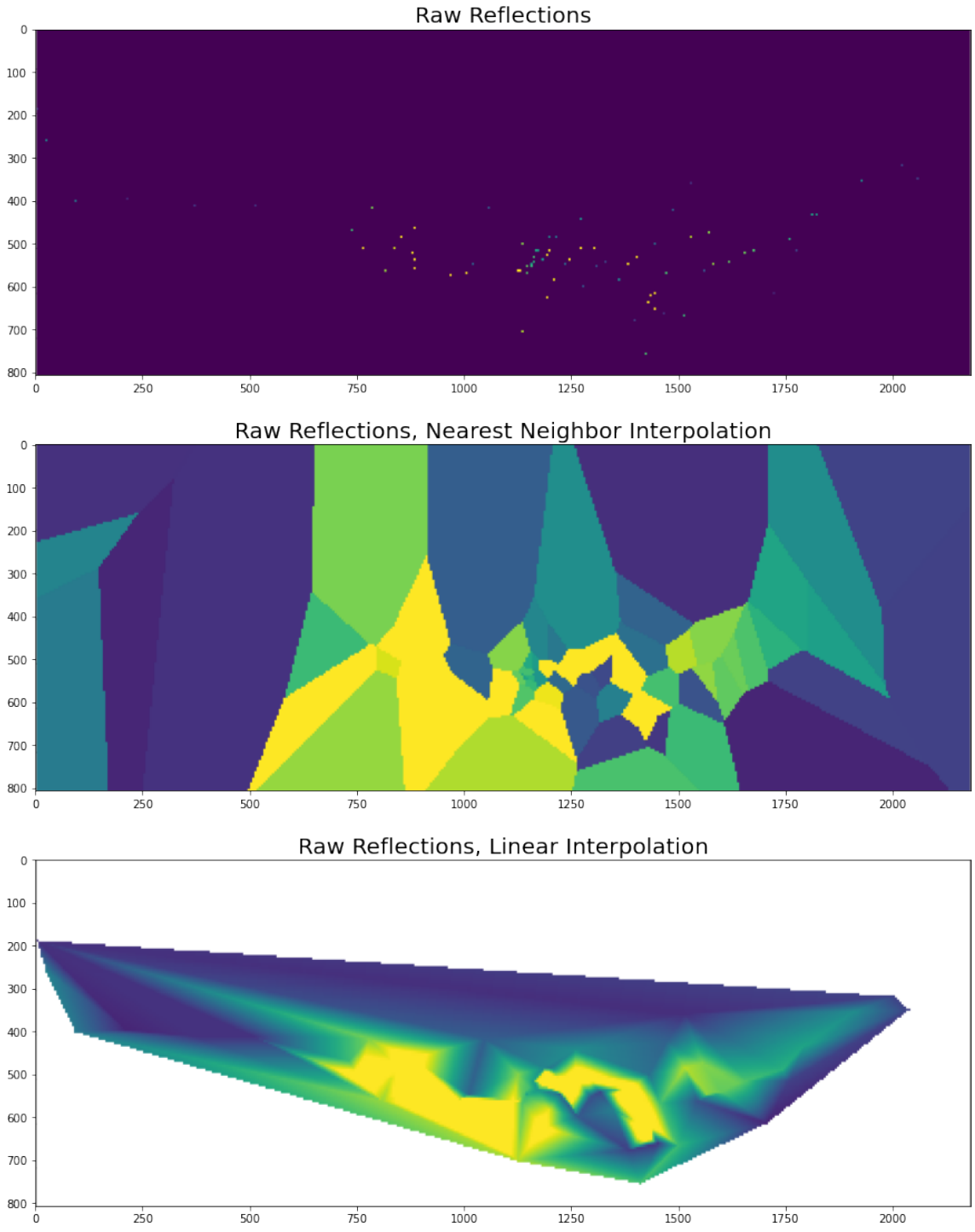

Figure 1.2. Depth maps created using raw radar reflections and naive interpolations (linear and nearest neighbor), distance is color-coded. Numbers on the axes denote pixel coordinates.

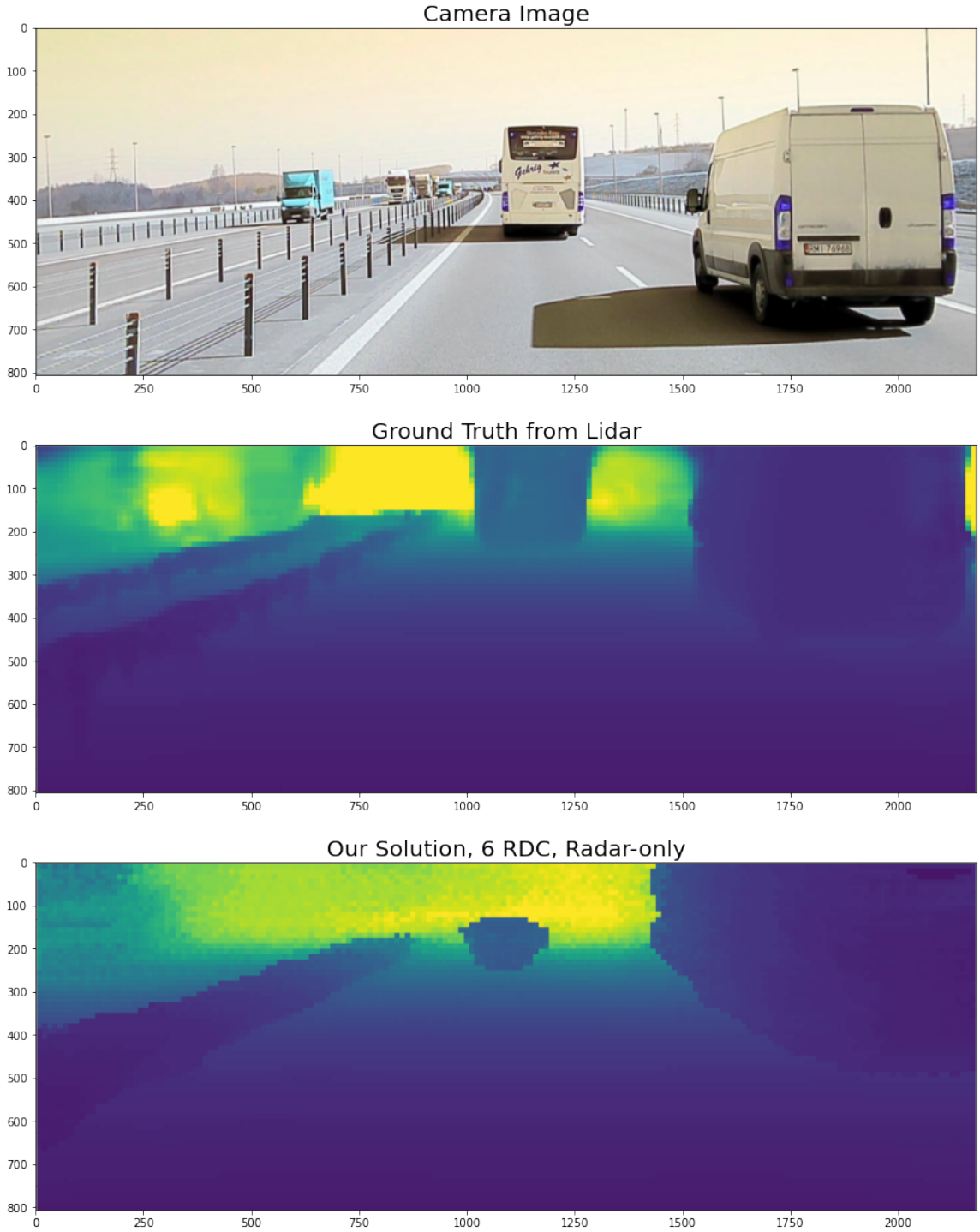

Figure 1.3. Camera image, ground truth (GT) from lidar and a depth map reconstruction using low-level radar data (my solution), distance is color-coded. Numbers on axes denote pixel coordinates.

1.3 Theses

The goal of this dissertation is to show the feasibility of depth completion (i.e. producing a dense depth map from a sparse depth input) for automotive Frequency Modulated Continous Wave (FMCW) radars and the feasibility of camera - FMCW radar fusion for depth completion. It can be distilled in the following theses:

a Low-level FMCW radar output can be processed in such a way, that a dense depth map is produced.

b A model fusing low-level FMCW radar output with camera images can produce a higher quality depth map than an analogous model utilizing only camera images.

1.4 Scope of Work and Organization of the Thesis

The end goal of this work has always been creating a dense depth map using the outputs of an automotive radar. However, to achieve this goal, I had to use quite a circuitous path. Graphically, it is represented in Figure 1.4. The first thing needed to develop a machine learning model is the dataset. I had access to data collected using a car with a forward-facing FMCW radar, a forward-facing camera, and a lidar on top. The 'independent variables' (RDCs from radar and camera images) were easy to measure directly, however, the lidar output is not dense, but only semi-dense. For example in the KITTI dataset (Uhrig et al. [2017], Geiger et al. [2013]), Velodyne HDL-64E lidar provides approximately 20,000 detections in the field of view of the front camera, while the camera itself provides an image with 465,750 pixels (375x1242). Lidar depth completion is a known problem, with known solutions, however, none of them could be safely applied to my case directly. I needed to create a dense depth map from the pointcloud from a Pandora lidar projected on a plane located closer to the grille (radar location) than the lidar itself. Such arrangement results in an unequal distribution of the measurements in the depth completion plane. Therefore, I had to create a lidar depth completion solution that is indifferent to the measurement density. I did it, by defining a WeaveNet neural network architecture, and a specialized universal sparsity training procedure. I trained that network on the publicly available KITTI lidar depth completion dataset (Uhrig et al. [2017]). During the work on WeaveNet, I noticed that an uncommon training procedure was very beneficial in its training. This procedure consisted of utilizing zero-centered, additive weight perturbations during neural network training. I dug deeper into it, demonstrated its usefulness for a range of neural architectures, and showed why it helps to better utilize the neural network weights. When I completed the work on WeaveNet, I used it to create dense depth maps using the data collected using the test car.

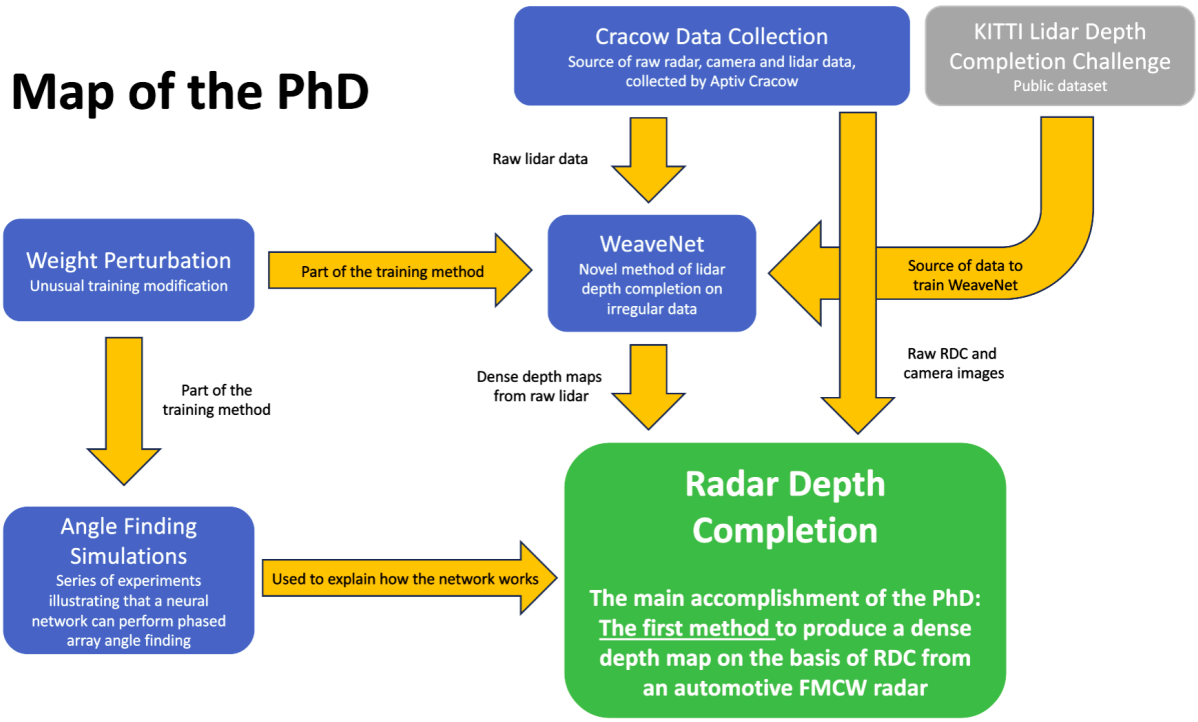

Figure 1.4. A graphical representation of the contents of the PhD.

Once I had the dataset, I could delve into developing a solution to the problem of radar depth completion. In particular, I defined a novel neural network architecture suitable for transforming the data from RDC into abstract channels in the azimuth-elevation plane and trained it for the task of depth completion. I also created a neural network fusing the RDC and visual data and trained it to produce dense depth maps. I have demonstrated that my network is able to extract useful information from RDC by showing that a radar+vision network achieves significantly better results than an analogous network utilizing only the camera images. Finally, I wanted to do something to 'illuminate the black box' of my radar depth completion solution. I created a toy model of the FMCW radar anglefinding when reflections from 2 different targets are superimposed. I have shown that a neural network is capable of disentangling

the 2 sources of signal and successfully perform anglefinding in such a scenario. The zero-centered, additive weight perturbations were also used for training this network.

2 Weight Perturbation

Using the toy example of a neural network approximating a XOR gate I have shown that SGD-based optimizers are unlikely to achieve good results if the weights in a particular layer are too similar to each other. This effect is more pronounced in smaller neural networks, but does not disappear in the larger versions. Moreover, the effect appears to be stable, regardless of the length of training (so it does not only slow training down, but rather makes it impossible in some cases).

To combat this effect I proposed to apply noise to network weights during training. I have shown that using a training procedure utilizing additive weight perturbation causes visible performance improvement in some of the tested network architectures (MobileNetV2, DenseNet and Xception), when compared to the vanilla training procedure. The results are particularly significant for MobileNetV2, with weight perturbations clearly improving the performance of both the standard network and a network transformed according to the lottery ticket training subroutine.

The weight perturbation procedure is easily adaptable to the training of any neural network and consequently provides a potentially useful addition to a deep learning practitioner's toolbox. The place of the weight perturbation procedure in such toolbox is comparable with the place of the lottery ticket subroutine. While for some network architectures, the use of lottery tickets brings significant improvements, it is conjectured not to bring performance gains (Frankle and Carbin [2018]) in other architectures. My experiments show the same to be true for weight perturbations (it improved on vanilla training for MobileNetV2, DenseNet and Xception - 3 out of 5 cases). It is also worth noting, that training with weight perturbations works very well in tandem with the lottery ticket subroutine (it causes very clear improvement over the lottery ticket alone when applied to MobileNetV2, ResNet152V2 and EfficientNet B0 - 3 out of 5 cases, in the remaining 2 cases it does not cause the performance of the network to drop).

I used this training method to train WeaveNet and to train the neural networks used in the angle finding experiments.

3 Lidar Depth Completion - WeaveNet

I presented a novel WeaveNet architecture (see Fig. 3.1) capable of performing the depth completion task on very sparse input data. The results obtained by the version of the network trained using variable input sparsity are particularly promising since they point to the possibility of using depth completion methods using data from sensors producing a highly variable number of irregularly spaced measurements. An example of such a sensor that might in the future benefit from depth completion is a high-resolution imaging radar.

Another contribution of this work is the interpretation of the channels at the intermediate layers of the network. The knowledge that the channels in the intermediate layers of the network either represent the interpolated depth data or are estimating the positions of object boundaries might help improve future depth completion networks by incorporating auxiliary losses using the intermediate layer output.

I used the WeaveNet network to create the radar depth completion dataset. The robustness of the method to the varying density of depth measurements was of crucial importance, as to collect that dataset we used the Pandora lidar (instead of the Velodyne one). This enabled the training of the radar depth completion network.

4 Neural Network Angle Finding Simulations

I have shown that a neural network can deal with the task of angle finding in a phased array radar. I have shown, that angle finding can be performed reasonably well even when 2 waves are superimposed at the antenna. I have analyzed the network performance at different levels of the amplitude of the weaker wave (which can be thought of as noise from the perspective of estimating strong wave parameters). I

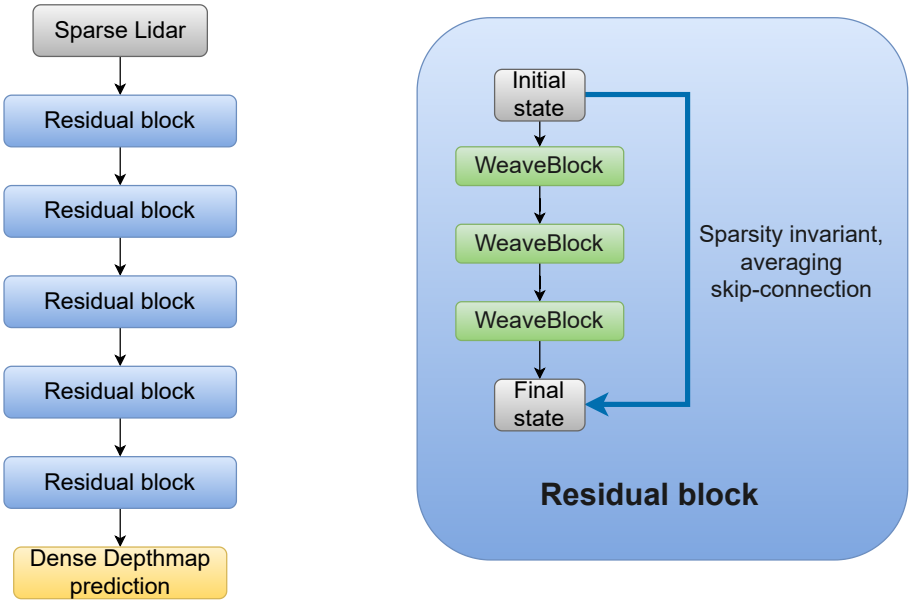

Figure 3.1. A schematic representation of the unguided WeaveNet architecture on the left, and a close-up on a single residual block on the right.

have illustrated the well-known fact that larger antennae are better at angle finding. The angle finding performance is better for larger antennae, since for a small antenna the same pattern at receiving elements may be the result of waves coming from different directions. This is illustrated in Figure 4.1. Please note, that for the small antennae the network predictions have multiple peaks, for different possible angles of arrival, and for the 20 element antenna, there is one very pronounced peak, at the correct angle. This result is very relevant to my experiments on radar depth completion, as the automotive radar antennae are significantly larger in the horizontal plane than in the vertical plane, thereby making the azimuth estimation much easier than elevation estimation. I believe that these methods can be easily used to perform a fast evaluation of different radar antenna designs. Neural networks are universal approximators

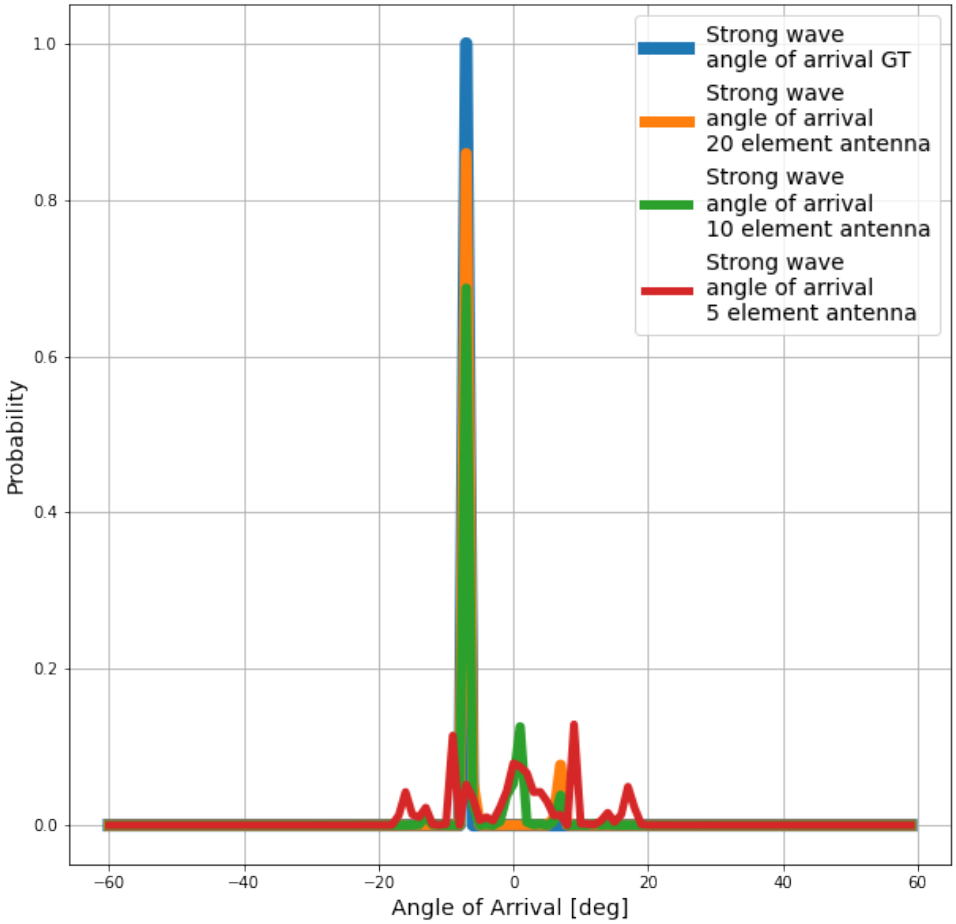

Figure 4.1. Estimation of the strong wave angle of arrival using antennae of different sizes.

(Hornik et al. [1989]). Hence, they can be used to quickly put some bounds on the level of performance that can be obtained using a specific antenna design.

5 Dataset

We have collected a dataset that consists of approximately 1,000,000 radar frames of a forward-facing radar, which corresponds to roughly 14 hours of driving. Radar had 12 receiving antennae and utilized 77 GHz (3.9 mm) waves. The dataset was collected during 7 drives, 2 of which were done on a highway and 5 in an urban setting. All the drives were in good weather conditions. For each of the rides, the

first 90% of frames (chronologically) were assigned to the train set, and the last 10% were assigned to the test set. The PhD dissertation contains histograms characterizing the dataset. The biggest differences are between the urban and the highway drives. The distribution of the radar distance measurements is very different than the distribution of the lidar distance measurements. This is mainly caused by the fact that radar can detect only one unambiguous object for a given distance-velocity combination and by the CFAR thresholding. The size of the range bins is slightly different for the radar short look and the long look, but the distribution of the detections in bins is very similar in both operating modes. Similarly, the distribution of velocities measured in both radar operating modes is very similar.

6 Radar Depth Completion

I presented a novel neural network architecture capable of processing low-level radar data from an FMCW radar commonly used in automotive. My neural network (shown in Figure 6.1) may be logically divided into 6 modules (not counting inputs as one of them), they are enumerated below.

0. Inputs.
1. Trainable angle-finding module.
2. Trainable encoder for the range and relative velocity dimensions of the RDC.
3. Fixed module dealing with reshaping the output of the RDC encoder to project it onto a 2D azimuth-elevation plane.
4. Trainable encoder for the camera images.
5. Trainable sensor and temporal fusion module.
6. Trainable module jointly processing the encoded RDC data and encoded camera image in the azimuth-elevation plane.

In the case of the radar-only experiments, the network structure was kept, but the vision processing stream was fed with constant dummy input. Similarly, in the case of vision-only experiments, the network structure was kept, but the radar processing stream was fed with constant dummy input. The proposed network architecture is capable of inferring significantly more information than standard algorithms used in automotive radars. In particular, I would like to note that my radar-only networks are able to produce visually understandable scene images using a sensor that was not designed to produce such output (see

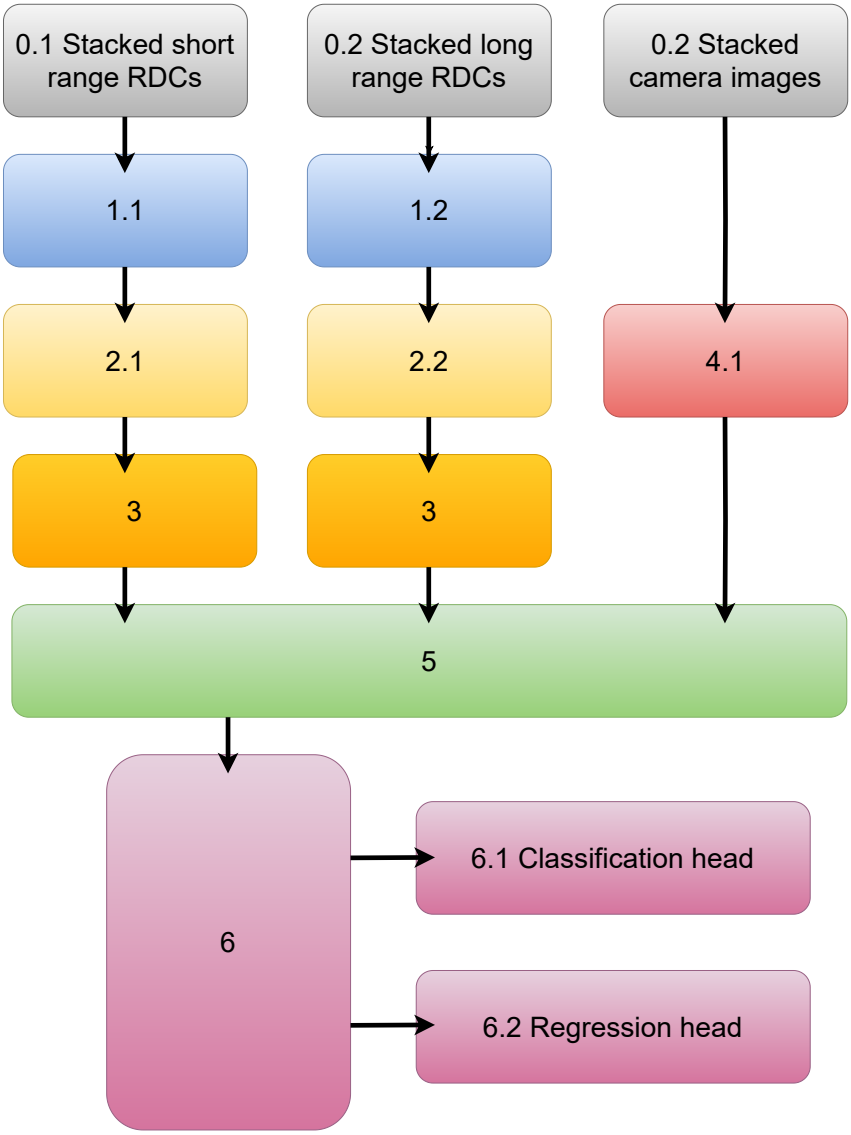

Figure 6.1. Diagram of the high-level network structure. Elements of the diagram are numbered according to the convention from subsection **??**. In the case of the elements whose number consists of 2 digits (e.g. 1.1), the second digit is introduced to emphasize that the multiple elements sharing the same leading digit do not share weights.

Figures 6.2) and 6.3). To the best of my knowledge, this is the first solution producing a dense depth map on the basis of automotive FMCW radar RDC.

My solution transforms the RDC data into the azimuth-elevation plane, making fusion with visual data relatively easy. My radar+vision models perform measurably better than vision-only controls, proving that my model architecture is capable of extracting useful information from RDC (see Tables 6.1 and 6.2).

Table 6.1. Relative L1 results for different datasets (no distance cap).

Dataset	Radar only 6 RDCs	Radar only 8 RDCs	Radar+Vision 6 RDCs	Vision only	Percentage improvement over vision only
Highway drive 1	0.2803	0.1414	0.0842	0.0895	5.92%
Highway drive 2	0.161	0.1553	0.1076	0.1164	7.56%
Urban drive 1	0.2699	0.2862	0.1135	0.1207	5.97%
Urban drive 2	0.3462	0.2265	0.1087	0.1121	3.03%
Urban drive 3	0.2789	0.2914	0.1020	0.1162	12.22%
Urban drive 4	0.5179	0.4112	0.1057	0.1347	21.53%
Urban drive 5	0.3847	0.2546	0.1048	0.1114	5.92%

Table 6.2. Relative L1 results for different datasets (distance capped at 70 m).

Dataset	Radar only 6 RDCs	Radar only 8 RDCs	Radar+Vision 6 RDCs	Vision only	Percentage improvement over vision only
Highway drive 1	0.2505	0.1114	0.0600	0.0664	9.64%
Highway drive 2	0.1365	0.1247	0.0857	0.0928	7.65%
Urban drive 1	0.2553	0.2647	0.0971	0.1047	7.26%
Urban drive 2	0.3284	0.2046	0.0927	0.0958	3.24%
Urban drive 3	0.2679	0.2764	0.0916	0.1055	13.18%
Urban drive 4	0.4935	0.3766	0.0858	0.1149	25.33%
Urban drive 5	0.3598	0.2268	0.0814	0.0891	8.64%

I performed an analysis of the influence of the number of non-empty RDC cells on my model performance. The results of the analysis point out that the number of 'catastrophic failures' (i.e. very high Relative L1 error) can be reduced by providing the networks with more radar measurements. In practice, it can be done by using more lenient CFAR thresholding, resulting in more non-empty RDC cells.

The labels for my depth prediction network were created without the need for manual labeling, which makes the creation of large-scale datasets very affordable. I believe that pretraining on a large-scale depth prediction dataset may be beneficial for training radar-processing neural networks for different tasks (e.g. occupancy grid estimation), thereby reducing the need for expensive manual labeling.

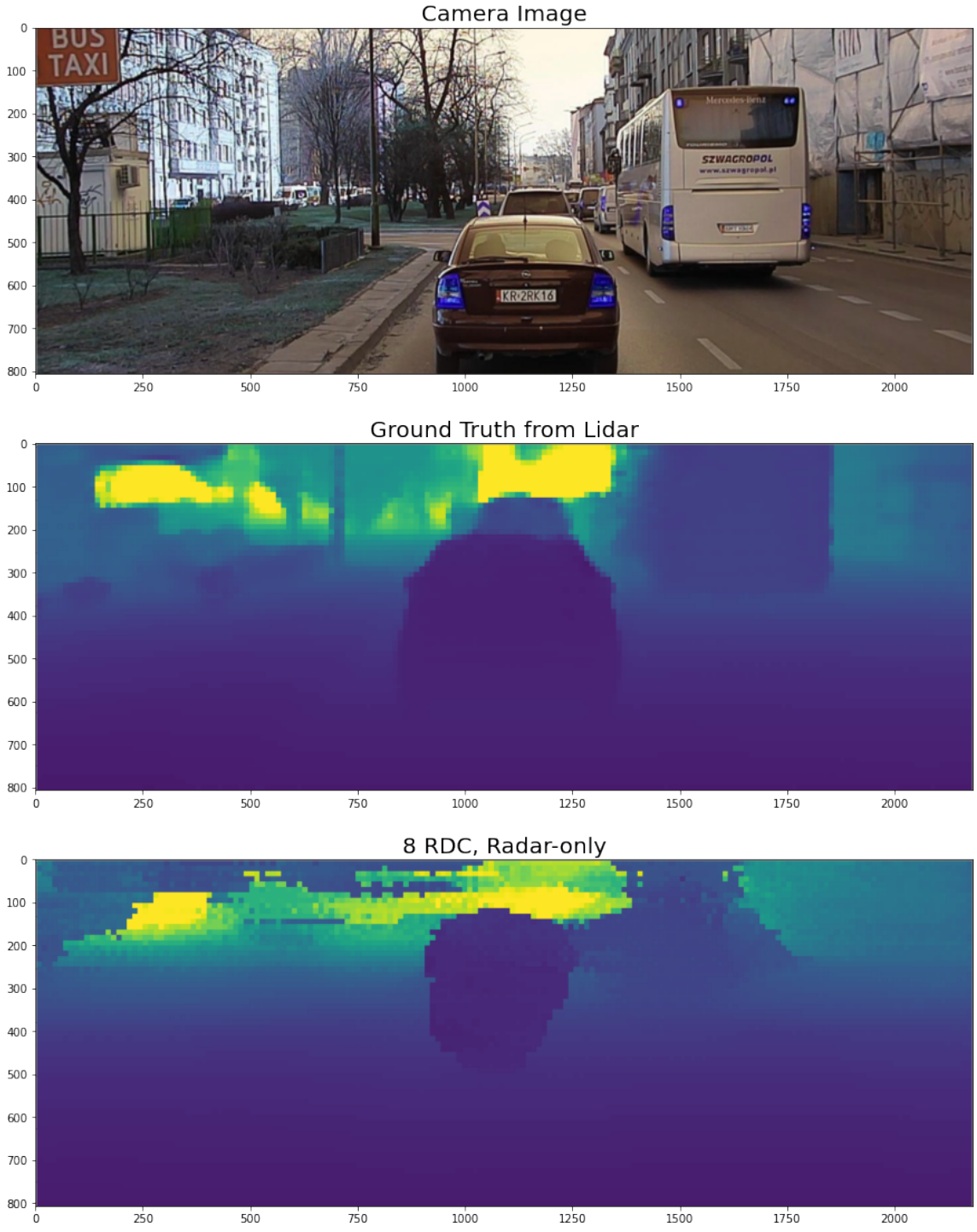

Figure 6.2. Comparison of the output of different versions of the network, frame from Urban Drive 1 (part 1). Depth is color coded, numbers on the axes denote pixel coordinates.

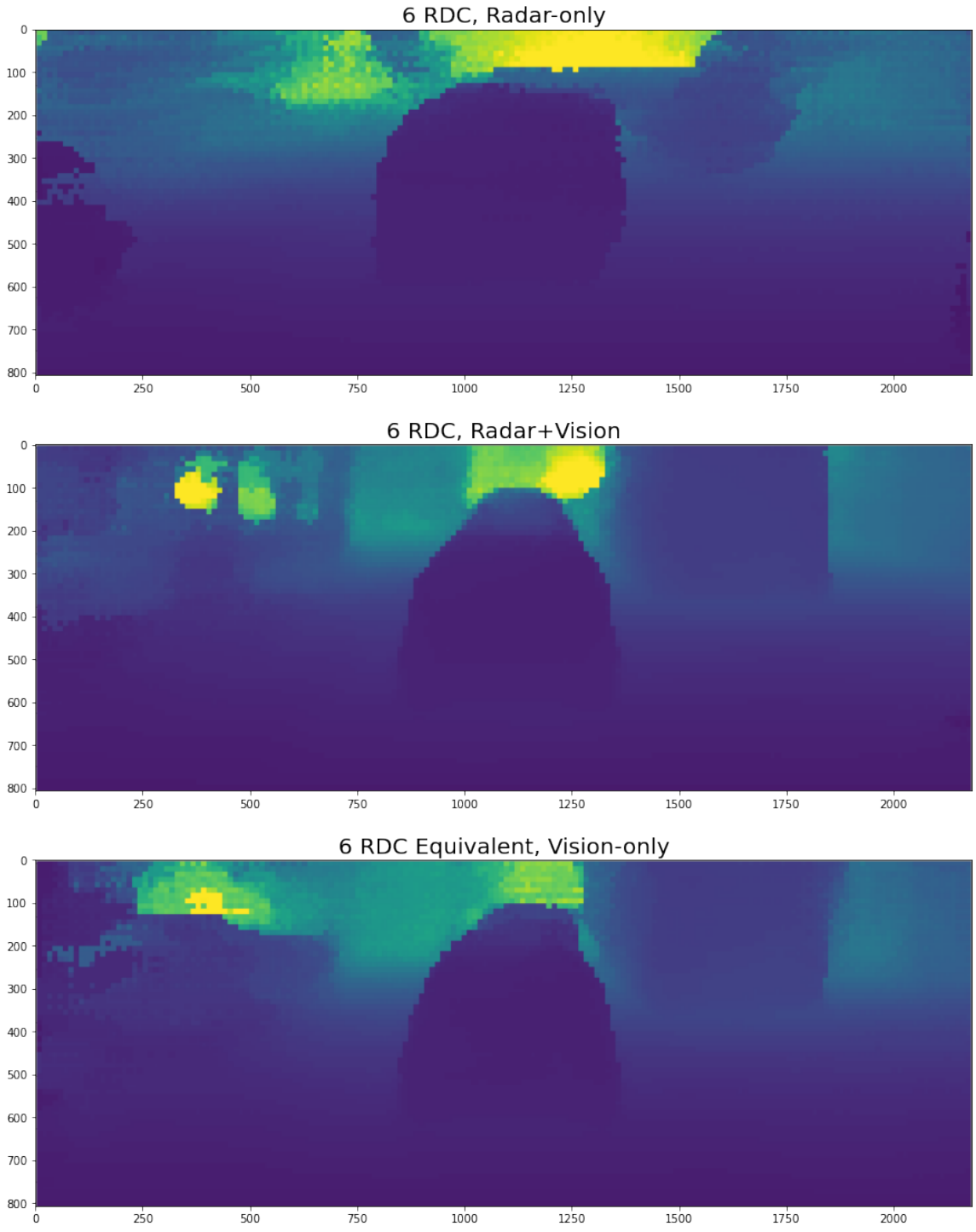

Figure 6.3. Comparison of the output of different versions of the network, frame from Urban Drive 1 (part 2). Depth is color coded, numbers on the axes denote pixel coordinates.

Bibliography

[1] J. Frankle and M. Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

[2] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[3] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[4] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger. Sparsity invariant cnns. In *International Conference on 3D Vision (3DV)*, 2017.